Published in Journals: Applied Sciences, Entropy,
Sustainability, Electronics and Energies

Topic Reprint

Artificial Intelligence and Sustainable Energy Systems

Volume II

Edited by
Luis Hernández-Callejo, Sergio Nesmachnow and Sara Gallardo Saavedra

www.mdpi.com/topics

MDPI

Artificial Intelligence and Sustainable Energy Systems

Artificial Intelligence and Sustainable Energy Systems

Volume II

Editors

Luis Hernández-Callejo
Sergio Nesmachnow
Sara Gallardo Saavedra

MDPI • Basel • Beijing • Wuhan • Barcelona • Belgrade • Manchester • Tokyo • Cluj • Tianjin

Editors

Luis Hernández-Callejo
University of Valladolid
Spain

Sergio Nesmachnow
Universidad de la República
Uruguay

Sara Gallardo Saavedra
Universidad de Valladolid
Spain

Editorial Office
MDPI
St. Alban-Anlage 66
4052 Basel, Switzerland

This is a reprint of Topic published online in the open access journal *Applied Sciences* (ISSN 2076-3417), *Entropy* (ISSN 1099-4300), *Sustainability* (ISSN 2071-1050), *Electronics* (ISSN 2079-9292), and *Energies* (ISSN 1996-1073) (available at: https://www.mdpi.com/topics/Artificial_Intelligence_Energy_Systems).

For citation purposes, cite each article independently as indicated on the article page online and as indicated below:

LastName, A.A.; LastName, B.B.; LastName, C.C. Article Title. *Journal Name* **Year**, *Volume Number*, Page Range.

Volume II
ISBN 978-3-0365-7646-6 (Hbk)
ISBN 978-3-0365-7647-3 (PDF)

Volume I-III
ISBN 978-3-0365-7642-8 (Hbk)
ISBN 978-3-0365-7643-5 (PDF)

Contents

About the Editors

Luis Hernández-Callejo

Luis Hernández-Callejo is an electrical engineer at the Universidad Nacional de Educación a Distancia (UNED, Spain), a computer engineer at UNED, and a PhD candidate at the Universidad de Valladolid (Spain). Professor and researcher at the Universidad de Valladolid. His areas of interest are renewable energy, microgrids, photovoltaic energy, wind energy, smart cities, and artificial intelligence. He has participated in numerous research projects, directed many doctoral theses, and is the author of hundreds of scientific articles.

Sergio Nesmachnow

Sergio Nesmachnow is a full professor at the Faculty of Engineering, Universidad de la República, Uruguay. He is a level III researcher (the maximum level) of the National System of Researchers in Uruguay and a visiting professor at renowned universities and research centers in America and Europe. He has more than 400 publications in scientific journals and international conferences and is responsible for more than 50 research projects.

Sara Gallardo Saavedra

Sara Gallardo Saavedra is a professor and researcher at the Campus Duques de Soria of the University of Valladolid, Spain. Her research focuses on the detection, characterization, and classification of defects in photovoltaic (PV) modules through the use of thermography, electroluminescence, I-V curves, and visual analysis. She has participated in numerous national and international R+D+I projects, carrying out active dissemination of the results with high regularity in the scientific production, including scientific publications in high impact factor journals, book chapters, and contributing to congresses on advanced maintenance in PV. She has made a predoctoral and a postdoctoral stay in the Unit of Solar PV Energy in the Energy Department of the Energy Research Center, Environment, and Technology (CIEMAT) in Madrid, and the researcher has collaborated with different institutions such as the University of Gavle in Sweden, the Universidad del Valle in Colombia, the National Polytechnic Institute of Mexico, and the University of Cuenca in Ecuador.

Preface to ”Artificial Intelligence and Sustainable Energy Systems”

The problems that affect humanity are numerous and occur in different areas. Energy sustainability, climate change, and the effects derived from pollutants and viruses are some of the most relevant problems. The main objective of researchers is to provide solutions to these and other problems.

In recent years, the use of artificial intelligence has increased considerably. Artificial intelligence is used in different areas: energy, sustainability, medicine, health, mobility, industry, etc. Therefore, it is necessary to continue advancing in the application of artificial intelligence to the aforementioned problems. Energy is a precious commodity, and it is increasingly difficult to dispose of it in a sustainable way. In this sense, renewable energy sources are essential, although the use of conventional energy cannot be forgotten. Therefore, sustainable energy systems, integrating renewable and non-renewable energy sources, smart systems, and new business models, are crucial.

Therefore, in this book, the best accepted and published articles on the topic ”Artificial Intelligence and Sustainable Energy Systems” are presented. All articles refer to the themes indicated above.

Luis Hernández-Callejo, Sergio Nesmachnow , and Sara Gallardo Saavedra
Editors

entropy

MDPI

Article

Minimum Distribution Support Vector Clustering

Yan Wang [1,2], Jiali Chen [1], Xuping Xie [1], Sen Yang [1], Wei Pang [3], Lan Huang [1,*], Shuangquan Zhang [1] and Shishun Zhao [4]

[1] Key Laboratory of Symbol Computation and Knowledge Engineering, Ministry of Education, Colleague of Computer Science and Technology, Jilin University, Changchun 130012, China; wy6868@jlu.edu.cn (Y.W.); jiali19@mails.jlu.edu.cn (J.C.); xiexp21@mails.jlu.edu.cn (X.X.); ystop2020@gmail.com (S.Y.); shuangquan18@mails.jlu.edu.cn (S.Z.)
[2] School of Artificial Intelligence, Jilin University, Changchun 130012, China
[3] School of Mathematical and Computer Sciences, Heriot-Watt University, Edinburgh EH14 4AS, UK; w.pang@hw.ac.uk
[4] College of Mathematics, Jilin University, Changchun 130012, China; zhaoss@jlu.edu.cn
* Correspondence: Huanglan@jlu.edu.cn

Abstract: Support vector clustering (SVC) is a boundary-based algorithm, which has several advantages over other clustering methods, including identifying clusters of arbitrary shapes and numbers. Leveraged by the high generalization ability of the large margin distribution machine (LDM) and the optimal margin distribution clustering (ODMC), we propose a new clustering method: minimum distribution for support vector clustering (MDSVC), for improving the robustness of boundary point recognition, which characterizes the optimal hypersphere by the first-order and second-order statistics and tries to minimize the mean and variance simultaneously. In addition, we further prove, theoretically, that our algorithm can obtain better generalization performance. Some instructive insights for adjusting the number of support vector points are gained. For the optimization problem of MDSVC, we propose a double coordinate descent algorithm for small and medium samples. The experimental results on both artificial and real datasets indicate that our MDSVC has a significant improvement in generalization performance compared to SVC.

Keywords: support vector clustering; margin theory; mean; variance; dual coordinate descent

Citation: Wang, Y.; Chen, J.; Xie, X.; Yang, S.; Pang, W.; Huang, L.; Zhang, S.; Zhao, S. Minimum Distribution Support Vector Clustering. *Entropy* **2021**, *23*, 1473. https://doi.org/10.3390/e23111473

Academic Editors: Luis Hernández-Callejo, Sergio Nesmachnow and Sara Gallardo Saavedra

Received: 6 October 2021
Accepted: 4 November 2021
Published: 8 November 2021

1. Introduction

Cluster analysis groups a dataset into clusters according to the correlations of data. To date, many clustering algorithms have emerged, such as plane-based clustering algorithm, spectral clustering, density-based DBSCAN [1], OPTICS [2], Density Peak algorithm (DP) characterizing the center of clusters [3], and partition-based k-means algorithm [4]. In particular, the support vector machine (SVM) has become an important tool for data mining. As a classical machine learning algorithm, SVM can well address the issue of local extremum and high dimensionality of data in the process of model optimization, and it makes data separable in feature space through nonlinear transformation [5].

In particular, Tax and Duin proposed a novel method in which the decision boundaries are constructed by a set of support vectors, the so-called support vector domain description (SVDD) [6]. Leveraged by the kernel theory and SVDD, support vector clustering (SVC) was proposed based on contour clustering, which has many advantages over other clustering algorithms [7]. SVC is robust to noise and does not need to pre-specify the number of clusters in advance. For SVC, it is feasible to adjust its parameter C to obtain better performance, but this comes at the cost of increasing outliers, and it only introduces a soft boundary for optimization. Several insights into understanding the features of SVC have been offered in [8,9]. After studying the relevant literature, we found that these insights mainly cover two aspects: the first aspect is the selection of parameters q and C. Lee and Daniels chose a method similar to a secant to generate monotone increasing sequences of *q* and establish the monotone function of *q* and radius R, which can be applied to high dimensions; the second aspect is optimizing the cluster assignments. Considering the high cost of the second stage of SVC, several methods have been proposed for improving the cluster partition of SVC. First, Ben et al. improved the original Complete Graph (CG) partition by using the adjacency matrix partition based on SV points, which simplified the original calculation, but this method failed to avoid random sampling. Yang et al. elaborated on the Proximity Graph (PG) to model the proximity structure of the m samples with time complexity of O(m) or O(mlog(m)). However, the complexity of this algorithm increases with the increase in dimensionality [10]. Lee et al. studied a cone cluster labeling (CCL) method by using the geometry of the feature space to assign clusters in the data space. If two cones intersect, the samples in these cones belong to the same cluster [9]. However, the performance of CCL is sensitive to kernel parameter q for the cones decided by q. More recently, Peng et al. designed a partition method that utilized the clustering algorithm of similarity segmentation-based point sorting (CASS-PS) and considered the geometrical properties of support vectors in the feature space to avoid the downsides of SVC and CASS-PS [11]. However, CASS-PS is sensitive to the number and distribution of the support vector points recognized. Jennath and Asharaf proposed an efficient cluster assignment algorithm for SVC using the similarity of feature set for data points utilizing an efficient MEB approximation algorithm [12].

It is well known from the margin theory that maximizing the minimum margin is often not the best way for further improving the learning performance. Regarding this, the introduction of the margin mean and margin variance in distribution can make the model achieve better generalization performance, as revealed by Gao and Zhou [13,14]. In classification and regression analysis, there are many methods for improving the learning performance by considering the statistical information of the data. Zhang and Zhou proposed the large margin distribution machine (LDM) and optimal margin distribution machine (ODM) for data classification, which adjusted the mean and variance to improve the performance of the model [15,16]. In regression analysis, MDR, ε-SVR, LDMR, and v-MDAR considers the marginal distribution to achieve better performance. MDR, proposed by Liu et al., minimizes the regression deviation mean and the regression deviation variance, which introduced the statistics of regression deviation into ε-SVR [17]. To deal with this issue, Wang et al. characterized the absolute regression deviation mean and the absolute regression deviation variance and proposed the v-minimum absolute deviation distribution regression (v-MADR) machine [18]. However, it is not very appropriate when both positive-label and negative-label samples are present. Inspired by LDM, Rastogi et al. also proposed a large margin distribution machine-based regression model (LDMR) [19].

In clustering analysis, for a good clustering, when the labels are consistent with the clustering results, SVM can obtain a larger minimum margin. Inspired by this, maximum margin clustering (MMC) considered the large margin heuristic from SVM and added the maximum margin to all possible markers [20]. Improved versions of MMC are also proposed [21]. The optimal margin distribution clustering (ODMC) proposed by Zhang et al. forms the optimal marginal distribution during the clustering process, which characterizes the margin distribution by the first- and second-order statistics. It also has the same convergence rate as state-of-the-art cutting plane-based algorithms [22].

The success of the aforementioned models suggests that there may still exist room for further improving SVC. These models do not involve the improvement in the generalization performance of SVC, that is, the reconstruction of hyperplane, when the distribution of data is fixed in feature space. In this research, we propose a novel approach called minimum distribution support vector clustering (MDSVC), and our novel contributions are as follows:

- We characterize the envelope radius of minimum hypersphere by the first- and second-order statistics, i.e., the mean and variance. By minimizing these two statistics, it can avoid the problem of too many or too few support vector points caused by the inappropriate kernel width coefficient q to some extent, form a better cluster contour, and, thus, improve the accuracy.

- We enhance the generalization ability and robustness of the algorithm by introducing these statistics while the distribution of data is fixed for the given q in feature space.
- We further prove that our method has better performance inspired by the expectation of the probability of test error proposed in SVDD.
- We customize a dual coordinate descent (DCD) algorithm to optimize the objective function of MDSVC for our experiments.

The remainder of this paper is organized as follows. Section 2 introduces the notations, the recent progress in the margin theory, and the SVC algorithm. In Section 3, we present the MDSVC algorithm, which minimizes the mean and the variance, and propose a DCD algorithm to solve the objective function of MDSVC. Section 4 reports our experimental results on both artificial and real datasets. We discuss our method in Section 5 and draw conclusions in Section 6.

2. Background

Suppose $\mathrm{D} = [x_1, \ldots, x_{\mathrm{m}}]$ is a dataset of m samples, and each column is a sample of a d-dimensional vector. $\phi(x)$ is the mapping function induced by a kernel k, i.e., $k(x_i, x_j) = \phi(x_i)^T \phi(x_j)$. SVC used the nonlinear Gaussian kernel function $k(x_i, x_j) = \exp(-q * \|x_i - x_j\|^2)$. Obviously, we have $k(x, x) = 1$. Both MDSVC and SVC aim to obtain the radius R of the sphere, center a of the hypersphere, and the radius of each point in feature space. Formally, we denote X the matrix whose i-th column is $\phi(x_i)$, i.e., $x = [\phi(x_1), \ldots, \phi(x_{\mathrm{m}})]$. In this paper, we use the Gaussian kernel as our nonlinear transformation approach to map data points to feature space.

Recent Progress in Margin Theory

Recent margin theory indicates that maximizing the minimum margin may not lead to an optimal result and better generalization performance. In the SVC algorithm, when the kernel width coefficient q is selected, the distribution of data points mapped to the feature space is determined. If the distribution of boundary data is different from that of internal data, the hyperplane constructed by SVC may not make better use of the data information, thus reducing the performance of SVC. Additionally, we note that SVC is always overfitting with too many support vector points in practice. Gao and Zhou have already demonstrated that marginal distribution is critical to the generalization performance [13]. The high generalization ability of margin has been shown in v-MADR, which minimizes both the absolute regression deviation mean and the absolute regression deviation variance [18]. We also note that SVC can be regarded as a binary classifier divided by the division hyperplane. Inspired by the aforementioned research, we introduce the mean and variance of the marginal distribution and minimize them to reduce the number of support vector points.

For the convenience of readers, a more detailed description of SVC is presented in Appendix A.

3. Minimum Distribution Support Vector Clustering

In this section, we briefly delineate the process of MDSVC, including three subsections, the formula of MDSVC, which minimizes both the mean and the variance, the optimization algorithms based on dual coordinate descent method, and the statistical property of MDSVC that shows the upper bound of the expectation of error. In this research, as mentioned before, we take the Gaussian kernel as a nonlinear transformation approach to map data points to the feature space, and then we derive $k(x, x) = 1$, which is critical for us to simplify the variance and solve the objective function. In addition, we define the mean and variance based on the Euclidean distance. The reason we employ the Euclidean distance is that we can take the objective function as the convex quadratic programming function and the Euclidean norm represents the actual distance between two points rather than the distance on the surface.

We delineate the idea of our algorithm in the feature space in Figure 1 roughly, and more detailed descriptions are given in Sections 3.1.1 and 3.1.2. First, the hyperplanes of MDSVC, SVC, and the unit ball are shown in Figure 1a. By characterizing and minimizing our mean and variance, we can, thus, have the hypersphere of MDSVC as an inclined curved surface in the feature space, as indicated in red in Figure 1a. The intersection of the SVC's hypersphere and the unit sphere is a cap-like area. We further illustrate the main difference between MDSVC and SVC through a lateral view and top view, which are shown in Figure 1b,c, respectively. Figure 1b is the schematic diagram of the MDSVC's Cap and the SVC's Cap. We can find that the center $\boldsymbol{a}$ of MDSVC's hypersphere moves away from the center of the ball and inclines to the distribution of the overall data because of the mean and variance. In Figure 1c, we use *Soft-*R_{svc} to represent the soft boundary of SVC. The centers of the three spheres, namely the unit ball, SVC's hypersphere, and MDSVC's hypersphere, are denoted by $\boldsymbol{o}$, $\boldsymbol{a}_{svc}$, and $\boldsymbol{a}$, respectively. We also use red points to indicate the SVs of MDSVC. As shown in Figure 1c, we can see how the boundary of MDSVC R is determined. Finally, we use Figure 1d to show the distribution of data points and the details of the Cap formed by SVC.

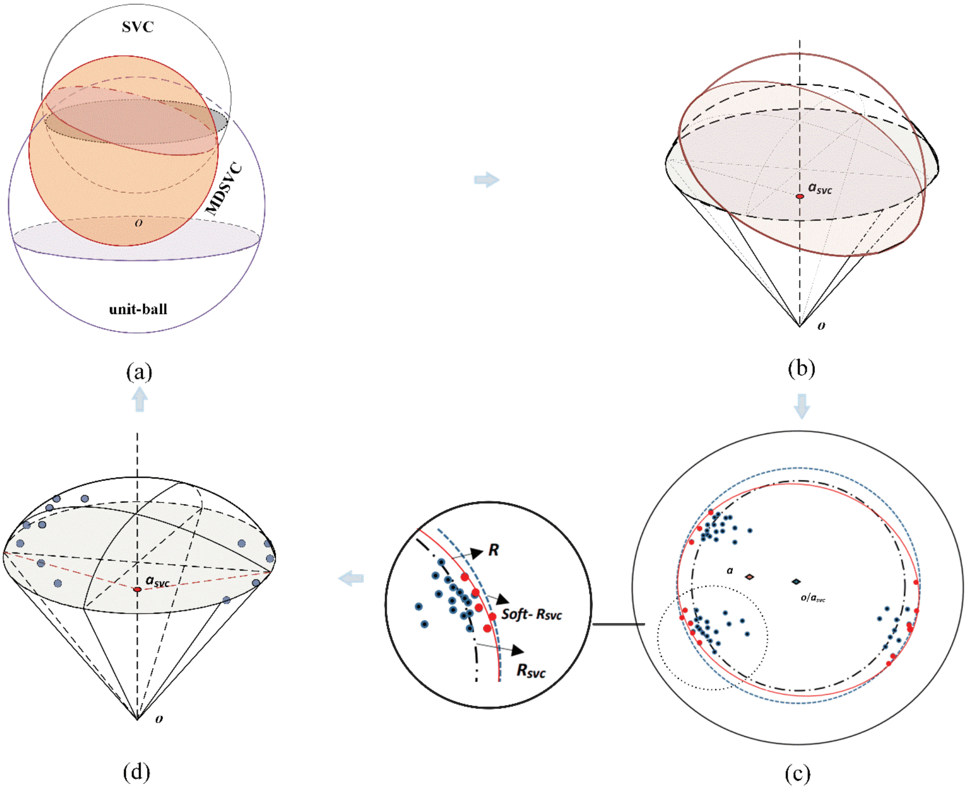


Figure 1. (**a**) Hyperplanes of SVC and MDSVC. (**b**) Two caps formed by SVC and MDSVC with the unit-ball respect tively. (**c**) Top view of Figure 1a. (**d**) Data distribution in the cap.

3.1. Formula of MDSVC

3.1.1. Preliminary

Let $\phi(x)$ be the mapping function induced by a kernel k, i.e., $k(x_i, x_j) = \phi(x_i)^T\phi(x_j)$. In the feature space, we use the Gaussian kernel, and we derive $k(x, x) = 1$. The distance between $\boldsymbol{a}$ and $\boldsymbol{x}$ is $\|\phi(x) - \boldsymbol{a}\|^2$, where $\|.\|$ is the Euclidean norm and $\boldsymbol{a}$ is the center of the sphere. We denote $\boldsymbol{X}$ as the matrix whose i-th column is $\phi(x_i)$. In what follows in the rest of this subsection, we first give the definitions of statistics of mean and variance in clustering; we then present Theorems 1 and 2 to facilitate the formation of the variance; next, we employ the mean and variance (Equations (1) and (2)) to obtain and elucidate the final formula as a convex quadratic programming problem.

Definition 1. *The margin mean is defined as follows.*

$$\bar{\gamma} = \frac{1}{\mathrm{m}}\sum_{i} \|\phi(\boldsymbol{x}_i) - \boldsymbol{a}\|^2 = 1 - \frac{2}{\mathrm{m}}\boldsymbol{a}^T\boldsymbol{X}\boldsymbol{e} + \boldsymbol{a}^2 \tag{1}$$

where ***e*** stands for the all-one column vector of *m* dimensions. Because we use the Gaussian kernel, we have $\boldsymbol{k}(\boldsymbol{x}, \boldsymbol{x}) = 1$, which can facilitate the calculation. The reason for choosing this form of mean is that we incline to make the center of the MDSVC's sphere close to the denser part of the samples. Next, we define the margin variance.

Definition 2. *The margin variance is defined as follows.*

$$\begin{aligned}
\overset{\wedge}{\gamma} &= \tfrac{1}{\mathrm{m}^2}\sum_{i=1}^{m}\sum_{j=1}^{m}\left(\|\phi(\boldsymbol{x}_i) - \boldsymbol{a}\|^2 - \|\phi(\boldsymbol{x}_j) - \boldsymbol{a}\|^2\right)^2 \\
&= \tfrac{4}{\mathrm{m}^2}\sum_{i=1}^{m}\sum_{j=1}^{m}\left(\boldsymbol{a}^T\phi(\boldsymbol{x}_i) - \boldsymbol{a}^T\phi(\boldsymbol{x}_j)\right)^2 \\
&= \tfrac{4}{\mathrm{m}^2}\sum_{i=1}^{m}\sum_{j=1}^{m}\left(\boldsymbol{a}^T\phi(\boldsymbol{x}_i)\phi(\boldsymbol{x}_i)^T\boldsymbol{a} - 2\boldsymbol{a}^T\phi(\boldsymbol{x}_i)\phi(\boldsymbol{x}_j)^T\boldsymbol{a} + \boldsymbol{a}^T\phi(\boldsymbol{x}_j)\phi(\boldsymbol{x}_j)^T\boldsymbol{a}\right) \\
&= \tfrac{8}{\mathrm{m}}\sum_{i=1}^{m}\boldsymbol{a}^T\phi(\boldsymbol{x}_i)\phi(\boldsymbol{x}_i)^T\boldsymbol{a} + \tfrac{8}{\mathrm{m}^2}\sum_{i=1}^{m}\sum_{j=1}^{m}\boldsymbol{a}^T\phi(\boldsymbol{x}_i)\phi(\boldsymbol{x}_j)^T\boldsymbol{a}
\end{aligned} \tag{2}$$

The variance considers the distribution of the overall data rather than the distribution of SVs. Note that if we only characterize the mean in our method, the hyperplane would incline to dense clusters and there may appear more support vectors for the high density of the clusters, which will result in unbalance. However, we should realize that the mean is just the first step to adjusting the sphere of MDSVC. Next, we introduce the variance to adjust the boundary with less volatility. We can find that the variance quantifies the scatter of clustering. Additionally, we denote kernel matrix $Q = X^TX$, where $\boldsymbol{Q}_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \phi(\boldsymbol{x}_i)^T\phi(\boldsymbol{x}_j)$. Note that $\phi(\boldsymbol{x}_i)\phi(\boldsymbol{x}_j)^T$, different from $\phi(\boldsymbol{x}_i)^T\phi(\boldsymbol{x}_j)$, is difficult to obtain due to its complicated form, so we have to use an alternative way to address this issue. Thus, we use the following Theorem 1. Note that the formula of variance can be further simplified, so we employ Theorem 2 to elucidate and facilitate the form of the variance. Finally, we obtain the simplified form for the margin variance as in Equation (8).

Theorem 1. *The center of hypersphere a can be represented as follows,*

$$\boldsymbol{a} = \sum_{i=1}^{m}\alpha_i\phi(\boldsymbol{x}_i) = \boldsymbol{X}\boldsymbol{\alpha} \tag{3}$$

Proof of Theorem 1. Suppose that ***a*** can be decomposed into the span of $\phi(\boldsymbol{x}_i)$ and an orthogonal vector ***v***, that is

$$\boldsymbol{a} = \sum_{i=1}^{m}\alpha_i\phi(\boldsymbol{x}_i) + v = \boldsymbol{X}\boldsymbol{\alpha} + \boldsymbol{v}, \qquad \boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_m]^T \tag{4}$$

where ***v*** satisfies $\phi(\boldsymbol{x}_i)^T\boldsymbol{v} = 0$ for all ***i***, i.e., $\boldsymbol{x}^T\boldsymbol{v} = 0$. Then we have the following formula

$$\boldsymbol{a}^2 = \boldsymbol{\alpha}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\alpha} + \boldsymbol{v}^T\boldsymbol{v} \geq \boldsymbol{\alpha}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\alpha} \tag{5}$$

Therefore, when minimizing $\boldsymbol{a}$, $\boldsymbol{v}$ = 0 does not affect its value. The formula of mean is then derived as follows

$$\begin{aligned}\bar{\gamma} &= \tfrac{1}{\mathrm{m}}\sum_{i} \|\phi(\boldsymbol{x}_i) - \boldsymbol{a}\|^2 = 1 - \tfrac{2}{\mathrm{m}}\boldsymbol{\alpha}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{e} + \boldsymbol{\alpha}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\alpha} + \boldsymbol{v}^T\boldsymbol{v} \\ &\geq 1 - \tfrac{2}{\mathrm{m}}\boldsymbol{\alpha}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{e} + \boldsymbol{\alpha}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\alpha}\end{aligned}$$

From the aforementioned formula, the mean is equivalent to modulus $\boldsymbol{a}$ in optimization, that is, $\bar{\gamma} \Leftrightarrow \boldsymbol{a}^T\boldsymbol{a}$. For variance, we have the following form

$$\begin{aligned}\hat{\gamma} &= \tfrac{1}{\mathrm{m}^2}\sum_{i=1}^{m}\sum_{j=1}^{m}(\|\phi(\boldsymbol{x}_i) - \boldsymbol{a}\|^2 - \|\phi(\boldsymbol{x}_j) - \boldsymbol{a}\|^2)^2 \\ &= \tfrac{8}{\mathrm{m}}\sum_{i=1}^{m}\boldsymbol{a}^T\phi(\boldsymbol{x}_i)\phi(\boldsymbol{x}_i)^T\boldsymbol{a} \ + \ \tfrac{8}{\mathrm{m}^2}\sum_{i=1}^{m}\sum_{j=1}^{m}\boldsymbol{a}^T\phi(\boldsymbol{x}_i)\phi(\boldsymbol{x}_j)^T\boldsymbol{a} \\ &= \tfrac{8}{\mathrm{m}}\sum_{i=1}^{m}\alpha^T\boldsymbol{X}^T\phi(\boldsymbol{x}_i)\phi(\boldsymbol{x}_i)^T\boldsymbol{X}\alpha + \tfrac{8}{\mathrm{m}^2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha^T\boldsymbol{X}^T\phi(\boldsymbol{x}_i)\phi(\boldsymbol{x}_j)^T\boldsymbol{X}\alpha\end{aligned} \tag{6}$$

Thus, the variance is independent of $\boldsymbol{v}$. The rest of the optimization objectives are also independent of $\boldsymbol{v}$. Based on all of the aforementioned equations, $\boldsymbol{a}$ can be represented as the form of Equation (3). □

Theorem 2. *$\boldsymbol{Q}_i\boldsymbol{Q}_i{}^T$, $\sum_{i=1}^{m}\sum_{j=1}^{m}\boldsymbol{Q}_i\boldsymbol{Q}_j{}^T$, $\boldsymbol{H}$, $\boldsymbol{P}$, $\boldsymbol{QG}$ are symmetric matrices where*

$$\boldsymbol{Q}_i = \begin{bmatrix} k(\boldsymbol{x}_1, \boldsymbol{x}_i) \\ \vdots \\ k(\boldsymbol{x}_m, \boldsymbol{x}_i) \end{bmatrix}, \boldsymbol{H} = \tfrac{8\lambda_2}{\mathrm{m}}\sum_{i=1}^{m}\boldsymbol{Q}_i\boldsymbol{Q}_i{}^T$$

$$\boldsymbol{P} = \tfrac{8\lambda_2}{\mathrm{m}^2}\sum_{i=1}^{m}\sum_{j=1}^{m}\boldsymbol{Q}_i\boldsymbol{Q}_j{}^T, \boldsymbol{G} = ((\lambda_1+1)\boldsymbol{Q} + \boldsymbol{H} + \boldsymbol{P})^{-1}\boldsymbol{Q}$$

$$((\lambda_1+1)\boldsymbol{Q} + \boldsymbol{H} + \boldsymbol{P})^{-1} \textit{ refers to the inverse matrix of } (\lambda_1+1)\boldsymbol{Q} + \boldsymbol{H} + \boldsymbol{P}$$

Proof of Theorem 2. $\boldsymbol{Q}_{i(m\times 1)}$ is a column vector of the kernel matrix $\boldsymbol{Q}$ with the following form

$$\boldsymbol{Q}_{i(m\times 1)} = \begin{bmatrix} k(\boldsymbol{x}_1, \boldsymbol{x}_i) \\ \vdots \\ k(\boldsymbol{x}_m, \boldsymbol{x}_i) \end{bmatrix}$$

$$\begin{aligned}\boldsymbol{Q}_i\boldsymbol{Q}_i{}^T &= \begin{bmatrix} k(\boldsymbol{x}_1, \boldsymbol{x}_i) \\ \vdots \\ k(\boldsymbol{x}_\mathrm{m}, \boldsymbol{x}_i) \end{bmatrix}\begin{bmatrix} k(\boldsymbol{x}_1, \boldsymbol{x}_i) & \cdots & k(\boldsymbol{x}_\mathrm{m}, \boldsymbol{x}_i) \end{bmatrix} \\ &= \begin{bmatrix} k(\boldsymbol{x}_1, \boldsymbol{x}_i)^2 & \cdots & k(\boldsymbol{x}_1, \boldsymbol{x}_i)k(\boldsymbol{x}_\mathrm{m}, \boldsymbol{x}_i) \\ \vdots & \ddots & \vdots \\ k(\boldsymbol{x}_1, \boldsymbol{x}_i)k(\boldsymbol{x}_\mathrm{m}, \boldsymbol{x}_i) & \cdots & k(\boldsymbol{x}_\mathrm{m}, \boldsymbol{x}_i)^2 \end{bmatrix}\end{aligned}$$

Note that $\boldsymbol{Q}_i\boldsymbol{Q}_i{}^T$ is a symmetric matrix from the above form. Obviously, $\sum_{i=1}^{m}\sum_{j=1}^{m}\boldsymbol{Q}_i\boldsymbol{Q}_j{}^T$ is a symmetric matrix. Therefore, H and P are both symmetric matrices. We deduce $\boldsymbol{QG}$ as follows

$$\begin{aligned}
&\boldsymbol{QG} = \boldsymbol{Q}((\lambda_1+1)\boldsymbol{Q}+\boldsymbol{H}+\boldsymbol{P})^{-1}\boldsymbol{Q}\\
&\Rightarrow (\boldsymbol{QG})^T = (\boldsymbol{Q}((\lambda_1+1)\boldsymbol{Q}+\boldsymbol{H}+\boldsymbol{P})^{-1}\boldsymbol{Q})^T\\
&\qquad\qquad\quad = \boldsymbol{Q}((\lambda_1+1)\boldsymbol{Q}+\boldsymbol{H}+\boldsymbol{P})T)^{-1}\boldsymbol{Q} = \boldsymbol{Q}((\lambda_1+1)\boldsymbol{Q}+\boldsymbol{H}+\boldsymbol{P})^{-1}\boldsymbol{Q}\\
&\Rightarrow (\boldsymbol{QG})^T = \boldsymbol{G}^T\boldsymbol{Q} = \boldsymbol{QG}
\end{aligned}$$

Therefore, $\boldsymbol{QG}$ is a symmetric matrix. □

According to Theorem 1, we have the following form of mean and variance

$$\overline{\gamma} = \frac{1}{\mathrm{m}}\sum_i \|\phi(\boldsymbol{x}_i)-\boldsymbol{a}\|^2 = 1-\frac{2}{\mathrm{m}}\boldsymbol{\alpha}^T\boldsymbol{Qe}+\boldsymbol{\alpha}^T\boldsymbol{Q\alpha} \tag{7}$$

$$\begin{aligned}
\overset{\wedge}{\gamma} &= \tfrac{8}{\mathrm{m}}\boldsymbol{\alpha}^T\sum_{i=1}^{m}\boldsymbol{Q}_i\boldsymbol{Q}_i{}^T\boldsymbol{\alpha}+\tfrac{8}{\mathrm{m}^2}\boldsymbol{\alpha}^T\sum_{i=1}^{m}\sum_{j=1}^{m}\boldsymbol{Q}_i\boldsymbol{Q}_j{}^T\boldsymbol{\alpha}\\
&= \boldsymbol{\alpha}^T\left(\tfrac{8}{\mathrm{m}}\sum_{i=1}^{m}\boldsymbol{Q}_i\boldsymbol{Q}_i{}^T+\tfrac{8}{\mathrm{m}^2}\sum_{i=1}^{m}\sum_{j=1}^{m}\boldsymbol{Q}_i\boldsymbol{Q}_j{}^T\right)\boldsymbol{\alpha}
\end{aligned} \tag{8}$$

3.1.2. Minimizing the Mean and Variance

Referring to the above subsections, we define the formula of MDSVC as follows

$$\begin{aligned}
&\min_{R,\boldsymbol{a}} R^2+\lambda_1\overline{\gamma}+\lambda_2\overset{\wedge}{\gamma}+C\sum_{i=1}^{m}\xi_i\\
&\text{s.t. } \|\phi(\boldsymbol{x}_i)-\boldsymbol{a}\|^2 \le R^2+\xi_i,\\
&\quad\ \ \xi_i \ge 0
\end{aligned} \tag{9}$$

Consider that the center $\boldsymbol{a}$ of the sphere is closer to the denser part in the feature space as minimizing the mean, and then we minimize the value of λ_2 to make more points closer to $\boldsymbol{a}$, resulting in fewer support vector points. Next, we simplify Equation (9).

Based on Theorem 1, Equation (9) leads to

$$\begin{aligned}
&\min_{\mathrm{R},\alpha} R^2+\boldsymbol{\alpha}^T(\lambda_1\boldsymbol{Q}+\boldsymbol{H}+\boldsymbol{P})\boldsymbol{\alpha}-\tfrac{2\lambda_1}{\mathrm{m}}\boldsymbol{e}^T\boldsymbol{Q\alpha}+C\sum_{i=1}^{m}\xi_i\\
&\text{s.t. } \|\phi(\boldsymbol{x}_i)-\boldsymbol{X\alpha}\|^2 \le R^2+\xi_i,\\
&\quad\ \ \xi_i \ge 0
\end{aligned} \tag{10}$$

By introducing Lagrange multipliers β_i, μ_i, the Lagrange function of Equation (12) is given as follows

$$\begin{aligned}
L(R,\boldsymbol{\alpha},\boldsymbol{\xi},\boldsymbol{\beta},\boldsymbol{\mu}) = {}&\boldsymbol{\alpha}^T((\lambda+1)_1\boldsymbol{Q}+\boldsymbol{H}+\boldsymbol{P})\boldsymbol{\alpha}\\
&-(\tfrac{2\lambda_1}{\mathrm{m}}\boldsymbol{e}^T\boldsymbol{Q}+2\beta^T\boldsymbol{Q})\boldsymbol{\alpha}+R^2(1-\sum_{i=1}^{m}\beta_i)+\sum_{i=1}^{m}(C-\mu_i-\beta_i)\xi_i
\end{aligned} \tag{11}$$

By setting the partial derivatives $\{R,\boldsymbol{\alpha},\boldsymbol{\xi}\}$ to zero for satisfying the KKT conditions, we have the following equations of derivatives

$$\frac{\partial L}{\partial R} = 2R-2R\sum_{i=1}^{\mathrm{m}}\beta_i = 0 \tag{12}$$

$$\frac{\partial L}{\partial \boldsymbol{\alpha}} = 2\boldsymbol{\alpha}^T((\lambda_1+1)\boldsymbol{Q}+\boldsymbol{H}+\boldsymbol{P})-(\frac{2\lambda_1}{\mathrm{m}}\boldsymbol{e}^T\boldsymbol{Q}+2\boldsymbol{\beta}^T\boldsymbol{Q}) = 0 \tag{13}$$

$$\frac{\partial L}{\partial \xi_i} = C-\mu_i-\beta_i = 0 \tag{14}$$

Thus, we adopt $\boldsymbol{G} = ((\lambda_1 + 1)\boldsymbol{Q} + \boldsymbol{H} + \boldsymbol{P})^{-1}\boldsymbol{Q}$, where $((\lambda_1 + 1)\boldsymbol{Q} + \boldsymbol{H} + \boldsymbol{P})^{-1}$ refers to the inverse matrix of $((\lambda_1 + 1)\boldsymbol{Q} + \boldsymbol{H} + \boldsymbol{P})$. On the basis of these equations, we obtain vector $\boldsymbol{A}$ as follows

$$\boldsymbol{A} = \frac{\lambda_1}{\mathrm{m}}((\lambda_1 + 1)\boldsymbol{Q} + \boldsymbol{H} + \boldsymbol{P})^{-1}\boldsymbol{Qe} = \frac{\lambda_1}{\mathrm{m}}\boldsymbol{Ge} \tag{15}$$

Substituting Equation (15) into Equation (13), we thus have

$$\boldsymbol{\alpha} = \boldsymbol{A} + \boldsymbol{G\beta} \tag{16}$$

By substituting Equations (12)–(14) into Equation (11), Equation (11) is re-written as follows

$$\begin{aligned} L(\boldsymbol{\beta}) &= (\boldsymbol{A} + \boldsymbol{G\beta})^T((\lambda_1 + 1)\boldsymbol{Q} + \boldsymbol{H} + \boldsymbol{P})(\boldsymbol{A} + \boldsymbol{G\beta}) - (\tfrac{2\lambda_1}{\mathrm{m}}\boldsymbol{e}^T\boldsymbol{Q} + 2\boldsymbol{\beta}^T\boldsymbol{Q})(\boldsymbol{A} + \boldsymbol{G\beta}) \\ &= \min_{\boldsymbol{\beta}} \tfrac{1}{2}\boldsymbol{\beta}^T\boldsymbol{D\beta} + \boldsymbol{F\beta} \end{aligned} \tag{17}$$

We notice that $\boldsymbol{G} = ((\lambda_1 + 1)\boldsymbol{Q} + \boldsymbol{H} + \boldsymbol{P})^{-1}\boldsymbol{Q}$, so $\boldsymbol{D}$ and $\boldsymbol{F}$ have the following form

$$\begin{aligned} &\boldsymbol{D} = 4\boldsymbol{QG} - 2\boldsymbol{G}^T\boldsymbol{Q} = 2\boldsymbol{G}^T\boldsymbol{Q} = 2\boldsymbol{QG} \\ &\boldsymbol{F} = \tfrac{2\lambda_1}{\mathrm{m}}\boldsymbol{e}^T\boldsymbol{QG} \end{aligned} \tag{18}$$

Referring to the above equations, thus, we derive our formula of MDSVC as follows

$$\begin{gathered} \min_{\boldsymbol{\beta}} \tfrac{1}{2}\boldsymbol{\beta}^T\boldsymbol{D\beta} + \boldsymbol{F\beta} \\ \text{s.t. } 0 \le \beta_i \le C \end{gathered} \tag{19}$$

Based on Theorem 2, $\boldsymbol{D}$ is symmetric and consists of positive elements. We can then make a conclusion that Equation (19) is a convex quadratic problem resulting from the convex objective function and convex domain $\beta \in [0, C]$. Thus, we can solve the objective function with convex quadratic programming.

3.2. The MDSVC Algorithm

Due to the simple box constraint and the convex quadratic objective function of our optimization problem, we adopt the DCD algorithm to minimize one of the variables continuously and keep the other variables fixed to obtain the closed form solution. For our problem, we adjust the value of β_i with a step size of t to make $f(\boldsymbol{\beta})$ reach the minimum value, while keeping other $\beta_{k \neq i}$ unchanged. Our sub-problem is thus as follows

$$\begin{cases} \min\limits_{\beta} f(\boldsymbol{\beta} + t\boldsymbol{e}_i) \\ 0 \le \beta_i + t \le C \end{cases} \tag{20}$$

where $\boldsymbol{e}_i = (0, \ldots, 1_i, .., 0)_m{}^T$ denotes the vector with 1 in the i-th element and 0 is elsewhere. For function f, we have

$$f(\boldsymbol{\beta} + t\boldsymbol{e}_i) = \frac{1}{2}d_{ii}t^2 + \nabla f(\boldsymbol{\beta})_i t + f(\boldsymbol{\beta}) \tag{21}$$

where $d_{ii} = \boldsymbol{e}_i{}^T\boldsymbol{D}\boldsymbol{e}_i$ is the diagonal entry of $\boldsymbol{D}$. Then we calculate the gradient by the following form

$$\nabla f(\boldsymbol{\beta})_i = \boldsymbol{e}_i{}^T\boldsymbol{D\beta} + \boldsymbol{e}_i{}^T\boldsymbol{F}^T \tag{22}$$

As $f(\boldsymbol{\beta})$ is independent of t, we can consider Equation (21) as a function of t. Hence, $f(\boldsymbol{\beta} + t\boldsymbol{e}_i)$ can be transformed into a simple quadratic function of t. Thus, we get the minimum value of Equation (21) by setting the derivation of the aforementioned function with respect to t to zero. Therefore, t is represented as follows

$$t = -\frac{\nabla f(\boldsymbol{\beta})_i}{d_{ii}} \tag{23}$$

We denote ${\beta_i}^{iter}$ as the value of β_i at the i-th iteration, thus, the value of ${\beta_i}^{iter+1}$ can be obtained as

$${\beta_i}^{iter+1} = {\beta_i}^{iter} - \frac{\nabla f(\boldsymbol{\beta})_i}{d_{ii}} \tag{24}$$

Considering the box constraint $0 \leq \beta_i \leq C$ of the problem, we can further obtain the final form of updating β_i

$$\beta_i \leftarrow \min\left(\max\left(\beta_i - \frac{\nabla f(\boldsymbol{\beta})_i}{d_{ii}}, 0\right), C\right) \tag{25}$$

According to Equations (16) and (19), we have $[\nabla f(\beta)]_i = 2{\boldsymbol{e}_i}^T \boldsymbol{Q\alpha}$. Algorithm 1 (MDSVC) describes the procedure of MDSVC with the Gaussian kernel.

Algorithm 1: MDSVC. The DCD Algorithm for our method MDSVC

```
Step 1. Input: Data set $\boldsymbol{X}$, parameters:$[\lambda_1, \lambda_2, C, q]$, maxIter, $m$
Step 2. Initialization: $\boldsymbol{\beta} = \frac{\lambda_1}{m}\boldsymbol{e}, \boldsymbol{\alpha} = \frac{2\lambda_1}{m}\boldsymbol{Ge},\ d_{ii} = 2{\boldsymbol{e}_i}^T\boldsymbol{QGe}_i,\ \boldsymbol{G} = ((\lambda_1+1)\boldsymbol{Q}+\boldsymbol{H}+\boldsymbol{P})^{-1}\boldsymbol{Q}$
Step 3. Iteration(1~maxIter): Iteration stops when the $\beta$ converges.
Step 3.1. Randomly disturb $\beta$ and then get the random index $i$
Step 3.2. Loop ($i$ = 1, 2, . . . , $m$): update gradient and update $\beta$, $\alpha$ alternately.
        $[\nabla f(\beta)]_i \leftarrow 2{e_i}^T Q\alpha$
        ${\beta_i}^{temp} \leftarrow \beta_i$
        $\beta_i \leftarrow \min\left(\max\left(\beta_i - \frac{\nabla f(\beta)_i}{d_{ii}}, 0\right), C\right)$
        $\alpha \leftarrow \alpha + \left(\beta_i - {\beta_i}^{temp}\right)Ge_i$
Step 4. Output: $\alpha$, $\beta$.
```

Meanwhile, we give the analysis of the computational complexity of Algorithm MDSVC, where m denotes the number of the examples and n represents the number of features. We set maxIter to 1000 during our experiments, the time complexity of DCD, thus, can be cast as maxIter*m*m. Furthermore, we can infer that the time complexity of DCD in this paper is the sum of time complexity as shown in Table 1. Considering that m is much greater than n, thus, the time complexity of DCD is $O(m^3)$, and the space complexity of DCD is $O(m^2)$.

Table 1. Time Complexity Calculation of formulas involved.

The Formula of MDSVC	Time Complexity of the Formula
$\boldsymbol{Q} = \boldsymbol{x}^T\boldsymbol{x}$	m*n*m
$\boldsymbol{H} = \frac{8\lambda_2}{\mathrm{m}}\sum\limits_{i=1}^{m}\boldsymbol{Q}_i{\boldsymbol{Q}_i}^T$	m^3
$\boldsymbol{P} = \frac{8\lambda_2}{\mathrm{m}^2}\sum\limits_{i=1}^{m}\sum\limits_{j=1}^{m}\boldsymbol{Q}_i{\boldsymbol{Q}_j}^T$	m^3
$\boldsymbol{G} = ((\lambda_1+1)\boldsymbol{Q}+\boldsymbol{H}+\boldsymbol{P})^{-1}\boldsymbol{Q}$	m^3
$\boldsymbol{A} = \frac{\lambda_1}{\mathrm{m}}((\lambda_1+1)\boldsymbol{Q}+\boldsymbol{H}+\boldsymbol{P})^{-1}\boldsymbol{Qe} = \frac{\lambda_1}{\mathrm{m}}\boldsymbol{Ge}$	m^2

3.3. The Properties of MDSVC

We briefly introduce the properties of MDSVC in this subsection. Hereinafter, the points with $0 < \beta_i < C$ will be referred to as support vectors (SVs); the points with $\beta_i = C$ will be called bounded support vectors (BSVs), which are the same as in SVC. Additionally, the SVDD [5] used cross-validation (leave-one-out) as the criterion to characterize the expectation of the probability of test error, and, then, they describe the expectation as follows

$$E(P(error)) = \frac{num(SV)}{m} \tag{26}$$

The above expectation is more suitable as a standard for adjusting the parameters in the experiments of SVDD rather than having a theoretical basis. It can only estimate the error of the first kind, i.e., the target class. By analyzing the above equation, we further infer that our algorithm can reduce the number of SVs to some extent compared with SVC. Thus, we can obtain better generalization performance compared with SVC theoretically. Inspired by SVDD and LDM, we give the expectation in a manner similar to the approach used in LDM.

Theorem 3. *The center. Let $\boldsymbol{\beta}$ represent the optimal solution of Equation (19) and $E[R(\boldsymbol{\beta})]$ be the expectation of the probability of error, and then we obtain*

$$E[R(\boldsymbol{\beta})] \leq \frac{E[d \sum\limits_{i \in I_1} \frac{\beta_i{}^*}{2(1-R^2)} + |I_2|]}{m}, \tag{27}$$

where $\boldsymbol{I}_1 \equiv \{i|0 < \alpha_i < C\}$, $\boldsymbol{I}_2 \equiv \{i|\alpha_i = C\}$, $d = max\{diag\{D\}\}$.

Proof of Theorem 3. Suppose

$$\begin{aligned} \boldsymbol{\beta}^* &= \underset{0 \leq \boldsymbol{\beta} \leq C}{\text{argmin}} f(\boldsymbol{\beta}), \\ \boldsymbol{\beta}^i &= \underset{0 \leq \boldsymbol{\beta} \leq C, \beta^i = 0}{\text{argmin}} \ f(\boldsymbol{\beta}), \, i = 1, \ldots, m, \end{aligned}$$

and the parameters of the sphere are R and $\boldsymbol{a}$, respectively. As in [16], the expectation is calculated as below

$$E[R(\boldsymbol{\beta})] = \frac{E[\gamma((\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_m, y_m))]}{m}, \tag{28}$$

where $\gamma((\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_m, y_m))$ is the number of errors produced during the leave-one-out procedure. Data points are divided into three categories. Note that if $\beta_i{}^* = 0$, the point is interior in the data space. The cluster of the interior points is totally up to the SVs regardless of the assignment of the cluster in the second stage of the MDSVC procedure based on the analysis of SVDD. Hence, we consider two cases as follows:

(1) $0 < \beta_i{}^* < C$, the data is the support point according to the SVC and KKT conditions, we have

$$f(\boldsymbol{\beta}^i) - \min_t f(\boldsymbol{\beta}^i + t\boldsymbol{e}_i) \leq f(\boldsymbol{\beta}^i) - f(\boldsymbol{\beta}^*) \leq f(\boldsymbol{\beta}^* - \beta_i{}^* \boldsymbol{e}_i) - f(\boldsymbol{\beta}^*), \tag{29}$$

where e_i is a vector with 1 in the i-th coordinate and 0 elsewhere. Incorporating Equation (16) into the aforementioned formula, we have $\langle \phi(\boldsymbol{x}_i, \boldsymbol{a}) \rangle \leq \frac{\beta_i * d_{ii}}{2}$, where $\boldsymbol{x}_i$ are SVs. Further, note that if $\boldsymbol{x}_i$ is an SV, we have $\langle \phi(\boldsymbol{x}_i, \boldsymbol{a}) \rangle = \|\boldsymbol{a}\|^2 = 1 - R^2$, which is a lemma proposed in CCL [9]. Thus, we rearrange $\langle \phi(\boldsymbol{x}_i, \boldsymbol{a}) \rangle \leq \frac{\beta_i * d_{ii}}{2}$, and then obtain $1 \leq \frac{\beta_i * d_{ii}}{2(1-R^2)}$.

(2) $\beta_i{}^* = C$, $\boldsymbol{x}_i$ is the bounded SV (SVs) and must be misclassified in the leave-one-out procedure. Hence we have

$$\gamma((\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_m, y_m)) \leq d \sum_{i \in \boldsymbol{I}_1} \frac{\beta_i{}^*}{2(1 - R^2)} + |\boldsymbol{I}_2| \tag{30}$$

where $\boldsymbol{I}_1 \equiv \{i|0 < \alpha_i < C\}$, $\boldsymbol{I}_2 \equiv \{i|\alpha_i = C\}$, $d = \max\{diag\{D\}\}$. Taking the mean of both sides of Equation (30) and with Equation (28), we finally obtain the result that Equation (27) holds. □

4. Experimental Study

In this section, MDSVC is compared with k-means (KM) [4], optimal margin distribution clustering (ODMC) [22], spectral clustering (SC) [23], mean shift (MS) [24], and hierarchical clustering (HC) [25]. We adopt the results of K-means acting as a baseline rather than maximum margin clustering (MMC) [20] since it could not return results in a reasonable time for most datasets. We experimentally evaluate the performance of our MDSVC compared with the original algorithms of SVC on classic artificial datasets and several medium-sized datasets; that is, we focus on the difference between MDSVC and SVC. Table 2 summarizes the statistics of these data sets. All real-world datasets used for our experiments can be found at UCI (http://archive.ics.uci.edu/ml, 2 February 2021). In Table 2, all of the samples of artificial datasets, namely convex, dbmoon, and ring, are added with Gaussian noises, which are representative of different types of datasets. All algorithms are implemented with MATLAB R2021a on a PC with a 2.50 GHz CPU and 64 GB memory.

Table 2. Experimental Datasets.

Source	Datasets	Samples	Feature
artificial	convex	150	3
	dbmoon	200	2
	ring	900	2
real	iris	150	3
	glass	214	9
	breast	277	9
	heart	303	13
	liver	345	6
	ionosphere	351	34
	vote	435	16
	balance	625	4

4.1. Evaluation Criteria

To evaluate the performance of MDSVC, we use two external indicators, clustering accuracy (Acc) and Adjusted Rand Index (ARI), as our performance metrics. Table 3 shows the definition of the metrics mentioned.

Table 3. Formula of metrics.

Metrics	Definition
Acc	$\mathrm{Acc} = \frac{\sum_{i=1}^{r} c_i}{\mathrm{m}}$
ARI	$\mathrm{ARI} = \frac{\mathrm{RI} - E[\mathrm{RI}]}{\max(\mathrm{RI}) - E[\mathrm{RI}]}$

Accuracy: m is the total number of samples. We use c_i to represent the number of the i-th cluster points classified correctly. We predict the clusters r by performing clustering methods and then measure the accuracy according to the true label.

Adjusted Rand Index: $[\mathrm{y}_1, \mathrm{y}_2, \ldots, \mathrm{y}_\mathrm{s}]$ stands for the true labels of datasets, while $[\mathrm{c}_1, \mathrm{c}_2, \ldots, \mathrm{c}_\mathrm{r}]$ stands for the clusters separated by MDSVC. The sum of TP and TN that we need to obtain can represent the consistency between the clustering result and the result of the original cluster labels. We can distinctly compute it through the confusion matrix. The Rand index (RI), which equals $(\mathrm{TP} + \mathrm{TN})/C_2^m$, represents the frequency of occurrence of agreements over all of the instance pairs. Finally, we can calculate the RI value. However, the RI value is not a constant close to zero for two random label assignments. The ARI, discounting the expected RI of random partition, can however address this issue.

4.2. Experimental Results and Analysis

In the process of SVC tuning, it is noted that there are often too many SVs or too few SVs, failing to form a better contour. Irrational SVs may not divide the clusters better and/or obtain higher precision. Based on this observation, we design experiments on the number of SVs with varying values of λ_1 and q. As mentioned before, the Gaussian kernel $k(x, y) = \exp(-q * \|x - y\|^2)$ is employed for nonlinear clustering, and we can derive $k(x, x) = 1$. We apply the commonly used dichotomy method to select the kernel width coefficient q.

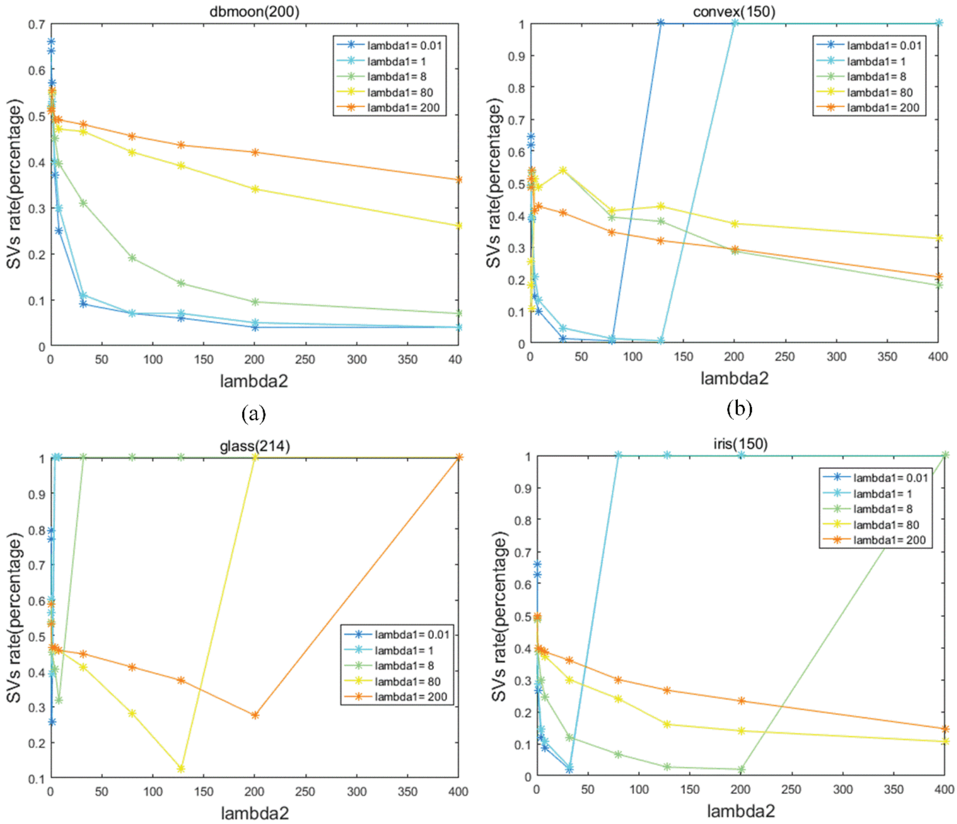
Before conducting experiments on the evaluations for MDVSC and other clustering methods, we analyze the relationship between λ_1 and λ_2 about SVs on two artificial datasets and two real datasets in Figure 2. For the appropriate range of these two parameters, we can realize that the number of SVs increases when λ_1 increases as $\boldsymbol{a}$ is closer to the denser data in the feature space. Furthermore, the increase in λ_2 leads to a decrease in the number of SVs for less volatility in terms of distance from $\boldsymbol{a}$ because the sphere is in the right place with fewer SVs. Thus, it is instructive for us to adjust λ_1 and λ_2 to solve the problem of too many or too few SVs when q and C are given.

We show the results with respect to the corresponding performance metrics in Tables 4 and 5, where PERCENTAGE represents the percentage of the average number of SVs to the total data. We adopt/to represent the method has no need to compute the PERCENTAGE. We summarize the win/tie/loss counts for MDSVC in the last row compared with other methods. For a clearer comparison between MDSVC and SVC, q is selected from the same range $[2^{-7}, 2^7]$ to compute the PERCENTAGE.

In particular, the evaluation of datasets is shown in Tables 4 and 5. Table 4 shows that MDSVC is almost on par with SVC on artificial datasets. It is worth noting that our MDSVC can reduce the number of SVs significantly under the same conditions compared to SVC, i.e., the same q and C. In Table 5, although we note that both SVC and MDSVC have worse Acc or ARI on some datasets, MDSVC still obtains better results than SVC and other methods on most real datasets. Based on the analysis of the experiments, we derive that we can change the SVs by changing the other parameters, λ_1 and λ_2, to achieve better performance when the parameters q and C are selected for MDSVC. In addition, in terms of the CPU time, MDSVC has superior performance on the datasets (ring, vehicle) with higher dimensions and larger size than SVC, as shown in Figure 3. Referring to the comparison of the CPU time between MDSVC and SVC, we indicate that MDSVC has two advantages: better performance and less running time.

The estimated clustering assignments on artificial datasets, convex, and ring, are shown in Figure 4. In order to show the clusters divided by SVs more intuitively and accurately, we draw the contour lines decided by R. We note that the SVC algorithm is almost always overfitting on artificial datasets when the boundary is optimal; that is, all data points are identified as SVs, and, thus, Figure 4 only shows the best non-fitting effect of SVC. Obviously, MDSVC is superior to SVC in terms of forming better boundaries on artificial datasets.

Considering Figure 4a–d, the boundaries of the convex and the dbmoon formed by MDSVC are more rational than SVC in terms of separating clusters. For the ring set, the challenge for SVC is to make rational boundaries with the appropriate number of SVs. MDSVC forms four more rational boundaries and, thus, separates the ring set into two clusters, as shown in Figure 4e, while SVC recognizes only two boundaries in Figure 4f. Moreover, the introduction of statistical items (non-negative), which makes the hyperplane closer to the denser part in the feature space, results in the value of R being larger than SVC. Therefore, it can be seen that we have obtained a greater boundary under the premise of not increasing outliers. In summary, MDSVC obtains better boundaries and a better presentation of the statistical information in the above datasets.



(a) (b) (c) (d)

Figure 2. (**a**): The relationship between λ_1 and λ_2 on dbmoon about SVs. (**b**): The relationship between λ_1 and λ_2 on convex. (**c**): The relationship between λ_1 and λ_2 on glass about SVs. (**d**): The relationship between λ_1 and λ_2 on iris about SVs.

Table 4. The result comparisons on artificial datasets.

Datasets	Metric	KM	SC	HC	ODMC	SVC	MDSVC
convex	ARI Acc PERCENTAGE	0.970 0.820 /	0.748 0.013 /	1.000 0.333 /	0.329 0.333 /	1.000 1.000 64.2%	1.000 1.000 35.0%
dbmoon	ARI Acc PERCENTAGE	0.638 0.900 /	0.324 0.185 /	0.516 0.140 /	0.498 0.500 /	0.928 0.990 79.7%	1.000 1.000 55.3%
ring	ARI Acc PERCENTAGE	0.113 0.322 /	0.171 0.338 /	1.000 0.500 /	0.420 0.511 /	1.000 1.000 95.8%	1.000 1.000 53.1%
MDSVC: w/t/l	ARI Acc PERCENTAGE	(3/0/0) (3/0/0) /	(3,0,0) (3,0,0) /	(3,0,0) (3,0,0) /	(3,0,0) (3,0,0) /	(1,2,0) (1,2,0) (3,0,0)	

For further evaluation, we assess the impact of parameters on ARI, Acc, and PERCENTAGE as the change of parameter values may have a significant influence on the clustering results. Percentage characterizes the level of SVs. For our MDSVC, there are three trade-off parameters λ_1, λ_2, C, and the kernel parameter q. We show the impact of λ_1 on ARI, Acc, and PERCENTAGE by varying it from 2^{-5} to 2^{5} while making the other parameters fixed as the optimal ones. As one can see from Figure 5e–h, the number of SVs is more sensitive to the kernel q and C compared to λ_1 and λ_2. In Figure 5b,d,f,h,j,l, we can see that the results are not sensitive to parameter λ_1 after reaching the optimal results on most datasets. To sum up, we indicate that the mean and variance are both the main factors that affect the performance of the algorithm.

Table 5. The result comparisons on real datasets.

Datasets	Metric	KM	SC	HC	ODMC	SVC	MDSVC
iris	ARI	0.730	0.474	0.558	0.329	0.848	0.828
	Acc	0.347	0.193	0.333	0.333	0.667	0.753
	PERCENTAGE	/	/	/	/	96.1%	51.8%
glass	ARI	0.230	0.067	0.259	0.260	0.750	0.751
	Acc	0.327	0.014	0.028	0.327	0.289	0.351
	PERCENTAGE	/	/	/	/	89.8%	12.5%
breast	ARI	0.171	0.177	0.062	0.585	0.542	0.612
	Acc	0.376	0.087	0.025	0.707	0.484	0.711
	PERCENTAGE	/	/	/	/	98.7%	71.5%
heart	ARI	0.564	0.074	0.058	0.637	0.571	0.580
	Acc	0.551	0.172	0.195	0.772	0.990	0.990
	PERCENTAGE	/	/	/	/	61.3%	55.1%
liver	ARI	0.001	0.002	0.009	0.511	0.489	0.512
	Acc	0.154	0.033	0.067	0.420	0.476	0.493
	PERCENTAGE	/	/	/	/	89.7%	50.4%
ionosphere	ARI	0.178	0.191	0.189	0.612	0.747	0.756
	Acc	0.477	0.393	0.171	0.738	0.687	0.734
	PERCENTAGE	/	/	/	/	90.6%	26.2%
vote	ARI	0.296	0.009	0.512	0.525	0.512	0.525
	Acc	0.540	0.112	0.356	0.386	0.361	0.387
	PERCENTAGE	/	/	/	/	95.2%	88.7%
balance	ARI	0.114	0.184	0.695	0.112	0.570	0.653
	Acc	0.294	0.075	0.016	0.147	0.278	0.356
	PERCENTAGE	/	/	/	/	61.4%	58.1%
MDSVC: w/t/l	ARI	(7,0,0)	(7,0,0)	(6,0,1)	(4,2,1)	(5,1.1)	
	Acc	(6,0,1)	(7,0,0)	(7,0,0)	(5,1,1)	(6,1,0)	
	PERCENTAGE	/	/	/	/	(7,0,0)	

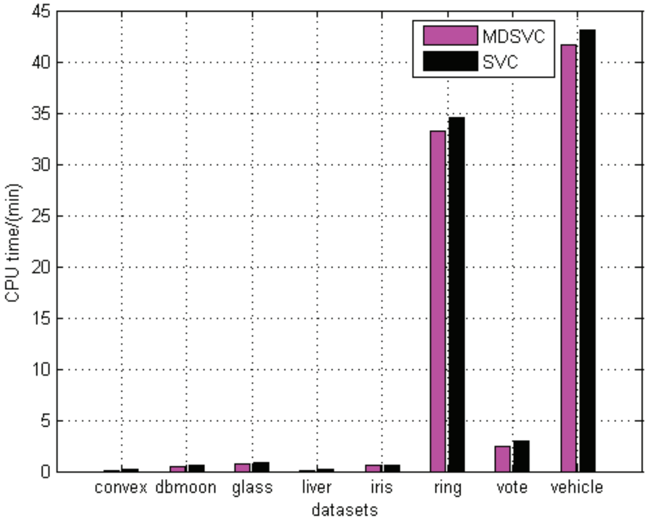
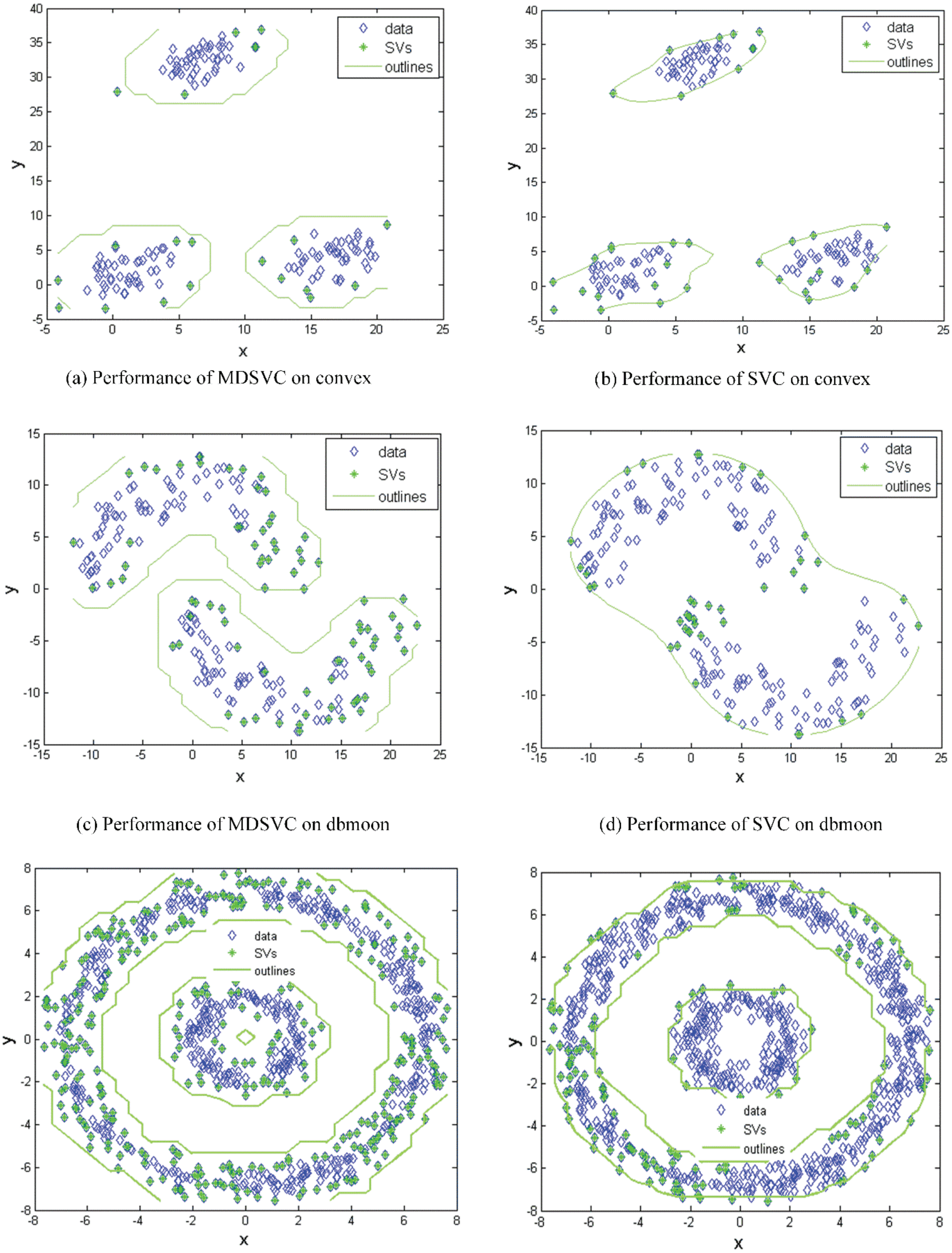


Figure 3. The CPU time of MDSVC and SVC.



(a) Performance of MDSVC on convex

(b) Performance of SVC on convex

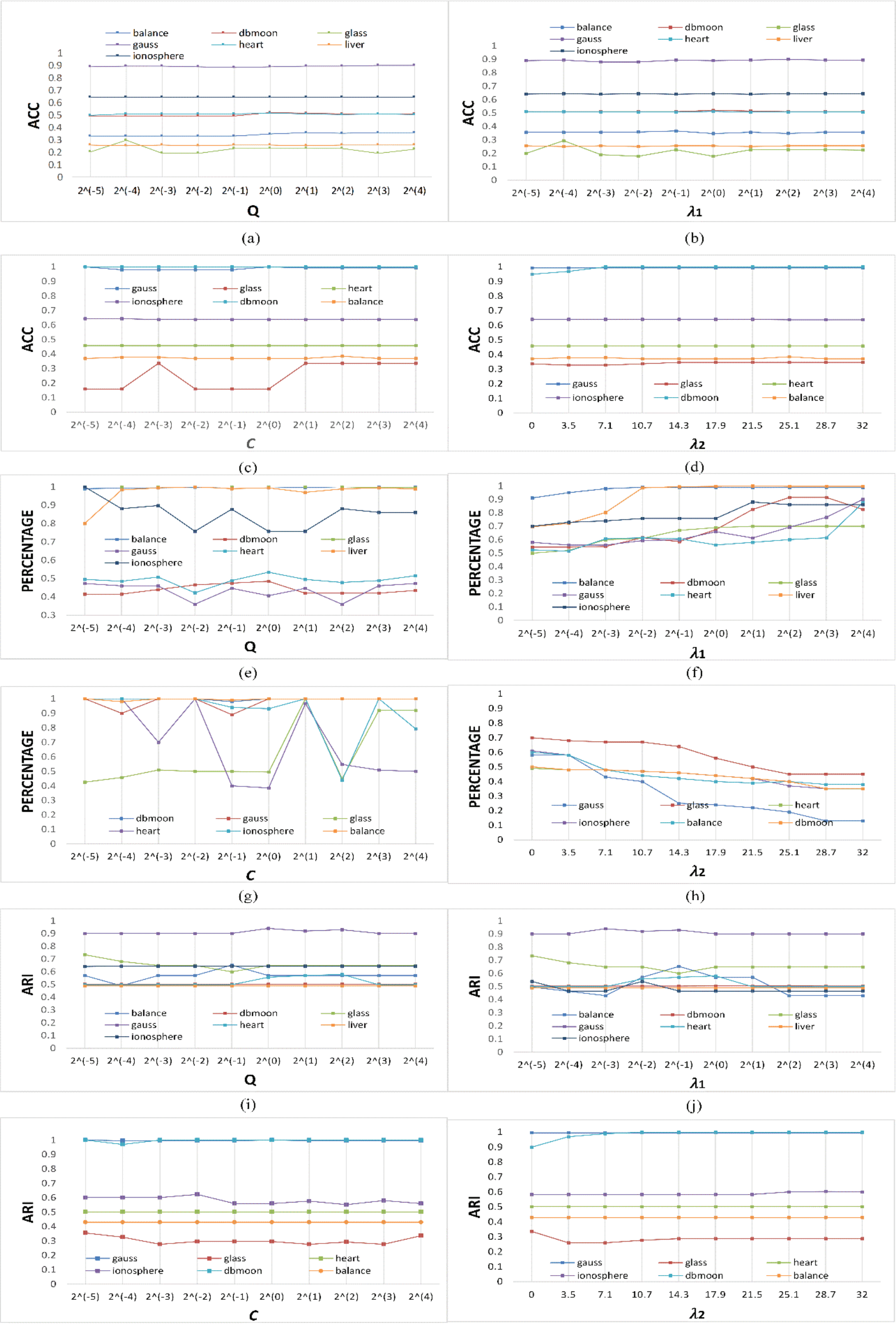
(c) Performance of MDSVC on dbmoon

(d) Performance of SVC on dbmoon

(e) Performance of MDSVC on ring

(f) Performance of SVC on ring

Figure 4. The result of MDSVC on three artificial datasets: convex, dbmoon, and ring. The parameters are set as follows: (**a**): $q = 0.1$; $\lambda_1 = 8$; $\lambda_2 = 32$; $C = 0.1$. (**b**) $q = 1$; $C = 0.1$. (**c**): $q = 0.1$; $\lambda_1 = 1$; $\lambda_2 = 4$; $C = 0.1$. (**d**) $q = 0.5$; $C = 0.1$. (**e**): $q = 2$; $\lambda_1 = 200$; $\lambda_2 = 300$; $C = 0.1$. (**f**) $q = 1$; $C = 0.5$.



(a) (b) (c) (d) (e) (f) (g) (h) (i) (j) (k) (l)

Figure 5. The impact of λ_1, λ_2, C, and kernel parameter Q on ARI, Acc, and PERCENTAGE for different datasets. (**a**–**d**): The impact of λ_1, λ_2, C, and kernel parameter Q on Acc. (**e**–**h**): The impact of λ_1, λ_2, C, and kernel parameter Q on PERCENTAGE. (**i**–**l**): The impact of λ_1, λ_2, C, and kernel parameter Q on Acc.

5. Discussion

It has been proved that trade-off parameters, q and C, have a significant impact on the results of SVC [5,7]. Obviously, we may spend more time in finding the optimal parameters that characterize a better boundary of clusters for SVC. This will result in a large number of SVs during the tuning process, which may affect the partition of clusters and is unreasonable, obviously. We know that it is feasible to adjust parameter C to obtain better performance, but it comes at the cost of increasing outliers. To solve these problems and inspired by the margin theory, we reconstruct a new hypersphere to identify the clusters to make denser sets more easily divided by employing the margin distribution, and then we establish the corresponding theory. We circumvent the high complexity resulting from the variance by demonstrating Theorem 1 and employing the Gaussian kernel, and then we derive the convex optimization problem.

As for the MDSVC algorithm, we design the customized DCD method to solve the convex optimization problem [25]. MDSVC has two other trade-off parameters compared to SVC, namely, λ_1, λ_2. Furthermore, we demonstrate that both of them play an important role in MDSVC through experiments shown in Figure 2 and equations about hypersphere we derive in Section 2. In Figure 4, we can obtain some useful instructive insights as an avenue for adjusting the number of SVs. Therefore, we can obtain better performance by increasing the λ_1 value while there are few SVs. Moreover, we can increase λ_2 value to reduce SVs. If one focuses on forming better outlines of clusters, the recommendation is to control the ratio of λ_1 and λ_2 to between 10^{-2} and 10^2. Once the number of SVs changes drastically, there is no need for us to increase the value of λ_1 and λ_2. Meanwhile, what we should be aware of is that λ_1 should not be zero. We further theoretically prove that the error has an upper bound in Section 3. Due to the lack of prior knowledge (true labels) of clustering algorithms, it is difficult for us to achieve our error bound in a manner similar to the approach used in LDM. We make it by taking the advantage of the error proposed in SVDD [6] and the lemma derived in CCL [9]. According to Figure 1b,c and Figure 4c–e, minimizing the mean and variance can make datasets properly outlined with a proper amount of SVs from a practical and theoretical perspective, while the outlines of SVC are inappropriate. However, we found that our method performed generally when the edge points of datasets are separated relatively densely, where edge points are a collection of relatively sparsely distributed points in the data space. Based on the experiments and formulas obtained; thus, we think that our method performs better on the datasets with edge points dispersing sparsely.

In short, the novel contribution of our work is that we redefine the hyperplane and the center in feature space considering the distribution of data to form better boundaries with a proper amount of SVs. Furthermore, experimental results in most datasets indicate that MDSVC achieves better performance, which further demonstrates the superiority of our method. In the future, we will design a corresponding method to improve the performance, which redefines the clustering partition.

6. Conclusions

In this research, we propose MDSVC, which employs the mean and variance, leveraged by marginal theory and SVM. The novelty of MDSVC lies in its reconstruction of the hyperplane, reducing the number of support vector points compared to SVC under the same conditions, and the improvement in generalization performance. We also have theoretically proven that our generalization performance has been improved, and the error has an upper bound. To optimize the objective function of MDSVC, we employ the DCD method with high applicability and efficiency. Experimental results in most datasets show that MDSVC achieves better performance, which indicates its superiority.

In our future work, we will study the partition of the second stage to further improve the performance of our method. At the same time, to assess the application potential of our algorithm, we will employ our model in more application scenarios.

Author Contributions: Data curation, X.X. and J.C.; Formal analysis, Y.W. and J.C.; Methodology, Y.W. and J.C.; Software, J.C. and S.Y.; Resources, Y.W.; Supervision, L.H.; Writing—original draft, J.C.; Writing—review and editing, W.P., S.Z. (Shuangquan Zhang) and S.Z. (Shishun Zhao). All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (No. 62072212), the Development Project of Jilin Province of China (Nos. 20200401083GX, 2020C003, 20200403172SF), and Guangdong Key Project for Applied Fundamental Research (No. 2018KZDXM076). This work was also supported by Jilin Province Key Laboratory of Big Data Intelligent Computing (No. 20180622002JC).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The implementation is publicly available at http://github.com/Galichen/MDSVC (accessed on 3 November 2021).

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Support Vector Clustering

Support vector clustering (SVC) introduces soft boundary as a tolerance mechanism to reduce the number of boundary support vector points. The algorithm is robust to noise and does not need to know the number of clusters. However, the effectiveness of the algorithm depends on the selection of the kernel width coefficient q and the soft boundary constant C. Clearly, parameter adjustment is time-consuming. SVC has the formulation as follows

$$\begin{aligned} &\min_{R,a} R^2 + C\sum_{i=1}^{m} \xi_i \\ &s.t.\|\phi(\boldsymbol{x}_i) - \boldsymbol{a}\|^2 \leq R^2 + \xi_i, \xi_i \geq 0 \end{aligned} \tag{A1}$$

where parameter C is used for controlling outliers and $C\sum_{i=1}^{m} \xi_i$ is a penalty term, and then the slack variables ξ_i are used as tolerance. SVC looks for the smallest enclosing sphere of radius R, under the constraints $\|\phi(\boldsymbol{x}_i) - \boldsymbol{a}\|^2 \leq R^2 + \xi_i$, where $||.||$ is the Euclidean norm and $\boldsymbol{a}$ is the center of the hypersphere. We can use the Lagrange function to solve the problem

$$L = R^2 + C\sum_{i=1}^{m} \xi_i - \sum_{i=1}^{m} \mu_i \xi_i - \sum_{i=1}^{m} \beta_i (R^2 + \xi_i - \|\phi(\boldsymbol{x}_i) - \boldsymbol{a}\|^2)$$

After we take the derivative of the above formula, the dual problem can be cast as follows

$$\begin{aligned} &\max_{\beta} L = \sum_i \beta_i \kappa(\boldsymbol{x}_i, \boldsymbol{x}_i) - \sum_i \sum_j \beta_i \beta_j \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) \\ &0 \leq \beta_i \leq C \end{aligned}$$

Thus, we can define the distance of each point in the feature space

$$R^2(\boldsymbol{x}) = \|\phi(\boldsymbol{x}) - \boldsymbol{a}\|^2$$

Finally, R^2 has the following form

$$R^2(\boldsymbol{x}) = \kappa(\boldsymbol{x}, \boldsymbol{x}) - 2\sum_i \beta_i \kappa(\boldsymbol{x}, \boldsymbol{x}_i) + \sum_{i,j} \beta_i \beta_j \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) \tag{A2}$$

The radius of the hypersphere is

$$R = \{R(\boldsymbol{x}_i) | \boldsymbol{x}_i \text{ is a support vector}\}$$

Here, the Lagrange multiplier $\beta_i \in (0, C)$, $\boldsymbol{x_i}$ is a support vector (SV). The point is a boundary support vector point (BSV) when $\beta_i = C$. SVC used the adjacency matrix $\boldsymbol{A_{ij}}$ to identify the connected components. For two points $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$

$$\boldsymbol{A}_{ij} = \begin{cases} 0 & \exists \boldsymbol{x}, \text{s.t.} \quad R^2(\boldsymbol{x}) > R, \text{and } \boldsymbol{x} - \boldsymbol{x}_i = t(\boldsymbol{x}_j - \boldsymbol{x}_i) \\ 1 & \text{otherwise.} \end{cases}$$

Finally, the clusters can be defined according to the adjacency matrix $\boldsymbol{A_{ij}}$. The time complexity of calculating the adjacency matrix is $O\left(vm^2\right)$, in which v is the number of samples for the line segment. The quadratic programming problem can be solved by the SMO algorithm, the memory requirements of which are low, and it can be implemented using $O\,(1)$ memory at the cost of a decrease in efficiency. The obvious shortcoming of SVC lies in the high cost of partition.

References

1. Chowdhury, A.; Mollah, M.E.; Rahman, M.A. An efficient method for subjectively choosing parameter 'k' automatically in VDBSCAN (Varied Density Based Spatial Clustering of Applications with Noise) algorithm. *Int. Conf. Comput. Autom. Eng.* **2010**, *1*, 38–41. [CrossRef]
2. Nag, A.; Karforma, S. An Efficient Clustering Algorithm for Spatial Datasets with Noise. *Int. J. Mod. Educ. Comput. Sci.* **2018**, *10*, 29–36. [CrossRef]
3. Tong, W.; Liu, S.; Gao, X.-Z. A density-peak-based clustering algorithm of automatically determining the number of clusters. *Neurocomputing* **2020**, *458*, 655–666. [CrossRef]
4. Kumar, K.M.; Reddy, A.R.M. An efficient k-means clustering filtering algorithm using density based initial cluster centers. *Inf. Sci.* **2017**, *418*, 286–301. [CrossRef]
5. Jiang, W.E.I.; Siddiqui, S. Hyper-parameter optimization for support vector machines using stochastic gradient descent and dual coordinate descent. *EURO J. Comput. Optim.* **2020**, *8*, 85–101. [CrossRef]
6. Tax, D.M.J.; Duin, R.P.W. Support vector domain description. *Pattern Recognit. Lett.* **1999**, *20*, 1191–1199. [CrossRef]
7. Ben-Hur, A.; Horn, D.; Siegelmann, H.T.; Vapnik, V. A Support Vector Method for Clustering. In *Advances in Neural Information Processing Systems 13*; MIT Press: Cambridge, MA, USA, 2001; pp. 367–373.
8. Lee, S.-H.; Daniels, K. Gaussian Kernel Width Generator for Support Vector Clustering. In *Advances in Bioinformatics and Its Applications*; Series in Mathematical Biology and Medicine; World Scientific: Singapore, 2005; Volume 8, pp. 151–162. [CrossRef]
9. Lee, S.-H.; Daniels, K.M. Cone Cluster Labeling for Support Vector Clustering. In Proceedings of the 2006 SIAM International Conference on Data Mining, Bethesda, MD, USA, 12–22 April 2006; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 2006; pp. 484–488. [CrossRef]
10. Yang, J.; Estivill-Castro, V.; Chalup, S. Support Vector Clustering Through Proximity Graph Modeling. In Proceedings of the 9th International Conference on Neural Information Processing, Singapore, 18–22 November 2002; Volume 2, pp. 898–903. [CrossRef]
11. Peng, Q.; Wang, Y.; Ou, G.; Tian, Y.; Huang, L.; Pang, W. Partitioning Clustering Based on Support Vector Ranking. *Adv. Data Min. Appl.* **2016**, *10086*, 726–737. [CrossRef]
12. Jennath, H.S.; Asharaf, S. An Efficient Cluster Assignment Algorithm for Scaling Support Vector Clustering. In *International Conference on Innovative Computing and Communications*; Springer: Singapore, 2022; pp. 285–297. [CrossRef]
13. Gao, W.; Zhou, Z.-H. On the doubt about margin explanation of boosting. *Artif. Intell.* **2013**, *203*, 1–18. [CrossRef]
14. Guo, Y.; Zhang, C. Recent Advances in Large Margin Learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, in press. [CrossRef] [PubMed]
15. Zhang, T.; Zhou, Z.-H. Optimal Margin Distribution Machine. *IEEE Trans. Knowl. Data Eng.* **2020**, *32*, 1143–1156. [CrossRef]
16. Zhang, T.; Zhou, Z.-H. Large Margin Distribution Machine. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Chicago, IL, USA, 11–14 August 2013; pp. 313–322. [CrossRef]
17. Liu, M.-Z.; Shao, Y.-H.; Wang, Z.; Li, C.-N.; Chen, W.-J. Minimum deviation distribution machine for large scale regression. *Knowl.-Based Syst.* **2018**, *146*, 167–180. [CrossRef]
18. Wang, Y.; Wang, Y.; Song, Y.; Xie, X.; Huang, L.; Pang, W.; Coghill, G.M. An Efficient v-minimum Absolute Deviation Distribution Regression Machine. *IEEE Access* **2020**, *8*, 85533–85551. [CrossRef]
19. Rastogi, R.; Anand, P.; Chandra, S. Large-margin Distribution Machine-based regression. *Neural Comput. Appl.* **2020**, *32*, 3633–3648. [CrossRef]
20. Zhang, K.; Tsang, I.W.; Kwok, J.T. Maximum Margin Clustering Made Practical. *IEEE Trans. Neural Netw.* **2009**, *20*, 583–596. [CrossRef] [PubMed]
21. Saradhi, V.V.; Abraham, P.C. Incremental maximum margin clustering. *Pattern Anal. Appl.* **2016**, *19*, 1057–1067. [CrossRef]
22. Zhang, T.; Zhou, Z.-H. Optimal Margin Distribution Clustering. *Natl. Conf. Artif. Intell.* **2018**, *32*, 4474–4481.
23. Luxburg, U. A tutorial on spectral clustering. *Stat. Comput.* **2007**, *17*, 395–416. [CrossRef]

24. Saragih, J.M.; Lucey, S.; Cohn, J.F. Deformable Model Fitting by Regularized Landmark Mean-Shift. *Int. J. Comput. Vis.* **2011**, *91*, 200–215. [CrossRef]
25. Berkhin, P. A Survey of Clustering Data Mining Techniques. In *Grouping Multidimensional Data*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 25–71. [CrossRef]

electronics

MDPI

Article

An Improved Generating Energy Prediction Method Based on Bi-LSTM and Attention Mechanism

Bo He [1], Runze Ma [2], Wenwei Zhang [2,3], Jun Zhu [4,*] and Xingyuan Zhang [1,*]

[1] Department of Polymer Science and Engineering, University of Science and Technology of China, Hefei 230026, China; hebo01@zts.com.cn

[2] Key Laboratory of Wireless Sensor Network and Communication of Chinese Academy of Sciences, Shanghai Institute of Microsystem and Information Technology, Shanghai 201899, China; marunze@mail.sim.ac.cn (R.M.); wenweizhang@mail.sim.ac.cn (W.Z.)

[3] University of Chinese Academy of Sciences, Beijing 100049, China

[4] Special Display and Imaging Technology Innovation Center of Anhui Province, Academy of Opto-Electric Technology, Hefei University of Technology, Hefei 230009, China

[*] Correspondence: jzhu@hfut.edu.cn (J.Z.); zxym@ustc.edu.cn (X.Z.)

Abstract: The energy generated by a photovoltaic power station is affected by environmental factors, and the prediction of the generating energy would be helpful for power grid scheduling. Recently, many power generation prediction models (PGPM) based on machine learning have been proposed, but few existing methods use the attention mechanism to improve the prediction accuracy of generating energy. In the paper, a PGPM based on the Bi-LSTM model and attention mechanism was proposed. Firstly, the environmental factors with respect to the generating energy were selected through the Pearson coefficient, and then the principle and implementation of the proposed PGPM were detailed. Finally, the performance of the proposed PGPM was evaluated through an actual data set collected from a photovoltaic power station in Suzhou, China. The experimental results showed that the prediction error of proposed PGPM was only 8.6 kWh, and the fitting accuracy was more than 0.99, which is better than existing methods.

Keywords: Bi-LSTM; artificial neural networks; generating energy prediction

Citation: He, B.; Ma, R.; Zhang, W.; Zhu, J.; Zhang, X. An Improved Generating Energy Prediction Method Based on Bi-LSTM and Attention Mechanism. *Electronics* **2022**, *11*, 1885. https://doi.org/10.3390/electronics11121885

Academic Editors: Luis Hernández-Callejo and Javid Taheri

Received: 18 April 2022
Accepted: 13 June 2022
Published: 15 June 2022

1. Introduction

The daily generating energy of a photovoltaic power station affects the power consumption of the local area [1–3], while the photovoltaic power generation has a relationship with environmental factors, such as sunshine duration, temperature, etc. Thus, the prediction of the generating energy helps the local power grid system to improve foreseeability and to create a proper generating schedule [4–7]. Since the main facility of a photovoltaic power station works outdoors, the environmental factors would affect the device's working state, making it meaningful to study this effect. For example, the characteristics of temperature changes on the quality of output current in solar power plants are studied in Indonesia [8]. In the global viewpoint, temperature and sunshine duration vary in different countries around the world, which makes the characteristics of solar plants generation different. It is a research focus to predict the generation based on environmental variation.

Generally, prediction is essentially a regression problem, the purpose of which is to build the relationship between environmental factors and generating energy. Hence, the machine learning-based methods have been widely used to achieve power generation prediction, such as outage forecasting, wind power prediction, stability forecasting, peak load prediction, etc.

The machine learning algorithm can treat big data efficiently [9], which can obtain the optimal parameters for PGPMs based on a lot of historical data, as well as make a prediction to generating energy through a trained model. Recently, the PGPMs based on machine learning have been proposed for different types of power stations, such as wind power, thermal power, solar power, nuclear power, etc. Moreover, in order to achieve accurate prediction of daily generating energy of power stations, the input data set of existing PGPMs based on machine learning algorithm usually adopt all the environmental parameters that affect the power generation, which makes the computational complexity of such PGPM very high.

A PGPM based on support vector machine (SVM), one of the most commonly used algorithms in machine learning, was proposed in ref. [10], which applied an improved grid search method to optimize the parameters of C and g to improve the accuracy in forecasting wind power generation. The experimental results showed that the model was able to predict the real-time (15 min) wind power, and the accuracy was up to 78.49%. However, since the computational complexity is very high in scenarios with larger training samples, the SVM-based prediction model is only suitable for small-sample scenarios that can obtain the global optimization parameters.

In order to solve the limitations of the SVM-based PGPM in the large-sample condition, a lightweight PGPM based on ensemble decision tree haswas proposed in ref. [11], which can predict a power system's operating states in a real-time and in an on-line environment. In the proposed solution, an ensemble security predictor (ENSP) was developed and trained to predict and classify power system's dynamic operating states into secure, insecure, and intermediate transitional classes. Finally, the performance was evaluated with two different case studies performed on IEEE 118-bus and IEEE 300-bus test systems, and the experimental results showed that the prediction accuracy was up to 94.4%. However, in some circumstances, for the ensemble decision tree model, it is a challenge to find appropriate pruning schemes to remedy the decision tree due to the overfitting problem, which means the proposed model is only optimized for the existing data, namely, the proposed model is not quite suitable for unknown, new data.

Moreover, to improve the performance of the decision tree-based power generation prediction model, the random forest-based PGPM [12] is developed to forecast medium–long-term power load. In the proposed model, the total load is decomposed into the basic load affected by the economy and meteorological sensitive load affected by meteorological factors, and the prediction results are intelligently corrected by the wavelet neural network algorithm. The experimental results showed that the mean absolute percent error (MAPE) of the random forest-based PGPM was up to 1.43%, which is much better than decision tree-based model proposed in ref. [11]. However, the random forest-based model is equivalent to running multiple decision trees at the same time, which will inevitably have higher computational complexity than decision trees.

Apart from the above-mentioned statistical learning methods, the artificial neural network (ANN), which can simulate the human brain, has been widely used in the power generation prediction field in the recent years [13]. To improve the power production prediction for solar power stations, a PGPM based on the optimized and diversified artificial Neural Networks was proposed in ref. [14]. The method is optimized in terms of the number of hidden neurons and improved in terms of diverse training datasets used to build ANN. The simulation results showed that the proposed approach outperformed three benchmark models, with a performance gain reaching up to 11% for RMSE (root-mean-square error) metric, and the confidence level reaches up to 84%. However, such methods employ classical neural networks, which may not be suitable for some time-varying sequence data of environmental factors.

Generally, for time-varying sequence data, the model based on recurrent neural network (RNN) can provide higher prediction accuracy [15]. The Long Short-Term Memory (LSTM) [16], an improved RNN, could solve the problems of gradient disappearance and gradient explosion when training long sequence data in RNN, making it superior in time sequence forecasting problems [17]. The LSTM network has a strong memory function, which can establish the correlation between the data before and after, thereby improving the prediction accuracy. Based on the above advantages of LSTM, a PGPM based on the high-performance K-Means-long-short-term-memory (K-Means-LSTM) was proposed to predict the power point of wind power in ref. [18], and the simulation results showed that the prediction error (RMSE) of the proposed PGPM reached 62 kW, achieving higher accuracy than RNN-based methods.

However, the LSTM-based PGPM can only capture the data features of the former part of the time sequence, which in turn leads to very limited performance of such methods in some scenarios. As an improved version of LSTM, the Bidirectional LSTM (Bi-LSTM) has better performance via adding a reverse-calculation module. Hence, a Bi-LSTM-based PGPM, which is used to predict the abnormal electricity consumption in power grids, was proposed in [19]. In the Bi-LSTM-based PGPM, the framework of Tensorflow was used to achieve feature extraction and power generation prediction. Final experimental results showed that the accuracy of the Bi-LSTM-based PGPM reached up to 96.1%, which is better than that of the LSTM-based PGPM proposed in ref. [18] (94.5%).

Generally, the Bi-LSTM model can enhance the mining of correlation information of time series feature to some extent; however, it can only extract local features, and it is difficult to obtain global correlation, resulting in the loss of feature correlation information. Simultaneously, such a model only focuses on the inherent relationship between the input features and the target feature, so the input features of each time are assigned the same weight. Nevertheless, the correlation between the input and target characteristics of electricity consumption varies with time, which puts forward higher requirements for the mining of time series correlation of input features.

Hence, in order to improve the performance of PGPMs based on Bi-LSTM, an Attention-Bi-LSTM PGPM based on attention mechanism and Bi-LSTM is proposed in this paper, which adequately employs the advantages of the attention mechanism and Bi-LSTM network. The main contribution of this paper is the way in which the attention mechanism is introduced. To solve this, appropriate attention layers have to be selected and designed to efficiently utilize historical data.

Moreover, existing machine learning-based PGPMs usually use all environmental parameters that affect power generation as input data sets, which can inevitably increase the computational burden of computers. In order to improve computational efficiency, the feature selection algorithm based on Pearson correlation theory [20] is proposed before constructing the proposed PGPM.

The remaining of this paper is organized as follows. Section 2 details the principle of environmental factors selection method based on Pearson coefficient theory. Section 3 presents the methodology of the prediction method. Section 4 elaborates data processing procedures. Section 5 shows experimental layout and relative results. Section 6 concludes the paper and looks forward to future work.

2. Feature Selection

According to the previous analysis, the daily generating energy is related to environmental factors for photovoltaic power stations, and there are correlations between the above-mentioned environmental factors. Therefore, finding the correlation between various environmental factors and selecting appropriate environmental factors as the input dataset can inevitably reduce the computational complexity of prediction models.

Generally, the environmental factors such as daily average temperature, maximum temperature, minimum temperature, daily sunshine duration, average cloud cover, average humidity, minimum humidity, precipitation from 8:00 a.m. to 8:00 p.m., etc., can affect power generation. Under normal circumstances, the more environmental factors, the larger the processing of high-dimensional vectors, as these factors would constitute the input feature vector, and the complexity of calculations will be improved greatly. To reduce the calculation complexity, these environmental factors should be properly selected, and the Pearson correlation coefficients that can evaluate the correlation between environmental factors and generating energy are introduced into the paper.

Pearson correlation coefficient is a value between −1 and 1 that denotes the similar trend between two datasets. For two random variables X and Y, the Pearson correlation coefficient can be expressed by:

$$\rho_{XY} = \frac{cov(X,Y)}{r_X r_Y} = \frac{E(XY) - E(X)E(Y)}{\sqrt{E(X^2) - E^2(X)}\sqrt{E(Y^2) - E^2(Y)}} \tag{1}$$

where $cov(X,Y)$ means the covariance between X and Y; ρ_X and ρ_Y are the standard deviation of X and Y respectively; $E(.)$ function means the random variable's expectation.

In the paper, the Pearson correlation coefficient between environmental factors and generating energy can be calculated by:

$$r = \frac{N\sum x_i y_i - \sum x_i \sum y_i}{\sqrt{N\sum {x_i}^2 - \left(\sum x_i\right)^2}\sqrt{N\sum {y_i}^2 - \left(\sum y_i\right)^2}} \tag{2}$$

where r is the Pearson coefficient; x_i and y_i are the environmental factors and corresponding generating energy respectively; N is the amount of historical data samples.

Hence, in order to select the optimal environmental factors to construct the input dataset, the Pearson coefficients between environmental factors and generating energy obtained from a photovoltaic power station in Suzhou, China (Supplementary Materials), were used and the results are shown in Table 1.

Table 1. Pearson coefficients between environmental factors and generating energy.

Environmental Factors	Pearson Coefficient
Daily average temperature	0.42551
Maximum temperature	0.54173
Minimum temperature	0.27529
Average humidity	−0.69062
Minimum humidity	−0.74763
Precipitation from 8:00 a.m. to 8:00 p.m.	−0.33582
Daily sunshine duration	0.83609
Average cloud cover	−0.59997

According to Pearson coefficient theory, factors with positive Pearson coefficients have good correlation with the generating energy, which means they are suitable to be regarded as the input data features to predict the generating energy. As can been found in Table 1, some factors such as average humidity, minimum humidity, precipitation from 8:00 a.m. to 8:00 p.m., and average cloud cover could be filtered because they have a weak correlation with generating energy. Hence, the remaining four environmental factors are taken to compose the input feature vector, which means the data feature vectors are four-dimensional.

3. The Methodology

3.1. Bi-LSTM Model

Generally, Bi-LSTM is composed by two LSTM models of the forward and backward direction, which can capture long-term dependencies in one direction. Hence, the Bi-LSTM allows more information to be preserved by capturing long-term dependencies in both directions, which is suitable for power generation forecasting scenarios that require big data processing. The architecture of Bi-LSTM model can be shown as Figure 1.

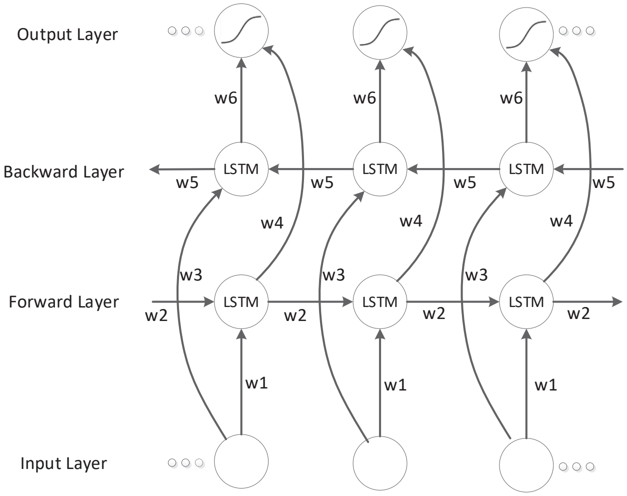


Figure 1. Architecture of the Bi-LSTM model.

From Figure 1, it can be found that the forward directional LSTM is used to produce the past information of input sequences, while the backward directional LSTM can gain the future information of input sequences. Finally, the final output is obtained by combining the corresponding time output of forward directional LSTM and backward directional LSTM at each time, which can be expressed by:

$$h_t = f(w_1 x_t + w_2 h_{t-1}) \tag{3}$$

$$h'_t = f\left(w_3 x_t + w_5 h'_{t+1}\right) \tag{4}$$

$$o_t = g\left(w_4 h_t + w_6 h'_t\right) \tag{5}$$

where h_t and h'_t are current node outputs of the forward and backward direction respectively; o_t is the output of current cell; w_1, w_2, w_3, w_4, w_5 and w_6 are the weight coefficients.

According to Equations (3)–(5), w_1 and w_3 are the weights of the input to the forward and backward hidden layers, w_2 and w_5 are the weights between the same hidden layers, while w_4 and w_6 are the weights of the forward and backward hidden layers to the output layers. Compared with LSTM, Bi-LSTM improves the globality and integrity of feature extraction.

3.2. Feature Attention Mechanism

Generally, the feature attention mechanism can improve the performance of Bi-LSTM by dynamically assigning the attention weight to input features, as well as the correlation between hidden layer and target features being mined, which can effectively reduce the loss of feature correlations. The architecture of the feature attention mechanism is shown in Figure 2.

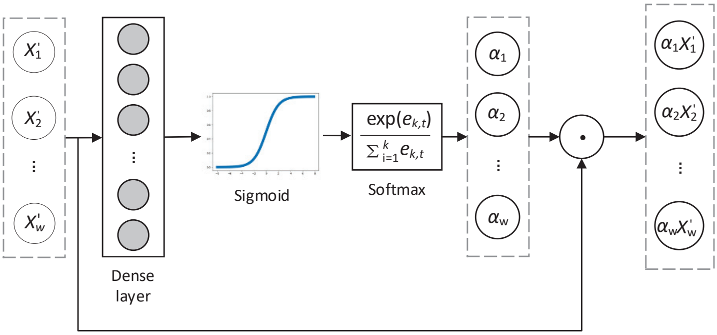


Figure 2. Architecture of feature attention mechanism.

From Figure 2, the input feature vector of time sequences with K hidden layer features can be described as $X_t = [X_{1,t}, X_{2,t}, \ldots, X_{k,t}]$. Then, a single layer neural network is used to calculate the attention weight vector, which can be expressed by:

$$e_{k,t} = \sigma(W_e X_t + b_e) \tag{6}$$

where t is the time length of input sequences depending on sampling rates, and $e_{k,t} = [e_{1,t}, e_{2,t}, \ldots, e_{k,t}]$ is regarded as the combination of attention weight coefficients corresponding to the input characteristics of current moments. W_e is the trainable weight matrix, b_e is an offset vector, and $\sigma(.)$ is a sigmoid activation function.

The data sequence generated by the sigmoid activation function is normalized by the softmax function, which is denoted as:

$$\alpha_{k,t} = \frac{\exp(e_{k,t})}{\sum\limits_{i=1}^{k} e_{i,t}} \tag{7}$$

where $\alpha_{k,t}$ is the attention weight of character k, and the resulting attention weight $\alpha_{k,t}$ and hidden layer feature vector X'_t are recalculated as a weighted feature vector X'_{a_t}, which can be expressed by:

$$X'_{a\_t} = a_t \odot X'_t = [a_{1,t}x_{1,t}, a_{2,t}x_{2,t}, \cdots, a_{k,t}x_{k,t}] \tag{8}$$

3.3. Temporal Attention Mechanism

Apart from the feature attention mechanism, the temporal attention mechanism can allocate attention weight to the temporal information carried by each historical moment of the input sequence to distinguish its influence on the output of the current time. At the same time, the time sequence of each historical moment can be extracted independently and the information expression of the critical moment can be enhanced; the architecture of the temporal attention mechanism is shown in Figure 3.

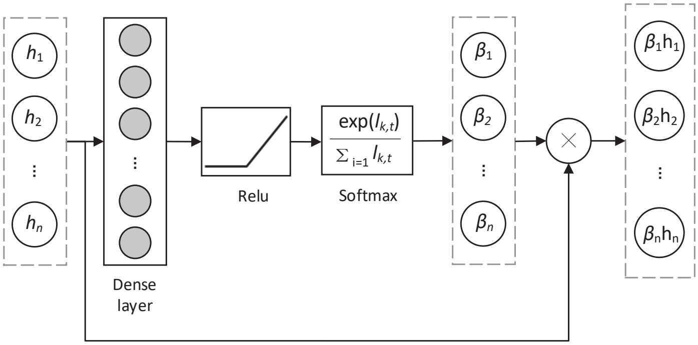


Figure 3. Architecture of the temporal attention mechanism.

From Figure 3, it can be found that the input is the hidden layer state of the Bi-LSTM network iterated to time, which can be expressed by $h_t = [h_{1,t}, h_{2,t}, \ldots, h_{n,t}]$, where n is the time window length of input sequences. The temporal attention weight vector l_t of the current moment corresponding to each historical moment can be described as:

$$l_t = \mathrm{Re}LU(W_d X_t + b_d) \tag{9}$$

where $l_t = [l_{1,t}, l_{2,t}, \ldots, l_{k,t}]$; W_d is a trainable weight matrix; b_d is a bias vector; and $\mathrm{Re}LU(.)$ is an activation function to increase feature differences and make the weight distribution more centralized.

Moreover, from Figure 3, it can be seen that the input sequence generated by the activation function is normalized by the softmax function to obtain the temporal attention weight, which can be expressed by $\beta_t = [\beta_{1,t}, \beta_{2,t}, \ldots, \beta_{k,t}]$, where $\beta_{k,t}$ is the attention weight of character k, which can be denoted as:

$$\beta_{k,t} = \frac{\exp(l_{k,t})}{\sum\limits_{\mathrm{i}=1}^{k} l_{i,t}} \tag{10}$$

Hence, the weighted feature vector h'_t can be recalculated via data feature vector h_t generated by the hidden layer, which can be expressed by:

$$h'_t = \beta_t \otimes h_t = \sum_{i=1}^{k} \beta_{i,t} h_{i,t} \tag{11}$$

3.4. The Proposed Attention-Bi-LSTM PGPM

In the paper, the Attention-Bi-LSTM PGPM based on the attention mechanism and Bi-LSTM network is proposed, which consists of an input layer, feature attention layer, Bi-LSTM layer, temporal attention layer, residual connected layer, and fully connected layer, and the architecture of the Attention-Bi-LSTM PGPM is shown in Figure 4.

From Figure 4, it can be found that a Bi-LSTM network is built to extract the hidden temporal correlation information from the input sample X_t, which is composed of the history sequence and related four-dimensional input feature vector extracted from environmental factors. The sample is fed into first Bi-LSTM network and the hidden layer feature X'_t is obtained.

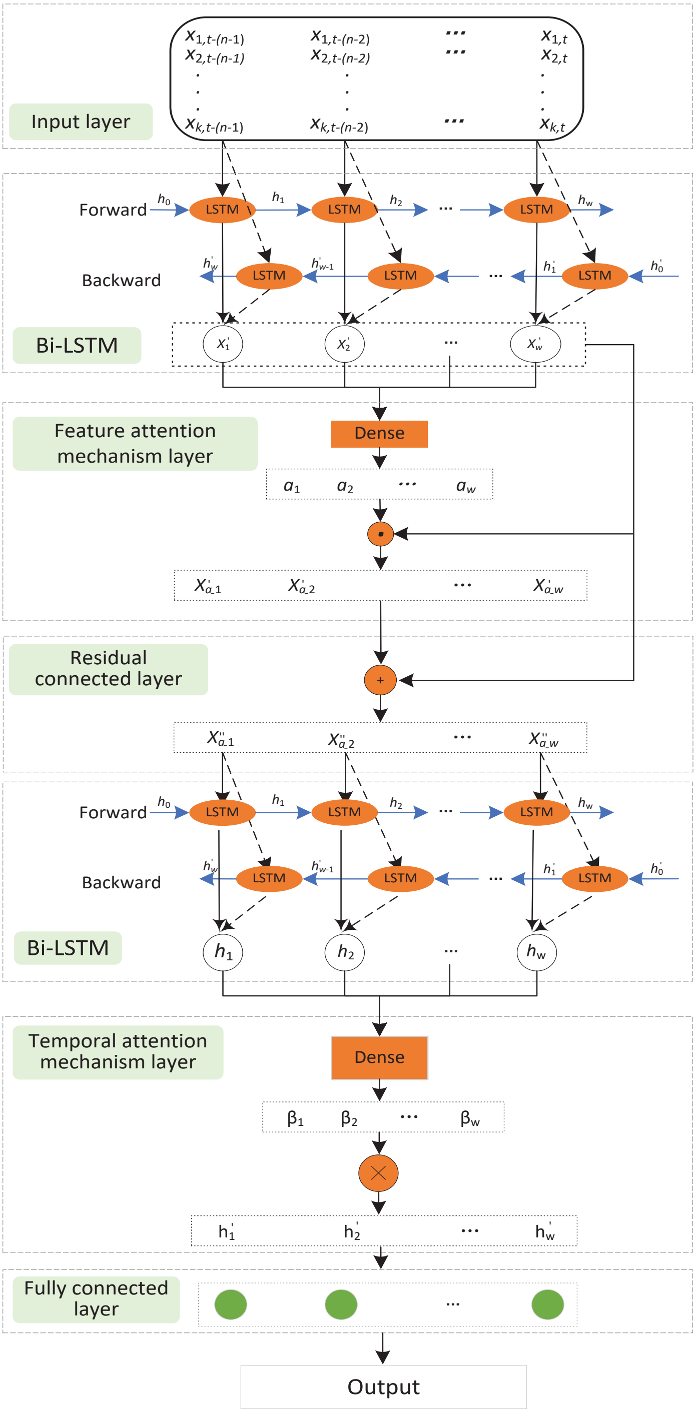

Input layer
$x_{1,t\text{-}(n\text{-}1)}$ $x_{2,t\text{-}(n\text{-}1)}$ $x_{k,t\text{-}(n\text{-}1)}$
$x_{1,t\text{-}(n\text{-}2)}$ $x_{2,t\text{-}(n\text{-}2)}$ $x_{k,t\text{-}(n\text{-}2)}$
$x_{1,t}$ $x_{2,t}$ $x_{k,t}$
Forward
Backward
LSTM
h_0 h_1 h_2 h_w
h'_w $h'_{w\text{-}1}$ h'_1 h'_0
Bi-LSTM
X'_1 X'_2 X'_w
Feature attention mechanism layer
Dense
a_1 a_2 ··· a_w
X'_{a_1} X'_{a_2} ··· X'_{a_w}
Residual connected layer
+
X''_{a_1} X''_{a_2} ··· X''_{a_w}
Forward
Backward
LSTM
Bi-LSTM
h_1 h_2 ··· h_w
Temporal attention mechanism layer
Dense
β_1 β_2 ··· β_w
h'_1 h'_2 ··· h'_w
Fully connected layer
Output

Figure 4. Architecture of Attention-Bi-LSTM PGPM.

Then, the feature attention mechanism was used to explore the potential correlation between hidden layer features X_t'. The features X_t' extracted from the first Bi-LSTM were sent to the feature attention layer. In order to extract the hidden layer features X_t', the attention weight of features was allocated dynamically. Based on the above statements, the weighted hidden layer feature X_{a_t}' was obtained by dynamic distribution of the feature attention weight.

Next, the weighted feature X_{a_t}' was residually linked to the original feature X_t', which was fed into the second Bi-LSTM to obtain the hidden layer feature h_t. Moreover, the correlation between the historical sequence and the feature h_t was mined in the second Bi-LSTM's hidden layer, as well as the weighted feature vector h_t' being mined in the temporal attention layer. Finally, the power generation was predicted in the fully connected layer with the above-mentioned parameters.

4. Data Processing

4.1. Data Cleaning

In this paper, a historical dataset collected from a photovoltaic power station with a sampling rate of 1 day, which includes daily average temperature, maximum temperature, minimum temperature, daily sunshine duration, and daily generating energy, was introduced into the experiment [SM]. The input data sample is a 4-dimensional vector, which denotes the above-mentioned four environmental features, and every input feature vector corresponds to a daily generating energy, as the output value.

For data cleaning, firstly, the data sample with missing or invalid features was preprocessed. In this paper, the data sample with invalid features was eliminated directly.

Secondly, different features have values of different ranges, making it necessary to normalize the feature data. The normalized value could be calculated by:

$$\begin{cases} \overline{x} = \frac{1}{n}\sum\limits_{i=1}^{n} x_i \\ std(x) = \sqrt{\frac{1}{n}\sum\limits_{i=1}^{n}(x_i - \overline{x})^2} \\ y_i = \frac{x_i - \overline{x}}{std(x)} \end{cases} \tag{12}$$

where x_i is the i-th original feature value; y_i is the i-th normalized feature value; n is the amount of data samples.

4.2. Division of Dataset

To train the prediction model parameters, which are mainly some structural weight values, 75% of historical data samples were recognized as the training dataset, and the remaining 25% of data samples were taken as the testing dataset to examine the prediction efficiency. The ensemble division of dataset is shown in Figure 5.

As shown in Figure 5, the training process adopts a cross validation mechanism, composed by many epochs. In each epoch, 90% of the training samples are regarded as a sub-training set, and the remaining 10% of the training samples are regarded as the sub-testing dataset. The partition scheme of the sub-training dataset and sub-testing dataset is to divide them randomly. From Figure 5, it can be found that the optimal parameters are obtained through multiple cross-validation, which was used to provide a basis for the subsequent experiments.

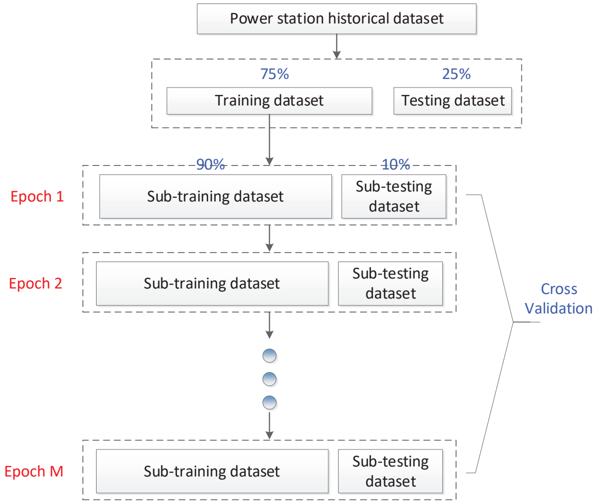


Figure 5. Division of the dataset.

5. Experimental Results and Analysis

5.1. Parameter Tuning and Statistical Analysis

In order to illustrate the advantages of the proposed PGPM, the performance of existing PGPMs based on Support Vector Regression (SVR) [21], Decision Tree [22], Random Forest [23], LSTM [24], and Bi-LSTM [25] were compared with the Attention-Bi-LSTM PGPM proposed in the paper, and the main experimental parameters of PGPMs based on SVR, Decision Tree, and Random Forest were tuned, as shown in Tables 2–4, respectively.

Table 2. Parameter tuning of PGPM based on SVR.

Penalty C	RBF Gamma	Prediction Error (kWh)
100	1	238.9
1	0.01	479.3
0.1	0.01	489.1

Table 3. Parameter tuning of PGPM based on Decision Tree.

Max Depth	Prediction Error (kWh)
4	255.9
5	243.6
6	236.0
10	291.7
90	305.6

Table 4. Parameter tuning of PGPM based on Random Forest.

Number of Estimators	Minimum Samples of Subtree	Minimum Samples of Leaf	Prediction Error (kWh)
200	2	1	231.8
200	2	4	232.1
100	2	1	232.9
400	4	1	232.9
400	4	2	232.8

From Tables 2–4, the best parameters of each algorithm could be determined, for the best prediction accuracy was achieved.

Moreover, the essence of proposed Attention-Bi-LSTM PGPM is an improved version of PGPMs based on LSTM and Bi-LSTM. In order to ensure the comparability and accuracy of subsequent experimental results, the experiments parameters of the above three LSTM-based PGPMs are the same in the paper, and the related parameters are shown in Table 5.

Table 5. Related parameters of LSTM-based PGPMs.

Category	Parameter
Length of Time Sequence	4
Bi-LSTM Hidden Layer Neurons	350
Learning Rate	0.01
Batch Size	64
Optimization Algorithm	Adam
Loss Function	Mean Squared Error (MSE)
Neuron Loss Rate	0.1

Furthermore, the statistical analysis was performed for the selected parameter configurations, the way of which is to run the model training and prediction 50 times. Each time, the training dataset and testing dataset were partitioned randomly to evaluate the statistical stability of these models, and the results are shown in Table 6.

Table 6. Statistical analysis on the studied methods.

Method	Average of RMSE (kWh)	Standard Deviation of RMSE (kWh)
SVR	238.9	2.3
Decision Tree	236.0	2.7
Random Forest	231.8	1.9
LSTM	29.7	1.5
Bi-LSTM	18.3	1.8
Attention-Bi-LSTM (Ours)	8.6	1.2

Table 6 shows the standard deviation for each algorithm is only 1~2 kWh, which means the prediction result is stable when the parameters are determined. Therefore, the subsequent comparison of parameter-dependent results could reflect the performance gaps of different methods from the statistical viewpoint.

Moreover, in order to evaluate the performance of the above algorithms, the Python scikit-learn library was employed to implement the PGPMs based on SVR, Decision Tree, and Random Forest algorithms, while the Tensorflow library was employed to implement the PGPMs based on LSTM, Bi-LSTM, and the proposed Attention-Bi-LSTM.

5.2. Experimental Results

According to above-mentioned relevant experimental parameters shown in Table 2 to Table 5 and experimental layouts, the visualized experimental results within half a year output by six PGPMs mentioned above are shown in Figure 6.

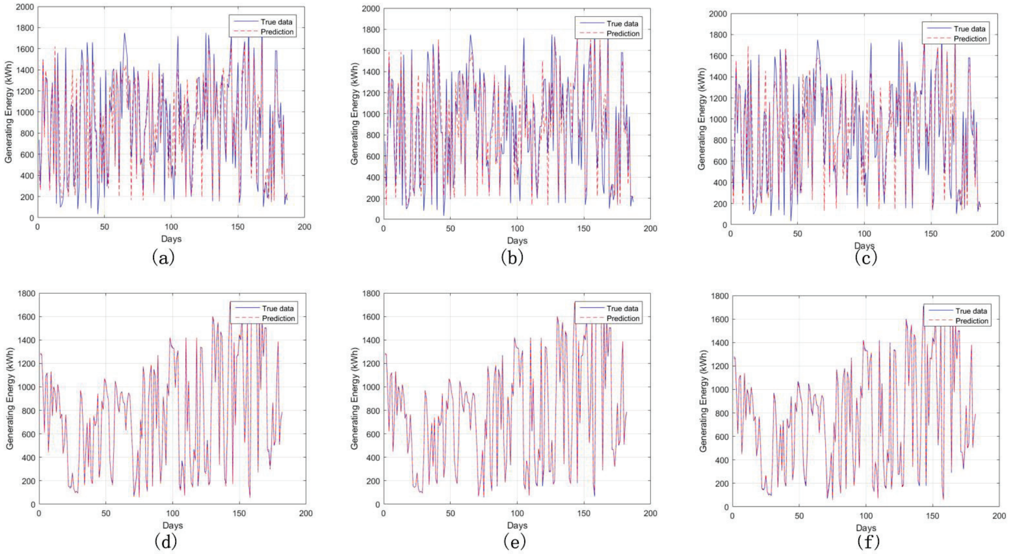


Figure 6. Prediction results of different algorithms. (**a**) PGPM based on SVR; (**b**) PGPM based on Decision Tree; (**c**) PGPM based on Random Forest; (**d**) PGPM based on LSTM; (**e**) PGPM based on Bi-LSTM; (**f**) Ours.

As shown in Figure 6, it can be found that the deviations between the true data and experimental results of PGPMs based on SVR, Decision Tree, and Random Forest were more obvious than that generated of PGPMs based on LSTM, Bi-LSTM, and Attention-Bi-LSTM. Summarily, the LSTM-based PGPMs are very suitable for power generation forecasting scenarios. However, according to Figure 6d–f, it can be seen that from the visualization point of view, the performance of Attention-Bi-LSTM PGPM proposed in this paper is basically the same as that of the other LSTM-based PGPMs. Therefore, to further illustrate the advantages of the proposed PGPM, this paper evaluates the performance of above-mentioned PGPMs from a quantitative perspective.

Besides, in the training procedure of the proposed PGPM, the model converges very quickly, as presented in Figure 7.

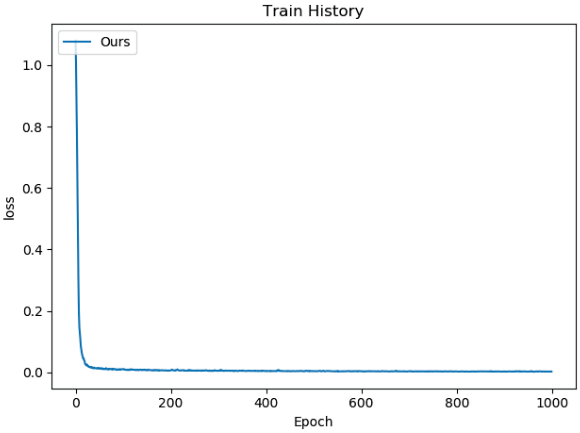


Figure 7. Convergence curve of the proposed PGPM.

As Figure 7 shows, the loss function of the model decreased quickly, and converged nearly to zero within the first 100 epochs, which means in the practical training procedure it could be finished very fast.

5.3. The Quantitative Comparison of Results

To evaluate the performances of above-mentioned PGPMs more precisely, Mean Absolute Error, Root of Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE) of each PGPM were evaluated and compared. Moreover, a R-square coefficient [26] is also introduced into the paper to calculate the fitting accuracy, which can be expressed by

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i| \tag{13}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2} \tag{14}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{\hat{y}_i - y_i}{y_i}\right| \times 100\% \tag{15}$$

$$R - square = \frac{\sum_{i=1}^{n}(\hat{y}_i - \overline{y}_i)^2}{\sum_{i=1}^{n}(y_i - \overline{y}_i)^2} \tag{16}$$

where y_i is the generating energy (true data) of the i-th sample; $\hat{y}_i$ is the prediction of the i-th sample; $R - square$ is a coefficient with a range of [0 1], and the closer this value is to 1, the higher the fitting accuracy.

According to Equations (13) to (16), the prediction errors and fitting accuracy of above-mentioned PGPMs are shown in Table 7.

Table 7. Comparison of different PGPMs.

Method	MAE (kWh)	RMSE (kWh)	MAPE (%)	Fitting Accuracy (R-Square)
SVR	166.7	238.9	40.7	0.7617
Decision Tree	160.3	236.0	37.9	0.7675
Random Forest	160.6	231.8	38.8	0.7591
LSTM	25.5	29.7	5.7	0.9959
Bi-LSTM	13.7	18.3	3.6	0.9984
Ours	10.2	8.6	2.8	0.9997

As Table 7 shows, the prediction errors of the proposed PGPM were 10.2 kWh, 8.6 kWh, and 2.8%, which were the smallest among these six algorithms. Moreover, from Table 6, taking RMSE as an example, it can be found that the prediction errors of the PGPMs based on SVR, Decision Tree, and Random Forest were 238.9 kWh, 236.0 kWh, and 231.8 kWh, respectively, which are generally more than 200 kWh, as well as that of the PGPMs based on LSTM and Bi-LSTM being less than 30 kWh. Hence, the performances of LSTM- and Bi-LSTM-based PGPM are better than that of SVR-, Decision Tree-, and Random Forest-based PGPMs. Simultaneously, with the introduction of the attention mechanism, the proposed PGPM also achieved better prediction accuracy than that of LSTM- and Bi-LSTM-based PGPMs. The metrics of MAE and MAPE showed similar results.

Additionally, the fitting accuracy was also evaluated in this paper. Fitting accuracy is another indicator for evaluating prediction efficiency, which represents the relative prediction error and can be used as a sign of the similarity between the predicted value and the true value. From Table 6, it can be found that the fitting accuracy of the proposed PGPM was 0.9997, slightly more than that based on LSTM and Bi-LSTM, but obviously more than that of SVR-, Decision Tree-, and Random Forest-based PGPMs. Therefore, in the metric of fitting accuracy, the proposed Attention-Bi-LSTM PGPM achieves the best performance, and is consequently very suitable for application in power generation forecasting scenarios.

5.4. Comparison of Multi-Step Prediction Results

Moreover, in order to evaluate the influence of proposed PGPM with the input time sequences of various step lengths, an experiment was also implemented based on different time steps, and the experimental results are shown in Table 7.

From Table 8, it can be found that there was a positive correlation between the prediction error and step size; in other words, the prediction error increased with respect to step length increases. Synchronously, the fitting accuracy had a negative correlation with step length, that is, the fitting accuracy decreased as the step length increased. The reason for the above phenomenon is that the dependence between the power generation and time sequences is weakened with the increase of step length. In summary, when the time step of input time sequences is four, the PGPM proposed in this paper can meet the demand for power generation forecasting.

Table 8. Comparison of multi-step prediction results.

Time Step	Evaluation Criteria	
	Prediction Error (kWh)	**Fitting Accuracy (R-Square)**
4	8.6408	0.9997
8	15.2754	0.9989
10	18.0235	0.9985
14	23.8192	0.9974

6. Conclusions

The contribution of this paper was to propose a generating energy prediction model based on the attention mechanism and Bi-LSTM, which improve the prediction accuracy, and the experimental results showed that the performance of the proposed PGPM is much better than that of PGPMs based on SVR, Decision Tree, Random Forest, LSTM, and Bi-LSTM. The challenge of this work was how to employ attention mechanism efficiently. To solve this, feature attention layer and temporal attention layer were introduced to enhance the prediction performance, because these attention layers could help the algorithm to utilize the most important features and the most critical moments.

Moreover, compared with the existing PGPMs, this paper mines the correlation of environmental factors that affect photovoltaic power generation before implementing the proposed PGPM, and thereby the calculation efficiency can be improved by eliminating environmental factors that are weakly related to power generation.

However, the data features of the proposed PGPM are few, and only the meteorological factors are considered as the input source. In the future, to further optimize the accuracy of the prediction method, other data features can be introduced to construct a more accurate input source.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/electronics11121885/s1; also in ftp://simitPublic:Simit123@47.116.99.105 (accessed on 12 June 2022), the generation data from a power station in Suzhou, China.

Author Contributions: Conceptualization, B.H., R.M. and X.Z.; data curation, W.Z.; formal analysis, R.M. and W.Z.; investigation, B.H.; methodology, R.M. and W.Z.; project administration, B.H.; resources, B.H.; software, W.Z.; supervision, J.Z.; validation, J.Z. and X.Z.; visualization, W.Z.; writing—original draft, R.M. and W.Z.; writing—review and editing, B.H. and R.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: Thanks are due to Wujiang photovoltaic power station for assistance with generation and environmental data.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kalkal, P.; Teja, A.V.R. A Sustainable Business Framework Using Solar and Bio-Energy to Instate Incessant Power in Rural India: Optimal Scheduling, Smart Metering, and Economic Viability. *IEEE Access* **2022**, *10*, 11021–11035. [CrossRef]
2. Glavan, M.; Gradisar, D.; Humar, I.; Vrancic, D. Refrigeration Control Algorithm for Managing Supermarket's Overall Peak Power Demand. *IEEE Trans. Control Syst. Technol.* **2019**, *27*, 2279–2286. [CrossRef]
3. Zheng, L.; Wenbo, X.; Kui, W.; Dawei, Y.; Jin, Z.; Yuanyuan, L.; Jia, S.; Chengdi, D.; Zhongyi, L.; Lu, L. Research on power demand forecasting based on the relationship Between economic development and power demand. In Proceedings of the 2018 China International Conference on Electricity Distribution (CICED), Tianjin, China, 17–19 September 2018; pp. 2710–2713. [CrossRef]
4. Jia, Y.; Wan, C.; Cui, W.; Song, Y.; Ju, P. Peer-to-Peer Energy Trading Using Prediction Intervals of Renewable Energy Generation. *IEEE Trans. Smart Grid* **2022**. [CrossRef]
5. Chen, H.; Xuan, P.; Wang, Y.; Tan, K.; Jin, X. Key Technologies for Integration of Multitype Renewable Energy Sources—Research on Multi-Timeframe Robust Scheduling/Dispatch. *IEEE Trans. Smart Grid* **2016**, *7*, 471–480. [CrossRef]
6. Eskandarpour, R.; Khodaei, A. Machine Learning Based Power Grid Outage Prediction in Response to Extreme Events. *IEEE Trans. Power Syst.* **2017**, *32*, 3315–3316. [CrossRef]
7. Tao, L.; Rongfa, Q.; Zhenjiang, G.; Zhongchuang, S.; Guohu, X.; Qi, G.; Zheng, X.; Hongyu, J. Research on Power System Active Power Scheduling Strategy Adapted to Large-Scale Wind Power Grid-Connected. In Proceedings of the 2018 3rd International Conference on Smart City and Systems Engineering (ICSCSE), Xiamen, China, 29–30 December 2018; pp. 482–486. [CrossRef]
8. Makkulau, A.; Christiono; Samsurizal. Characteristics of Temperature Changes Measurement on Photovoltaic Surfaces Against Quality of Output Current on Solar Power Plants. In Proceedings of the 2019 International Conference on Technologies and Policies in Electric Power & Energy, Yogyakarta, Indonesia, 21–22 October 2019; pp. 1–4. [CrossRef]
9. L'Heureux, A.; Grolinger, K.; Elyamany, H.F.; Capretz, M.A.M. Machine Learning with Big Data: Challenges and Approaches. *IEEE Access* **2017**, *5*, 7776–7797. [CrossRef]
10. Meng, L.; Shi, J.-W.; Wang, H.; Wen, X.-Q. SVM with improved grid search and its application to wind power prediction. In Proceedings of the 2013 International Conference on Machine Learning and Cybernetics, Tianjin, China, 14–17 July 2013; Volume 2, pp. 603–609. [CrossRef]
11. Mukherjee, R.; De, A. Development of an Ensemble Decision Tree-Based Power System Dynamic Security State Predictor. *IEEE Syst. J.* **2020**, *14*, 3836–3843. [CrossRef]
12. Chen, P.; Cheng, H.; Yao, Y.; Li, X.; Zhang, J.; Yang, Z. Research on Medium-Long Term Power Load Forecasting Method Based on Load Decomposition and Big Data Technology. In Proceedings of the 2018 International Conference on Smart Grid and Electrical Automation (ICSGEA), Changsha, China, 9–10 June 2018; pp. 50–54. [CrossRef]
13. Ullah, I.; Fayaz, M.; Naveed, N.; Kim, D. ANN Based Learning to Kalman Filter Algorithm for Indoor Environment Prediction in Smart Greenhouse. *IEEE Access* **2020**, *8*, 159371–159388. [CrossRef]
14. Al-Dahidi, S.; Ayadi, O.; Alrbai, M.; Adeeb, J. Ensemble Approach of Optimized Artificial Neural Networks for Solar Photovoltaic Power Prediction. *IEEE Access* **2019**, *7*, 81741–81758. [CrossRef]
15. Fan, M.; Si, Z.; Xie, X.; Liu, Y.; Liu, T. Text Backdoor Detection Using an Interpretable RNN Abstract Model. *IEEE Trans. Inf. Forensics Secur.* **2021**, *16*, 4117–4132. [CrossRef]
16. Dai, S.; Li, L.; Li, Z. Modeling Vehicle Interactions via Modified LSTM Models for Trajectory Prediction. *IEEE Access* **2019**, *7*, 38287–38296. [CrossRef]
17. Moharm, K.; Eltahan, M.; Elsaadany, E. Wind Speed Forecast using LSTM and Bi-LSTM Algorithms over Gabal El-Zayt Wind Farm. In Proceedings of the 2020 International Conference on Smart Grids and Energy Systems (SGES), Perth, Australia, 23–26 November 2020; pp. 922–927. [CrossRef]
18. Zhou, B.; Ma, X.; Luo, Y.; Yang, D. Wind Power Prediction Based on LSTM Networks and Nonparametric Kernel Density Estimation. *IEEE Access* **2019**, *7*, 165279–165292. [CrossRef]
19. Fang, Z.; Huang, Y.; Chen, X.; Gong, K.; Zhou, H. Identification of Abnormal Electricity Consumption Behavior Based on Bi-LSTM Recurrent Neural Network. In Proceedings of the 2019 4th International Conference on Power and Renewable Energy (ICPRE), Chengdu, China, 21–23 September 2019; pp. 342–346. [CrossRef]
20. Zhenya, Z.; Tingting, P.; Xia, Y.; Mengfei, Y.; Yan, Y.; Zhijun, L. Correlation between Meteorological Factors and Water Temperature in Yibin Section of Lower Jinsha River under Influence of Human Activities. *Express Water Resour. Hydropower Inf. (EWRHI)* **2021**, *42*, 12–17. [CrossRef]
21. Liu, D.; Xu, Q.; Tang, Y.; Jian, Y. Prediction of Water Inrush in Long-Lasting Shutdown Karst Tunnels Based on the HGWO-SVR Model. *IEEE Access* **2021**, *9*, 6368–6378. [CrossRef]

22. Wang, T.; Bi, T.; Wang, H.; Liu, J. Decision tree based online stability assessment scheme for power systems with renewable generations. *CSEE J. Power Energy Syst.* **2015**, *1*, 53–61. [CrossRef]
23. Gupta, V.K.; Gupta, A.; Kumar, D.; Sardana, A. Prediction of COVID-19 confirmed, death, and cured cases in India using random forest model. *Big Data Min. Anal.* **2021**, *4*, 116–123. [CrossRef]
24. Fang, W.; Jiang, J.; Lu, S.; Gong, Y.; Tao, Y.; Tang, Y.; Yan, P.; Luo, H.; Liu, J. A LSTM Algorithm Estimating Pseudo Measurements for Aiding INS during GNSS Signal Outages. *Remote Sens.* **2020**, *12*, 256. [CrossRef]
25. Hao, Z.; Hu, C.; Du, D.; Pei, H.; Zhang, J. Remaining Useful Life Prediction Method of Lithium-Ion Battery Based on Bi-LSTM Network under Multi-State Influence. *Acta Electron. Sin.* **2021**, *50*, 619–624.
26. Gouda, O.E.; El-Hoshy, S.H. Diagnostic technique for analysing the internal faults within power transformers based on sweep frequency response using adjusted R-square methodology. *IET Sci. Meas. Technol.* **2021**, *14*, 1057–1068. [CrossRef]

applied sciences

MDPI

Article

Feature Selection to Predict LED Light Energy Consumption with Specific Light Recipes in Closed Plant Production Systems

Martín Montes Rivera [1,*], Nivia Escalante-Garcia [2,*], José Alonso Dena-Aguilar [3], Ernesto Olvera-Gonzalez [2] and Paulino Vacas-Jacques [3]

[1] Dirección de Posgrados e Investigación, Universidad Politécnica de Aguascalientes, Calle Paseo San Gerardo #201, Fracc. San Gerardo, Aguascalientes 20342, Mexico

[2] Laboratorio de Iluminación Artificial, Tecnológico Nacional de México/IT de Pabellón de Arteaga, Carretera a la Estación de Rincón Km. 1, Aguascalientes 20670, Mexico; jose.og@pabellon.tecnm.mx

[3] Departamento de Ingenierías, Tecnológico Nacional de México/IT de Pabellón de Arteaga, Carretera a la Estación de Rincón Km. 1, Aguascalientes 20670, Mexico; jose.da@pabellon.tecnm.mx (J.A.D.-A.); paulino.vj@pabellon.tecnm.mx (P.V.-J.)

* Correspondence: martin.montes@upa.edu.mx (M.M.R.); nivia.eg@pabellon.tecnm.mx (N.E.-G.)

Abstract: The use of closed growth environments, such as greenhouses, plant factories, and vertical farms, represents a sustainable alternative for fresh food production. Closed plant production systems (CPPSs) allow growing of any plant variety, no matter the year’s season. Artificial lighting plays an essential role in CPPSs as it promotes growth by providing optimal conditions for plant development. Nevertheless, it is a model with a high demand for electricity, which is required for artificial radiation systems to enhance the developing plants. A high percentage (40% to 50%) of the costs in CPPSs point to artificial lighting systems. Due to this, lighting strategies are essential to improve sustainability and profitability in closed plant production systems. However, no tools have been applied in the literature to contribute to energy savings in LED-type artificial radiation systems through the configuration of light recipes (wavelengths combination. For CPPS to be cost-effective and sustainable, a pre-evaluation of energy consumption for plant cultivation must consider. Artificial intelligence (AI) methods integrated into the prediction crucial variables such as each input-variable light color or specific wavelengths like red, green, blue, and white along with light intensity (quantity), frequency (pulsed light), and duty cycle. This paper focuses on the feature-selection stage, in which a regression model is trained to predict energy consumption in LED lights with specific light recipes in CPPSs. This stage is critical because it identifies the most representative features for training the model, and the other stages depend on it. These tools can enable further in-depth analysis of the energy savings that can be obtained with light recipes and pulsed and continuous operation light modes in artificial LED lighting systems.

Keywords: light wavelength; energy efficiency; features selection; machine learning

Citation: Montes Rivera, M.; Escalante-Garcia, N.; Dena-Aguilar, J.A.; Olvera-Gonzalez, E.; Vacas-Jacques, P. Feature Selection to Predict LED Light Energy Consumption with Specific Light Recipes in Closed Plant Production Systems. *Appl. Sci.* **2022**, *12*, 5901. https://doi.org/10.3390/app12125901

Academic Editors: Luis Hernández-Callejo, Sergio Nesmachnow and Sara Gallardo Saavedra

Received: 3 May 2022
Accepted: 26 May 2022
Published: 9 June 2022

1. Introduction

1.1. LED Lights in Closed Plant Production Systems

Agriculture in 2050 will have to produce almost 50% more output to meet the demand for food supplies, presenting it with a crucial challenge in meeting the increase in demand [1]. Technological development and innovation can offer alternatives to ensure food security sustainably. The use of closed growth environments, such as greenhouses, plant factories, and vertical farms [2–6], represents a sustainable alternative for fresh food production. In closed plant production systems (CPPSs), several variables can be controlled and optimized, such as water, fertilizers, CO_2 injection, and temperature, as well as the quantity and quality of light thus ensuring minimum greenhouse gas emissions [3]. CPPSs allow growing of any plant variety, no matter the season of the year. Artificial lighting plays an essential role in CPPSs, as it promotes growth by providing optimal conditions for plant development. LEDs are energy-efficient replacements that contribute to plant growth in agriculture. An outstanding advantage of LED lamps is their ability to operate with specific wavelengths (λ) that considerably reduce energy consumption. LEDs regularly generate continuous light. Likewise, they can radiate pulsed light (on/off in microseconds (µs)) with high power and low energy consumption at a specific frequency and duty cycle without upsetting the vegetative development of plants [4,7,8]. LED technology can produce different colors of light—that is, different qualities—called light recipes (different wavelength combinations). The wavelength combinations (red, blue, green, ultraviolet, and infrared) and the photosynthetic photon flux density (PPFD, given in µmol m^{-2} s^{-1}) are the components that constitute the light recipes. Light recipes impact crop growth from branching to flowering; optimize the biomass; and increase the antioxidant capacity levels of calcium, potassium, magnesium, chlorophyll, iron, vitamins A, B, and E, and other substances [7–9]. Crop quality and productivity rely upon the time and the light quantity supplied to the plants.

CPPSs can offer several advantages (improved management control of all variables involved—temperature, CO_2, radiation—and increased productivity, growth, and yield) and generate an impact on humanity. Nevertheless, it is a model with a high demand for electricity for the artificial radiation systems needed to enhance the developing plants. Environmental control (refrigeration), the air required to remove the heat produced, and artificial lighting account for approximately 32%, 11%, and 57% of the total energy demand, respectively [10]. Furthermore, according to Avgoustaki and Xydis [11], the artificial lighting system accounts for 80% of the electrical demand, since the overall operability of the CPPS accounts for 40% of the total energy consumption.

Innovative approaches, such as fluid dynamics, evolutionary algorithms [12,13], the derivative integral model, and derivative model [14–16], control the resources in CPPSs. Artificial neural networks predict weather conditions and energy consumption [13–15,17]. Other techniques predict energy consumption performance for plant production [18,19]. Finally, other techniques focus on in the optimization of resources and reducing energy demand in CPPSs [20,21].

1.2. Machine-Learning Modeling

Physical modeling approaches are the most common approaches for predicting system behaviors, but they rely on descriptions of physics concepts. Thus, they tend to be complex, as the detail of the model increased. Therefore, as the principle of Occam's razor states, physical modeling must balance complexity with assumptions in order to produce simplified and representative models [22,23].

On the other hand, artificial intelligence (AI) researchers have proposed several techniques that allow automatic generation of the models and equations based on measurements arranged in datasets. Furthermore, machine learning (ML), a field of AI, applies deterministic and heuristic methods to produce models with less complexity established in the raw measurements [22].

During the last two decades, ML models have exhibited high effectivity, accuracy, and performance in several fields, including energy applications. Furthermore, ML results for modeling have motivated researchers to apply its models to accurately predict the behavior of physical phenomena [22–30].

The ML modeling process can involve several stages, depending on its application, but a general description would include collecting data, preprocessing data, building a model, training, and testing. Furthermore, all the stages must be continually tuned to improve the results; i.e., the stages can repeatedly change across the entire process if the model requires efficiency improvements, as represented in Figure 1 [22].

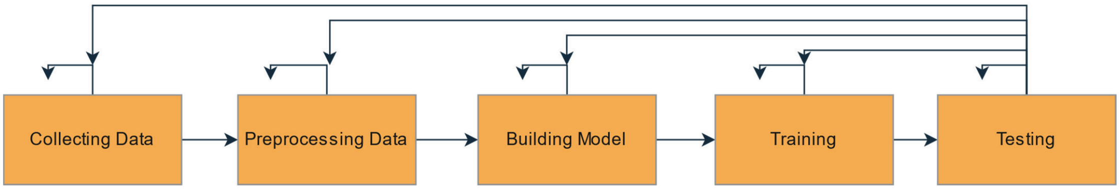


Figure 1. Modeling process with ML.

1.2.1. Collecting Data

ML modeling uses algorithms, statistics, and measurements structured in a dataset to identify the process behaviors and mimic them in a model [31]. The data generation stage depends on the processes contained in the chosen model. They may include electrical, mechanical, optical, thermal, psychic, or chemical variables [22,24,25,32,33]; derive from statistical analysis [26,27]; or be constructed with text, multimedia, or even real-time reports [32–35]. Nevertheless, the datasets can be associated with a specific time and/or frequency domain [36,37].

1.2.2. Preprocessing Data

After collecting and structuring the dataset, its variables need to be cleaned, processed, and filtered for the ML model. The processing stage includes several techniques, which can be human- or AI-designed, and they depend on the nature of the training data. For example, in natural language processing with text, preprocessing removes capitals [38]; in signal processing, wavelet transforms separate signals into their main components [39]; in image processing, convolution with the image filters extracts features [40]; in big data and data mining, dimensionality reduction is achieved [41].

The preprocessing data stages include normalization based on algorithms, such as MIN-MAX normalization, decimal scaling, and Z-scores; filtering redundant and inconsistent data; transformations such as linear, quadratic, polynomial, and histogram transformations; removing noisy data with techniques such as ensemble filtering, cross-validated filtering, and interactive partitioning; feature selection with exhaustive, heuristic, filter, and wrapper methods; and discretization to change from analog systems to digital ones [42].

Input features in ML modeling are representative when their information affects the output of the modeled system. Additionally, removing characteristics that are irrelevant or have low correlations from the results produces search spaces with lower complexity, boosting the capabilities of the training algorithm and improving the final model's efficiency [43,44].

One of the most used commonly techniques for removing redundant and inconsistent data in the second stage is feature Selection (FS). FS also makes it possible to reduce size, increase the efficiency and accuracy of predictive learning, and reduce the complexity of the final model [42]. The different FS approaches reported in the literature are constituted theoretically and apply methods such as filtering, wrapping, and embedding through techniques involving search algorithms, statistical criteria, and information, distance, dependency, and consistency measures [42].

1.2.3. Building Model

ML includes several models for predicting behavior that are supported by statistics and artificial intelligence. Different proposals have obtained different results depending on the ML model's application. The most common models are artificial neural networks, evolutionary algorithms, swarm intelligence algorithms, decision trees, naive Bayesian algorithms, logistic regression, fuzzy systems, gradient boosting machines, support vector machines, support vector regression, random forest algorithms, AdaBoost, simulated annealing, and hybrids of these models [22,24,26–28,31,34,44].

1.2.4. Training Model

Each ML model tunes its internal parameters with a training algorithm designed for the learning type. The most common learning types are supervised, unsupervised, reinforced, semi-supervised, transductive, self-trained, ensemble learning, boosting, and generative [31].

1.2.5. Testing Model

The metrics used to evaluate quality in the process of ML modeling depend on the nature of the model, which may be for classification or regression. In regression models, the metrics quantify the reliability of the model and the error between the model output and the real-world system. The most common regression metrics are the root mean square error (RMSE), mean error (ME), mean absolute error (MAE), mean average percentage error (MAPE), and the Nash coefficients E and R^2 [22,44].

1.3. Feature Selection

As mentioned in Section 1.2.2, feature selection is one of the most critical stages of ML modeling since it makes it possible to identify the best relation to the required complexity of the model and its quality at the preprocessing stage. In addition, feature selection makes it possible to find the more representative inputs in the real-world system and to eliminate no representative inputs or those that are redundant. ML models and training algorithms that consider only representative features improve their efficiency and reduce the time required for training [26,27,38,45]. A feature is an observable property in a system. Feature selection aims to select a specific subset of features that maximize the performance of the ML model.

The feature selection (FS) used here applied one of the most common techniques for removing irrelevant data, reducing dimensionality, increasing predictive accuracy and learning efficiency, and reducing the complexity of the final model [42]. Although there are different approaches for FS, all have theoretical support in their use of different methods, such as filtering, wrapping, and embedding, and involve techniques that use search algorithms, statistical criteria, and information, distance, dependence, and consistency measures [42]. The aim was to use linear and nonlinear methods to implement FS with a dataset acquired from an illumination radiation system.

This paper focuses on the feature selection stage in order to train a regression model to predict energy consumption in LED lights with specific light recipes in CPPSs. This stage is critical because it identifies the most representative features for training the model, and the other stages depend on it. These tools can enable further in-depth analysis of the energy savings that can be obtained with light recipes and pulsed and continuous light operation modes in artificial LED lighting systems.

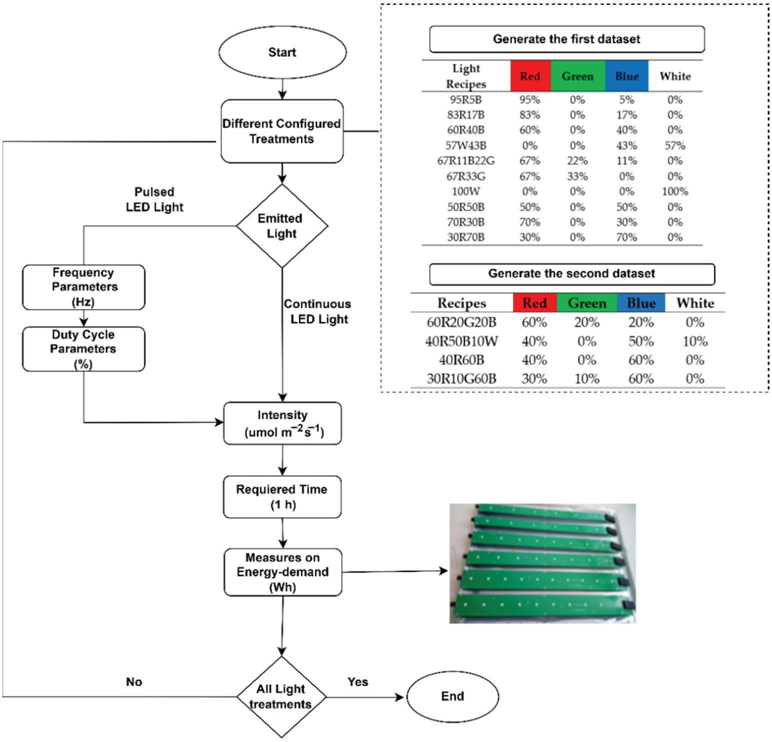
2. Materials and Methods

2.1. Lighting System Features

The Artificial Lighting Laboratory (LIA) at Instituto Tecnológico de Pabellón de Arteaga in Aguascalientes, Mexico, developed the lighting system. An array of eight lamps formed the artificial lighting system. The wattage of each lamp was 25 watts. The ultra-bright LEDs emitted continuous and pulsed irradiation with different qualities (red, blue, green, and white). A programmed controller (a field-programmable gate array (FPGA)) allowed us to configure functions such as pulse frequency, duty cycle, intensity, wavelength, and on–off time.

2.2. Construction of Experiment

This study evaluated 10 light recipes from the literature, as can be seen in Figure 2. After that, the LED artificial radiation system was configured for continuous and pulsed emission to generate the first dataset (see table at the top of figure) to be analyzed. The intensities parameters were 50, 65, 80, 95, 110, 125, 140, 155, 170, and 185 µmol m^{-2} s^{-1}, as determined by a quantum sensor, and the frequency was set to 100, 500, and 1000 Hz with 40%, 50%, 60%, 70%, 80%, and 90% duty cycles for different treatments.


Start
Different Configured Treatments
Pulsed LED Light
Emitted Light
Frequency Parameters (Hz)
Duty Cycle Parameters (%)
Continuous LED Light
Intensity (umol m^{-2}s^{-1})
Requiered Time (1 h)
Measures on Energy-demand (Wh)
No
All Light treatments
Yes
End

Generate the first dataset

Light Recipes	Red	Green	Blue	White
95R5B	95%	0%	5%	0%
83R17B	83%	0%	17%	0%
60R40B	60%	0%	40%	0%
57W43B	0%	0%	43%	57%
67R11B22G	67%	22%	11%	0%
67R33G	67%	33%	0%	0%
100W	0%	0%	0%	100%
50R50B	50%	0%	50%	0%
70R30B	70%	0%	30%	0%
30R70B	30%	0%	70%	0%

Generate the second dataset

Recipes	Red	Green	Blue	White
60R20G20B	60%	20%	20%	0%
40R50B10W	40%	0%	50%	10%
40R60B	40%	0%	60%	0%
30R10G60B	30%	10%	60%	0%

Figure 2. Flow diagram for collection of CPPS measurements for the dataset.

In the generation of the second dataset, four different light recipes were set at intensities of 60, 70, 85, 90, 90, 100, 120, 130, 150, 160, and 180 µmol m^{-2} s^{-1}, the frequency was set at 100, 500, and 1000 Hz, and duty cycles were randomly selected at 60%, 70%, and 80%, depending on the treatment (see table at the bottom of Figure 2).

The artificial illumination system included 14 light recipes (see tables in Figure 2) with all combinations of parameters. After 60 min of radiation, we registered the measurements for the energy demand with a hook-on AC ammeter (Peak Teach, Salerno, Italy) in watts × hours (Wh). Then, the artificial radiation system was turned off for 15 min to cool down.

2.3. Min-Max Normalization

Normalization linearly transforms variables within specific ranges based on the minimum and maximum median absolute deviations of the variable values, avoiding changes to priorities in the variables because of the scale. Equation (1) represents the standard deviation required in the transformation as X_{std}, and Equation (2) indicates the variable scaling [46,47].

$$X_{std} = \frac{x - X_{min}}{X_{max} - X_{min}} \tag{1}$$

$$X_{scaled} = X_{std} \times (X_{max} - X_{min}) + X_{min} \tag{2}$$

where X_{scaled} is the new value transformed from the original value $x \in X$ and X_{max} and X_{min} are the maximum and minimum values, respectively.

2.4. Pearson Correlation

This association method's primary goal is to identify two or more correlated variables [45].

The Pearson correlation coefficient measures the degree of correlation between two variables in a linear approach. Let X and Y be those variables, with measurements given by $\{x_1, x_2, x_3, \ldots, x_n\}$ and $\{y_1, y_2, y_3, \ldots, y_n\}$ and means $\bar{x}$ and $\bar{y}$. Then, the Pearson coefficient is given by Equation (3) [42].

$$\rho(X,Y) = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\left[\sum_{i=1}^{n} (x_i - \bar{x})^2 \sum_{i=1}^{n} (y_i - \bar{y})^2\right]^{\frac{1}{2}}} \tag{3}$$

A Pearson coefficient with the range $\rho = [-1, 1]$ represents the level of correlation when ρ is positive and correlation is direct, and the negative is the inverse [42].

When two variables are highly correlated, one can be redundant. The Pearson correlation works only for linear relations and results in incorrectly measured correlations for nonlinear systems. When classifying with binary outputs, it is possible to identify using Pearson coefficients how an attribute correlates with the target class [42].

Additionally, one can perform a correlation statistical significance test using the p_{value} coefficient, such as a test of the probability that the correlation coefficient ρ is a wrong hypothesis; for example, as a convention from the literature, if $p_{value} > 0.05$, it is unreliable. The alternatives for such a determination include statistical tests, such as the t_{value}, variance analysis (ANOVA), and 1_{tailed} or 2_{tailed} tests [48].

2.5. Variance Threshold

This method is used to identify features with variance. The features eliminated based on variance are those with zero value, near to zero value, or below a specific threshold [49–51].

The variability in a group given with $\{x_1, x_2, x_3, \ldots, x_n\}$ is the standard error; in other words, it is the difference between the samples and the average value of the group $\bar{x}$, as in Equation (4) [52].

$$\sigma^2 = \frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{n - 1} \tag{4}$$

2.6. Mutual Information Gain

Feature selection with mutual information gain enables the discrimination of features based on their interaction measurement, both for linear and nonlinear models [53]. Mutual information measures the uncertainty based on the entropy H of one variable while observing the other one. Let X be a random variable with values $\{x_1, x_2, x_3, \ldots, x_n\}$; its entropy is given by Equation (5) [54].

$$H(X) = -\sum_{i=1}^{n} P(x_i) log_2[P(x_i)] \tag{5}$$

Let Y be an output variable with values $\{y_1, y_2, y_3, \ldots, y_n\}$ and let X be a features array with values $\{x_1, x_2, x_3, \ldots, x_n\}$; $H(X|Y)$ is then given by Equation (6) [54].

$$H(X \mid Y) = -\sum_{j=1}^{n} [P(x_i)] \sum_{i=1}^{n} P(x_i \mid y_j) \log_2 [P(x_i \mid y_j)] \tag{6}$$

The mutual information in Equation (7) measures the reduction in the uncertainty of X given Y [54,55].

$$MI(X|Y) = H(X) - H(X|Y) \tag{7}$$

2.7. Univariate Linear F-Regression Selection

This method uses a linear model to measure the degree of linear dependence between two random variables; in other words, it measures the significance of a feature in a linear model [56].

The F-regression equations use the null hypothesis H_0, indicating that the data only intercept the model, and the alternative hypothesis H_1, indicating the compatibility of the data with the model. The selection of the true hypothesis relies on the F_{score} given in Equation (8), the explained variance from Equation (9), and the unexplained variance from Equation (10) [56].

$$F = \frac{\text{explained variance}}{\text{unexplained variance}} \tag{8}$$

$$\text{explained variance} = \sum_{i=1}^{K} n_i \frac{\left(\overline{Y}_{i\cdot} - \overline{Y}\right)^2}{(K-1)} \tag{9}$$

$$\text{unexplained variance} = \sum_{i=1}^{K} \sum_{j=1}^{n_i} \frac{\left(Y_{ij} - \overline{Y}_{i\cdot}\right)^2}{(N-K)} \tag{10}$$

where Y_{ij} is the j th observation in the i out group in K, which is the number of out groups. N is the overall sample size and n_i is the number of observations.

Additionally, following Section 2.4, one can determine a p_{value} for the hypothesis conclusion, and, like with the Pearson correlation, if $p_{value} > 0.05$, the conclusion is unreliable [56].

2.8. Sequential Feature Selection

Sequential feature selection algorithms are a subset of wrapper algorithms that use greedy search algorithms. They evaluate a solution with certain features in a specific model and decide which feature to remove based on its quality. This technique can use a feedforward or backward approach; i.e., adding or removing features in the model. Figure 3 displays the searching schema for feedforward and backward sequential selection with three features [57,58].

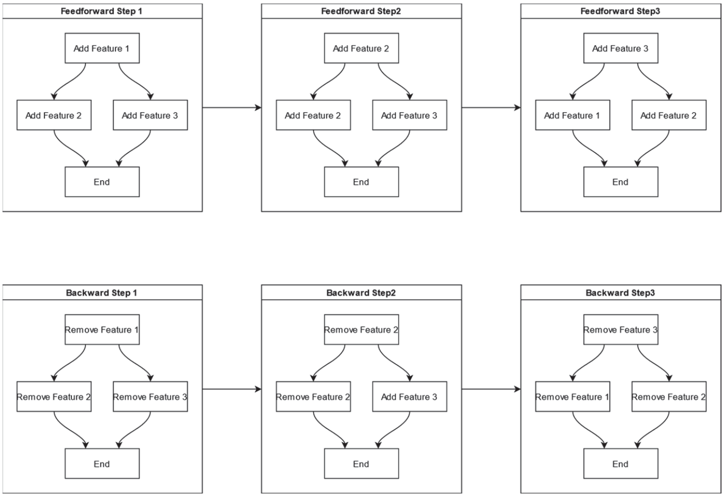


Figure 3. Flow diagram for feedforward and backward sequential feature selection.

For this study, backward sequential feature selection served to remove the worst variables in the energy consumption dataset for the LED lamps of a CPPS.

2.8.1. Linear Regression Model

The linear model structure includes $Y \in \mathbb{R}^{n\times 1}$, where $Y = (y_1, y_2, y_3, \ldots, y_n)^T$ is the response variable; $X \in \mathbb{R}^{n\times p}$, where $X = (x_1, x_2, x_3, \ldots, x_n)$ represents the design matrix; and $x_i = (x_{i,1}, x_{i,2}, x_{i,3}, \ldots, x_{i,p})$ and $\beta \in \mathbb{R}^{p\times 1}$, where n is the number of observations and p is similar to the number of features. Then, the linear regression model is given by Equation (11) [59].

$$Y = \mu + \epsilon \tag{11}$$

where $\mu = \beta \times X$ and ϵ is the regression error.

Then, with a given predictor Y and the design matrix X, Equation (12) solves the β model parameters that reduce ϵ [59].

$$\beta = \left(X^T X\right)^{-1} X^T Y \tag{12}$$

2.8.2. Decision Tree Regression Model

Decision trees are hierarchical structures with nodes representing tests of the data with specific attributes and branches representing the test results. Decision tree models include IDS, C4.5, CART, and regression models. For example, the regression decision trees predict continuous random variables by finding the attributes that reduce the mean square error (MSE), obtained with Equation (13) [60].

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \overline{y}_i)^2 \tag{13}$$

where $Y = (y_1, y_2, y_3, \ldots, y_n)$ is the raw data output variable and $\overline{Y} = (\overline{y}_1, \overline{y}_2, \overline{y}_3, \ldots, y_n)$ represents the decision tree model output [60].

For this application, the regression variable used a decision tree with the energy consumption and the node attributes as the features for the energy consumption dataset.

3. Results

3.1. Energy Consumption Dataset

We registered the power consumption emitted by the artificial lighting system as a function of the light recipe, including parameters such as intensity; R, G, B, and W quality; frequency; and duty cycle. Table 1 represents the first dataset obtained through the process described in Section 2 (Figure 2). The evaluated energy consumption contained different ranges depending on the directly configured parameters. However, applying specific value ranges to the inputs affects the priority assigned to each one.

Endeavoring not to affect the input priorities, a new scaled dataset with min-max normalization was generated according to the equations defined in Section 2.3. Table 2 shows the data obtained after applying the equations corresponding to each input and output variable. The data represent the ranges from 0 to 1 after normalization. A value of 0 corresponds to the minimum value identified for that variable, while 1 is the maximum.

Table 1. First 15 measurements of the dataset generated.

Intensity (A) (μmol m^{-2} s^{-1})	Light Color Percentage (%)				Frequency (Hz)	Duty Cycle (%)	Energy Consumption (Wh)
	R	G	B	W			
50	45	0	5	0	0	0	23.5
50	41.5	0	8.5	0	0	0	23.4
50	30	0	20	0	0	0	23.9
50	0	0	21.5	28.5	0	0	25.1
50	33.5	11	5.5	0	0	0	24.4
50	33.5	16.5	0	0	0	0	23.4
50	0	0	0	50	0	0	24.5
50	25	0	25	0	0	0	23.9
50	35	0	15	0	0	0	33.5
50	15	0	35	0	0	0	24.1
50	45	0	5	0	100	40	20.7
50	41.5	0	8.5	0	100	40	20.6
50	30	0	20	0	100	40	20.9
50	0	0	21.5	28.5	100	40	22.2
50	33.5	11	5.5	0	100	40	21.1

Table 2. First 15 scaled dataset measurements.

Intensity (A) (μmol m^{-2} s^{-1})	R	G	B	W	Frequency (Hz)	Duty (%)	Energy Consumption (Wh)
0.000	0.256	0.000	0.039	0.000	0.000	0.000	0.085
0.000	0.236	0.000	0.066	0.000	0.000	0.000	0.082
0.000	0.171	0.000	0.154	0.000	0.000	0.000	0.097
0.000	0.000	0.000	0.166	0.154	0.000	0.000	0.132
0.000	0.191	0.180	0.042	0.000	0.000	0.000	0.111
0.000	0.191	0.270	0.000	0.000	0.000	0.000	0.082
0.000	0.000	0.000	0.000	0.270	0.000	0.000	0.114
0.000	0.142	0.000	0.193	0.000	0.000	0.000	0.097
0.000	0.199	0.000	0.116	0.000	0.000	0.000	0.378
0.000	0.085	0.000	0.270	0.000	0.000	0.000	0.103
0.000	0.256	0.000	0.039	0.000	0.100	0.444	0.003
0.000	0.236	0.000	0.066	0.000	0.100	0.444	0.000
0.000	0.171	0.000	0.154	0.000	0.100	0.444	0.009
0.000	0.000	0.000	0.166	0.154	0.100	0.444	0.047
0.000	0.191	0.180	0.042	0.000	0.100	0.444	0.015

3.2. Person Correlation Results

The next step was determining the Pearson correlation level with the coefficient ρ supported by the statistic test p_{value} using the equations in Section 2.4. The evidence for Pearson correlation with intensity, R, G, B, W, and frequency was sufficient as $p_{value} \leq 0.05$, but the duty correlation was unreliable because $p_{value} > 0.05$. Thus, duty cycle was the first variable eliminated (Table 3).

Table 3. ρ and p_{value} correlation with energy consumption per input variable.

Elimination Order	Input	ρ	p_{value}
7th	Intensity	0.865312	0
3rd	R	0.091069	5.64×10^{-12}
2nd	G	0.043198	0.001106
5th	B	0.372963	1.3×10^{-187}
6th	W	0.522086	0
4th	Frequency	0.110005	8.18×10^{-17}
1st	Duty cycle	0.014195	0.283926

Figure 4 shows a correlation heat map of the input variables and the energy consumption output to identify the strongest correlations graphically. There is a lower correlation where the graph color is darker.

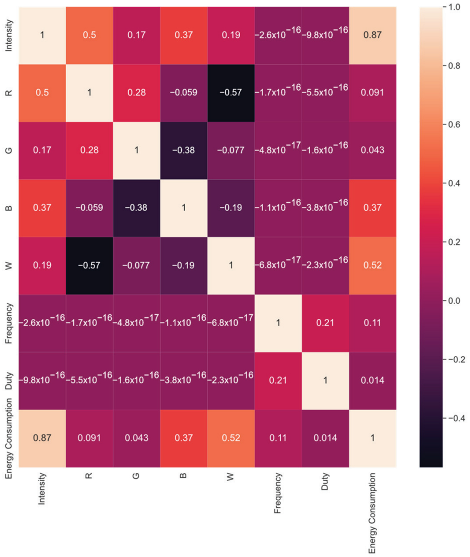


Figure 4. Pearson correlation heat map results.

3.3. Variance Threshold Results

The dataset energy consumption variables were dismissed against the variance threshold value as it gradually increased. The method for eliminating the variables with lower variance was described in Section 2.5. Figure 5 indicates the color associated with each variable in the variance threshold selection.

Table 4 displays the feature variance, the threshold value, and a bar plot showing the eliminated variable. Each threshold value was increased by 0.01 steps until a feature was eliminated from the energy consumption dataset.

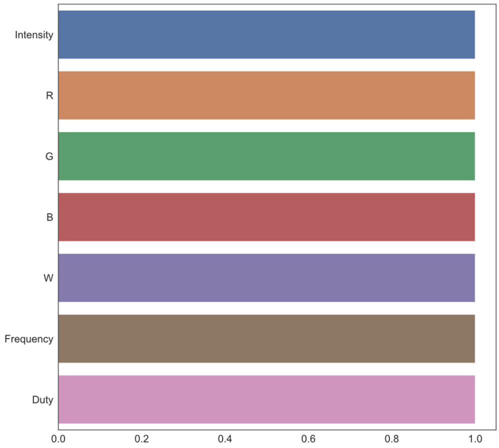


Figure 5. Colors per variable for variance threshold feature selection.

Table 4. Elimination order for features using variance threshold selection.

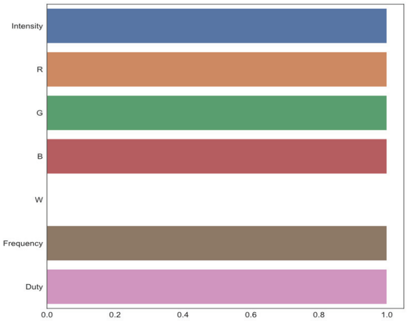
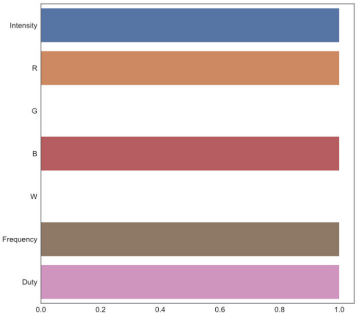
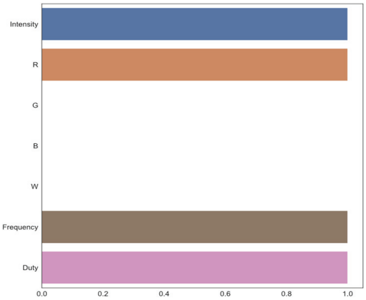
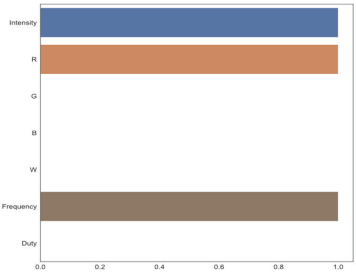
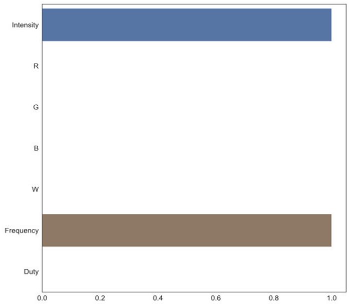
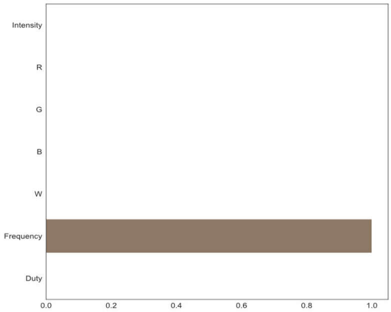
Elimination Order	Variable	Variance	Threshold	Image
1st	W	0.05079	0.051	
2nd	G	0.05490	0.055	
3rd	B	0.05546	0.056	
4th	Duty cycle	0.06012	0.061	

Table 4. *Cont.*

Elimination Order	Variable	Variance	Threshold	Image
5th	R	0.06479	0.065	
6th	Intensity	0.10185	0.110	
7th	Frequency	0.14260	N/A	N/A

3.4. Mutual Information Gain Results

Mutual information gain feature selection for the energy consumption dataset, set as the output variable, and all the other variables, assigned to the features array, was applied according to the description in Section 2.6. The results of this analysis are represented in Figure 6 and Table 5.

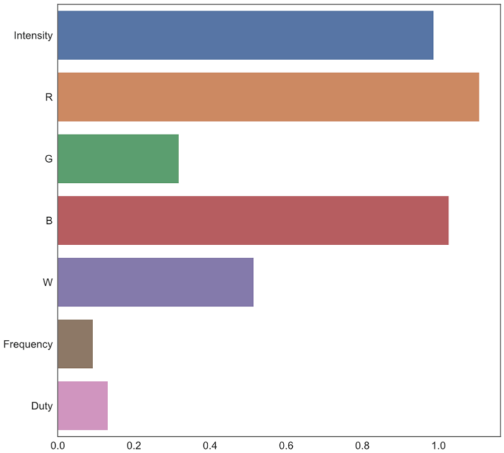


Figure 6. Mutual information evaluated versus the features in the dataset.

Table 5. Mutual information gain values for the energy consumption dataset.

Elimination Order	Input	MI(X∣Y)
5th	Intensity	0.987600
7th	R	1.107432
3rd	G	0.318185
6th	B	1.027326
4th	W	0.514607
1st	Frequency	0.092839
2nd	Duty	0.131858

3.5. Univariate Linear F-Regression Results

Once again, feature selection by F-regression in the generated dataset employed the energy consumption, such as the outcome variable in the features array (Section 2.7). Figure 7 displays the F_{score} value for hypothesis H_0 for each variable in the features array, and Table 6 shows the F_{score} and p_{value} calculated by H_0 for all input parameters.

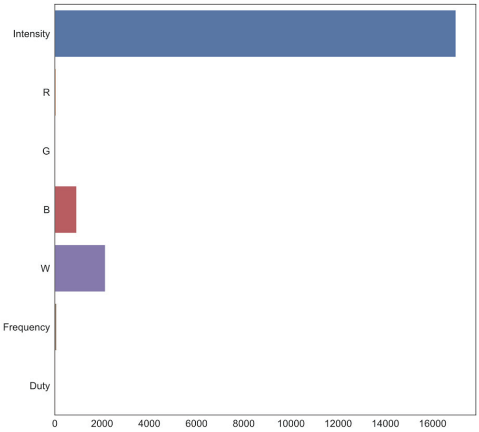


Figure 7. F_{score} comparison and the energy consumption dataset features.

Table 6. F_{score} and p_{value} feature selection with F-regression.

Elimination Order	Input	F_{score}	p_{value}
7th	Intensity	16,981.875086	0
3rd	R	47.651943	5.643556×10^{-12}
2nd	G	10.652646	1.105620×10^{-3}
5th	B	920.664903	$1.349950 \times 10^{-187}$
6th	W	2135.097576	0
4th	Frequency	69.796609	8.176545×10^{-17}
1st	Duty cycle	1.148417	2.839262×10^{-1}

The calculated p_{value} is indicated in Table 6. The Pearson correlations for intensity, R, G, B, W, and frequency showed reliable results since $p_{value} \leq 0.05$, but the duty cycle correlation was unreliable because $p_{value} > 0.05$; that is, the duty cycle was the first eliminated.

3.6. Sequential Feature Selection Results

We used backward sequential feature selection under a linear model and a decision tree regression (no linear model). However, implementation of the sequential feature selection through the decision tree may have generated overfitting; thus, a 10-fold cross-validation allowed the recognition of the accepted characteristics by modifying the tree depth from 2 to 5.

3.6.1. Sequential Feature Selection with Linear Regression Model

Table 7 presents the results obtained for the linear regression model (Section 2.8.1) determining the feature elimination sequence, which used as attributes the admitted range from 1 to 6. The table follows the logic of Figure 5, showing each color in the elimination ranking with the variance threshold.

Table 7. Sequential feature deletion from the linear regression.

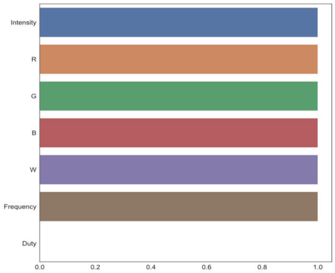
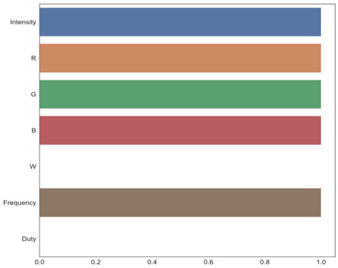
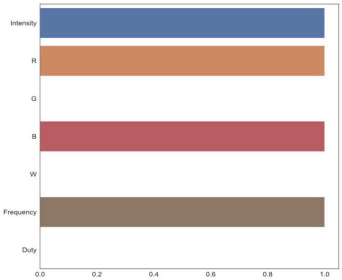
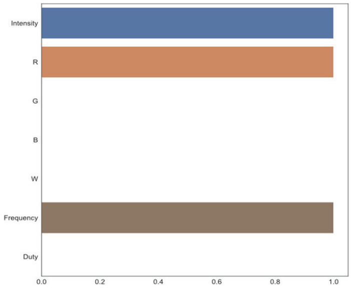
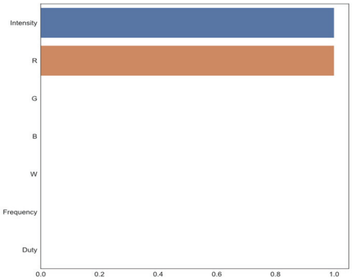
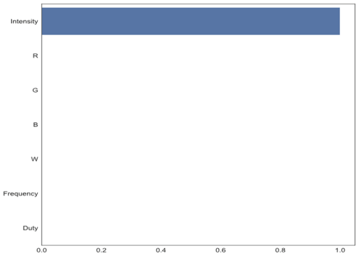
Elimination Order	Variable	Image
1st	Duty cycle	
2nd	W	
3rd	G	

Table 7. *Cont.*

Elimination Order	Variable	Image
4th	B	
5th	Frequency	
6th	R	
7th	Intensity	N/A

3.6.2. Sequential Feature Selection with Decision Tree Regression Model

The feature recognition through the decision tree regression model used the variance in a specific feature, which ranged from 1 to 6, allowing the elimination order for each input variable (Tables 8–11) to be obtained from the tree depth configuration. A color image of each feature, following the structure for variance threshold selection, is shown in Figure 5.

Table 8. Sequential feature selection by decision tree for depth = 2.

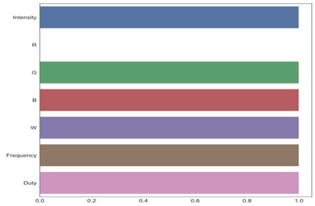
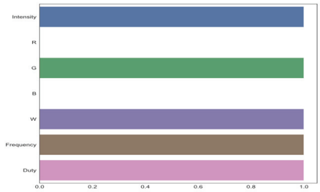
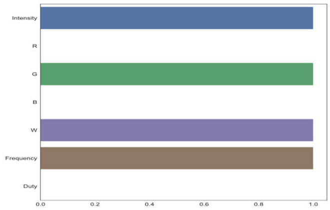
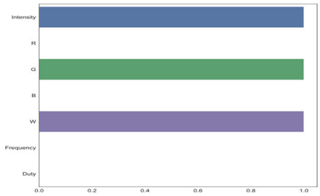
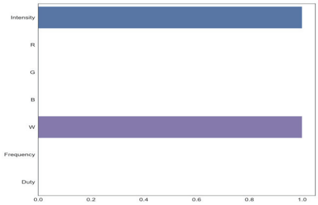
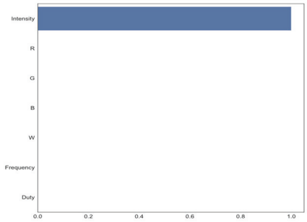
Elimination Order	Variable	Image
1st	R	
2nd	B	
3rd	Duty cycle	
4th	Frequency	
5th	G	
6th	W	
7th	Intensity	N/A

Table 9. Sequential feature selection by decision tree for depth = 3.

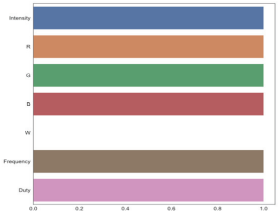
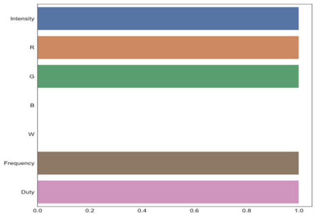
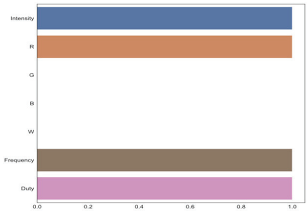
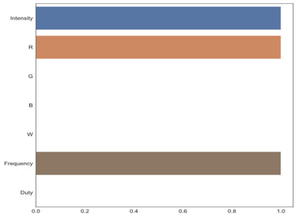
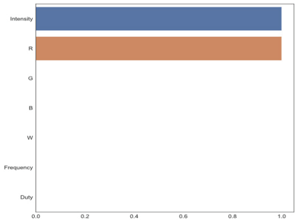
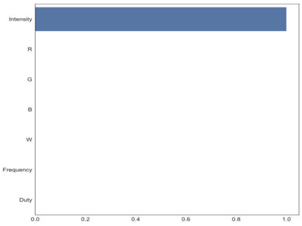
Elimination Order	Variable	Image
1st	W	
2nd	B	
3rd	G	
4th	Duty cycle	
5th	Frequency	
6th	R	
7th	Intensity	N/A

Table 10. Sequential feature selection by decision tree for depth = 4.

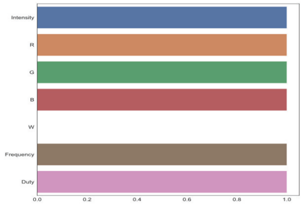
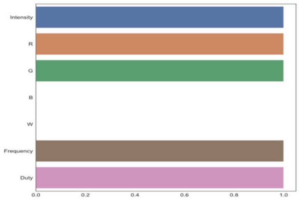
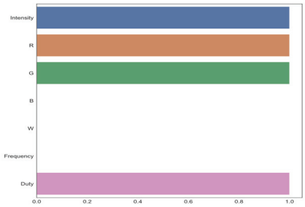
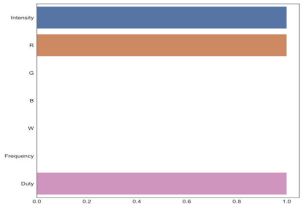
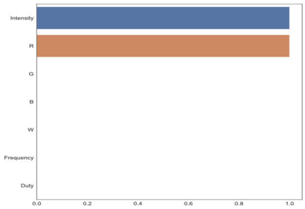
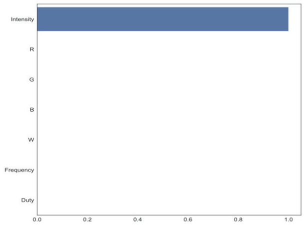
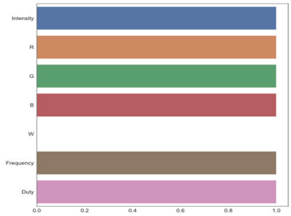
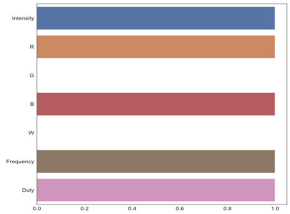
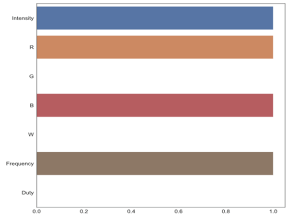
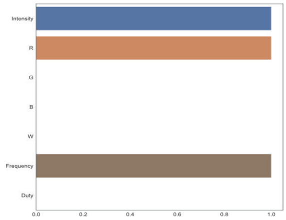
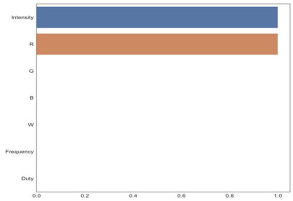
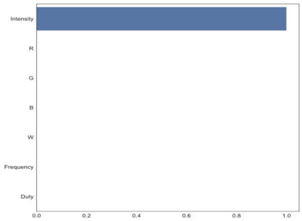
Elimination Order	Variable	Image
1st	W	
2nd	B	
3rd	Frequency	
4th	G	
5th	Duty cycle	
6th	R	
7th	Intensity	N/A

Table 11. Sequential feature selection by decision tree for depth = 5.

Elimination Order	Variable	Image
1st	W	
2nd	G	
3rd	Duty cycle	
4th	B	
5th	Frequency	
6th	R	
7th	Intensity	N/A

4. Discussion

The results obtained require division into linear and nonlinear model selection algorithms. The division generated makes it possible to analyze the results according to the model type and to identify the sequence of each feature. Tables 12 and 13 show the algorithms by group, the feature selection order, and the mean.

Table 12. Elimination order for the linear model group.

Feature	Pearson Correlation	Variance Threshold	Univariate Linear F-Regression	Sequential Backward Linear	Mean
Intensity	7	6	7	7	6.8
R	3	5	3	6	4
G	2	2	2	3	2.2
B	5	3	5	4	4.4
W	6	1	6	2	4.2
Frequency	4	7	4	5	4.8
Duty cycle	1	4	1	1	1.6

Table 13. Elimination order for the nonlinear model group.

Feature	Variance Threshold	Mutual Information Gain	Sequential Backward Deep Tree Values				Mean
			2	3	4	5	
Intensity	6	5	7	7	7	7	6.5
R	5	7	1	6	6	6	5.17
G	2	3	5	3	4	2	3.17
B	3	6	2	2	2	4	3.17
W	1	4	6	1	1	1	2.33
Frequency	7	1	4	5	3	5	4.17
Duty cycle	4	2	3	4	5	3	3.5

The averages calculated and reported in Tables 12 and 13 indicate two different behaviors depending on the model performance (linear or nonlinear). Figure 8 shows an alternative way to visualize the performance between linear and nonlinear models.

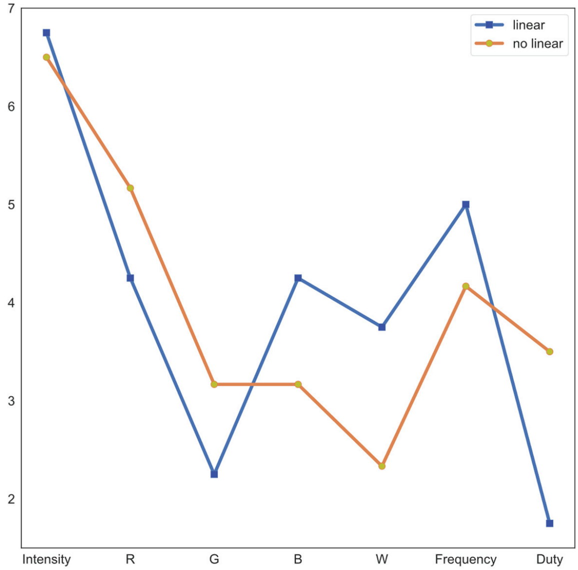


Figure 8. Feature elimination order distributions with algorithms from the two models used.

After dividing the models into linear and nonlinear groups, we validated the ordinal elimination variables on a scale from 1 to 7 and tested the distributions with the Kruskal–Wallis test (Table 14) [61]. The sequence elimination distribution for the proposed models is shown in Figures 9 and 10 (linear and nonlinear models, respectively).

Table 14. Values obtained with the Kruskal–Wallis test.

Group	F_{score}	p_{value}
Linear	16.27232	0.012364
Nonlinear	17.65278	0.007161

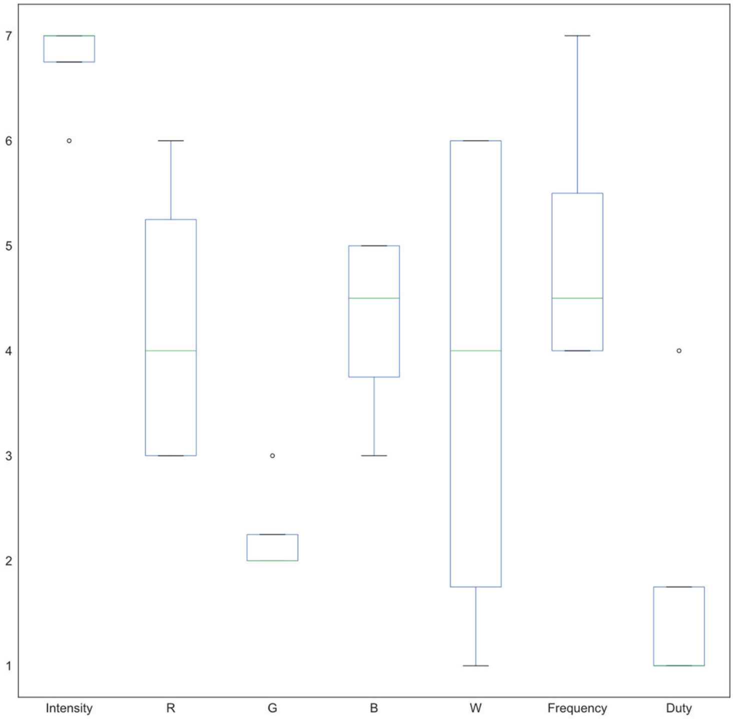


Figure 9. Order of elimination for features in linear models.

The linear model indicated that the essential characteristic was intensity, while the least significant was the duty cycle (Figure 9 and Table 14). If the appropriate sequence for any variable is required, the mean value can be found in Table 12. This means that the elimination order for the linear models was duty cycle, G, R, W, B, frequency, and intensity.

The nonlinear model found that the most crucial characteristic was intensity, while the least important was W (white color), with sufficient significance $p < 0.05$. If the correct sequence of the other variables is required, we can rely on the mean values for the feature distribution (Table 13). Overall, the elimination sequence was W, G, B, duty cycle, frequency, R, and intensity.

The elimination order for the duty cycle and R in the linear and nonlinear models suggests that they are nonlinear features, mainly because several linear algorithms selected them as the first variables to eliminate but nonlinear algorithms selected them as the most important ones.

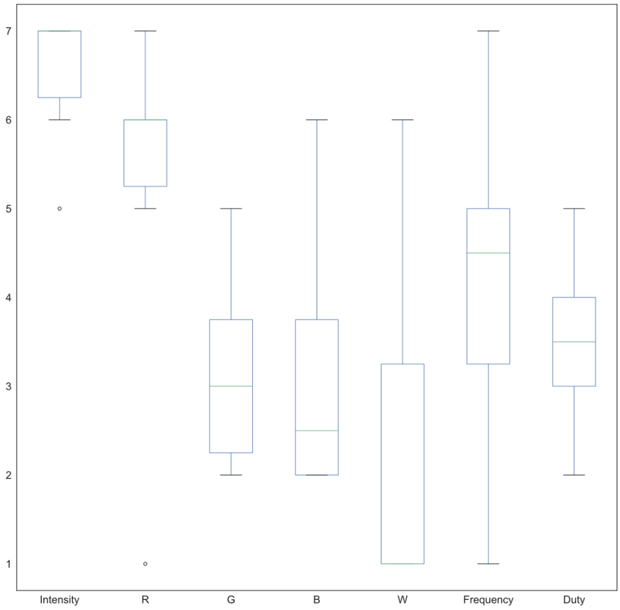


Figure 10. Order of elimination for features in nonlinear models.

5. Conclusions

In this study, we performed feature selection in order to prioritize inputs in the predication of energy consumption in an artificial illumination system for a CPPS using linear and nonlinear regression models. A dataset was generated with electrical measurements for proprieties such as intensity, light wavelength (RGB and W), frequency, and duty cycle.

The algorithms used for the linear models to identify the elimination order of the features included the variance threshold, Pearson correlation, univariate liner F-regression, and sequential backward feature selection with linear regression.

On the other hand, for nonlinear models, the algorithms used were the variance threshold, mutual information gain, and sequential backward feature selection with tree decision regression, using a tree depth from 2–5. The Kruskal–Wallis test served to validate the elimination order distributions.

The best order for eliminating features with the linear model was duty cycle, light color, frequency, and intensity, with $p_{value} = 0.012364$. The best order with nonlinear models was white, green, blue, duty cycle, frequency, red, and intensity, with significance at $p_{value} = 0.007161$. The elimination order for the duty cycle and R in the linear and nonlinear models differed enormously because the linear algorithms considered them the most suitable elimination features, while nonlinear algorithms marked them as essential features. This discrepancy was because the duty cycle and R were nonlinear features. Thus, only nonlinear models could map them correctly. Moreover, this supports the hypothesis that the energy consumption in LED lamps for CPPSs has nonlinear behavior and that nonlinear models should be used to predict it.

This technique allows various deductions to be drawn from the analysis of the data obtained, including the estimation of the average energy consumption and its comparison with the quality of the crop, as well as the determination of the circumstances under which energy use is efficient. The selection of characteristics can be used as a reference for the agro-industrial community.

Author Contributions: All authors conceived the experiments; E.O.-G., N.E.-G. and J.A.D.-A. collected the data for the experiments; M.M.R., N.E.-G., E.O.-G. and P.V.-J. conducted the experiments, performed the statistical analysis, and generated the figures. All authors wrote and reviewed the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: We acknowledge the support of the Consejo Nacional de Ciencia y Tecnología (CONACYT) in Mexico for supporting this work through funds for the projects INFRA-2016-01, Project No. 270665, and CB-2016-01, Project No. 287818.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. FAO. *FAO Publications Catalogue*; FAO: Quebec City, QC, Canada, 2021.
2. Massa, G.D.; Kim, H.H.; Wheeler, R.M.; Mitchell, C.A. Plant Productivity in Response to LED Lighting. *HortScience* **2008**, *43*, 1951–1956. [CrossRef]
3. Kozai, T.; Fujiwara, K.; Runkle, E.S. *LED Lighting for Urban Agriculture*; Springer: Singapore, 2016; ISBN 9789811018480.
4. Domurath, N.; Schroeder, F.G.; Glatzel, S. Light Response Curves of Selected Plants under Different Light Conditions. *Acta Hortic.* **2012**, *956*, 291–298. [CrossRef]
5. Eaves, J.; Eaves, S. Comparing the Profitability of a Greenhouse to a Vertical Farm in Quebec. *Can. J. Agric. Econ.* **2018**, *66*, 43–54. [CrossRef]
6. Benke, K.; Tomkins, B. Future Food-Production Systems: Vertical Farming and Controlled-Environment Agriculture. *Sustain. Sci. Pract. Policy* **2017**, *13*, 13–26. [CrossRef]
7. Mickens, M.A.; Skoog, E.J.; Reese, L.E.; Barnwell, P.L.; Spencer, L.E.; Massa, G.D.; Wheeler, R.M. A Strategic Approach for Investigating Light Recipes for 'Outredgeous' Red Romaine Lettuce Using White and Monochromatic LEDs. *Life Sci. Sp. Res.* **2018**, *19*, 53–62. [CrossRef] [PubMed]
8. Ahmed, H.A.; Yu-Xin, T.; Qi-Chang, Y. Optimal Control of Environmental Conditions Affecting Lettuce Plant Growth in a Controlled Environment with Artificial Lighting: A Review. *S. Afr. J. Bot.* **2020**, *130*, 75–89. [CrossRef]
9. Meng, Q.; Kelly, N.; Runkle, E.S. Substituting Green or Far-Red Radiation for Blue Radiation Induces Shade Avoidance and Promotes Growth in Lettuce and Kale. *Environ. Exp. Bot.* **2019**, *162*, 383–391. [CrossRef]
10. Graamans, L.; Baeza, E.; van den Dobbelsteen, A.; Tsafaras, I.; Stanghellini, C. Plant Factories versus Greenhouses: Comparison of Resource Use Efficiency. *Agric. Syst.* **2018**, *160*, 31–43. [CrossRef]
11. Avgoustaki, D.D.; Xydis, G. Energy Cost Reduction by Shifting Electricity Demand in Indoor Vertical Farms with Artificial Lighting. *Biosyst. Eng.* **2021**, *211*, 219–229. [CrossRef]
12. Hwang, P.W.; Chen, C.H.; Chang, Y.J. A Study on Energy Strategy of a Plant Factory Using Sustainable Energy Combined with Computational Fluid Dynamics Simulation: An Innovative Practice of Green Information Systems. In Proceedings of the Proceedings of Computing Conference, London, UK, 18–20 July 2017; IEEE: Piscataway, NJ, USA, 2018; pp. 517–522.
13. Sørensen, J.C.; Kjaer, K.H.; Ottosen, C.O.; Jørgensen, B.N. DynaGrow—Multi-Objective Optimization for Energy Cost-Efficient Control of Supplemental Light in Greenhouses. In Proceedings of the 8th International Joint Conference on Computational Intelligence (IJCCI 2016), Porto, Portugal, 9–11 November 2016; pp. 41–48.
14. Francik, S.; Kurpaska, S. The Use of Artificial Neural Networks for Forecasting of Air Temperature inside a Heated Foil Tunnel. *Sensors* **2020**, *20*, 652. [CrossRef]
15. Jung, D.H.; Kim, H.S.; Jhin, C.; Kim, H.J.; Park, S.H. Time-Serial Analysis of Deep Neural Network Models for Prediction of Climatic Conditions inside a Greenhouse. *Comput. Electron. Agric.* **2020**, *173*, 105402. [CrossRef]
16. Escamilla-García, A.; Soto-Zarazúa, G.M.; Toledano-Ayala, M.; Rivas-Araiza, E.; Gastélum-Barrios, A. Applications of Artificial Neural Networks in Greenhouse Technology and Overview for Smart Agriculture Development. *Appl. Sci.* **2020**, *10*, 3835. [CrossRef]
17. Singh, V.K.; Tiwari, K.N. Prediction of Greenhouse Micro-Climate Using Artificial Neural Network. *Appl. Ecol. Environ. Res.* **2017**, *15*, 767–778. [CrossRef]
18. Gros, S.; Zanon, M.; Quirynen, R.; Bemporad, A.; Diehl, M. From Linear to Nonlinear MPC: Bridging the Gap via the Real-Time Iteration. *Int. J. Control* **2020**, *93*, 62–80. [CrossRef]
19. Ouammi, A.; Achour, Y.; Zejli, D.; Dagdougui, H. Supervisory Model Predictive Control for Optimal Energy Management of Networked Smart Greenhouses Integrated Microgrid. *IEEE Trans. Autom. Sci. Eng.* **2020**, *17*, 117–128. [CrossRef]
20. Xu, H.; Zhai, Z.; Wang, K.; Ren, S.; Wang, H. Multiobjective Distributed Model Predictive Control Method for Facility Environment Control Based on Cooperative Game Theory. *Turk. J. Electr. Eng. Comput. Sci.* **2017**, *25*, 4160–4171. [CrossRef]
21. Lin, D.; Zhang, L.; Xia, X. Hierarchical Model Predictive Control of Venlo-Type Greenhouse Climate for Improving Energy Efficiency and Reducing Operating Cost. *J. Clean. Prod.* **2020**, *264*, 121513. [CrossRef]

22. Mosavi, A.; Ozturk, P.; Chau, K.W. Flood Prediction Using Machine Learning Models: Literature Review. *Water* **2018**, *10*, 1536. [CrossRef]
23. Hosseinzadeh, A.; Zhou, J.L.; Altaee, A.; Li, D. Machine Learning Modeling and Analysis of Biohydrogen Production from Wastewater by Dark Fermentation Process. *Bioresour. Technol.* **2022**, *343*, 126111. [CrossRef]
24. Alizamir, M.; Kisi, O.; Ahmed, A.N.; Mert, C.; Fai, C.M.; Kim, S.; Kim, N.W.; El-Shafie, A. Advanced Machine Learning Model for Better Prediction Accuracy of Soil Temperature at Different Depths. *PLoS ONE* **2020**, *15*, e0231055. [CrossRef]
25. Nemati, S.; Holder, A.; Razmi, F.; Stanley, M.D.; Clifford, G.D.; Buchman, T.G. An Interpretable Machine Learning Model for Accurate Prediction of Sepsis in the ICU. *Crit. Care Med.* **2018**, *46*, 547. [CrossRef] [PubMed]
26. Sneha, N.; Gangil, T. Analysis of Diabetes Mellitus for Early Prediction Using Optimal Features Selection. *J. Big Data* **2019**, *6*, 13. [CrossRef]
27. Haq, A.U.; Li, J.; Memon, M.H.; Hunain Memon, M.; Khan, J.; Marium, S.M. Heart Disease Prediction System Using Model of Machine Learning and Sequential Backward Selection Algorithm for Features Selection. In Proceedings of the 2019 IEEE 5th International Conference for Convergence in Technology, Bombay, India, 29–31 March 2019. [CrossRef]
28. Fan, X.; Wang, X.; Zhang, X.; Yu, P.A. Machine Learning Based Water Pipe Failure Prediction: The Effects of Engineering, Geology, Climate and Socio-Economic Factors. *Reliab. Eng. Syst. Saf.* **2022**, *219*, 108185. [CrossRef]
29. Ahmed, H.W.; Alamire, J.H. A Review of Machine Learning Models in the Air Quality Research. *Int. J. Adv. Res. Comput. Eng. Technol.* **2020**, *9*, 30–36.
30. Zoabi, Y.; Deri-Rozov, S.; Shomron, N. Machine Learning-Based Prediction of COVID-19 Diagnosis Based on Symptoms. *npj Digit. Med.* **2021**, *4*, 3. [CrossRef] [PubMed]
31. Mahesh, B. Machine Learning Algorithms—A Review. *Int. J. Sci. Res.* **2020**, *9*, 381–386. [CrossRef]
32. Ahmad, A.; Khan, M.; Paul, A.; Din, S.; Rathore, M.M.; Jeon, G.; Choi, G.S. Toward Modeling and Optimization of Features Selection in Big Data Based Social Internet of Things. *Future Gener. Comput. Syst.* **2018**, *82*, 715–726. [CrossRef]
33. Khan, M.A.; Kadry, S.; Alhaisoni, M.; Nam, Y.; Zhang, Y.; Rajinikanth, V.; Sarfraz, M.S. Computer-Aided Gastrointestinal Diseases Analysis from Wireless Capsule Endoscopy: A Framework of Best Features Selection. *IEEE Access* **2020**, *8*, 132850–132859. [CrossRef]
34. Genova, K.; Cole, F.; Maschinot, A.; Sarna, A.; Vlasic, D.; Freeman, W.T. Unsupervised Training for 3D Morphable Model Regression. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 8377–8386.
35. Gehrig, D.; Gehrig, M.; Hidalgo-Carrio, J.; Scaramuzza, D. Video to Events: Recycling Video Datasets for Event Cameras. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 3586–3595.
36. Simao, M.; Mendes, N.; Gibaru, O.; Neto, P. A Review on Electromyography Decoding and Pattern Recognition for Human-Machine Interaction. *IEEE Access* **2019**, *7*, 39564–39582. [CrossRef]
37. Combes, P.P.; Gobillon, L.; Zylberberg, Y. Urban Economics in a Historical Perspective: Recovering Data with Machine Learning. *Reg. Sci. Urban Econ.* **2021**, *94*, 103711. [CrossRef]
38. Uysal, A.K.; Gunal, S. The Impact of Preprocessing on Text Classification. *Inf. Process. Manag.* **2014**, *50*, 104–112. [CrossRef]
39. Zhang, X.; Zhao, Z.; Wang, Z.; Wang, X. Fault Detection and Identification Method for Quadcopter Based on Airframe Vibration Signals. *Sensors* **2021**, *21*, 581. [CrossRef] [PubMed]
40. Choras, R.S. A Survey on Methods of Image Processing and Recognition for Personal Identification. In *Machine Learning and Biometrics*; IntechOpen: Vienna, Austria, 2018. [CrossRef]
41. Mohammed, B.; Hasan, S.; Mohsin Abdulazeez, A. A Review of Principal Component Analysis Algorithm for Dimensionality Reduction. *J. Soft Comput. Data Min.* **2021**, *2*, 20–30. [CrossRef]
42. García, S.; Luengo, J.; Herrera, F. *Data Preprocessing in Data Mining*; Springer: Cham, Switzerland, 2015; Volume 72, ISBN 9783319102467.
43. Arslan, S.; Ozturk, C. Feature Selection for Classification with Artificial Bee Colony Programming. In *Swarm Intelligence-Recent Advances, New Perspectives and Applications*; IntechOpen: Vienna, Austria, 2019. [CrossRef]
44. Olvera-Gonzalez, E.; Rivera, M.M.; Escalante-Garcia, N.; Flores-Gallegos, E. Modeling Energy LED Light Consumption Based on an Artificial Intelligent Method Applied to Closed Plant Production System. *Appl. Sci.* **2021**, *11*, 2735. [CrossRef]
45. Jia, K.; Yang, L.; Liang, S.; Xiao, Z.; Zhao, X.; Yao, Y.; Zhang, X.; Jiang, B.; Liu, D. Long-Term Global Land Surface Satellite (GLASS) Fractional Vegetation Cover Product Derived From MODIS and AVHRR Data. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2018**, *12*, 508–518. [CrossRef]
46. Kappal, S. Data Normalization Using Median Median Absolute Deviation MMAD Based Z-Score for Robust Predictions vs. Min—Max Normalization. *Lond. J. Res. Sci. Nat. Form.* **2019**, *19*, 39–44.
47. Saranya, C.; Manikandan, G. A Study on Normalization Techniques for Privacy Preserving Data Mining. *Int. J. Eng. Technol.* **2013**, *5*, 2701–2704.
48. Curran-Everett, D. Explorations in Statistics: Hypothesis Tests and P Values. *Am. J. Physiol. Adv. Physiol. Educ.* **2009**, *33*, 81–86. [CrossRef]

49. Kumar, S.C.; Ramasree, R.J. Dimensionality Reduction in Automated Evaluation of Descriptive Answers through Zero Variance, near Zero Variance and Non Frequent Words Techniques-a Comparison. In Proceedings of the 2015 IEEE 9th International Conference on Intelligent Systems and Control (ISCO), Coimbatore, India, 9–10 January 2015. [CrossRef]
50. Roberts, A.G.K.; Catchpoole, D.R.; Kennedy, P.J. Variance-Based Feature Selection for Classification of Cancer Subtypes Using Gene Expression Data. In Proceedings of the International Joint Conference on Neural Networks, Rio de Janeiro, Brazil, 8–13 July 2018. [CrossRef]
51. Siti Ambarwati, Y.; Uyun, S. Feature Selection on Magelang Duck Egg Candling Image Using Variance Threshold Method. In Proceedings of the 2020 3rd International Seminar on Research of Information Technology and Intelligent Systems, Yogyakarta, Indonesia, 10–11 December 2020; pp. 694–699. [CrossRef]
52. Bewick, V.; Cheek, L.; Ball, J. Statistics Review 9: One-Way Analysis of Variance. *Crit. Care* **2004**, *8*, 130–136. [CrossRef]
53. Chehreh Chelgani, S.; Shahbazi, B.; Hadavandi, E. Support Vector Regression Modeling of Coal Flotation Based on Variable Importance Measurements by Mutual Information Method. *Measurement* **2018**, *114*, 102–108. [CrossRef]
54. Mamun, M.M.R.K.; Alouani, A.T. Cuffless Blood Pressure Measurement Using Linear and Nonlinear Optimized Feature Selection. *Diagnostics* **2022**, *12*, 408. [CrossRef] [PubMed]
55. Xiong, H.; Fan, C.; Chen, H.; Yang, Y.; ANTWI, C.O.; Fan, X. A Novel Approach to Air Passenger Index Prediction: Based on Mutual Information Principle and Support Vector Regression Blended Model. *SAGE Open* **2022**, *12*. [CrossRef]
56. Toğaçar, M.; Ergen, B.; Cömert, Z. Classification of Flower Species by Using Features Extracted from the Intersection of Feature Selection Methods in Convolutional Neural Network Models. *Measurement* **2020**, *158*, 107703. [CrossRef]
57. Khaire, U.M.; Dhanalakshmi, R. Stability of Feature Selection Algorithm: A Review. *J. King Saud Univ. Comput. Inf. Sci.* **2019**, *34*, 1060–1073. [CrossRef]
58. Wang, M.; Lu, Y.; Qin, J. A Dynamic MLP-Based DDoS Attack Detection Method Using Feature Selection and Feedback. *Comput. Secur.* **2020**, *88*, 101645. [CrossRef]
59. Zhang, D.; Khalili, A.; Asgharian, M. Post-Model-Selection Inference in Linear Regression Models: An Integrated Review. *Stat. Surv.* **2022**, *16*, 86–136. [CrossRef]
60. Darwin, D.; Christian, D.; Chandra, W.; Nababan, M. Comparison of Decision Tree and Linear Regression Algorithms in the Case of Spread Prediction of COVID-19 in Indonesia. *J. Comput. Netw. Archit. High Perform. Comput.* **2022**, *4*, 1–12. [CrossRef]
61. Johnson, R.W. Alternate Forms of the One-Way ANOVA F and Kruskal-Wallis Test Statistics. *J. Stat. Data Sci. Educ.* **2022**, *30*, 82–85. [CrossRef]

electronics

MDPI

Article

Electric Vehicle Fire Trace Recognition Based on Multi-Task Semantic Segmentation

Jiankun Pu and Wei Zhang *

School of Microelectronics, Tianjin University, Tianjin 300072, China; 3015204014@tju.edu.cn
* Correspondence: tjuzhangwei@tju.edu.cn

Abstract: Conflagration is the major safety issue of electric vehicles (EVs). Due to their well-kept appearance and structure, which demonstrate salient visual changes after combustion, EV bodies are recognized as an important basis for on-spot inspection of burnt EVs and make application using semantic segmentation possible. The combination of deep learning-based semantic segmentation and recognition of visual traces of burnt EVs would provide preliminary analytical results of fire spread trends and output status descriptions of burnt EVs for further investigation. In this paper, a dataset of image traces of burnt EVs was built, and a two-branch network structure that splits the whole task into two sub-tasks separately concentrated on foreground extraction and severity segmentation is proposed. The proposed network is trained on the dataset via the transfer learning method and is tested using 5-fold cross validation. The foreground extraction branch achieved a mean intersection over union (mIoU) of 95.16% in the burnt EV foreground extraction task, and the burnt severity branch achieved a mIoU of 66.96% for the severity segmentation task. By jointly training two branches and applying a foreground mask to 3-class severity output, the mIoU was improved to 68.92%.

Keywords: deep learning; semantic segmentation; electric vehicle fire

Citation: Pu, J.; Zhang, W. Electric Vehicle Fire Trace Recognition Based on Multi-Task Semantic Segmentation. *Electronics* **2022**, *11*, 1738. https://doi.org/10.3390/electronics11111738

Academic Editors: Luis Hernández-Callejo, Sergio Nesmachnow and Sara Gallardo Saavedra

Received: 5 May 2022
Accepted: 27 May 2022
Published: 30 May 2022

1. Introduction

Vehicles are necessities in human life and are extensively utilized in logistics, transportation and travel. The termination of the production of traditional internal combustion engine vehicles (ICEVs) is being gradually implemented worldwide under the pressure of the global energy shortage and environment pollution issues, and electric vehicles (EVs) are recognized ideal alternatives in this situation. Partially or fully driven by Li-ion batteries, EVs have presented the potential hazard of fire, which heavily affects the safety of passengers under various scenarios, e.g., parking, charging and driving. Fire incidents in EVs and plug-in hybrid electric vehicles (PHEVs) mostly begin in the battery power system. Compared with gasoline-caused vehicle fires, battery-caused vehicle fires contain more energy, extremely high temperatures, and the release of combustible and toxic gas, thus leading to higher risks and difficulty in extinguishing the fire [1,2].

In order to eliminate potential fire hazards and improve the manufacturing safety of EVs, correlative research should not only focus on prevention of combustion, but also on analysis and research of existing cases of burnt EVs. Recently, the on-spot investigation of burnt EVs has become an important method for analysis and research. Fire or damage traces remaining on the body panels and vehicle frames are frequently used to locate the origin of fire [3]. When the vehicle is not burnt extensively, traces with salient appearances, e.g., burnt-off paint and rusted metal, can provide reliable clues for the determination of fire origin [4]. Due to the similarity of material and paint utilized in EVs and conventional vehicles, fire traces of bodies of burnt EVs are also applicable and credible for investigation. Moreover, fire traces can be conveniently captured as digital images, which also provides possibilities for using a computer vision method for recognition.

Semantic segmentation is one of the major computer vision tasks that applies end-to-end classification of every pixel of the image input and outputs a corresponding segmentation map, in which a cluster of pixels classified as the same class is called semantic. With fully convolutional network (FCN) [5] first introduce convolutional neural network (CNN) into semantic segmentation, multiple advanced network structures with various optimization methods were proposed, e.g., contextual information-reinforced PSPNet [6] and DeepLab [7,8] and attention mechanism-based DANet [9] and PSANet [10]. Multiple backbones are also implemented in semantic segmentation tasks for different purposes, e.g., ResNet [11,12] with deep architecture, MobileNet [13] as a lightweight framework, and HRNet [14] for high-resolution feature extraction.

With the improvement of computer performance and the emergence of in-depth research on deep learning, semantic segmentation has been utilized in various practical tasks and has achieved par excellence performance. In the medical field, Ronneberger et al. [15] proposed U-Net with an encoder-decoder architecture for biomedical segmentation tasks. Milletari et al. [16] proposed a variant called V-Net that utilized residual blocks. Zhou et al. [17] proposed a much more complex UNet++ with sub-networks connected through a series of nested, dense skip pathways. Apart from the structures, the targets for medical segmentation also varies, e.g., lungs, lesions, lobes, tumours, and vessels. In the scene parsing and automatic driving field, Zhao et al. [6] proposed PSPNet with a classic pyramid pooling module. Charles et al. [18] expanded the input of the network to 3d point sets and proposed a related structure named PointNet. Kirillov et al. [19] combined sematic segmentation and instance segmentation tasks and proposed a new task called panoptic segmentation. Semantic segmentation is also in large-scale use for fire and smoke detection and recognition. Wang et al. [20] proposed a model concentrated on small fire and smoke regions in video data. Zhang et al. [21] proposed a lightweight U-Net-based network for forest fire detection and recognition. Mseddi et al. [22] proposed a method combining YOLOV5 and U-Net for fire detection and segmentation. Moreover, in the remote sensing field, Chen et al. [23] proposed symmetrical dense-shortcut frameworks for very-high-resolution images, and Zhang et al. [24] proposed a dual lightweight attention network for high-resolution remote sensing images.

Currently, no semantic segmentation-based research on the recognition of EV fire traces has been implemented, and no corresponding dataset has been built for the task. However, according to the forementioned analogous tasks, semantic segmentation would be compatible with the EV fire trace recognition task of this paper. The combination of semantic segmentation would not only output a preliminary analytical result of burnt EVs by collecting images conveniently, but also make its output a status description of burnt EVs for further archive and research. In summary, the main contributions of this paper can be summarized as follows:

1. A deep learning-based semantic segmentation technique was novelly applied to the recognition of fire image traces on EVs, and a dataset was labeled according to the different visual appearances of burnt EVs for corresponding tasks;
2. A multi-task learning-based two-branch network architecture was proposed. The first branch of the network was used for the foreground extraction task, and the other was built for distinguishing different severities of the burnt vehicle body. The best configuration of training and output of this architecture was found;
3. A connectivity-based weighted cross entropy loss function was proposed in the foreground branch for eliminating false true regions and keeping the main vehicle body for further processing;
4. A densely connected module with the expectation maximum attention (EMA) mechanism was proposed for better extracting multi-scale features in the severity segmentation branch.

The proposed model and an executable demo are available in Supplementary Materials at: https://github.com/Jkreat/EVFTR (accessed on 27 May 2022).

2. Materials and Methods

2.1. Dataset of Burnt EVs

Original images of burnt EVs were collected from various accident cases of EV combustion in China and burning tests conducted by Tianjin Fire Research Institute of M.E.M. The dataset contains 314 raw images with pixel-level annotations of burnt EVs. Vehicle bodies of the dataset are labeled into 3 different levels of severity and background into pixel-level according to their visual appearance after combustion. Blue stands for intact (IN), brown stands for mild and moderate burnt (MB) regions, red stands for severely burnt (SB) regions, and black stands for background (BG). The proportion of the numbers of pixels in different classes is shown in Table 1. Detailed regions of different labels are shown in Figure 1. The distinction between MB and SB is mainly based on the visual appearance of the painting. In short, regions with painting burnt into yellow or black were labeled as MB, and regions with painting entirely burnt out and bottom metal exposed were labeled as SB. As for tires and glasses, MB and SB were labeled according to whether their basic structure were kept after burning. All images with labeled masks were resized to 560 × 420 to fit the input of the proposed network. Moreover, the whole dataset was divided into five folds uniformly for five-fold cross validation. While training the foreground extraction branch, the labeled images were transferred into foreground masks. More images with corresponding labeled masks for different tasks are shown in Figure 2.

Table 1. Proportion of numbers of pixels in different classes (%).

BG	IN	MB	SB
57.09	25.70	7.55	9.67

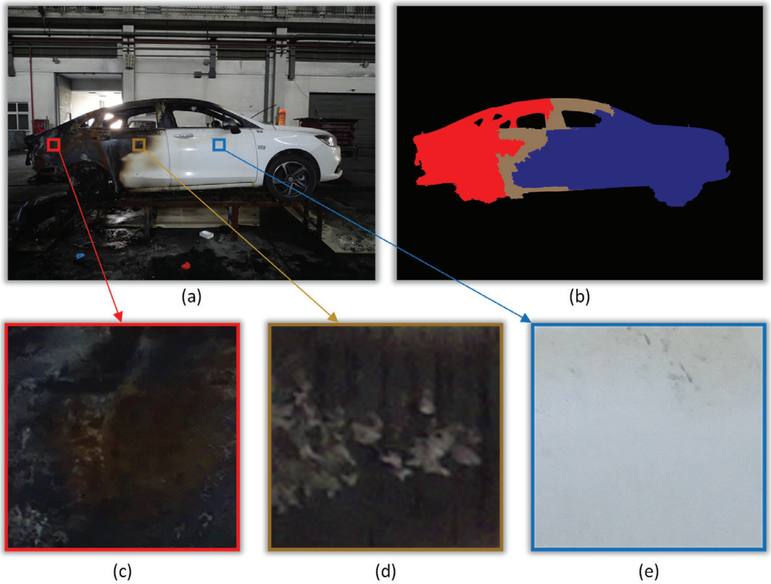


Figure 1. Original image, labeled image and details. (**a**) Original image of burnt EV. (**b**) Labeled image of different severity. (**c**) Detail of region labeled as SB. (**d**) Detail of region labeled as MB. (**e**) Detail of region labeled as IN.

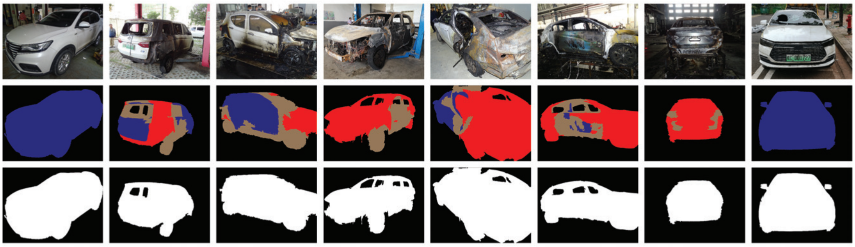


Figure 2. Images from the dataset with corresponding labels. First row: original images, second row: labeled masks for fire trace segmentation, third row: labeled masks for foreground extraction.

2.2. Backbone and Transfer Learning

Many public datasets for semantic segmentation task contain classes annotated as vehicles or cars. Due to the similarity of burnt vehicles in the tasks of this paper and intact vehicles annotated in public datasets, initializing pretrained weights from these public datasets for training the proposed network of this paper via fine-tuning method will not only lead to quick convergence, but significantly improve the overall accuracy by transferring knowledge learned from abundant corresponding data. Therefore, rather than training from scratch, transfer learning was used for training. To obtain benefits from pretrained weights and extract features better, a mainstream backbone network with deep architecture was needed. Therefore, ResNet101 with dilated convolution was selected as the backbone of the proposed architecture. Weights of the backbone were initialized using pretrained weights from COCO dataset.

Compared with the original ResNet101, the dilated version has the same number of layers and number of parameters but replaces the normal convolution operation with the dilated convolution operation in the last two groups of convolution blocks. Such a replacement increased the resolution of the output feature map without reducing the reception field. As for the semantic segmentation task, the feature map with higher spatial resolution contains more context representation; thus, the dilated ResNet101 better fits the task of this paper. The detailed configuration of the selected backbone is listed in Table 2.

Table 2. Configuration of backbone.

Layer Name	Block Configuration	Number of Blocks	Output Size
Layer0	$\begin{bmatrix} Conv, (7 \times 7), 64, stride = 2 \\ Maxpool, (3 \times 3), 64, stride = 2 \end{bmatrix}$	1	280 × 210
Layer1	$\begin{bmatrix} Conv, (3 \times 3), 64, stride = 1 \\ Conv, (3 \times 3), 64, stride = 1 \\ Conv, (3 \times 3), 256, stride = 1 \end{bmatrix}$	3	140 × 105
Layer2	$\begin{bmatrix} Conv, (3 \times 3), 128, stride = 1 \\ Conv, (3 \times 3), 128, stride = 1 \\ Conv, (3 \times 3), 512, stride = 1 \end{bmatrix}$	4	70 × 53
Layer3	$\begin{bmatrix} Conv, (3 \times 3), 256, stride = 1 \\ Conv, (3 \times 3), 256, dilation = 2 \\ Conv, (3 \times 3), 1024, stride = 1 \end{bmatrix}$	23	70 × 53
Layer4	$\begin{bmatrix} Conv, (3 \times 3), 512, stride = 1 \\ Conv, (3 \times 3), 512, dilation = 4 \\ Conv, (3 \times 3), 2048, stride = 1 \end{bmatrix}$	3	70 × 53

2.3. Foreground Extraction Branch

2.3.1. Network Structure

A modified atrous spatial pyramid pooling (ASPP) module from DeeplabV3 was connected after the backbone in this branch for capturing the multi-scale context. To fit the size of the feature map from the backbone, the original ASPP module with a dilation rate of

(6, 12, 18) was modified to a larger module with a dilation rate of (4, 11, 18, 25). Moreover, the number of output channels of each layer was promoted from 256 to 512. The overall structure of the foreground extraction branch is shown in Figure 3.

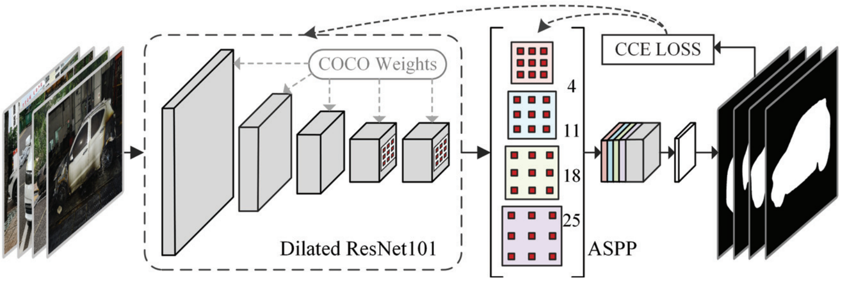


Figure 3. Structure of the foreground extraction branch.

2.3.2. CCE Loss Function

The characteristics of the foreground extraction task in this paper are summarized as follows:

- Every input image has only one main EV body as the target for processing. Other partially or fully captured vehicle bodies in the image should all be regarded as background and be minimized;
- The body of the target EV in each image is always at the center of image, i.e., the farther a predicted foreground pixel cluster is from the center of image, the less possible it would be for it to be considered the main vehicle target;
- Compared to false negative (FN) areas, false positive (FP) areas are a major issue that influence overall accuracy and should be eliminated.

To restrain the FP areas of the results from the foreground branch, a cross entropy loss function with connectivity-based weights was proposed to increase the penalization of FP domains according to their area and distance from center of the image.

The proposed loss function works when the model is "nearly converged", i.e., $N < \textit{threshold}$ connected domains exist in the output image. In this condition, a connectivity analysis algorithm is applied to split output foreground into N sorted domains according to their area, and the domain with the largest area is regarded as the main body of the vehicle.

$$\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_N\} \tag{1}$$

The weighted binary cross entropy loss function for 2-class segmentation task could be described as below:

$$L = -\frac{1}{N}\sum_{i=1}^{N}(y_i \log p_i + w(1-y_i)\log(1-p_i)) \tag{2}$$

In the equation above, w is the weight value. When $w < 1$, the function concentrates more on FNs; on the contrary, the function pays more attention on FPs when $w > 1$. Moreover, the function degenerates into normal cross entropy loss if w tends to 1. When one pixel belongs to the domain $\mathcal{D}_k$, w is calculated as follows:

$$w = 1 + \log\left(1 + \frac{d_k}{\gamma}\sqrt{\frac{A_k}{A_1}}\right) \tag{3}$$

In the equation above, d_k stands for the distance between the centroid of the minimum bounding rectangle of $\mathcal{D}_k$ and the center of image, A_k is the area of $\mathcal{D}_k$, and A_1 is the domain that possesses the largest area, i.e., the main body of the EV. γ is a hyperparameter for controlling the value of the weight.

2.4. Severity Segmentation Branch

Considering that the features of burnt EV bodies are close to the features of intact vehicles from the source dataset used for pretraining, the transfer learning method is effective in the foreground extraction task, and a simple ASPP module would result in good accuracy. However, in the severity segmentation task, the features of burnt regions are amorphous and abstract, and the number of classes for classification also increase from 2 to 4. Therefore, a network architecture with a better feature representation capability is in need.

Contextual information reinforcement and attention mechanism utilization are two major research priorities in semantic segmentation research. Inspired by DenseASPP, a densely connected multi-scale structure with an attention module named DA-EMA was proposed in this paper. The overall structure of the severity segmentation branch, including the DA-EMA module, is shown in Figure 4.

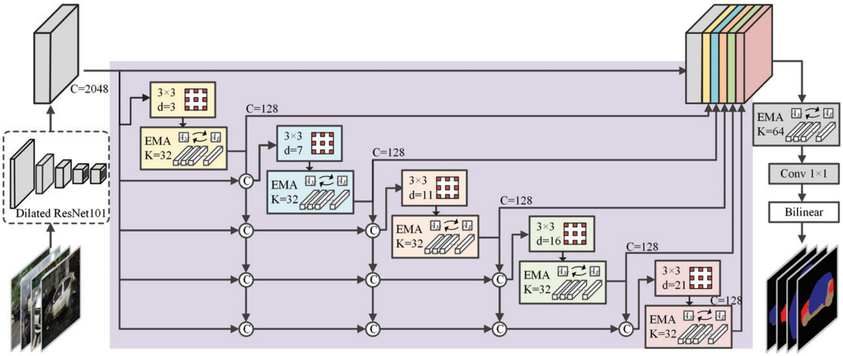


Figure 4. Structure of the severity segmentation branch. C is the output channels, K is number of bases contained in the EMA unit, and d is the dilation rate.

Simply improving the dilation rate of the ASPP module to improve the receptive field may cause a drop in overall model performance caused by the loss of modelling capability. To solve the problem and enlarge the receptive field further, Yang et al. [25] proposed a DenseNet [26]-like densely connected ASPP (DenseASPP) module.

Attention mechanisms have been proven effective in many semantic segmentation scenarios by performing feature recalibration and feature enhancement [27]. In this paper, an attention module is added to every level of a densely connected structure for enhancing multi-scale feature representation. However, traditional attention-based modules need to generate a large attention map that has high computation complexity and high GPU memory cost. A lightweight expectation maximization attention (EMA) module [28] is a good alternative in this case. Instead of treating all pixels as the reconstruction bases of the attention map, the EMA module uses the expectation maximization algorithm to find a set of compact basis in an iterative manner and then largely reduces computational complexity. A typical EMA unit consists of three operations, including responsibility estimation (A_E), likelihood maximization (A_M) and data re-estimation (A_R). Given the input $\mathbf{X} \in \mathbb{R}^{N \times C}$ and the initial bases $\boldsymbol{\mu} \in \mathbb{R}^{K \times C}$, A_E estimates the latent variables $\mathbf{Z} \in \mathbb{R}^{N \times K}$ as 'responsibility', the step functions as the E step in the expectation maximization (EM) algorithm. A_M uses the estimation to update the bases $\boldsymbol{\mu}$, which works as the M step in the EM algorithm. The A_E and A_M steps execute alternately for a pre-specified number of iterations. Then, with the converged $\boldsymbol{\mu}$ and $\mathbf{Z}$, A_R reconstructs the original $\mathbf{X}$ as $\mathbf{Y}$ and outputs it. The detailed structure of one EMA unit is shown in Figure 5.

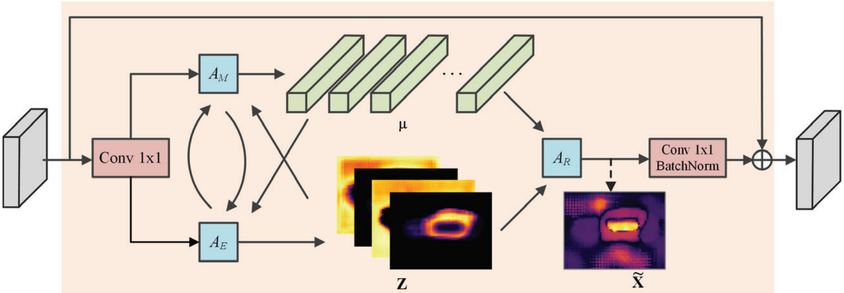


Figure 5. Detailed structure of an expectation maximization attention unit.

To improve the contextual representation, dilated convolution is frequently utilized in the proposed network. Wang et al. [29] found a "gridding" issue in the dilated convolution framework: as zeros are padded in the dilated convolution layer, the receptive field of the kernel only covers locations with a non-zero value and makes other neighboring information become lost. In this paper, dilation rates in the proposed DA-EMA module were modified from $(3, 6, 12, 18, 24)$ to $(3, 7, 11, 16, 21)$, which had no common divisor larger than 1 to improve the information used in the densely connected convolution layers with alleviation of the gridding effect. According to Figure 4, the overall DA-EMA module contains 5 EMA units with dilated convolution, and the sixth EMA unit is utilized to process the concatenated feature map. The detailed configuration of the dilated convolution layers and EMA units is shown in Table 3.

Table 3. Detailed configuration of DA-EMA units.

Block Name	Convolution Kernel Size	Dilation	Number of EMA Bases	Input Channels	Output Channels
DA-EMA1	3 × 3	3	32	2048	128
DA-EMA2	3 × 3	7	32	2048 + 128 × 1	128
DA-EMA3	3 × 3	11	32	2048 + 128 × 2	128
DA-EMA4	3 × 3	16	32	2048 + 128 × 3	128
DA-EMA5	3 × 3	21	32	2048 + 128 × 4	128
Output EMA	1 × 1	1	64	2048 + 128 × 5	4

2.5. Multi-Task Learning-Based Two-Branch Architecture

Multi-task learning is a learning mechanism that enables multiple learning tasks to improve their generalization performance by sharing common knowledge learned from other tasks and maintaining their own features. The proposed model combines branches introduced above together with a shared backbone feature. In the foreground extraction branch, the result is accurate enough by training with the transfer learning method; thus, the output of this branch is used as a mask for further processes. In the severity branch, the background class is set as ignored, i.e., the parameters of the background class are not reckoned in back propagation; only parameters of three different severity levels are learned. Finally, to get the final results, the mask from the foreground extraction branch is applied to the output image of the severity segmentation branch.

Two different training methods were adopted for comparison to get better results. The overall architecture and training methods are listed in Figure 6.

Two-stage training: Train the backbone and foreground extraction branch using transfer learning first, then fully freeze parameters of the backbone and train the severity segmentation branch.

Joint training: Train the two branches and background together, then calculate the weighted sum of loss from the two branches for back propagation. Assuming L_1 is the loss from the foreground extraction branch, and L_2 is the loss from the severity segmentation branch, the overall loss is calculated as:

$$L = \lambda L_1 + (1 - \lambda) L_2 \tag{4}$$

Moreover, two output methods were also implemented and taken into comparison. The first output method did not set the background label as ignored; thus, the severity branch also output the prediction of the background, and the number of classes of this branch output is 4. On the contrary, the second method set the background label as ignored, i.e., background was not included for back propagation; thus the severity branch barely output the prediction result containing the background class. Two different methods are shown in Figure 7.

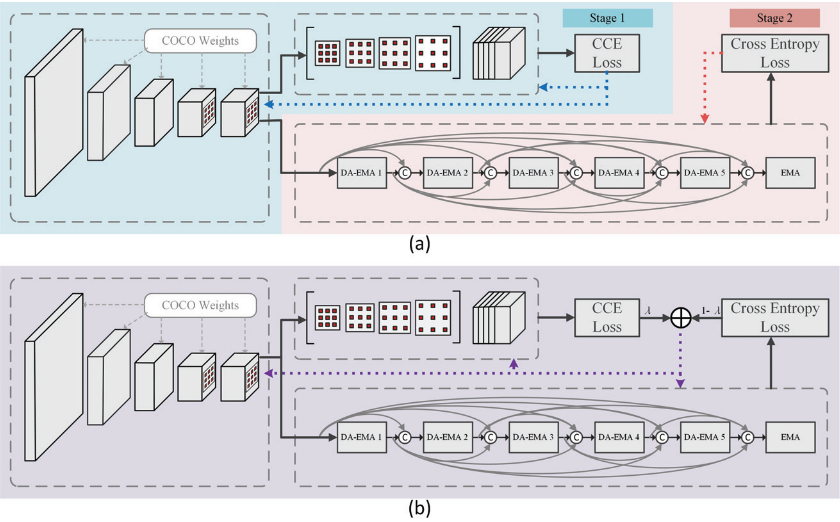


Figure 6. Two training methods implemented in this paper; dotted lines stand for back propagation. (**a**) Two-stage training; (**b**) joint training.

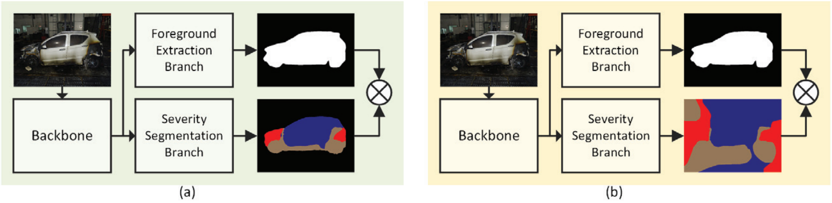

Backbone
Foreground Extraction Branch
Severity Segmentation Branch
(a)
Backbone
Foreground Extraction Branch
Severity Segmentation Branch
(b)

Figure 7. Two output methods (for the severity segmentation branch) in this paper: (**a**) 4-class output; (**b**) 3-class output.

2.6. K-fold Cross Validation

Generally, to evaluate the performance of a model, the dataset is randomly split into two subsets for training and testing according to a certain ratio. Test set obtained through this method may be unreliable to estimate the real performance of the model, especially when the size of the dataset is relatively small. K-fold cross validation utilizes all data to test the model, and thus could better estimate the generalization ability of the model. The fold number K is usually set to 5 or 10 [30,31]. In this paper, K was set to 5 as a trade-off between the bias of the result and time consumption for training. The leave-one-out method, a special case of K-fold cross validation, was utilized. In this case, the number of folds equals the number of instances.

3. Results

3.1. Experimental Configuration and Evaluation Metrics

All experiments were conducted on a server running the Ubuntu 16.04 operation system. The server was equipped with two Tesla p40 GPUs and a Xeon Gold 5118 CPU. The resolutions of images from the dataset were resized to 560 × 420. Due to the utilization of transfer learning, the model converged rapidly, and the number of training epochs was set to 10 while each branch was separately trained. When two branches were trained jointly, the number was increased to 20. For all experiments, the initial learning rate was set to 0.0001 and the Adam optimizer was used. Additionally, 5-fold cross validation was implemented. The training group with fold K set for testing was named training group K.

Intersection over union (IoU) was utilized as the metric form of segmentation tasks of this paper to evaluate the accuracy of the outputs. IoU is calculated as follows:

$$IoU = \frac{TP}{TP + FP + FN} \tag{5}$$

In experiments of the foreground extraction task, only the IoU of foreground that represented bodies of target vehicles were counted.

In the EV fire trace recognition task, the number of classes was set to 4, so the mean IoU (mIoU) of 4 classes was calculated to evaluate the performance. As discussed in 3.1, the 4 classes were IN, MB, SB, BG, and the mIoU could be calculated as follows:

$$mIoU = \frac{1}{4}(IoU_{BG} + IoU_{IN} + IoU_{MB} + IoU_{SB}) \tag{6}$$

Additionally, to evaluate the accuracy of vehicle body segmentation, the union regions of IN, MB and SB were regarded as "Vehicle Body" (VB) regions; to evaluate the segmentation accuracy of burnt regions as a whole, the union of MB regions and SB regions were regarded as "Fire Trace" (FT) regions. Their IoU was thus calculated as follows:

$$IoU_{VB} = \frac{I_{IN \cup MB \cup SB}}{U_{IN \cup MB \cup SB}} \tag{7}$$

$$IoU_{FT} = \frac{I_{MB \cup SB}}{U_{MB \cup SB}} \tag{8}$$

3.2. Experiments of the Foreground Extraction Branch

In this group of experiments, to evaluate the performance of the foreground extraction branch, the backbone was connected to the modified ASPP module only, and the proposed CCE loss function was utilized.

3.2.1. Parameter Experiments of the CCE Loss Function

γ is an important component of the proposed CCE loss function in the foreground extraction branch. The value of γ was adjusted in a reasonable range, and the results obtained from different values are shown in Table 4.

3.2.2. Ablation Study

To conduct an ablation study for the foreground extraction branch, we compared the impact of the modified ASPP module and the proposed CCE loss function. The value of γ in this experiment was set to 20 according to the results above. The comparison results are listed in Table 5.

Table 4. Detailed configuration of DA-EMA units.

γ	Training Group 1	Training Group 2	Training Group 3	Training Group 4	Training Group 5	Average	Standard Deviation
1	95.69	95.33	94.31	94.78	93.88	94.80	0.74
2	95.14	95.15	94.89	95.09	94.24	94.90	0.39
3	95.51	95.35	95.31	94.75	94.20	95.02	0.54
5	95.85	95.27	95.04	94.61	94.04	94.96	0.68
10	95.80	95.53	94.96	94.63	94.22	95.03	0.65
15	95.84	95.47	95.19	94.78	94.20	95.10	0.63
20	96.03	95.62	95.24	94.80	94.11	95.16	0.74
30	95.90	95.42	95.20	94.70	94.06	95.06	0.70
50	95.66	95.33	94.63	95.02	93.91	94.91	0.68

Table 5. Detailed configuration of DA-EMA units.

Modified ASPP	CCE Loss	Training Group 1	Training Group 2	Training Group 3	Training Group 4	Training Group 5	Average	Standard Deviation
		95.34	95.27	94.58	94.65	93.81	94.73	0.62
✓		95.81	95.37	94.69	94.78	94.02	94.93	0.69
	✓	95.48	95.59	94.88	94.73	94.09	94.95	0.61
✓	✓	96.03	95.62	95.24	94.80	94.11	95.16	0.74

3.3. Experiments of the Severity Segmentation Branch

In this group of experiments, to evaluate the performance of the severity segmentation branch, the backbone was connected to the proposed DA-EMA module only, and the number of classes for training and output was set to 4, i.e., no class was ignored in the back propagation process.

3.3.1. Performance Comparison

The proposed DA-EMA module and multiple mainstream semantic segmentation network structures were trained in the same configuration including the same backbone network. The results are shown in Table 6 and Figure 8.

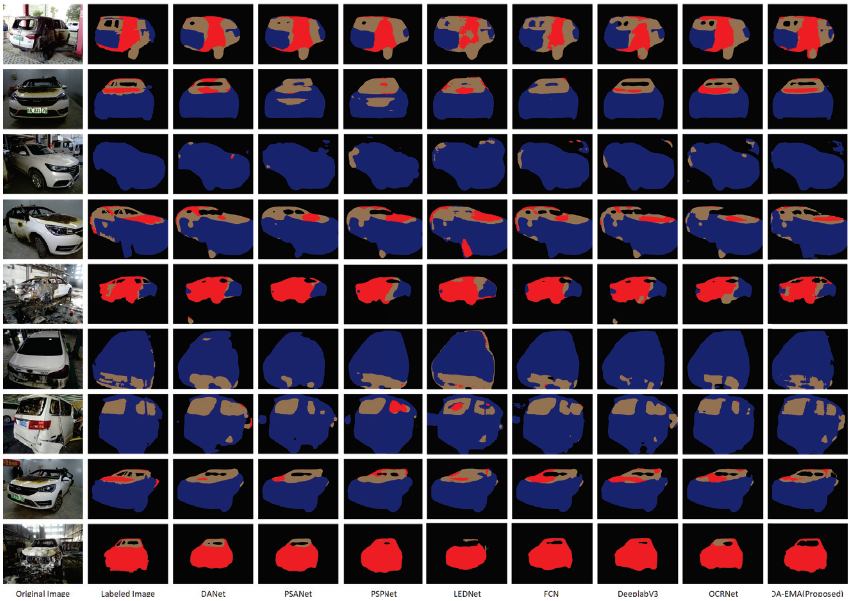


Figure 8. Results of the proposed DA-EMA and other semantic segmentation models.

Table 6. Comparison of the proposed DA-EMA and other semantic segmentation models.

Method	BG	IN	MB	SB	mIoU	VB	FT
FCN [5]	93.97	76.07	38.89	48.56	64.37	93.61	67.95
PSPNet [6]	94.15	75.07	37.63	46.90	63.44	92.82	67.85
DeepLabV3 [7]	93.84	76.23	39.11	49.06	64.56	93.44	68.47
DANet [9]	94.85	76.52	42.74	49.58	65.92	93.42	70.71
PSANet [10]	94.61	76.00	36.92	49.00	64.13	93.29	67.88
LEDNet [32]	93.18	76.91	37.82	48.82	64.18	91.25	65.75
OCRNet [33]	94.76	76.67	39.65	44.89	63.99	93.33	67.98
DA-EMA	95.30	77.12	42.68	52.73	66.96	94.03	71.00

3.3.2. Ablation Study

To examine the contribution of different modules in the proposed DA-EMA module, an ablation study was conducted. The first experiment used the structure of DenseASPP with a modified dilation rate without the EMA module (DA), the second experiment only utilized one EMA module to process the feature map from the backbone (EMA), and the third experiment was conducted using the proposed DA-EMA module. As per the results shown in Table 7, both the EMA module and the densely connected structure helped to improve the overall performance.

Table 7. Detailed configuration of DA-EMA units.

DA	EMA	BG	IN	MB	SB	mIoU	VB	FT
✓		95.16	76.92	40.03	51.22	65.83	93.86	69.15
	✓	94.52	75.78	40.79	48.51	64.90	92.87	69.78
✓	✓	95.30	77.12	42.68	52.73	66.96	94.03	71.00

3.3.3. Responsibility Map Visualization

In the EMA module, each basis corresponds to an abstract concept of the image. To examine whether the EMA mechanism functioned in the proposed DA-EMA module, multiple responsibility maps, i.e., latent variables **Z** generated from different EMA bases, were extracted. These were concluded from responsibility maps from different levels of the EMA module, as shown in Figure 9.

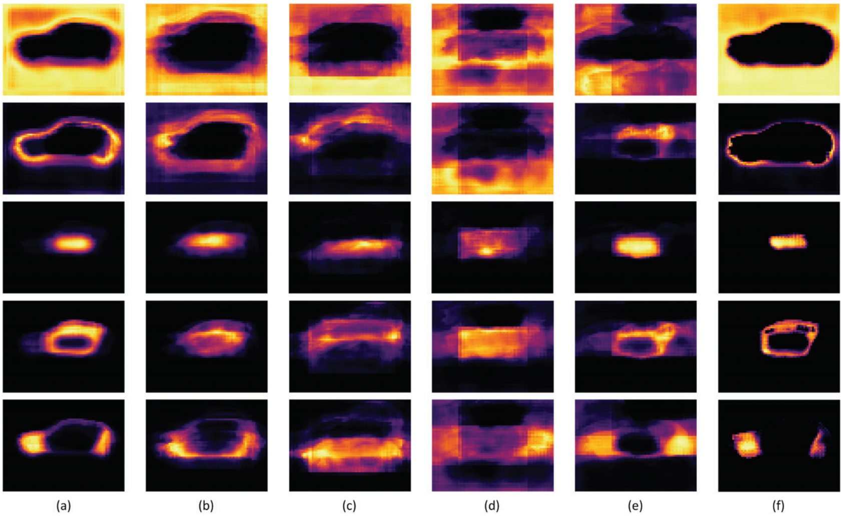


Figure 9. Converged responsibility maps collected from EMA units from different levels. (**a**) Maps from DA-EMA block 1. (**b**) Maps from DA-EMA block 2. (**c**) Maps from DA-EMA block 3. (**d**) Maps from DA-EMA block 4. (**e**) Maps from DA-EMA block 5. (**f**) Maps from output EMA block.

3.4. Experiment of the Entire Network

Benefiting from the multi-task learning mechanism, the entire network for EV fire trace recognition combined two branches and achieved better performance than using the single severity segmentation branch only. To demonstrate this improvement, different configurations of training and output were implemented using the proposed network, and the results are shown in Table 8 and Figure 10. In the joint training method, the λ value for loss calculation was set to 0.25 based on the ratio of the loss value while each branc converged.

Table 8. Results of different training methods and output methods. "Branch#2" stands for training the severity segmentation branch only.

Training Method	Output Classes	BG	IN	MB	SB	mIoU	VB	FT
Branch#2	4	95.30	77.12	42.68	52.73	66.96	94.03	71.00
2-Stage	4	95.84	78.52	45.54	52.59	68.12	94.72	72.44
Joint	4	95.63	77.86	43.81	53.57	67.72	94.86	72.79
2-Stage	3	96.15	79.17	45.11	53.80	68.56	95.10	71.79
Joint	3	95.70	78.92	45.96	55.11	68.92	94.50	73.17

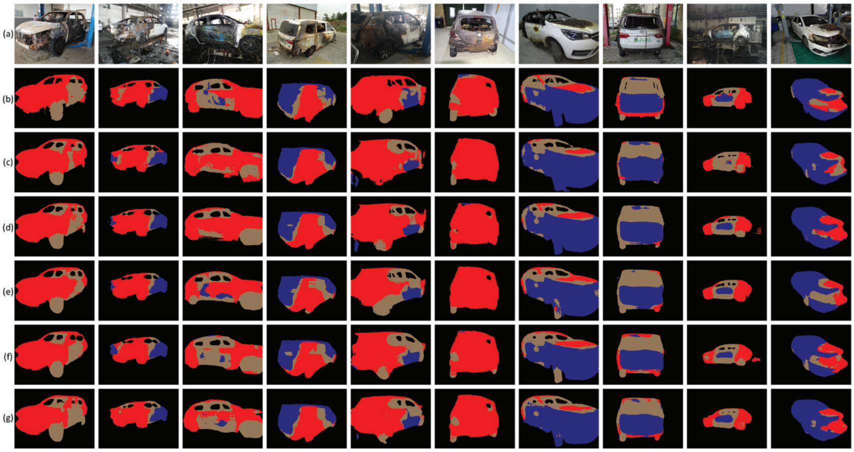


Figure 10. Results of the proposed DA-EMA in different training and output configurations. (**a**) Original images. (**b**) Labeled images. (**c**) Results of the single severity segmentation branch. (**d**) Results of the two-stage training and 4-class output. (**e**) Results of the joint training and 4-class output. (**f**) Results of the two-stage training and 3-class output. (**g**) Results of the joint training and 3-class output.

4. Discussion

To evaluate the foreground extraction branch, two experiments were conducted: a loss function parameter experiment and an ablation experiment. By tuning the value of hyperparameters in the proposed CCE loss function, we concluded that by using the loss function with an appropriate value of hyperparameters, the performance of the foreground extraction branch was improved. While the value is relatively small, it, on the contrary, hindered the convergence of the network. Once the value was extremely big, the function degenerated into normal cross entropy loss and lost its ability. Moreover, by conducting the ablation experiment, we found both the modified ASPP module and the CCE loss function had a positive effect on the branch.

The transfer learning method is essential in this paper, especially in the foreground extraction branch. By using weights pretrained on an enormous public dataset including labeled intact vehicles, the branch converged rapidly, and obtained great IoU results of over 95%. Due to the fine results obtained from the selected backbone and modified ASPP using pretrained weights, it is enough to use the simple ASPP module and CCE loss function for the foreground extraction task. More complicated models could not cause considerable excessive improvement. However, the severity segmentation task would be less benefited from the transfer learning method, which was also the reason for splitting the whole fire trace recognition task into two sub-tasks and focusing on a new module for the enhancement of feature extraction and expression. Therefore, the DA-EMA module with densely connected dilated convolution layers and a lightweight expectation maximization attention mechanism was proposed in the severity segmentation branch for the EV fire trace recognition task.

Regarding the experiments on the severity segmentation branch, we first compared the performance of the proposed DA-EMA module and other mainstream semantic segmentation models. The results in Table 6 showed that the proposed DA-EMA module achieved better accuracy in comparison to many mainstream networks. Moreover, according to Figure 8, due to the combination of the contextual mechanism and attention mechanism, outputs of the proposed DA-EMA module were more detailed than models with attention models, e.g., DANet and PSANet, and emphasized burnt regions more than models with contextual information, e.g., PSPNet and DeeplabV3. In addition, for EVs with slightly burnt bodies, the proposed DA-EMA module generated less error when classifying intact regions into burnt regions. For EVs with windows broken and internal structures or background exposed behind the glass, the proposed DA-EMA could better recognize regions behind the broken windows. Moreover, some models might wrongly recognize components, e.g., air inlets and intact tires, as burnt regions, but these issues were barely present with the proposed DA-EMA module. The other experiment evaluating the performance of the proposed DA-EMA module was an ablation experiment conducted by separately utilizing the DenseASPP-like structure with multiple dilated convolution layers and only one EMA module without a multi-scale structure. As a result, both the dense structure and EMA module had a positive impact on the overall performance. Moreover, the visualization of responsibility maps showed that bases of EMA units were converged to a certain concept of the input image, e.g., regions of different severities, contours of EV, and backgrounds. Though responsibility maps became more abstract and diffused as dilation rate increased, representations of different concepts were not reduced.

To prove that the performance improvement benefited from the multi-task learning mechanism by combining two branches, different training methods and number of classes of the severity branch were tested. According to the results shown in Table 8, by setting the background as an ignored label and predicting only three classes of severity levels, the severity segmentation branch output fewer errors than when taking the background class into consideration. When the two-stage training method was applied, backbone parameters were frozen after the foreground branch was trained, and the parameters did not change while training the severity branch. Therefore, the output of the foreground mask was much more close to the best performance achieved by training the foreground only. However, by training the two branches jointly and making the severity segmentation branch output only three classes, the whole model achieved the best performance.

Although the proposed DA-EMA module achieved better accuracy than other mainstream semantic segmentation models and the two-branched model also improved the overall performance further, the model still has some room for improvement. Firstly, the number of parameters of the network, especially the number of parameters of the backbone and the modified ASPP with more output channels in the foreground extraction branch is large, thus raising the time consumption of model training and inference. Though the task of this paper does not have a real-time requirement, there is still room for simplifying the model by reducing redundant components. Secondly, the size of dataset is relatively small, and white is the major color of EV bodies. Therefore, a lack of EV samples of different colors may lead to error when inferring EVs with rare colors or complicated paintings. Thirdly, restricted by the computing capacity, the resolution of images was relatively insufficient for expressing many detailed features. To solve this problem, a modified model with the capacity of processing larger images should be implemented.

5. Conclusions

In this paper, we used semantic segmentation techniques for recognizing traces of different severity levels from burnt EV images. A corresponding model with two branches separately concentrating on the foreground extraction task and the severity segmentation task was proposed, the backbone of which was ResNet101 with dilated convolution. Benefiting from the feature similarity between intact vehicles from a public dataset for pretraining and burnt vehicles from a dataset built in this paper, transfer learning considerably improved the overall accuracy of the foreground extraction task. Along with the modified ASPP module and proposed CCE loss function, the foreground extraction branch achieved an IoU of 95.16%. In the severity segmentation branch, to better enhance the feature representation capacity, a module combining the DenseASPP-like dense architecture and attention module named EMA was proposed. Achieving a mIoU of 66.96%, the proposed severity segmentation branch was tested and found to fit the task of the paper better than the other mainstream networks. Finally, by combining the two branches together, the whole multi-task based model was evaluated under different configurations of training and output, and the mIoU was finally improved to 68.92% while jointly training two branches and setting the background as ignored in the severity segmentation branch.

However, the proposed model has some limitations in certain scenarios. First, it is limited by the scale of dataset, as the majority of EV bodies are white. The lack of images of EVs with rare colors in dataset may cause errors when recognizing fire traces on EVs with these colors. To solve this problem, continuing to expand the dataset is the most efficient method. Second, although the gridding effect of the DA-EMA module was alleviated by modifying the dilation rates, the dilated convolution layers of the backbone were not optimized, and thus, the gridding effect still existed, especially in the foreground mask output from the foreground extraction branch. Third, the proposed CCE loss function in the foreground extraction branch did assist in eliminating FP areas, but when jointly training two branches, the λ was set to 0.25, which may weaken the function of CCE loss. As many FP areas were caused by other vehicle bodies, the best solution would be to apply the instance segmentation method to the foreground segmentation branch. Instance segmentation would classify pixel clusters of vehicle and distinguish which cluster belongs to which vehicle. By using this, the FP areas of other vehicle bodies can be conveniently removed. The problems above are shown in Figure 11.

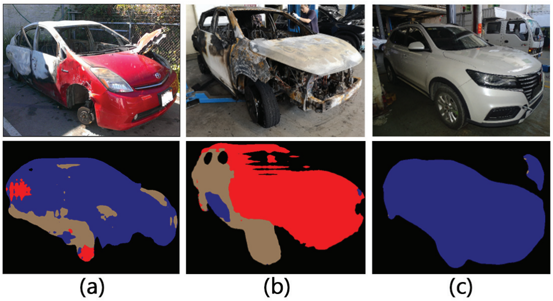


Figure 11. Limitations of the model. (**a**) Error of a red vehicle. (**b**) Gridding. (**c**) FP area from other vehicles.

Supplementary Materials: The proposed model and an executable demo are available at: https://github.com/Jkreat/EVFTR (accessed on 27 May 2022).

Author Contributions: Conceptualization, W.Z. and J.P.; methodology, J.P.; formal analysis, J.P.; investigation, J.P.; resources, W.Z.; software, J.P.; validation, J.P.; data curation, J.P.; writing—original draft preparation, J.P.; writing—review and editing, W.Z.; visualization, J.P.; supervision, W.Z.; project administration, W.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Restrictions apply to the availability of these data. Data was obtained from Tianjin Fire Research Institute of M.E.M. and are available from the authors with the permission of Tianjin Fire Research Institute of M.E.M.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Sun, P.; Bisschop, R.; Niu, H.; Huang, X. A Review of Battery Fires in Electric Vehicles. *Fire Technol.* **2020**, *56*, 1361–1410. [CrossRef]
2. Li, H.; Peng, W.; Yang, X.; Chen, H.; Sun, J.; Wang, Q. Full-scale experimental study on the combustion behavior of lithium ion battery pack used for electric vehicle. *Fire Technol.* **2020**, *56*, 2545–2564 [CrossRef]
3. Nicholas, J.S.; William, H.; Gregory, E.G.; Ronald, L.H.; Patrick, M.K. Vehicle Fire Burn Pattern Study. In Proceedings of the International Symposium on Fire Investigation Science and Technology, Adelphi, MD, USA, 27–29 September 2010.
4. Shields, L.E.; Scheibe, R.R. Computer-Based Training in Vehicle Fire Investigation Part 2: Fuel Sources and Burn Patterns. *SAE Tech. Paper* **2006**. [CrossRef]
5. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015.
6. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid scene parsing network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
7. Chen, L.C.; Papandreou, G.; Schroff, F.; Adam, H. Rethinking atrous convolution for semantic image segmentation. *arXiv* **2017**, arXiv:1706.05587.
8. Chen, L.C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-decoder with atrous separable convolution for semantic image segmentation. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 801–818.
9. Fu, J.; Liu, J.; Tian, H.; Li, Y.; Bao, Y.; Fang, Z.; Lu, H. Dual attention network for scene segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 3146–3154.
10. Zhao, H.; Zhang, Y.; Liu, S.; Shi, J.; Loy, C.C.; Lin, D.; Jia, J. Psanet: Point-wise spatial attention network for scene parsing. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 267–283.
11. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
12. Yu, F.; Koltun, V.; Funkhouser, T. Dilated residual networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–27 July 2017; pp. 472–480.
13. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
14. Wang, J.; Sun, K.; Cheng, T.; Jiang, B.; Deng, C.; Zhao, Y.; Liu, D.; Mu, Y.; Tan, M.; Wang, X.; et al. Deep high-resolution representation learning for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *43*, 3349–3364. [CrossRef] [PubMed]
15. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Munich, Germany, 5–9 October 2015; pp. 234–241.
16. Milletari, F.; Navab, N.; Ahmadi, S.A. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In Proceedings of the International Conference on 3D Vision, Stanford, CA, USA, 25–28 October 2016; pp. 565–571.
17. Zhou, Z.; Siddiquee, M.M.R.; Tajbakhsh, N.; Liang, J. Unet++: A nested U-net architecture for medical image segmentation. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*; Springer: Granada, Spain, 2018; pp. 3–11.
18. Charles, R.Q.; Su, H.; Kaichun, M.; Guibas, L.J. PointNet: Deep learning on point sets for 3D classification and segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Los Alamitos, CA, USA, 21–26 July 2017; pp. 77–85.
19. Kirillov, A.; He, K.; Girshick, R.; Rother, C.; Dollár, P. Panoptic segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 9396–9405.
20. Wang, G.; Zhang, Y.; Qu, Y.; Chen, Y.; Maqsood, H. Early forest fire region segmentation based on deep learning. In Proceedings of the Chinese Control And Decision Conference, Nanchang, China, 3–5 June 2019; pp. 6237–6241.

21. Zhang, J.; Zhu, H.; Wang, P.; Ling, X. Att squeeze u-net: A lightweight network for forest fire detection and recognition. *IEEE Access* **2021**, *9*, 10858–10870. [CrossRef]
22. Mseddi, W.S.; Ghali, R.; Jmal, M.; Attia, R. Fire detection and segmentation using YOLOv5 and U-net. In Proceedings of the European Signal Processing Conference, Dublin, Ireland, 23–27 August 2021; pp. 741–745.
23. Chen, G.; Zhang, X.; Wang, Q.; Dai, F.; Gong, Y.; Zhu, K. Symmetrical dense-shortcut deep fully convolutional networks for semantic segmentation of very-high-resolution remote sensing images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2018**, *11*, 1633–1644. [CrossRef]
24. Zhang, Y.; Chen, Y.; Ma, Q.; He, C.; Cheng, J. Dual lightweight network with attention and feature fusion for semantic segmentation of high-Resolution remote sensing images. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, Brussels, Belgium, 11–16 July 2021; pp. 2755–2758.
25. Yang, M.; Yu, K.; Zhang, C.; Li, Z.; Yang, K. Denseaspp for semantic segmentation in street scenes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 3684–3692.
26. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
27. Ji, Y.; Zhang, H.; Wu, Q.J. Salient object detection via multi-scale attention CNN. *Neurocomputing* **2018**, *322*, 130–140. [CrossRef]
28. Li, X.; Zhong, Z.; Wu, J.; Yang, Y.; Lin, Z.; Liu, H. Expectation-maximization attention networks for semantic segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27–28 October 2019; pp. 9167–9176.
29. Wang, P.; Chen, P.; Yuan, Y.; Liu, D.; Huang, Z.; Hou, X.; Cottrell, G. Understanding convolution for semantic segmentation. In Proceedings of the IEEE Winter Conference on Applications of Computer Vision, Lake Tahoe, NV, USA 12–15 March 2018; pp. 1451–1460.
30. Kuhn, M.; Johnson, K. *Applied Predictive Modeling*; Springer: New York, NY, USA, 2013.
31. Rodriguez, J.D.; Perez, A.; Lozano, J.A. Sensitivity analysis of k-fold cross validation in prediction error estimation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *32*, 569–575. [CrossRef] [PubMed]
32. Wang, Y.; Zhou, Q.; Liu, J.; Xiong, J.; Gao, G.; Wu, X.; Latecki, L.J. Lednet: A lightweight encoder-decoder network for real-time semantic segmentation. In Proceedings of the IEEE International Conference on Image Processing, Taipei, Taiwan, 22–25 September 2019; pp. 1860–1864.
33. Yuan, Y.; Chen, X.; Wang, J. Object-contextual representations for semantic segmentation. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 173–190.

sustainability

Article

Detection and Prevention of False Data Injection Attacks in the Measurement Infrastructure of Smart Grids

Muhammad Awais Shahid [1,*], Fiaz Ahmad [1,*], Fahad R. Albogamy [2], Ghulam Hafeez [3,*] and Zahid Ullah [4]

[1] Department of Electrical & Computer Engineering, Air University, Islamabad 44230, Pakistan
[2] Computer Sciences Program, Turabah University College, Taif University, P.O. Box 11099, Taif 21944, Saudi Arabia; f.alhammdani@tu.edu.sa
[3] Department of Electrical Engineering, University of Engineering and Technology, Mardan 23200, Pakistan
[4] Department of Electrical Engineering, University of Management and Technology Lahore, Sialkot Campus, Sialkot 51310, Pakistan; zahid.ullah@skt.umt.edu.pk
[*] Correspondence: awaisshahid.au.2020@gmail.com (M.A.S.); fiaz.ahmad@mail.au.edu.pk (F.A.); ghulamhafeez393@gmail.com (G.H.)

Abstract: The smart grid has become a cyber-physical system and the more cyber it becomes, the more prone it is to cyber-attacks. One of the most important cyber-attacks in smart grids is false data injection (FDI) into its measurement infrastructure. This attack could manipulate the control center in a way to execute wrong control actions on various generating units, causing system instabilities that could ultimately lead to power system blackouts. In this study, a novel false data detection and prevention paradigm was proposed for the measurement infrastructure in smart grids. Two techniques were devised to manage cyber-attacks, namely, the fixed dummy value model and the variable dummy value model. Limitations of the fixed dummy value model were identified and addressed in the variable dummy value model. Both methods were tested on an IEEE 14 bus system and it was shown through the results that an FDI attack that easily bypassed the bad data filter of the state estimator was successfully identified by the fixed dummy model. Second, attacks that were overlooked by the fixed dummy model were identified by the variable dummy method. In this way, the power system was protected from FDI attacks.

Keywords: smart grid; cyber-physical system; false data injection attacks; false data detection; cyber security

Citation: Shahid, M.A.; Ahmad, F.; Albogamy, F.R.; Hafeez, G.; Ullah, Z. Detection and Prevention of False Data Injection Attacks in the Measurement Infrastructure of Smart Grids. *Sustainability* **2022**, *14*, 6407. https://doi.org/10.3390/su14116407

Academic Editors: Luis Hernández-Callejo, Sergio Nesmachnow and Sara Gallardo Saavedra

Received: 30 March 2022
Accepted: 20 May 2022
Published: 24 May 2022

1. Introduction

"Smart grid" is taken as an umbrella term for different technologies. Those technologies are considered alternatives to the traditional methods used to operate the power system. Some of these technologies are advanced metering infrastructure (AMI), demand response, outage management, wide-area measurement system (WAMS), active fault level monitoring, etc. In a smart grid, the power resources can be used efficiently [1,2]. A smart grid has a high dependence on the advanced communication infrastructure, as there is an exchange of a huge amount of data for the proper operation of such a complex network [3–5]. In fact, the smart grid is taken as a network consisting of computers, as well as power infrastructure. All of these are used for monitoring and managing energy usage [6,7]. An automated and distributed energy network is created by the smart grid [8]. Self-monitoring is carried out in the case of a smart grid, which makes the smart grid distinct from a traditional grid [9]. Distributed power resources (DPR) can be accommodated in a smart grid [10,11].

In a power system, if there exists a mismatch between the generation and utilization of power, there will be a deviation of electrical quantities from their actual values. The two-way communication is carried out in a smart grid to have a safe and reliable power flow. That communication should be secure. Sometimes attackers hack these communication links to change the values of power flow in the power network. The hackers attack the power system to obtain different goals. Multiple purposes can be achieved by these attacks. Attacks can be used to obtain financial benefits; create technical problems, such as blackouts of power; and a combination of the two [12–18].

Considering the target of attacks, they can be further divided into three types. The first category involves attacks that target availability. In these attacks, the aim of the attackers is to corrupt, block, or delay the communication in the power system. The second type is attacks that target integrity. In these types of attacks, the attackers try to illegally disrupt data exchange in the smart grid. The final category is attacks that target confidentiality. In these attacks, the attackers try to obtain unauthorized information from the smart network [19].

1.1. Power System State Estimation

The power system state estimation (PSSE) technique is used for the detection of bad data received in the control room. All the received measurements are placed in a vector, which is denoted by **z**. The measurement vector contains the real forward powers, reactive forward powers, real backward powers, reactive backward powers, real powers injected into all the buses, reactive powers injected into all buses, voltage magnitudes, and voltage angles [20–23]. The measurement vector **z** and the state variable **x** have the following relationship:

$$\mathbf{z} = \mathbf{h}(\mathbf{x}) + \mathbf{e} \tag{1}$$

h(x) represents the non-linear function that gives the dependencies between measured values and the state variables, and it can be found using the power system topology. **e** represents random noise of Gaussian form with a zero mean and some known covariance.

In the case of AC state estimation (SE), the weighted least-squares method is adopted for solving the state variables with an objective function [24,25]:

$$\min F(\mathbf{x}) = (\mathbf{z} - \mathbf{h}(\mathbf{x}))^{\mathrm{T}}\mathbf{W}(\mathbf{z} - \mathbf{h}(\mathbf{x})) \tag{2}$$

where **W** is the weighting matrix, as given in [26]. This is an unconstrained optimization problem whose first-order optimality condition is given by:

$$\left.\frac{\partial F(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}} = -2\mathbf{H}^{\mathrm{T}}(\hat{\mathbf{x}})\mathbf{W}(\mathbf{z} - \mathbf{h}(\hat{\mathbf{x}})) = 0 \tag{3}$$

Here, **H** represents the Jacobian matrix and $\hat{\mathbf{x}}$ is taken as the vector of the estimated states. An iterative process can be used for solving this non-linear equation [27].

The non-linear function can be approximated by a linear function by using some DC assumptions. Those assumptions are given as follows:

1. The voltage magnitudes of all the buses are very close to each other and they are assumed to be "1 pu".
2. The active power transmission through the transmission lines is taken as lossless, i.e., there are no losses in the transmission lines.
3. The value of reactive power injected into all the buses, as well as flowing through the transmission lines, is taken as zero.
4. There is a small difference in the voltage angles of two buses such that "*Sin*($\delta\phi$) $\approx \delta\phi$"

After applying the DC assumptions, we can rewrite the above equations in this form:

$$\mathbf{z} = \mathbf{H}\mathbf{x} + \mathbf{e} \tag{4}$$

H is known as the Jacobian matrix of the power system topology. If the measurement vector has m values and the number of states is n, then the Jacobian matrix **H** will have an order of "$m \times n$". In (4), **x** contains the bus voltage angles. **z** contains the values of active powers flowing through the transmission lines and injected into all the buses.

The Jacobian matrix **H** is constant during each iteration of the linearization process. In the DC power flow model (4), the Jacobian matrix **H** is constant throughout. Equation (4) will be valid for each iteration of the linearization model (3). Therefore, the same notation is adopted for both the linearized model (3) and the DC power flow model (4).

The weighted least square (WLS) approach is used for estimating the states. In the WLS algorithm, the estimated state $\hat{\mathbf{x}}$ can be written as follows [19,22]:

$$\hat{\mathbf{x}} = \left(\mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H}\right)^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{z} \tag{5}$$

R represents the covariance matrix of **e**. The estimated states, as well as the measurement vector **z**, are used for the calculation of the measurement residue.

$$\mathbf{r} = \mathbf{z} - \mathbf{H}\hat{\mathbf{x}} \tag{6}$$

Then, the normalized L_2-norm is calculated for **r**.

$$L(\mathbf{r}) = \mathbf{r}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{r} \tag{7}$$

A comparison of $L(\mathbf{r})$ is done with the threshold $\boldsymbol{\tau}$ for finding the presence of bad data. The X_2—test is used for the determination of the threshold $\boldsymbol{\tau}$.

$$\mathbf{r}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{r} \leq \boldsymbol{\tau} \tag{8}$$

Bad data do not exist if the condition in (8) is satisfied. Similarly, when the condition is not satisfied, bad data exist in the system.

1.2. Stealth False Data Injection (FDI) Attack

A stealth attack is a special type of attack that bypasses the PSSE technique test. The residual test is not able to detect a stealth attack. This attack is also known as an unobservable attack or undetectable attack. In a stealth attack, the Jacobian matrix **H** is fully known to the attacker. **H** is used for the construction of an undetectable attack. Stealth false data injection (FDI) is given as follows [17,19,22,28–30]:

$$\mathbf{z_a} = \mathbf{z} + \mathbf{a} \tag{9}$$

where **a** represents the vector of false data that is added to the measurement vector **z**. The attacker hacks the data from the communication line and injects the attack vector **a** into it, where **a** = **Hc**.

The attack is done on the communication line by the attacker and all measurements of power are hacked. The Jacobian matrix **H** is determined with the help of those measurements of power. The whole power system topology can be understood with the help of **H**. The dependence of one power value on the other powers can be found using **H**. This leads the attacker to make an undetectable attack. In fact, it tells the attacker which specific values of power the attacker will have to change with one particular change in power. To understand the whole power network, the formation of the Jacobian matrix **H** is the most important component. The vector **c** is multiplied by matrix **H** and the resultant is added to the actual measurements when undertaking a stealth attack.

The stealth attack is executed against the PSSE in the power network and that is the attack of injecting false data into the system measurements. The state estimation technique is bypassed by the stealth attack [31]. In case of an attack, the estimated state becomes:

$$\hat{\mathbf{x}}_{\mathbf{a}} = \left(\mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H}\right)^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{z_a} \tag{10}$$

$$\hat{\mathbf{x}}_{\mathbf{a}} = \hat{\mathbf{x}} + \mathbf{c} \tag{11}$$

The estimated state is changed in the case of a stealth attack. Now, the estimated state is equal to the original estimated state plus the addition of a constant vector **c**. It is assumed that **c**~$N(0,\sigma_c^2)$, where the false state variance is represented by σ_c^2.

$$\mathbf{z_a} = \mathbf{Hx} + \mathbf{e} + \mathbf{a} = \mathbf{Hx} + \mathbf{e} + \mathbf{Hc} \tag{12}$$

$$\mathbf{z_a} = \mathbf{Hx_a} + \mathbf{e} \tag{13}$$

Therefore, the attack changes the state of the power system. The technique used in the system for bad data detection is bypassed by the stealth false data injection attack in this way:

$$\mathbf{r_a} = \mathbf{z_a} - \mathbf{H\hat{x}_a} = \mathbf{r} \tag{14}$$

The attacked residual is represented by $\mathbf{r_a}$. In the attack, the attacked residue is the same as that of the normal residue. Therefore, the technique of bad data detection using residue is bypassed by this attack and the defender is not able to detect the stealth attack.

1.3. Contributions

The key contributions of this study are the following:

1. It was shown that the bad data filter of the state estimation was only useful for detecting bad measurement data and could not efficiently detect a stealth FDI, making the system vulnerable to all such attacks.
2. A fixed dummy value model was proposed and it was shown that the false data attacks that went undetected by the bad data filter could be successfully detected.
3. Since the dummy value in the fixed dummy value model is kept fixed, the intruder may obtain a clue about it and may change the measurement, keeping the same dummy value, therefore causing this model to be vulnerable to FDI attacks. To address the vulnerability of the fixed dummy value model, another technique for the variable dummy value model was also proposed, which was shown to successfully counter such attacks.

In this work, a quasi-steady state system was assumed to carry out the AC state estimation model. Therefore, the dynamic model of the system is not discussed. For the AC state estimator, the Jacobian matrix H is obtained after the linearization of the measurement model at every iteration. Thus, Equation (5) is solved at each iteration until a stopping threshold is reached.

The organization of the rest of the paper is as follows. Section 2 contains a brief literature review of the different methods and frameworks used for the detection of attacks. Section 3 consists of the proposed model of the fixed dummy value model. Section 4 discusses the simulations and results of DC state estimation, AC state estimation, and fixed dummy value model. The limitations of the fixed dummy value model are also given in that section. Section 5 covers the variable dummy value model. The simulations and results of the variable dummy value model are present in Section 6. Moreover, Section 7 is devoted to the discussion of results and future work. Section 8 contains the conclusion.

2. Literature Review

A large variety of methods and algorithms have been used for detecting stealth attacks. Machine learning methods achieved significant success in this area. In [32], supervised learning based on recurrent neural networks (RNNs) was used for detecting FDI attacks. In [16], three supervised machine learning classifiers, namely, SVM, k-nearest neighbor (kNN), and the extended nearest neighbor (ENN), were used. Different machine learning algorithms are proposed in [28] for measurement classification. Measurements are classified as attacked or secure. Sparse logistic regression, SVM, and k-nearest neighbor methods were used in that study. Another technique was proposed in [33] for the detection of FDI attacks, which used the Gaussian mixture model. The contribution in [34] was based on unsupervised learning. Four machine learning methods, namely, a one-class SVM, local outlier factor, isolation forest, and robust covariance estimation, were employed for FDI attack detection. In [35], a machine-learning-based scheme was used that employed ensemble learning. In ensemble learning, there is a use of multiple classifiers, and the decisions obtained by the individual classifiers are further classified. The proposed scheme used two ensembles. Supervised classifiers were used in the first ensemble and the unsupervised classifiers were employed in the second ensemble. Supervised learning was proposed in [36], which used a two-layer hierarchical framework. The first layer distinguished the mode of operation, such as a normal state or cyberattack. The second layer classified the type of cyberattack. An approach based on machine learning was adopted in [37] for cyber-attacks, which used an extremely randomized trees algorithm. In [38], three machine learning techniques, namely, a support vector machine (SVM), k-nearest neighbor, and artificial neural network, were implemented for detecting FDI attacks. Each technique was used with three different feature selection techniques.

An extreme learning machine framework was used in [39] for detecting FDI attacks. In [40], auto-encoders were used for detecting FDI attacks. The hidden correlation structures were learned in the data by using auto-encoders. The correlation was learned in two dimensions, namely, the time and the spatial dimensions. Denoising auto-encoders were also used to clean the corrupted data. The approaches based on the auto-encoder neural network [41] and attention-based auto-encoders [42] were also used for the detection of attacks.

The contribution of [15] distinguished the normal function of the power system from the function in which there was a stealth attack. The stealth attacks were detected by using two machine-learning-based techniques. In the first technique, supervised learning was used for a set of labeled data. That data was used for the training of a support vector machine (SVM). The second technique did not use any training data and the deviation of the measurements was detected. An anomaly detection algorithm was applied to detect stealth attacks.

Deep learning models were also used for the purpose of detecting FDI attacks. The deep neural network (DNN) model was used [43] for the classification of cyber-attacks in a smart grid. Another deep learning-based method was proposed in [44] to detect FDI attacks. The proposed approach consisted of a convolutional neural network (CNN) and a long short-term memory (LSTM) network for the detection of attacks. The data integrity attacks in AC power systems can be detected by using a deep Q-network detection (DQND) scheme proposed in [45]. It is a deep reinforcement learning approach. A neural network model was used in [46] for detecting false data. In this case, the residual elements obtained from state estimation were the inputs given to the perceptron model. An algorithm based on deep learning was proposed in [47] to detect FDI attacks. The dimensionality reduction, as well as feature extraction from measurement datasets, was done by using auto-encoders. Then auto-encoders were integrated into an advanced generative adversarial network (GAN) framework, which was used for detecting the FDI attacks.

The methods based on machine learning had great success in the detection of FDI attacks. However, at the same time, they have certain limitations and drawbacks. The methods based on supervised learning need a labeled dataset. They are built on some conventional attack assumptions. Similarly, deep learning techniques also have some limitations. In these methods, there is a need for extensive training. More memory space is also required for deep learning methods.

The main aim of the detection frameworks is to protect the whole communication system against attacks. One of the key features of microgrids is a secure communication network. For the development of a communication network, its design has vital importance. For the deployment of a heterogeneous automation and monitoring system, a multi-layered architecture was proposed in [48]. For the organization of hardware, as well as software equipment in an integrated manner, six functional layers were structured in the proposed architecture. In [49], a clear description of a smart grid and the type of communication methods were given. The communication methods were explained based on their advantages and the lacking feature. The contribution of [50] was based on the hybrid communication simulation model. In hybrid network architectures, both wireless and dedicated wired media are used. A suite of hybrid communication simulation models was developed for the validation of critical system design criteria.

A mathematical model of the power system was presented in [51] and a robust security framework was proposed. A Kalman filter was used to estimate variables in the model. In [52], an online data-driven algorithm was presented for detecting FDI attacks toward synchrophasor measurements. The proposed algorithm applied density-based LOF (local outlier factor) analysis for detecting anomalies in the data. Another method was proposed in [53] in which the modeling of the system was done as a discrete-time linear dynamic system. There was the use of the Kalman filter for performing the state estimation (SE). A generalized cumulative sum algorithm achieved the quickest detection of the attacks. In [18,19], the economic impact due to stealth FDI attacks on the market operations in real-time was considered. The construction of a profitable attacking plan for the attacker was also shown. In [20], it was explained that the attacker can construct the stealth FDI attack without knowing the structure of the system. The attacker can find the system structure and make an attack.

In [54], a distributed state estimation method based on the alternating direction method of multipliers (ADMM) was presented for detecting cyber-attacks. In this case, the partitioning of regional subsystems was done using the K-means method. An online detection algorithm was proposed in [55] for detecting cyber-attacks. The online estimation of the unknown and time-varying attack parameters was provided by the algorithm. The FDI attacks were detected by proposing an active data modification scheme in [56]. In that scheme, there was an amendment of measurements and control data before they are transmitted through communication networks. In [57], an FDI attack detection method was proposed that was based on the equivalent model of a load frequency control (LFC) system and a Kalman filter algorithm.

The work of [21] formulated the problem of false data detection as a low-rank matrix recovery. Convex optimization was used for solving the problem. The adopted methodology normalized the combination of the l_1 norm and nuclear norm. This mixed norm optimization problem was solved using the augmented Lagrange method of multipliers in order to obtain a good convergence rate. In [22], the false data detection problem was considered a matrix separation problem. FDI attacks are sparse in nature. To separate the states of the power system from the anomalies, a mechanism was developed. The problem was solved using two methods, namely, low-rank matrix factorization and nuclear norm minimization.

3. Proposed Model

The methods used in the literature for the detection of attacks are successful up to a certain limit. If the attacker knows the whole network of the smart grid and makes an attack, it becomes difficult to detect those attacks. Therefore, we proposed a new power system model for an AC power flow network that is safe against stealth FDI attacks and the control room is able to detect these attacks in an efficient manner. The introduced model was based on the concept of dummy value. The smart grid meters will transmit both values, i.e., the actual value and the dummy value. No additional transmission lines and no extra buses will be used. There is no need for any extra meters in the proposed model. The vulnerabilities of the communication networks in supervisory control and data acquisition (SCADA) systems in the smart grid, such as unsophisticated bugs or communication failures, were not considered in this work. The application of the measured value and the dummy value in this article did not consider the error caused by the measurement equipment itself or any other reason. In this work, the error due to parametric variation of the meter or any other unknown reason was not taken into consideration. However, it may be incorporated into our future work. Moreover, this work focused on false data injection attacks in which the intruder hacks the measurement vector and injects the attack vector into the measurement vector before it is received by the control room. Therefore, this study only considered targeted attacks.

The measurement vector for the AC power flow network contains the active and reactive powers injected into all the buses, active and reactive powers flowing through transmission lines in the forward direction, and active and reactive powers flowing in the backward direction. If a system has *b* number of buses and *t* number of transmission lines, then the measurement vector for the AC power flow network is given by

$$\mathbf{z_y} = \begin{bmatrix} \mathbf{p_{v(y)}} & \mathbf{q_{v(y)}} & \mathbf{p_{vw(y)}} & \mathbf{q_{vw(y)}} & \mathbf{p_{wv(y)}} & \mathbf{q_{wv(y)}} \end{bmatrix}^{\mathrm{T}} \tag{15}$$

where $\mathbf{z}_y$ is the measurement vector at the *y*th instant and $\mathbf{y}$ = 1, 2, 3, . . . , *mt*. Here, *mt* represents the total number of instances. $\mathbf{p_{v(y)}}$ and $\mathbf{q_{v(y)}}$ are the vectors containing the active and reactive powers injected to all the buses at the *y*th instant. Both vectors will have a dimension of 1 × *b*. Similarly, $\mathbf{p_{vw(y)}}$ and $\mathbf{q_{vw(y)}}$ denote vectors having the active and reactive powers flowing through all the transmission in the forward direction at the *y*th instant. Both vectors have dimensions of 1 × *t*. Moreover, $\mathbf{p_{wv(y)}}$ and $\mathbf{q_{wv(y)}}$ represent the vectors of the active and reactive powers flowing through all the transmission lines in the backward direction at the *y*th instant. The complete measurement vector will have a dimension of *m* × 1. The state vector **x** contains the voltage magnitudes and voltage angles of all the buses. However, the Jacobian matrix will have a dimension of *m* × *n*, where *m* is the total number of values in the measurement vector and *n* is the total number of values in the state vector. The measurement vectors at all the instances can be placed together to obtain the measurement matrix as follows:

$$\mathbf{Z} = [\mathbf{z}_1 \quad \mathbf{z}_2 \quad \mathbf{z}_3 \ldots\ldots\ldots\ldots\ldots\ldots \mathbf{z_{mt}}]^{\mathrm{T}} \tag{16}$$

The dimensions of the measurement matrix are *mt* × *m*. The measurement vector after implementing the proposed system will become like this:

$$\mathbf{z_{dy}} = \begin{bmatrix} p_{v(y)}(1); p'_{v(y)}(1); \ldots\ldots; q_{v(y)}(b); q'_{v(y)}(b); \\ p_{vw(y)}(1); p'_{vw(y)}(1); \ldots\ldots; q_{vw(y)}(t); q'_{vw(y)}(t); \\ p_{wv(y)}(1); p'_{wv(y)}(1); \ldots\ldots; q_{wv(y)}(t); q'_{wv(y)}(t) \end{bmatrix}$$

The measurement vector containing the actual and dummy values is represented by $\mathbf{z_{dy}}$. Here, $\mathbf{p_{v(y)}}(1)$ represents the first entry of the vector $\mathbf{p_{v(y)}}$ and $\mathbf{q_{v(y)}}(\mathbf{b})$ is the *b*th entry of the vector $\mathbf{q'_{v(y)}}$. The dummy values of the power are present on the even indexes of the new measurement vector. The vectors of the dummy values containing the active and reactive powers injected to all the buses at the *y*th instant are $\mathbf{p'_{v(y)}}$ and $\mathbf{q'_{v(y)}}$. Similarly, other vectors containing dummy values of the active and reactive powers for transmission lines at the *y*th instant are denoted by $\mathbf{p'_{vw(y)}}$, $\mathbf{q'_{vw(y)}}$, $\mathbf{p'_{wv(y)}}$, and $\mathbf{q'_{wv(y)}}$. $\mathbf{z_{dy}}$ will have dimensions of 2*m* × 1. The measurement matrix after including the dummy values will be

$$\mathbf{Z_d} = [\mathbf{z_{d1}} \quad \mathbf{z_{d2}} \quad \mathbf{z_{d3}} \ldots\ldots\ldots\ldots\ldots\ldots \mathbf{z_{dmt}}]^{\mathrm{T}} \tag{17}$$

This measurement vector will have dimensions of *mt* × 2*m*. The Jacobian matrix of the proposed system at the *y*th instant is represented by $\mathbf{H_{dy}}$ and its dimensions are 2*m* × *n*. There are different methods to find the Jacobian matrix. To make a stealth attack, it is necessary for the attacker to determine the Jacobian matrix. The attacker hacks both the dummy and actual values and creates a Jacobian matrix to attack the system.

Realistic data of the AC power flow network was used for implementing and evaluating the proposed model. For this purpose, the load curves of a transmission organization known as PJM, which serves 13 states of the United States and the District of Columbia, were taken as a reference to generate the data of the power flow network. These load curves were based on realistic data. Therefore, our generated data were very close to the realistic data of an AC power flow network. The data were generated for four different seasons, namely, summer, fall, winter, and spring, based on the standard realistic load curves given for each season.

The overall proposed model was divided into two scenarios. In the first scenario, a fixed dummy value was sent to the control room. However, in the second case, a variable dummy value was sent and it changed with the change of the actual value of power.

Fixed Dummy Value Model

In this case, a fixed dummy value, along with each of the actual values, was sent to the control room. The dummy value of the power was not dependent on the load. It did not vary with the variation in load or variation in the actual value of power. In the fixed dummy value model, to select the dummy value of a particular power, the average value was calculated from all the actual measured values that occurred for that value at all the instances. That average value was selected as the dummy value of power. In this case, the dummy values were determined by taking the mean of the last year's worth of historical measurement data, i.e., real-time measured values are stored from the past year and utilized for the calculation of fixed dummy values based on Equations (18)–(21). Later, these values were inserted into the memory of the meters and were simply appended or added to all the newly acquired measurements accordingly. It should be noted that the dummy value will no longer change with the newly acquired measurements. The calculation of the fixed dummy value was done by using these formulas:

$$\begin{gathered} p'_{v(y)}(l) = \frac{\sum_{s=1}^{mt} z_s(lp)}{mt} \\ l = 1,2,3,\ldots\ldots,b \quad and \quad lp = 1,2,3,\ldots\ldots,b \end{gathered} \tag{18}$$

$$\begin{gathered} q'_{v(y)}(l) = \frac{\sum_{s=1}^{mt} z_s(lq)}{mt} \\ l = 1,2,3,\ldots\ldots,b \quad and \quad lq = b+1,b+2,\ldots\ldots,2b \end{gathered} \tag{19}$$

$$\begin{gathered} p'_{vw(y)}(l) = \frac{\sum_{s=1}^{mt} z_s(lpv)}{mt} \\ l = 1,2,3,\ldots\ldots,t \quad and \quad lpv = 2b+1,2b+2,\ldots\ldots,2b+t \end{gathered} \tag{20}$$

$$\begin{gathered} q'_{vw(y)}(l) = \frac{\sum_{s=1}^{mt} z_s(lqv)}{mt} \\ l = 1,2,3,\ldots,t \quad and \quad lqv = 2b+t+1,2b+t+2,\ldots\ldots,2b+2t \end{gathered} \tag{21}$$

In (18), $p'_{v(y)}(l)$ represents the lth entry of the dummy values vector $\mathbf{p'_{v(y)}}$. $z_s(lp)$ denotes the lpth entry of the sth historical measurement vector. mt is the total number of instances for which the historical measurement vectors are obtained. To calculate the first entry of the dummy measurement vector $\mathbf{p'_{v(y)}}$, the sum of the first entries of all the historical measurement vectors is calculated and then divided by the total number of instances for which those historical measurement vectors are obtained. Similarly, the second entry of the dummy values vector $\mathbf{p'_{v(y)}}$ can be calculated by finding the mean of the second entries of mt historical measurement vectors. The same procedure is adopted for finding all the entries of $\mathbf{p'_{v(y)}}$ and the dummy values of all the active powers injected into the buses are calculated in this way. In (19), $q'_{v(y)}(l)$ denotes the lth entry of the dummy values vector $\mathbf{q'_{v(y)}}$, and $z_s(lq)$ represents the lqth entry of the sth historical measurement vector. mt gives the total number of historical measurement vectors. The lth entry of the dummy values vector $\mathbf{q'_{v(y)}}$ is found by calculating the mean of the lqth entry of mt historical measurement vectors. By using this procedure, the dummy values of all the reactive powers injected into the buses can be calculated. In (20) and (21), $p'_{vw(y)}(l)$ and $q'_{vw(y)}(l)$ represent the lth entry of each of the dummy measurement vectors $\mathbf{p'_{vw(y)}}$ and $\mathbf{q'_{vw(y)}}$, respectively. $z_s(lpv)$ and $z_s(lqv)$ denote the lpvth and lqvth entries of the sth historical measurement vector, respectively. The lth entry of each of the dummy measurement vectors $\mathbf{p'_{vw(y)}}$ and $\mathbf{q'_{vw(y)}}$ is calculated by finding the mean of the lpvth and lqvth entries of mt historical measurement vectors, respectively.

Therefore, the dummy values of the active and reactive powers flowing through all the transmission lines can be calculated by using Equations (20) and (21), respectively.

By applying Equations (18)–(21), the dummy values are calculated at a single instant by using *mt* historical measurement vectors and then those calculated dummy values are kept the same for all the instances, i.e., the dummy values do not change for the other instances. In fact, in the fixed dummy value model, the dummy values depend only on the historical measurement values and they do not depend on the real-time measurement values.

$\mathbf{p}'_{\mathbf{wv(y)}}$ and $\mathbf{q}'_{\mathbf{wv(y)}}$ can also be calculated using this method and all the dummy values are selected in this way. These dummy values are placed in $\mathbf{z_{dy}}$, which is embedded in the meters. These values are also placed in another vector **d** present in the control room. When the system is hacked by the attacker, 2*m* power values will be obtained by the attacker in $\mathbf{z_{dy}}$ instead of *m* values. In the next step, the Jacobian matrix will be constructed by the attacker and a stealthy attack will be done in this way:

$$\mathbf{z_{dyr}} = \mathbf{z_{dy}} + \mathbf{H_{dy}} * \mathbf{c} \tag{22}$$

Here, $\mathbf{z_{dyr}}$ denotes the measurement vector received in the control room at the *y*th instant. For the detection of an attack, a comparison is made between the dummy values obtained from $\mathbf{z_{dyr}}$ and those dummy values set by the control room. The following equation is used in the control room to detect the attack:

$$r(u) = d(u) - z_{dyr}(v) \tag{23}$$

where *u* = 1, 2, 3 . . . , *m* and

v = 2,4, 6 . . . , 2*m*

Here *d(u)* denotes the *u*th entry of the dummy values vector *d*, which is selected and set by the defender. Meanwhile, $z_{dyr}(v)$ denotes the *v*th entry of the received measurement vector. In the case of a secure system:

$$|r(u)| = 0 \tag{24}$$

where *u* = 1,2, 3, . . . , *m*.

During the case of no attack, $\mathbf{z_{dyr}} = \mathbf{z_{dy}}$.

To launch an attack, the attacker changes the actual and dummy values according to the construction of the stealth attack. As the dummy values are fixed, they should not change for a secure system. Therefore, for an attack, the value of $|r(u)|$ will come out to be greater than zero and the attack will be detected in this way.

The conventional technique for bad data detection (BDD), such as DC state estimation (SE), fails to detect a stealth FDI attack. Moreover, AC SE is also bypassed by this attack. However, our model with a fixed dummy value was capable enough to detect the FDI attacks in the AC power flow network and all the attacks could be detected by the control room.

4. Simulations and Results of the Fixed Dummy Value Model

We implemented the proposed model for the AC power flow network on an IEEE 14-bus system, which had 14 buses and 20 transmission lines. Therefore, at every instant, the measurement vector had 54 values of active power and 54 values of reactive power i.e., 108 measurement values in total. The system had 28 state variables. In this case, the voltage magnitudes and voltage angles were taken as the states of the system.

4.1. Data Generation

The seasonal data was generated for the IEEE 14-bus test system. The standard realistic load curve of every season was followed by the generated data. For this purpose, the measurements of the power flow network were varied with the variation of the load that was connected to buses. The measurements were taken after a time interval of one hour. Therefore, there was a time of one hour between the two measurement vectors. For a complete day, the values were recorded in the control room 24 times. We generated the data for one year, i.e., 365 days. Therefore, for the whole system, we obtained values for 8760 different instances. MATPOWER 7.0 was used for the simulation of the system model. The load was varied from 61% to 118.5% of its average value in the reference load curves. Therefore, we also varied the load between these two values. For one day, a specific pattern was followed for every different season by the load that was connected to the buses. We generated the data for one day of every season by using the pattern of that specific season. We selected all the load values in a particular range for a whole day to follow that specific pattern. After varying the total load of the power system, four load curves for one day of every season are shown in Figure 1. The values of the loads are shown at 24 different instances in one complete day. These load curves follow the standard realistic load curves and realistic data was generated according to these curves.

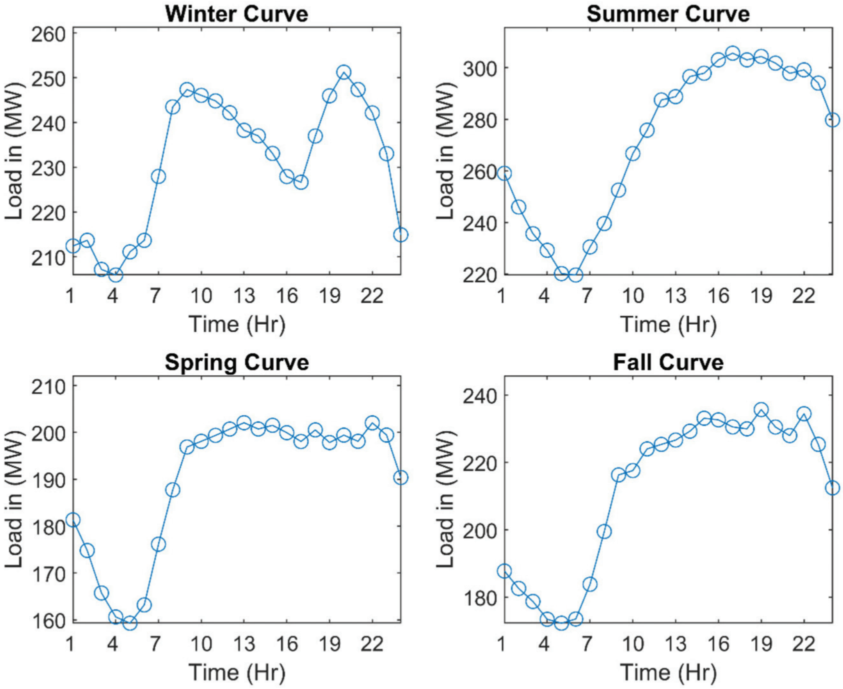


Figure 1. Seasonal load curves of four different seasons based on which data was generated.

4.2. DC State Estimation

By using the generated data, the DC state estimation, which was made by applying the DC assumptions, was implemented for the detection of simple and stealth attacks. For the simulations, the attacks were done according to a certain method, and that method was adopted in the whole manuscript wherever the attacks were made. For a complete day, 25% of the measurement vectors were considered as attacks, i.e., the attacks were made in the measurement vectors at six different instances. The choice of instances was made randomly to make it generalized. In 50% of the measurement vectors chosen for attacks, simple attacks were done. However, stealth attacks were made in the remaining 50% of the measurement vectors that were randomly chosen to be attacked. To create the simple attacks, the attack vector was constructed in such a way that at a particular instant, any value of power was randomly chosen between 0.5% of the maximum value and 0.5% of the minimum value of power at that instant. In the case of stealth attacks, the

Jacobian matrix was first constructed and then the Jacobian matrix was multiplied with a vector **c** to make the attack vector. The values of vector c were selected randomly between −1 and 1 such that it had zero mean and a variance of 2. The attack vector was added to the measurement vector to make the attack. The results of the DC state estimation are shown in Figures 2 and 3. Three types of measurements are shown in Figure 2, namely, safe measurements, simple attack measurements, and the measurements for a stealth attack. A safe zone based on the threshold is also shown in the figure. The measurements outside the safe zone are considered as attacked. The results show that the safe measurement points were present in the safe zone and points of simple attacks were outside the zone. However, measurements affected by stealth attacks are also found in the safe zone, i.e., they are declared as safe by the DC state estimation. They should have to appear outside the safe zone. Therefore, DC SE is not capable of detecting stealth FDI attacks. Similarly, Figure 3 also shows the results of DC state estimation in the form of a bar graph. The residue was calculated for every measurement and the difference of that residue from the threshold is plotted along the vertical axis. For a safe measurement, the value of the difference should be positive, as the residue of that measurement should be less than the threshold. In the graph, the safe measurements are labeled with 1, measurements of simple attacks are labeled with 0, and stealth-attacked measurements are labeled with −1. The results show that the safe measurements and stealth-attacked measurements had positive values of difference. However, the value of the difference was negative for simple attacks. This means that the simple-attacked measurements were termed as attacked by the DC state estimation but measurements having stealth attacks were considered safe. Therefore, simple attacks were detected by the DC state estimation, but stealth attacks bypassed detection.

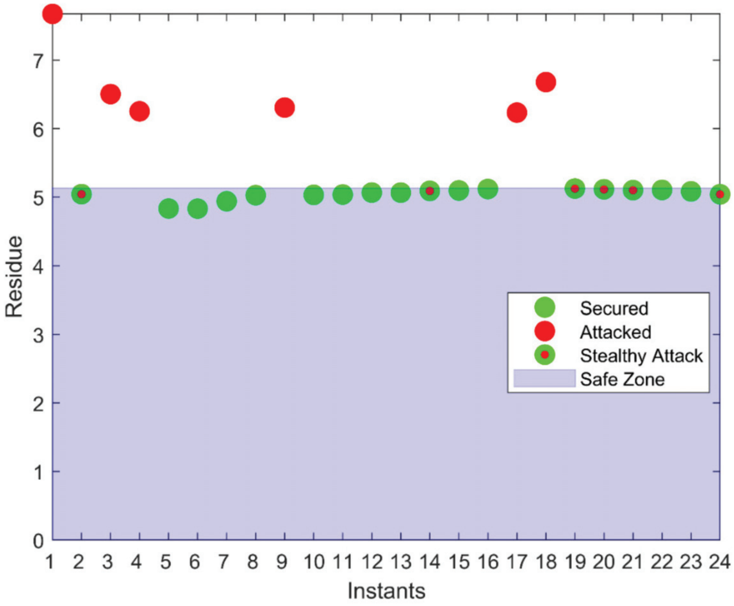


Figure 2. Categorization of safe measurements, simple attacks, and stealth attacks based on the threshold in a DC SE.

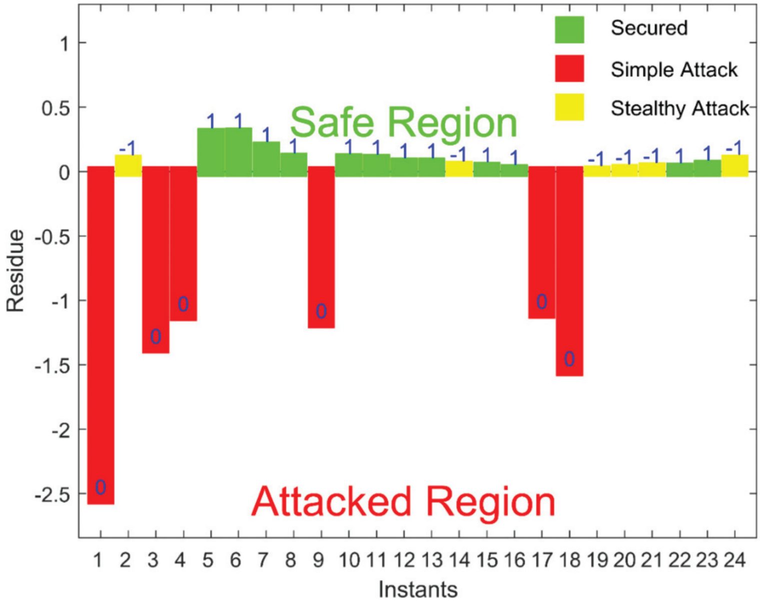


Figure 3. The results of a DC SE for simple and stealth attacks in the form of a bar graph.

4.3. AC State Estimation

For the AC power flow network, an AC SE is used for the detection of attacks. We also used AC SE for the detection of simple and stealth attacks, and the results are given in Figures 4 and 5. The conventions used in a DC state estimation for displaying the results are also used in these figures. In Figure 4, a safe zone, simple-attacked measurements, and stealth-attacked measurements are plotted. Safe measurements and stealth attacks are present in the safe zone, and simple-attacked measurements are outside the zone. It shows that stealth attacks were not detected by the AC state estimation. Similarly, in Figure 5, these results are shown in the form of a bar graph. The value of the difference in the residue from the threshold was positive for safe measurements and stealth-attacked measurements. However, the difference was negative for simple-attacked measurements. This indicated that the AC SE could detect a simple attack but it could not detect the stealth attack. Therefore, it is displayed in the results that simple attacks were detected by the DC state estimation, as well as the AC state estimation techniques, but stealth attacks bypassed these techniques.

4.4. Fixed Dummy Value Model (Results)

The proposed model of a fixed dummy value was implemented for the AC power flow network of the IEEE 14-bus system. The meters sent the actual value and the fixed dummy value to the control room in the form of a measurement vector. Table 1 shows the actual values and the fixed dummy values at the first instant for the first five buses and first five transmission lines. In the control room, the difference, i.e., the residue of two dummy values, was calculated. One dummy value was obtained from the measurement vector and the other was already present in the control room. As the dummy value was fixed in this case, it should not change at any instance. Therefore, for a secure system, the value of the residue should be zero.

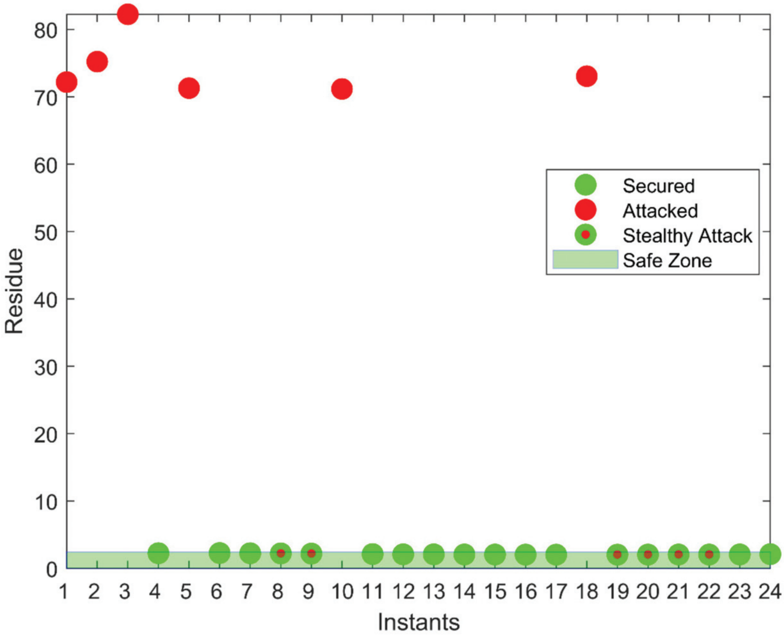


Figure 4. Categorization of safe measurements, simple attacks, and stealth attacks based on the threshold in an AC SE.

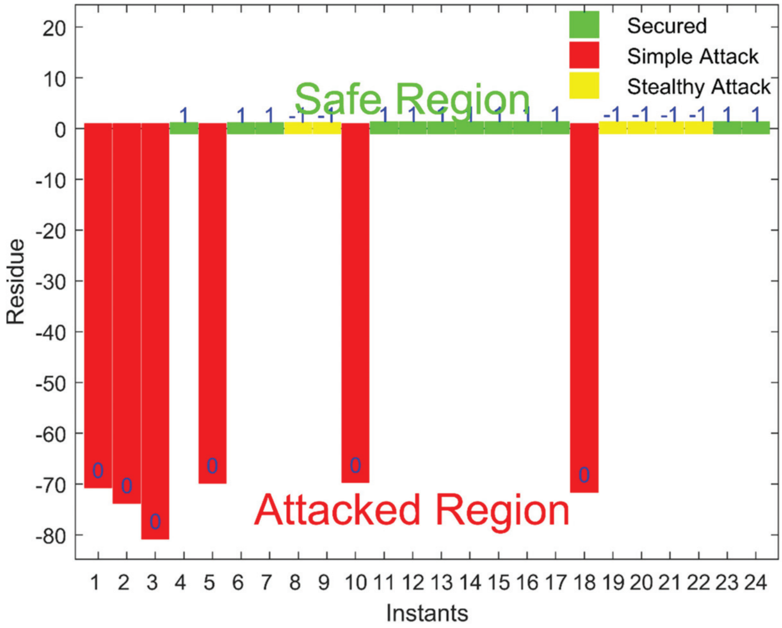


Figure 5. The results of an AC SE for simple and stealth attacks in the form of a bar graph.

Table 1. Active and reactive powers injected into the first 5 buses and the active and reactive powers flowing through the first 5 transmission lines in the forward and backward directions.

Active Powers Injected into the Buses			
Bus No.		**Actual Value (MW)**	**Dummy Value (MW)**
1		232.11	196.3
2		18.41	21.06
3		−93.94	−82.22
4		−47.88	−41.72
5		−7.58	−6.63
Reactive Powers Injected into the Buses			
Bus No.		Actual Value (MVAR)	Dummy Value (MVAR)
1		−16.49	−10.3
2		30.79	21.73
3		5.98	−0.27
4		3.9	3.9
5		−1.6	−1.6
Active Powers Flowing through the Transmission Lines in Forward Direction			
From	To	Actual Value (MW)	Dummy Value (MW)
1	2	156.65	131.57
1	5	75.46	64.73
2	3	73.11	63.66
2	4	56.14	49.21
2	5	41.53	36.6
Reactive Powers Flowing through the Transmission Lines in Forward Direction			
From	To	Actual Value (MVAR)	Dummy Value (MVAR)
1	2	−20.35	−14.04
1	5	3.86	3.74
2	3	3.57	4.71
2	4	−1.54	−1.62
2	5	1.17	0.81
Active Powers Flowing through the Transmission Lines in Backward Direction			
From	To	Actual Value (MW)	Dummy Value (MW)
1	2	−152.37	−128.41
1	5	−72.7	−62.63
2	3	−70.79	−61.85
2	4	−54.46	−47.89
2	5	−40.62	−35.88
Reactive Powers Flowing through the Transmission Lines in Backward Direction			
From	To	Actual Value (MVAR)	Dummy Value (MVAR)
1	2	27.58	17.83
1	5	2.21	−0.38
2	3	1.55	−1.68
2	4	3.01	2
2	5	−2.1	−2.31

The results are shown in Figure 6. The bar graph shows the results for safe measurements, simple-attacked measurements, and stealth-attacked measurements. Safe measurements are labeled as 1, simple-attacked measurements as 0, and stealth-attacked measurements as −1. The residue was calculated for each measurement and plotted along the vertical axis. For safe measurements, the value of the residue was zero, as shown in the bar graph. In the case of simple-attacked measurements and stealth-attacked measurements, the residue was not zero, as the dummy value was changed. Therefore, our proposed model could detect all kinds of attacks, such as simple attacks and stealth attacks. Stealth FDI attacks remained undetected by the DC and AC state estimation, but they could be detected by using our proposed approach.

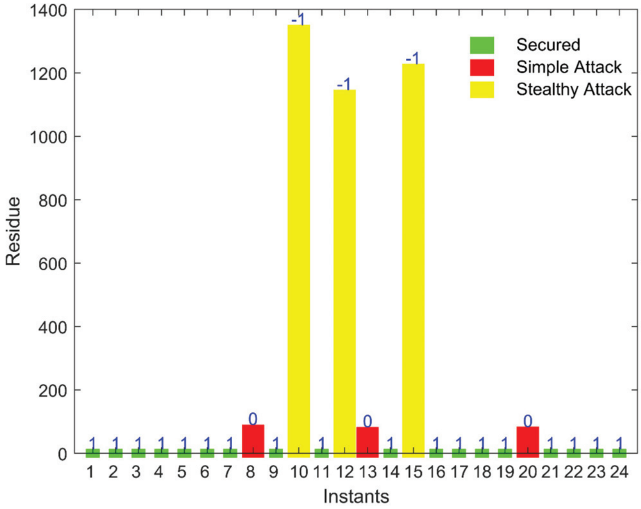


Figure 6. Detection results of the fixed dummy value model for simple and stealth attacks based on dummies.

4.5. Limitations of Fixed Dummy Value Model

The proposed fixed dummy value model could detect the stealth FDI attacks but, at the same time, there was a limitation of the model. As the dummy value does not change, by looking at the measurements continuously for some time, the attacker will come to know which one is the dummy value and the attacker will not change that value while doing an attack. In this way, the attack may be done such that it is unable to be detected. Figure 7 shows the results where the fixed dummy value model was bypassed by the stealth attack. It can be seen from the graph that the value of the residue in the case of attacks came out to be zero, as the attacker did not change the dummy values while launching the attack. Therefore, this limitation of the fixed dummy value was evaluated using the results.

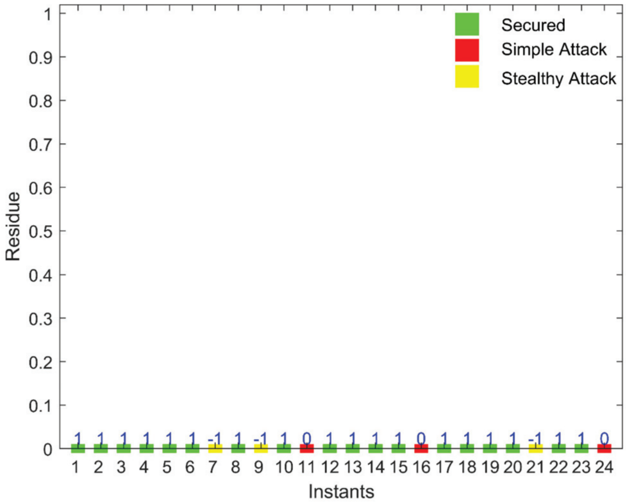


Figure 7. Limitations of the fixed dummy value model for simple and stealth attacks.

5. Variable Dummy Value Model

In the proposed model with a fixed dummy value, the attacker may attack the system by remaining undetected because the dummy value is fixed for all instances. Therefore, the dummy value should vary to protect the system against attacks. For this purpose, the model with a variable dummy value was introduced. In this scenario, the dummy value will change at every instant, and it will depend on the actual value, as well as some other values of the power in the system. Therefore, the dummy value of the power changes with the change in either of those values on which it depends. In the variable dummy value model, a linear function is implemented for the calculation of the dummy value and that function uses the actual measured value of that meter and the measured value of some other meter that has a relationship with that actual value. The function is only known to the control room. In the fixed dummy value model, the calculated dummy values are embedded into the meters. Similarly, in the variable dummy value model, the function used for the calculation of dummy values is embedded into the meters. This work assumed that the intruder does not have access to the meters, i.e., the intruder only has access to the measurements sent to the control room. The following functions are used in the case of the variable dummy value model for the calculation of dummy values for buses:

$$\begin{gathered} p'_{v(y)}(i) = \beta_{1vpi} p_{v(y)}(i) + \beta_{2vpi} p_{viz(y)} + \beta_{3vpi} \\ i = 1, 2, 3, \ldots\ldots, b \end{gathered} \tag{25}$$

$$\begin{gathered} q'_{v(y)}(i) = \beta_{1vqi} q_{v(y)}(i) + \beta_{2vqi} q_{viz(y)} + \beta_{3vqi} \\ i = 1, 2, 3, \ldots\ldots, b \end{gathered} \tag{26}$$

where $p'_{v(y)}(i)$ and $q'_{v(y)}(i)$ represent the *i*th entries of the dummy values vectors $\mathbf{p'_{v(y)}}$ and $\mathbf{q'_{v(y)}}$, respectively. $p_{v(y)}(i)$ and $q_{v(y)}(i)$ denote the *i*th entries of $\mathbf{p_{v(y)}}$ and $\mathbf{q_{v(y)}}$, respectively. Similarly, $p_{viz(y)}$ and $q_{viz(y)}$ represent the active power and reactive power flowing through the first transmission line connected to the *i*th bus at the *y*th instant, respectively. β_{1vpi}, β_{2vpi}, β_{3vpi}, β_{1vqi}, β_{2vqi}, and β_{3vqi} are the constants that have to be learned to calculate the dummy values. Similarly, the calculation of the dummy values of the active and reactive powers flowing through the transmission lines can be done by using the following functions:

$$\begin{gathered} p'_{vw(y)}(i) = \beta_{1vwpi} p_{v(y)}(i) + \beta_{2vwpi} p_{w(y)}(i) + \beta_{3vwpi} \\ i = 1, 2, 3, \ldots\ldots, t \end{gathered} \tag{27}$$

$$\begin{gathered} q'_{vw(y)}(i) = \beta_{1vwqi} q_{v(y)}(i) + \beta_{2vwqi} q_{w(y)}(i) + \beta_{3vwqi} \\ i = 1, 2, 3, \ldots\ldots, t \end{gathered} \tag{28}$$

$$\begin{gathered} p'_{wv(y)}(i) = \beta_{1wvpi} p_{w(y)}(i) + \beta_{2wvpi} p_{v(y)}(i) + \beta_{3wvpi} \\ i = 1, 2, 3, \ldots\ldots, t \end{gathered} \tag{29}$$

$$\begin{gathered} q'_{wv(y)}(i) = \beta_{1wvqi} q_{w(y)}(i) + \beta_{2wvqi} q_{v(y)}(i) + \beta_{3wvqi} \\ i = 1, 2, 3, \ldots\ldots, t \end{gathered} \tag{30}$$

$p'_{vw(y)}(i)$ and $q'_{vw(y)}(i)$ denote the *i*th entries of vectors $\mathbf{p'_{vw(y)}}$ and $\mathbf{q'_{vw(y)}}$, respectively, which contain the dummy values of powers flowing through the transmission lines in the forward direction at *y*th instant. $p_{w(y)}(i)$ and $q_{w(y)}(i)$ represent the active power and reactive power injected into *i*th bus at *y*th instant, respectively. $p_{w(y)}(i)$ and $q_{w(y)}(i)$ belong to $\mathbf{p_{v(y)}}$ and $\mathbf{q_{v(y)}}$, respectively. Similarly, $p'_{wv(y)}(i)$ and $q'_{wv(y)}(i)$ show the *i*th entries of vectors $\mathbf{p'_{wv(y)}}$ and $\mathbf{q'_{wv(y)}}$, respectively, which have the dummy values of the active power and reactive power flowing through transmission lines in the backward direction. Constants are also used in the equations proposed for the calculation of dummy values.

The Equations (25)–(30) are used for finding the dummy values at the *y*th instant. In the variable dummy value model, the dummy values depend on the real-time measurement values. As the real-time measurement values are used for the calculation of the dummy values, the dummy values change at every instant in this case.

There is a key point to consider while selecting the dummy value, which is that the dummy value of a meter should be close to its actual value. There should not be too much difference between the actual and dummy value such that the attacker can find the dummy value and construct an undetectable attack. Therefore, when these linear functions are implemented for the calculation of the dummy value, we may obtain a dummy value that is far away from its actual value. The reason for this is that these dummy values depend on two different values of the power and there might be a high variance in the values of a certain meter depending upon the load connected to a bus. If the variance of either of the two actual values is high for a whole day, the dummy value will not be close to the actual value.

This problem may be minimized due to the selection of appropriate values of the constants. The selection of constants is done in such a way that all dummy values of a specific power for the whole day must remain close to the actual value of that power. For this purpose, a machine-learning technique, namely, multivariate linear regression (MLR), was used for finding the best values of the constants. The procedure of MLR to find the constants of the equation used to calculate the dummy values of the active power injected to all the buses is explained here. In this case, the hypothesis is written as

$$\mathrm{g}_{\boldsymbol{\beta}_{\mathbf{k}}}(\mathbf{p_k}) = \beta_{1vpk}p_v(k) + \beta_{2vpk}p_{vkz} + \beta_{3vpk} \tag{31}$$

Here, $\mathrm{g}_{\boldsymbol{\beta}_{\mathbf{k}}}(\mathbf{p_k})$ is a function of $\mathbf{p_k}$ that is parameterized using $\boldsymbol{\beta}_{\mathbf{k}}$. $\mathbf{p_k}$ represents the *k*th input vector, where *k* = 1, 2, 3, . . . , b and $\mathbf{p_k}$ = [1 p_{vkz} $p_v(k)$]$^{\mathrm{T}}$. $\boldsymbol{\beta}_{\mathbf{k}}$ denotes the *k*th parameter vector and $\boldsymbol{\beta}_{\mathbf{k}}$ = [β_{3vpk} β_{2vpk} β_{1vpk}]$^{\mathrm{T}}$. β_{1vpk}, β_{2vpk}, and β_{3vpk} are the constants to be learned for each dummy value of the active power injected into the buses. Therefore, for each dummy value, a different vector of constants is used. Depending upon the hypothesis, the cost function for the multivariate linear regression can be written as

$$\mathrm{J}(\boldsymbol{\beta}_{\mathbf{k}}) = \frac{1}{2mt}\sum_{y=1}^{mt}\left(\left(\sum_{f=1}^{3}\beta_{kf}p_{kf(y)}\right) - p_{v(y)}(k)\right)^2 \tag{32}$$

Here, *mt* represents the total number of instances, i.e., the total number of training examples in this case. $p_{v(y)}(k)$ represents the output of the *y*th training example of the active power injected to the *k*th bus. We must minimize the cost function so that we obtain the best values of the parameters. For this purpose, the gradient descent algorithm was applied, which is based on the update rule. The gradient descent can be written as

$$\beta_{kf} := \beta_{kf} - \alpha\frac{1}{mt}\sum_{y=1}^{mt}\left(\mathrm{g}_{\boldsymbol{\beta}_{\mathbf{k}}}\left(\mathbf{p_{k(y)}}\right) - p_{v(y)}(k)\right)p_{kf(y)} \tag{33}$$

β_{kf} represents the *f*th entry of the *k*th parameter vector. $p_{kf(y)}$ denotes the *f*th entry of the *k*th input vector at the *y*th instant. The β's are calculated again and again, and those parameters are used to calculate the cost. The above process is repeated until convergence occurs. When the cost converges, this produces the best values of the parameters.

By adopting the same procedure, the constants for the remaining equations are also found and those constants are put in their respective functions to calculate the dummy values of the active and reactive power. Then, these functions are embedded into the meters for the calculation of the dummy values. The meters measure the actual values of power and then use those functions to calculate the dummy values of power to send them to the control room. These functions are only known to the control room.

In the control room, to detect the FDI attacks, these functions are used to recalculate the dummy value by using the actual values obtained from the measurement vector. Then, the recalculated dummy value is compared with the dummy value obtained from the measurement vector for attack detection. The following equations are used in the control room to compare the calculated dummy values and received dummy values of active and reactive powers injected into all the buses:

$$r_{vp(y)}(j) = p'_{vr(y)}(j) - (\beta_{1vpj}p_{vr(y)}(j) + \beta_{2vpj}p_{vrjz(y)} + \beta_{3vpj}) \\ j = 1, 2, 3, \ldots\ldots, b \tag{34}$$

$$r_{vq(y)}(j) = q'_{vr(y)}(j) - \left(\beta_{1vqj}q_{vr(y)}(j) + \beta_{2vqj}q_{vrjz(y)} + \beta_{3vqj}\right) \\ j = 1, 2, 3, \ldots\ldots, b \tag{35}$$

The measurement vector received in the control room at the yth instant is $\mathbf{z_{dyr}}$. Here, $p'_{vr(y)}(j)$ and $q'_{vr(y)}(j)$ represent the jth entries of the received vectors $\mathbf{p'_{vr(y)}}$ and $\mathbf{p'_{vr(y)}}$, respectively, which contain the dummy values of the active power and reactive power received in the control room at the yth instant. $p_{vr(y)}(j)$ and $q_{vr(y)}(j)$ denote the jth entries of the received vectors $\mathbf{p_{vr(y)}}$ and $\mathbf{q_{vr(y)}}$, respectively, which contain the actual values of the active power and reactive power received in the control room at the yth instant. $p_{vrjz(y)}$ and $q_{vrjz(y)}$ are taken from the received measurement vector. $r_{vp(y)}(j)$ and $r_{vq(y)}(j)$ represent the jth entries of the residue vectors $\mathbf{r_{vp(y)}}$ and $\mathbf{r_{vq(y)}}$, respectively, which contain the residues for the active and reactive powers injected into the buses at the yth instant. Similarly, the equations for calculating the residues for the forward and backward powers flowing through the transmission lines are given by:

$$r_{vwp(y)}(j) = p'_{vwr(y)}(j) - \left(\beta_{1vwpj}p_{vr(y)}(j) + \beta_{2vwpj}p_{wr(y)}(j) + \beta_{3vwpj}\right) \\ j = 1, 2, 3, \ldots\ldots, t \tag{36}$$

$$r_{vwq(y)}(j) = q'_{vwr(y)}(j) - \left(\beta_{1vwqj}q_{vr(y)}(j) + \beta_{2vwqj}q_{wr(y)}(j) + \beta_{3vwqj}\right) \\ j = 1, 2, 3, \ldots\ldots, t \tag{37}$$

$$r_{wvp(y)}(j) = p'_{wvr(y)}(j) - \left(\beta_{1wvpj}p_{wr(y)}(j) + \beta_{2wvpj}p_{vr(y)}(j) + \beta_{3wvpj}\right) \\ j = 1, 2, 3, \ldots\ldots, t \tag{38}$$

$$r_{wvq(y)}(j) = q'_{wvr(y)}(j) - \left(\beta_{1wvqj}q_{wr(y)}(j) + \beta_{2wvqj}q_{vr(y)}(j) + \beta_{3wvqj}\right) \\ j = 1, 2, 3, \ldots\ldots, t \tag{39}$$

In these equations, the dummy and actual values are obtained from the received measurement vector in the control room. $r_{vwp(y)}(j)$, $r_{vwq(y)}(j)$, $r_{wvp(y)}(j)$, and $r_{wvq(y)}(j)$ represent the jth entries of the residue vectors $\mathbf{r_{vwp(y)}}$, $\mathbf{r_{vwq(y)}}$, $\mathbf{r_{wvp(y)}}$, and $\mathbf{r_{wvq(y)}}$, respectively, which contain the residues for the active and reactive powers flowing through the transmission lines in the forward and backward directions at the yth instant. The overall residue at the yth instant is calculated using

$$r = \left|\mathbf{r_{vp(y)}}\right| + \left|\mathbf{r_{vq(y)}}\right| + \left|\mathbf{r_{vwp(y)}}\right| + \left|\mathbf{r_{vwq(y)}}\right| + \left|\mathbf{r_{wvp(y)}}\right| + \left|\mathbf{r_{wvq(y)}}\right| \tag{40}$$

For a secure system:

$$r = 0$$

If the total residue has some value other than zero, the system is considered attacked. The attacker hacks the measurement vector $\mathbf{z_{dy}}$ and sends the vector $\mathbf{z_{dyr}}$ to the control room after making the attack. As the attacker does not know which are the dummy values, the attacker will attack dummy values too. The attacker also does not know about the relationship used to calculate the dummy value. As a result, the attack is easily detected in the control room, as the value of r will not be equal to zero.

This proposed model of the variable dummy value can tackle the limitations of the fixed dummy value model and the stealth FDI attacks can be detected in an efficient way.

6. Results of the Variable Dummy Value Model

The proposed model with the fixed dummy value can be bypassed, as the dummy value is constant for all the instances. The dummy value should change with the change in the actual value. In the variable dummy value model, the limitation of the fixed value dummy model is overcome by changing the dummy value at every instant. The variable dummy value model was implemented for the AC power flow model of the IEEE 14-bus system. A dummy value was selected in such a way that it should remain close to the actual value of that power and this feature depends upon the values of constants used in the linear function. For the selection of the constants, the MLR model is built for the calculation of all dummy values. Before using the MLR model, we must select the dummy values as outputs to find the relationship between the input and output. To select the dummy value of power at any instant for MLR, any value is picked from its actual values that occurred for the whole one year prior to that instant. The multivariate linear regression model was run for all the linear equations, and we obtained the best values of the constants for a particular equation that gave the minimum cost for that equation. Table 2 shows the values of constants at the first instant for the first five buses and the first five transmission lines.

Figure 8 shows the learning of the MLR model when finding the parameters of the linear equation used to find the dummy values of P_1. The constants of the equation of the line that best fit the training data were found. Constants for all the linear equations were found in this way and those equations were embedded in the meters to calculate the dummy values. Table 3 shows the actual values and the dummy values for the variable dummy value model at a single instant for the first five buses and first five transmission lines.

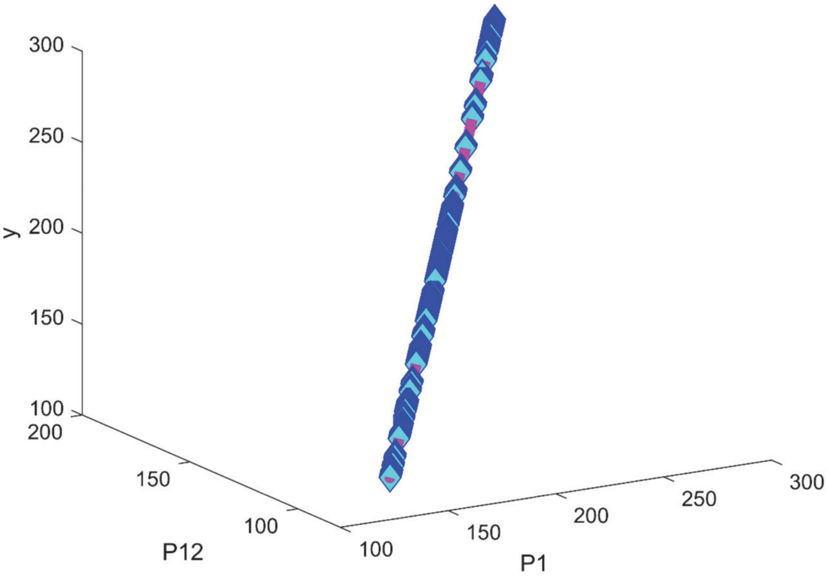


Figure 8. Training of the multivariate linear regression model to find the constants of the equation used to calculate the dummy values of P_1.

Table 2. Constants used for the calculation of dummy values of active and reactive powers injected into the first 5 buses and the active and reactive powers flowing through the first five transmission lines in the forward and backward directions.

Calculation of Dummy Values of Active Powers Injected into the Buses

Bus No.	$\boldsymbol{\beta_1}$	$\boldsymbol{\beta_2}$	$\boldsymbol{\beta_3}$
1	0	40.81	196.30
2	0	3.1	21.06
3	0	13.45	−82.22
4	0	6.83	−41.72
5	0	1.09	−6.63

Calculation of Dummy Values of Reactive Powers Injected into the Buses

Bus No.	β_1	β_2	β_3
1	0	6.73	−10.30
2	0	10.78	21.73
3	0	7.29	−0.27
4	0	1.13	5.13
5	0	8.25	−9.88

Calculation of Dummy Values of Active Powers Flowing through the Transmission Lines in Forward Direction

From	To	β_1	β_2	β_3
1	2	0.91	29.57	131.57
1	5	−1.22	10.93	64.73
2	3	−9.36	−1.44	63.66
2	4	−5.18	−2.62	49.21
2	5	−3.07	−2.48	36.6

Calculation of Dummy Values of Reactive Powers Flowing through the Transmission Lines in Forward Direction

From	To	β_1	β_2	β_3
1	2	5.97	−1.03	−14.04
1	5	4.91	5.21	3.74
2	3	1.08	0.13	4.71
2	4	−0.13	−0.03	−1.62
2	5	−0.22	−0.27	0.81

Calculation of Dummy Values of Active Powers Flowing through the Transmission Lines in Backward Direction

From	To	β_1	β_2	β_3
1	2	2.8	−24.45	−128.41
1	5	2.8	−8.55	−62.63
2	3	8.85	1.35	−61.85
2	4	4.9	2.48	−47.89
2	5	2.95	2.38	−35.88

Calculation of Dummy Values of Reactive Powers Flowing Through the Transmission Lines in Backward Direction

From	To	β_1	β_2	β_3
1	2	5.35	16.66	17.83
1	5	1.65	4.65	−0.38
2	3	−3.25	−0.53	−1.68
2	4	−0.74	−0.39	2
2	5	−0.16	−0.04	−2.31

Table 3. Active and reactive powers injected into the first 5 buses and the active and reactive powers flowing through the first 5 transmission lines in the forward and backward directions for the variable dummy value model.

Active Powers Injected to the Buses			
Bus No.		**Actual Value (MW)**	**Dummy Value (MW)**
1		232.11	232.02
2		18.41	14.74
3		−93.94	−154.82
4		−47.88	−87.64
5		−7.58	−9.95
Reactive Powers Injected into the Buses			
Bus No.		Actual Value (MVAR)	Dummy Value (MVAR)
1		−16.49	−45.96
2		30.79	−26.41
3		5.98	−32.63
4		3.9	0.58
5		−1.6	−44.16
Active Powers Flowing through the Transmission Lines in Forward Direction			
From	To	Actual Value (MW)	Dummy Value (MW)
1	2	156.65	131.57
1	5	75.46	64.73
2	3	73.11	63.66
2	4	56.14	49.21
2	5	41.53	36.6
Reactive Powers Flowing through the Transmission Lines in Forward Direction			
From	To	Actual Value (MVAR)	Dummy Value (MVAR)
1	2	−20.35	−14.04
1	5	3.86	3.74
2	3	3.57	4.71
2	4	−1.54	−1.62
2	5	1.17	0.81
Active Powers Flowing through the Transmission Lines in Backward Direction			
From	To	Actual Value (MW)	Dummy Value (MW)
1	2	−152.37	−128.41
1	5	−72.7	−62.63
2	3	−70.79	−61.85
2	4	−54.46	−47.89
2	5	−40.62	−35.88
Reactive Powers Flowing through the Transmission Lines in Backward Direction			
From	To	Actual Value (MVAR)	Dummy Value (MVAR)
1	2	27.58	17.83
1	5	2.21	−0.38
2	3	1.55	−1.68
2	4	3.01	2
2	5	−2.1	−2.31

In the control room, the dummy values were again calculated by using the obtained actual values from the measurement vector and those recalculated dummy values were subtracted from the obtained dummy values to find the residue. The residue should be zero for a secured system. The results of the proposed model of the variable dummy value are shown in Figure 9, where the model was evaluated using simple and stealth attacks. Safe measurements are also shown in the figure. The residue is plotted along the vertical axis. For safe measurements, the value of the residue was zero, as shown in the bar graph. However, for simple attacks and stealth attacks, the residue had some value greater than zero. Therefore, simple and stealth attacks were detected by our proposed variable dummy value model. As a result, the limitations of the fixed dummy value approach were handled by this variable dummy value model, and stealth FDI attacks were easily detected in the control room.

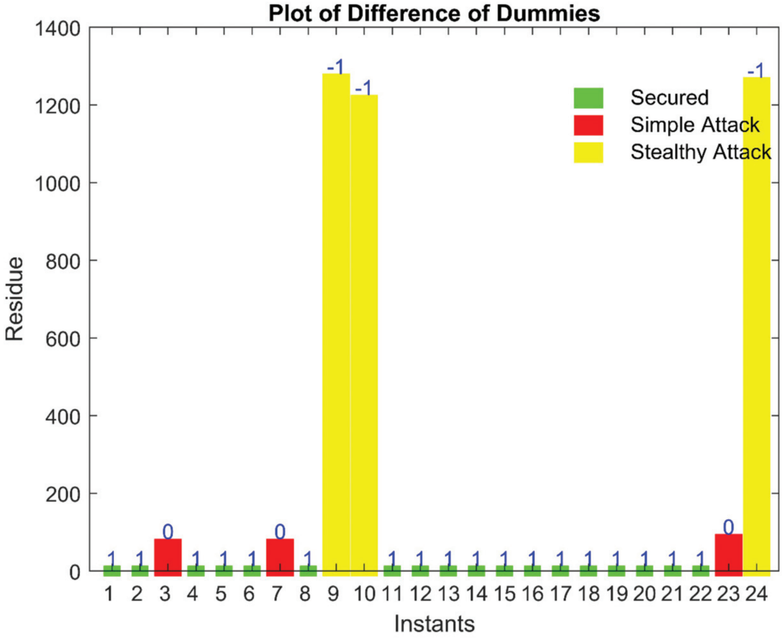


Figure 9. Results of the variable dummy value model for simple and stealth attacks.

7. Discussion

Stealth false data injection attacks create huge damage to a power system. Such kinds of attacks should be detected for a smooth and reliable power flow in a smart grid. The proposed model was applied for the detection of such attacks. The model was based on a dummy measurement. The meters in the smart grid send the actual measurement and the dummy measurement. There are two techniques in the proposed model, namely, the fixed dummy value model and the variable dummy value model.

Both techniques were validated through the experimental results. The first technique of the fixed dummy value model could detect FDI attacks. However, at the same time, this technique has some limitations. When the attacker does not attack the dummy value of measurement, i.e., only the actual measurement is attacked, the control room is not able to detect that attack. The second technique of the proposed model, such as the variable dummy value model overcomes the limitation and the FDI attacks that were left unnoticed by the fixed dummy value model were detected by the variable dummy value model, as validated by the results.

The proposed model does not require the installation of any extra buses or transmission lines. There is no need to install any extra meters. Therefore, the model can be effectively applied to a smart grid and is economically efficient. From the viewpoint of long-term operation, the proposed model can be applied to make a smart grid more protected and secured. In the future, an extension of this work can be done to practically implement the model for a smart grid, which will protect the smart grid from FDI attacks. Some other methods can be adopted to set the dummy values of the power in the future. The probability of launching the attack can be minimized in this way.

8. Conclusions

Two-way communication is one of the most important features of a smart grid. These communication links may be hacked by attackers to launch attacks. A lot of damage, as well as loss, can be caused by cyber-attacks on a power system. Financial benefits can be obtained by the attacker through these attacks. An attacker can also create technical problems. Power information can be corrupted or blocked. The values of power can be increased or decreased, which will cause power blackouts or power outages. The accurate and continuous power flow can be ensured by detecting and minimizing cyber-attacks. The methods of DC state estimation, as well as AC state estimation, are unable to detect stealth FDI attacks.

For this purpose, in this study, a model based on dummy measurement values was proposed and implemented for the detection of stealth FDI attacks. The overall proposed model consisted of a fixed dummy value model and a variable dummy value model. The fixed dummy value model showed promising results against FDI attacks but with some limitations. The variable dummy value model handled those limitations and the stealth FDI attacks were efficiently detected in the control room using our proposed model. Simulations were performed for the model and the results indicated that all the stealth FDI attacks were detected in the control room. We made the power system a secure one.

Author Contributions: Conceptualization, M.A.S., F.A. and F.R.A.; methodology, M.A.S., F.A. and G.H.; software, M.A.S. and Z.U.; validation, M.A.S., F.A. and Z.U.; investigation, F.A. and G.H.; data curation, M.A.S. and F.R.A.; writing—original draft preparation, M.A.S., F.A. and Z.U.; writing—review and editing, M.A.S., F.R.A. and G.H.; supervision, F.A. and G.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the Taif University Researchers Supporting Project Number (TURSP-2020/331), Taif University, Taif, Saudi Arabia.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Acknowledgments: The authors would like to acknowledge the support from Taif University Researchers Supporting Project Number (TURSP-2020/331), Taif University, Taif, Saudi Arabia.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Cui, L.; Qu, Y.; Gao, L.; Xie, G.; Yu, S. Detecting false data attacks using machine learning techniques in smart grid: A survey. *J. Netw. Comput. Appl.* **2020**, *170*, 102808. [CrossRef]
2. Hamidi, V.; Smith, K.S.; Wilson, R.C. Smart grid technology review within the transmission and distribution sector. In Proceedings of the IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT Europe), Gothenburg, Sweden, 11–13 October 2010; pp. 1–8.
3. Mohammadi, F. Emerging Challenges in Smart Grid Cybersecurity Enhancement: A Review. *Energies* **2021**, *14*, 1380. [CrossRef]
4. Aoufi, S.; Derhab, A.; Guerroumi, M. Survey of false data injection in smart power grid: Attacks, countermeasures and challenges. *J. Inf. Secur. Appl.* **2020**, *54*, 102518. [CrossRef]
5. Mohammadi, F.; Nazri, G.A.; Saif, M. A fast fault detection and identification approach in power distribution systems. In Proceedings of the International Conference on Power Generation Systems and Renewable Energy Technologies (PGSRET), Istanbul, Turkey, 26–27 August 2019; pp. 1–4.
6. Sen, Ö.; van der Velde, D.; Peters, S.N.; Henze, M. An Approach of Replicating Multi-Staged Cyber-Attacks and Countermeasures in a Smart Grid Co-Simulation Environment. *arXiv* **2021**, arXiv:2110.02040.
7. McDaniel, P.; McLaughlin, S. Security and privacy challenges in the smart grid. *IEEE Secur. Priv.* **2009**, *7*, 75–77. [CrossRef]
8. Fang, X.; Misra, S.; Xue, G.; Yang, D. Smart grid—The new and improved power grid: A survey. *IEEE Commun. Surv. Tutor.* **2011**, *14*, 944–980. [CrossRef]
9. Dileep, G. A survey on smart grid technologies and applications. *Renew. Energy* **2020**, *146*, 2589–2625. [CrossRef]

10. Kolhe, M. Smart grid: Charting a new energy future: Research, development and demonstration. *Electr. J.* **2012**, *25*, 88–93. [CrossRef]
11. Gunduz, M.Z.; Das, R. Cyber-security on smart grid: Threats and potential solutions. *Comput. Netw.* **2020**, *169*, 107094. [CrossRef]
12. Jayachandran, M.; Reddy, C.; Padmanaban, S.; Milyani, A.H. Operational planning steps in smart electric power delivery system. *Sci. Rep.* **2021**, *11*, 1–21.
13. Chen, J.; Mohamed, M.A.; Dampage, U.; Rezaei, M.; Salmen, S.H.; Obaid, S.A.; Annuk, A. A Multi-Layer Security Scheme for Mitigating Smart Grid Vulnerability against Faults and Cyber-Attacks. *Appl. Sci.* **2021**, *11*, 9972. [CrossRef]
14. Shaukat, N.; Ali, S.M.; Mehmood, C.A.; Khan, B.; Jawad, M.; Farid, U.; Ullah, Z.; Anwar, S.M.; Majid, M. A survey on consumers empowerment, communication technologies, and renewable generation penetration within Smart Grid. *Renew. Sustain. Energy Rev.* **2018**, *81*, 1453–1475. [CrossRef]
15. Esmalifalak, M.; Liu, L.; Nguyen, N.; Zheng, R.; Han, Z. Detecting stealthy false data injection using machine learning in smart grid. *IEEE Syst. J.* **2014**, *11*, 1644–1652. [CrossRef]
16. Yan, J.; Tang, B.; He, H. Detection of false data attacks in smart grid with supervised learning. In Proceedings of the International Joint Conference on Neural Networks (IJCNN), Vancouver, BC, Canada, 24–29 July 2016; pp. 1395–1402.
17. Liu, Y.; Ning, P.; Reiter, M.K. False data injection attacks against state estimation in electric power grids. *ACM Trans. Inf. Syst. Secur.* **2011**, *14*, 1–33. [CrossRef]
18. Xie, L.; Mo, Y.; Sinopoli, B. Integrity data attacks in power market operations. *IEEE Trans. Smart Grid.* **2011**, *2*, 659–666. [CrossRef]
19. Esmalifalak, M.; Han, Z.; Song, L. Effect of stealthy bad data injection on network congestion in market based power system. In Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC), Paris, France, 1–4 April 2012; pp. 2468–2472.
20. Esmalifalak, M.; Nguyen, H.; Zheng, R.; Han, Z. Stealth false data injection using independent component analysis in smart grid. In Proceedings of the IEEE International Conference on Smart Grid Communications (SmartGridComm), Brussels, Belgium, 17–20 October 2011; pp. 244–248.
21. Liu, L.; Esmalifalak, M.; Han, Z. Detection of false data injection in power grid exploiting low rank and sparsity. In Proceedings of the IEEE International Conference on Communications (ICC), Budapest, Hungary, 9–13 June 2013; pp. 4461–4465.
22. Liu, L.; Esmalifalak, M.; Ding, Q.; Emesih, V.A.; Han, Z. Detecting false data injection attacks on power grid by sparse optimization. *IEEE Trans. Smart Grid.* **2014**, *5*, 612–621. [CrossRef]
23. Sayghe, A.; Hu, Y.; Zografopoulos, I.; Liu, X.; Dutta, R.G.; Jin, Y.; Konstantinou, C. Survey of machine learning methods for detecting false data injection attacks in power systems. *IET Smart Grid.* **2020**, *3*, 581–595. [CrossRef]
24. Hug, G.; Giampapa, J.A. Vulnerability assessment of AC state estimation with respect to false data injection cyber-attacks. *IEEE Trans Smart Grid.* **2012**, *3*, 1362–1370. [CrossRef]
25. Liang, J.; Kosut, O.; Sankar, L. Cyber attacks on AC state estimation: Unobservability and physical consequences. In Proceedings of the IEEE PES General Meeting | Conference & Exposition, National Harbor, MD, USA, 27–31 July 2014; pp. 1–5.
26. Motiyani, M.R.; Chudasama, A.R.; Desai, M.A. Electrical Power System State Estimation: Theory and Implementation; 2015. Available online: https://www.semanticscholar.org/paper/ELECTRICAL-POWER-SYSTEM-STATE-ESTIMATION-%3A-THEORY-Motiyani-Chudasama/f5e8da4e8a6253575780b2ddc89725998eb35591 (accessed on 29 March 2022).
27. Monticelli, A. Electric power system state estimation. *Proc. IEEE* **2000**, *88*, 262–282. [CrossRef]
28. Ozay, M.; Esnaola, I.; Vural, F.T.; Kulkarni, S.R.; Poor, H.V. Machine learning methods for attack detection in the smart grid. *IEEE Trans. Neural Netw. Learn. Syst.* **2015**, *27*, 1773–1786. [CrossRef]
29. Huang, Y.; Esmalifalak, M.; Nguyen, H.; Zheng, R.; Han, Z.; Li, H.; Song, L. Bad data injection in smart grid: Attack and defense mechanisms. *IEEE Commun. Mag.* **2013**, *51*, 27–33. [CrossRef]
30. Kosut, O.; Jia, L.; Thomas, R.J.; Tong, L. Malicious data attacks on smart grid state estimation: Attack strategies and countermeasures. In Proceedings of the IEEE International Conference on Smart Grid Communications, Gaithersburg, MD, USA, 4–6 October 2010; pp. 220–225.
31. Chaojun, G.; Jirutitijaroen, P.; Motani, M. Detecting false data injection attacks in ac state estimation. *IEEE Trans. Smart Grid.* **2015**, *6*, 2476–2483. [CrossRef]
32. Ayad, A.; Farag, H.E.; Youssef, A.; El-Saadany, E.F. Detection of false data injection attacks in smart grids using recurrent neural networks. In Proceedings of the IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT), Washington, DC, USA, 19–22 February 2018; pp. 1–5.
33. Foroutan, S.A.; Salmasi, F.R. Detection of false data injection attacks against state estimation in smart grids based on a mixture Gaussian distribution learning method. *IET Cyber-Phys. Syst. Theory Appl.* **2017**, *2*, 161–171. [CrossRef]
34. Yang, C.; Wang, Y.; Zhou, Y.; Ruan, J.; Liu, W. False data injection attacks detection in power system using machine learning method. *J. Comput. Commun.* **2018**, *6*, 276. [CrossRef]
35. Ashrafuzzaman, M.; Das, S.; Chakhchoukh, Y.; Shiva, S.; Sheldon, F.T. Detecting stealthy false data injection attacks in the smart grid using ensemble-based machine learning. *Comput. Secur.* **2020**, *97*, 101994. [CrossRef]
36. Farrukh, Y.A.; Khan, I.; Ahmad, Z.; Elavarasan, R.M. A sequential supervised machine learning approach for cyber attack detection in a smart grid system. *arXiv* **2021**, arXiv:2108.00476.
37. Acosta, M.R.; Ahmed, S.; Garcia, C.E.; Koo, I. Extremely randomized trees-based scheme for stealthy cyber-attack detection in smart grid networks. *IEEE Access* **2020**, *8*, 19921–19933. [CrossRef]

38. Sakhnini, J.; Karimipour, H.; Dehghantanha, A. Smart grid cyber attacks detection using supervised learning and heuristic feature selection. In Proceedings of the IEEE 7th International Conference on Smart Energy Grid Engineering (SEGE), Oshawa, ON, Canada, 12–14 August 2019; pp. 108–112.
39. Xue, D.; Jing, X.; Liu, H. Detection of false data injection attacks in smart grid utilizing ELM-based OCON framework. *IEEE Access* **2019**, *7*, 31762–31773. [CrossRef]
40. Aboelwafa, M.M.; Seddik, K.G.; Eldefrawy, M.H.; Gadallah, Y.; Gidlund, M. A machine-learning-based technique for false data injection attacks detection in industrial IoT. *IEEE Internet Things J.* **2020**, *7*, 8462–8471. [CrossRef]
41. Wang, C.; Tindemans, S.; Pan, K.; Palensky, P. Detection of false data injection attacks using the autoencoder approach. In Proceedings of the International Conference on Probabilistic Methods Applied to Power Systems (PMAPS), Liège, Belgium, 18–21 August 2020; pp. 1–6.
42. Kundu, A.; Sahu, A.; Serpedin, E.; Davis, K. A3d: Attention-based auto-encoder anomaly detector for false data injection attacks. *Electr. Power Syst. Res.* **2020**, *189*, 106795. [CrossRef]
43. Zhou, L.; Ouyang, X.; Ying, H.; Han, L.; Cheng, Y.; Zhang, T. Cyber-attack classification in smart grid via deep neural network. In Proceedings of the 2nd International Conference on Computer Science and Application Engineering, Hohhot, China, 22–24 October 2018; pp. 1–5.
44. Niu, X.; Li, J.; Sun, J.; Tomsovic, K. Dynamic detection of false data injection attack in smart grid using deep learning. In Proceedings of the IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT), Washington, DC, USA, 17–20 February 2019; pp. 1–6.
45. An, D.; Yang, Q.; Liu, W.; Zhang, Y. Defending against data integrity attacks in smart grid: A deep reinforcement learning-based approach. *IEEE Access* **2019**, *7*, 110835–110845. [CrossRef]
46. Tabakhpour, A.; Abdelaziz, M.M. Neural network model for false data detection in power system state estimation. In Proceedings of the IEEE Canadian Conference of Electrical and Computer Engineering (CCECE), Edmonton, AB, Canada, 5–8 May 2019; pp. 1–5.
47. Zhang, Y.; Wang, J.; Chen, B. Detecting false data injection attacks in smart grids: A semi-supervised deep learning approach. *IEEE Trans. Smart Grid.* **2020**, *12*, 623–634. [CrossRef]
48. González, I.; Calderón, A.J.; Portalo, J.M. Innovative multi-layered architecture for heterogeneous automation and monitoring systems: Application case of a photovoltaic smart microgrid. *Sustainability* **2021**, *13*, 2234. [CrossRef]
49. Kabalci, Y. A survey on smart metering and smart grid communication. *Renew. Sustain. Energy Rev.* **2016**, *57*, 302–318. [CrossRef]
50. Zhang, J.; Hasandka, A.; Wei, J.; Alam, S.M.; Elgindy, T.; Florita, A.R.; Hodge, B.M. Hybrid communication architectures for distributed smart grid applications. *Energies* **2018**, *11*, 871. [CrossRef]
51. Manandhar, K.; Cao, X.; Hu, F.; Liu, Y. Detection of faults and attacks including false data injection attack in smart grid using Kalman filter. *IEEE Trans. Control Netw.* **2014**, *1*, 370–379. [CrossRef]
52. Wu, M.; Xie, L. Online detection of false data injection attacks to synchrophasor measurements: A data-driven approach. In Proceedings of the 50th Hawaii International Conference on System Sciences, Village, HI, USA, 4–7 January 2017.
53. Kurt, M.N.; Yılmaz, Y.; Wang, X. Distributed quickest detection of cyber-attacks in smart grid. *IEEE Trans. Inf. Forensics Secur.* **2018**, *13*, 2015–2030. [CrossRef]
54. Du, D.; Li, X.; Li, W.; Chen, R.; Fei, M.; Wu, L. ADMM-based distributed state estimation of smart grid under data deception and denial of service attacks. *IEEE Trans. Syst. Man Cybern. Syst.* **2019**, *49*, 1698–1711. [CrossRef]
55. Kurt, M.N.; Yılmaz, Y.; Wang, X. Real-time detection of hybrid and stealthy cyber-attacks in smart grid. *IEEE Trans. Inf. Forensics Secur.* **2018**, *14*, 498–513. [CrossRef]
56. Pang, Z.H.; Fan, L.Z.; Sun, J.; Liu, K.; Liu, G.P. Detection of stealthy false data injection attacks against networked control systems via active data modification. *Inf. Sci.* **2021**, *546*, 192–205. [CrossRef]
57. Zhu, R.; Huang, C.; Deng, S.; Li, Y. Detection of False Data Injection Attacks Based on Kalman Filter and Controller Design in Power System LFC. *J. Phys. Conf. Ser.* **2021**, *1861*, 012120. [CrossRef]

sustainability

MDPI

Article

Effect of Crop Establishment Methods and Microbial Inoculations on Augmenting the Energy Efficiency and Nutritional Status of Rice and Wheat in Cropping System Mode

Amit Anil Shahane [1,2], Yashbir Singh Shivay [1,*], Radha Prasanna [3], Dinesh Kumar [1] and Ram Swaroop Bana [1,*]

[1] Division of Agronomy, ICAR-Indian Agricultural Research Institute, New Delhi 110 012, India; amitiari89@gmail.com (A.A.S.); dineshctt@yahoo.com (D.K.)

[2] Department of Agronomy, College of Agriculture under (CAU, Imphal), Kyrdemkulai, Ri-Bhoi 793 105, India

[3] Division of Microbiology, ICAR-Indian Agricultural Research Institute, New Delhi 110 012, India; radhapr@gmail.com

* Correspondence: ysshivay@hotmail.com (Y.S.S.); rsbana@gmail.com (R.S.B.)

Abstract: A field experiment was conducted for two consecutive years with the aim to quantify the role of different nutrient management variables such as microbial inoculation, zinc (Zn) fertilization and optimal and sub-optimal fertilization of nitrogen and phosphorus on the energetic and nutritional status of the rice–wheat cropping system (RWCS). The said nutrient management variables were applied over six different crop establishment methods (CEMs) in RWCS viz. puddled transplanted rice (PTR), system of rice intensification (SRI) and aerobic rice system (ARS) in rice and conventional drill-sown wheat (CDW), system of wheat intensification (SWI) and zero-tillage wheat (ZTW) in wheat. Two microbial consortia viz. *Anabaena* sp. (CR1) + *Providencia* sp. (PR3) consortia (MC1) and *Anabaena-Pseudomonas* biofilmed formulations (MC2) were used in this study, while recommended dose of nitrogen (N) and phosphorus (P) (RDN) (120 kg N ha^{-1} and 25.8 kg P ha^{-1}), 75% RDN and Zn fertilization (soil applied 5 kg Zn ha^{-1} through zinc sulphate heptahydrate) were the other variables. The contribution of microbial consortia, Zn fertilization and RDN (over 75% RDN) to net energy production of RWCS was 12.9–16.1 × 10^3 MJ ha^{-1}, 10.1–11.0 × 10^3 MJ ha^{-1} and 11.7–15.3 × 10^3 MJ ha^{-1}. Among the CEMs, the highest gross and net energy production was recorded in ARS–ZTW with lowest energy required for production of one tonne of system yield (2366–2523 MJ). The system protein yield varies from 494.1 to 957.7 kg ha^{-1} with highest protein yield in 75% RDN + MC2 + Zn applied ARS–ZTW. Among micronutrients, the uptake of Zn and iron (Fe) is sensitive to all studied variables, while manganese (Mn) and cupper (Cu) uptake was found significantly affected by CEMs alone. The combination of 75% RDN + MC2 + Zn in ARS–ZTW was found superior in all respects with 288.3 and 286.9 MJ ha^{-1} net energy production and 2320 and 2473 MJ energy required for production of one tonne system yield in the first and second year of study, respectively.

Keywords: aerobic rice; energetics; nitrogen; protein yield; system of rice intensification; zero-tillage wheat; zinc

Citation: Shahane, A.A.; Shivay, Y.S.; Prasanna, R.; Kumar, D.; Bana, R.S. Effect of Crop Establishment Methods and Microbial Inoculations on Augmenting the Energy Efficiency and Nutritional Status of Rice and Wheat in Cropping System Mode. *Sustainability* **2022**, *14*, 5986. https://doi.org/10.3390/su14105986

Academic Editors: Luis Hernández-Callejo, Sergio Nesmachnow and Sara Gallardo Saavedra

Received: 28 March 2022
Accepted: 26 April 2022
Published: 15 May 2022

1. Introduction

Rice and wheat are the forerunner staple food crops in imparting the energy for humans, directly through carbohydrate and protein as the main components of foods and indirectly through different provisional services. Out of the total protein consumption in India, 56.7% is from cereals [1], while 20% of per-capita energy for humans and 13% protein in the diet of nearly half of the world population were contributed by rice, and this share is much higher in developing countries [2]. The share of both crops to food grain production is 75.11%, while the share in total cereal production was 81.3% [3]. This indicates the role of rice and wheat in meeting the protein requirement of the Indian population. On another side, the contribution of rice and wheat to resource utilization among all crops is the highest with 34.5% and 24.4% of the gross cropped area under rice and wheat, respectively [3]. At the same time, the share of rice in the total fertilizer consumption is 37% for nitrogen (6.98 million tonnes), 37% for phosphorus (2.76 million tonnes) and also 37% for potassium (0.977 million tonnes) in 2020–2021, respectively, and the same for wheat was nearly 24% for nitrogen (4.897 million tonnes) and 24% for phosphorus (2.155 million tonnes). Besides the above-mentioned natural resources, the monetary involvement is much higher in the cultivation of both crops with an average cost of cultivation for rice varying from Indian national rupee (INR) 1082.5 to 2732.6 for 100 kg grain yield, and the same for wheat varies from INR 1109.8 to 2233.9 for 100 kg grain yield, respectively [4]. The monetary criteria such as gross and net returns are used most commonly for calculating crop profitability, while for different artificial resources such as irrigation water, electricity, petroleum products, fertilizers, etc., which are purchased at a subsidized price, the present monetary evaluation is not complete. In this regard, the evaluation of all resources in a single unit, and with it the non-subsidized or original cost, is needed, and this can be carried out by the quantification of all inputs and outputs in terms of energetic and nutritional outcome. The need for accounting for energetics in crops and cropping systems along with monetary returns can be justified by increasing energy scarcity, increasing adoption of energy-efficient CEMs [5–7], the contribution of energy to greenhouse gas emissions and subsidies on fertilizers. As energy scarcity is aggravating and large variants for management practices and input additions are available, the study of these parameters for their energy efficiency will be an important scope and generate valuable scientific information. The requirement of energy per kg of crop produce and reduction in energy requirements for different field operations, and higher net energy production with the same level of resources, are useful criteria for judging efficiency in crop production. This high contribution of both crops to input consumption and meeting the energy and nutritional requirements of human beings creates scope to evaluate both crops in the cropping system mode for their energetic and nutritional outcomes.

Rice and wheat had significant variation in the crop establishment methods (CEMs) and cultivation methods and this can be explained by significant variations in hydrological regimes in rice ecosystems in India [8–11] and variation in tillage and land configuration in wheat [7,12]. The significance of energetics in a crop/cropping system has both economic and environmental bias. The largest contribution of the energy sector to global warming [13] with finite, limited and shrinking conventional (coal and petroleum-based) energy resources and increasing emphasis of policy makers on use of solar, wind and hydroelectric energy explain the environmental bias of energetics, while increasing the price per unit of energy leading to increasing prices of inputs, promotion of energy-efficient machines/equipment in crop production [14,15] and increasing wages of labour elucidate the economic bias of energy use. The energy equivalents given by different authors [16–18] indicate the highest energy equivalent per unit input was accounted by different nutrients. The energy equivalent for 1 kg nitrogen, phosphorus, potassium and Zn was 60.6 MJ, 11.1 MJ, 6.7 MJ and 20.2 MJ, respectively. The higher energy equivalent signifies the need for studying nutrient management variables for their role in energetics, while variation in CEMs and cultivation methods leading to variation in tillage requirements create scope for studying their energetics with varied levels of inputs. Along with the energy equivalent, the nutritional status of both crops needs to be studied considering their contribution to human nutrition and growing concerns of micronutrient deficiency [19,20] and other health-related risks [21,22]. The CEMs were studied for their energetics, while scientific information on the interactions of different CEMs and input additions (microbial inoculation, Zn fertilization and optimum and sub-optimum fertilization) on the energetic and nutritional status of RWCS is lacking, which was considered a research gap. Considering the increased number of crop establishment methods (CEMs) in RWCS with significant variations, the significant contribution of both rice and wheat to input consumption and human nutrition and the high energy equivalent of nutrients, the study was planned with two objectives: (i) to identify the energy-efficient CEMs in RWCS and the role of microbial inoculations and Zn fertilization in enhancing the energetics of RWCS; and (ii) to know about the micronutrient uptake in rice and wheat as affected by applied treatments, thereby increasing the nutritional status of grains.

2. Material and Methods

2.1. Experimental Site

The field experiment was conducted consecutively for two years (2013–2014 and 2014–2015) at Research Farm of ICAR-Indian Agricultural Research Institute, New Delhi (latitude of 28°38′ N, longitude of 77°10′ E and altitude of 228.6 m above the mean sea level). Two crops in a year including rice during wet season (June to September) and wheat during dry/winter season (November to April) were grown. The climate of New Delhi is of subtropical and semi-arid type with hot and dry summers followed by monsoon rains in July-September and cold winters in November–April and falls under the agro-climatic zone 'Trans-Gangetic plains'. The mean annual normal rainfall and evaporation are 650 and 850 mm, respectively. Amount of rainfall received during growing duration of first cycle of RWCS (2013–2014) was 1497.4 mm, out of which 1349.8 mm was received during rice growing season and 147.6 mm was received during wheat growing season. In second cycle (2014–2015), total rainfall was 760 mm, out of which 451.4 mm received in rice growing season and 308.6 mm during wheat growing season. The number of rainy days was higher during first rice growing season (39 days) than second rice growing season (22 days). The highest amount of rainfall during rice growing season was received during 33rd (196.1 mm) and 29th meteorological weeks (112.7 mm) in first and second year, while in case of wheat, 7th (53 mm) and 9th (135.4 mm) meteorological weeks received highest rainfall in first and second year, respectively (Supplementary Tables S1 and S2).

The soil was sandy clay loam (Typic Ustochrept) in texture with a mechanical composition [23] of 51.4% sand, 22.2% silt and 26.4% clay. The soils of experimental field had 0.54% organic C [24], 257 kg ha^{-1} alkaline permanganate oxidizable N [25], 17 kg ha^{-1} available P (Olsen's method) [26], 327 kg ha^{-1} 1 N ammonium acetate exchangeable K [27] and 0.85 mg kg^{-1} of available zinc [28]. The pH of the soil was 7.6 (1:2.5 soil-to-water ratio) [29].

2.2. Experimental Details

The field experiment was planned in split-plot design with six crop establishment methods (CEMs) with three for each rice (*Pusa Sugandh 5*) and wheat (*HD 2967*) as main plot (net area for each main plot was 256.5 m^2). The CEMs were arranged as puddled transplanted rice (PTR) followed by (*fb*) conventional drill-sown wheat (CDW), system of rice intensification (SRI) *fb* system of wheat intensification (SWI) and aerobic rice system (ARS) *fb* zero-tillage wheat (ZTW). In all these CEMs, nine subplot treatments were applied (net area for each sub-plot was 9.5 m^2), which include RDN (recommended dose of nutrients) (120 kg ha^{-1} N and 25. 8 kg ha^{-1} P), 75% RDN, 75% RDN + *Anabaena* sp. (CR1) + *Providencia* sp. (PR3) consortia (MC1) and 75% RDN + *Anabaena-Pseudomonas* biofilmed formulations (MC2). These four treatments were applied with and without Zn (soil applied 5 kg Zn ha^{-1} through zinc sulphate heptahydrate) making total eight treatments and one control (no fertilizer). All treatments were replicated thrice.

2.3. Crop Establishment Methods (CEMs)

The details for CEMs of rice and wheat are mentioned in Tables 1 and 2. In order to have the same crop growth duration in all three methods of cultivation, sowing of rice in main field for ARS and sowing rice in nursery for transplanting in both PTR and SRI was performed on the same date. The PTR is traditionally followed by the CEM in which rice is grown in standing water. The level of standing water is maintained by reduction in soil infiltration rate through soil cultivation in standing water before transplanting (puddling) and applying irrigation at frequent intervals. The level of water is maintained at 2–3 cm during vegetative growth stage and increased up to 5 cm during flowering and grain filling stage. In SRI [30–32], soil puddling is carried out the same as that of PTR and soil water level is maintained at saturation. The seedlings at13–14 days old were transplanted with 1–2 healthy seedlings per hill at a spacing of 20 cm × 20 cm. The ARS is growing of rice in unsaturated, unpuddled and arable soil conditions [33]. The soil is maintained at field capacity and direct sowing of pre-soaked rice grain was conducted through seed-drill. In case of wheat, drill-sowing of wheat is mostly followed in India in which row sowing of seed at 22.5 cm with seed drill is performed, while SWI [34–36] is a new CEM involving dibbling or transplanting of young seedlings at 20 cm × 20 cm spacing. The ZTW is gaining acceptance in Indo-Gangetic plains (IGPs) by the farmers due to energy and cost saving [12] and timely sowing [7].

2.4. Application of Microbial Inoculation and Fertilizers

Two microbial consortia were applied in present study (*Anabaena* sp. (CR1) + *Providencia* sp. (PR3) consortia (MC1) and *Anabaena-Pseudomonas* biofilmed formulations (MC1)) [37,38]. For application of microbial consortia, a thick paste of respective culture was made in carboxyl methyl cellulose and applied to rice seedlings in PTR and SRI by dipping roots in paste of respective culture for half an hour before transplanting. In ARS, pre-soaked seeds were treated with thick paste of culture made in carboxyl methyl cellulose half an hour before sowing. In wheat, thick paste of respective culture was made in carboxyl methyl cellulose (CMC) and seeds were treated with this thick paste in all CEMs for half an hour just before sowing. For application of N, P and K chemical fertilizers, urea, single super phosphate and muriate of potash were used, while zinc sulphate heptahydrate was used for supply of Zn. Among nutrients, P, K and Zn were applied at the time of sowing and N was split, applied in both rice and wheat (Tables 1 and 2).

2.5. Energy Calculation

For calculation of gross energy, grain and straw yield was measured and their cited energy equivalents [16,18] were considered. The energy equivalents mentioned in [16–18] were used to calculate the energy input (Table 3). The energy input consists of both direct (human labour, diesel and electricity) and indirect (seed, fertilizers and machinery) energy in rice and wheat. The net energy output is calculated by subtracting energy input from gross energy output. The energy input is also expressed as energy tonne^{-1} of economic yield produced.

2.6. Calculation of Grain Yield, Protein Yield and Micronutrient Uptake

Both rice and wheat were harvested at harvest maturity and threshed produce obtained from net plot areas were cleaned, dried and weighed at 14% moisture content and expressed as Mg ha^{-1}. The protein yield was calculated based on the nitrogen concentration in grain. For determining the nitrogen content, the plant sample (0.5 g each) was digested by using 10 mL of analytical grade concentrated sulphuric acid along with a pinch of digestion mixture ($CuSO_4$ + K_2SO_4) to determine total nitrogen content. The samples were analyzed by using Kjeldahl's apparatus [39] and were expressed as percentage. The zinc (Zn), iron (Fe), manganese (Mn) and copper (Cu) concentrations in rice and wheat plant samples were determined as per the procedure described by [40] using Atomic Absorption Spectrophotometer (AAS) and expressed as mg kg^{-1}. For calculating the uptake in grain, grain yield was measured at 12% moisture content. For rice, white rice kernel was used instead of rough rice.

Table 1. The details about methodologies of different CEM in rice.

Method of Cultivation	Method of Sowing	Seed Rate (kg ha^{-1})	Spacing (cm)	Age of Seedling	Seedling hill^{-1}	Land Preparation	Water Management	Number of Irrigation	Depth of Irrigation	Weed Management	Nutrient Application Method and Timing
PTR	Transplanting (manual)	20	20 × 15	23–25 days old	2–3	One ploughing, one harrowing and puddling twice	5 cm water applied at each irrigation; puddled and saturated	11 in first year and 18 in second year	5 cm puddled saturated	Two hand weeding in each crop	Broadcasting; 1/3 at 5 DAT, 1/3 at 25 days after transplanting (DAT) and 1/3 at 55 DAT for N; All dose of P, K and Zn at 5 DAT
SRI	Transplanting (manual)	5	20 × 20	13–14 days old	1	One ploughing, one harrowing and puddling twice	2 cm up to panicle initiation and 5 cm thereafter	11 in first year and 20 in second year	2 cm up to panicle initiation and 5 cm thereafter	Two hand weeding in each crop	Broadcasting; 1/3 at 5 DAT, 1/3 at 25 DAT and 1/3 at 55 DAT for N; All dose of P, K and Zn at 5 DAT
ARS	Direct sowing of seed in main field	60	20 cm row to row	Direct sowing of seed in field	-	One ploughing followed by harrowing	2 cm to maintain field capacity moisture level, non-saturated and non-puddled	10 in first year and 24 in second year	2 cm to maintain field capacity moisture level, non-saturated and non-puddled	Three hand weeding in each crop	Broadcasting; 1/3 at sowing, 1/3 at 30 days after sowing (DAS) and 1/3 at 60 DAS for N; All dose of P, K and Zn at sowing

Table 2. The details about methodologies of different CEMs in wheat.

Method of Cultivation	Method of Sowing	Seed Rate (kg ha^{-1})	Spacing	Land Preparation	Water Management	Weed Management	Nutrient Application Method and Timing
CDW	Sowing through seed drill	100	22.5 cm row to row	One ploughing followed by one harrowing and planking	Four and six irrigations at critical growth stages in first and second year, respectively, in all CEMs	Two hand weeding in each crop at 20 and 40 days after sowing (DAS) in all CEMs	Broadcasting; 1/3 at sowing, 1/3 at 30 DAS and 1/3 at 60 DAS for N; All dose of P, K and Zn at sowing
SWI	Dibbling of seeds (manual)	30	20 cm × 20 cm	One ploughing followed by one harrowing and planking			
ZTW	Sowing through seed drill	120	22.5 cm row to row	Direct sowing without cultivation			

Table 3. Energy equivalents used for calculation of energy input and output in production system [16–18].

S. No.	Input Used	Energy Equivalent (MJ Unit^{-1})
1.	Human labour	1.96
2.	Diesel (per litre)	56.31
3.	Farm machinery	62.7
4.	Fertilizer (Nitrogen MJ kg^{-1} N)	60.60
5.	Fertilizer (Phosphorus MJ kg^{-1} N)	11.2
6.	Fertilizer (Potassium MJ kg^{-1} N)	6.7
7.	Fertilizer (Zinc Sulphate Heptahydrate MJ kg^{-1} N)	20.2
8.	Electricity (per unit)	11.93
9.	Rice and wheat grain (MJ kg^{-1})	14.7
10.	Rice and wheat straw (MJ kg^{-1})	12.5

2.7. Statistical Analysis

The data obtained from the experiment were statistically analyzed using analysis of variance (ANOVA) using the IBM SPSS statistics package and the Duncan's multiple range test to quantify and evaluate the source of variation at the 5% level of significance.

3. Results

3.1. Energy Input

Energy requirement was higher in rice than wheat in both years (Figures 1 and 2). In both crops, the second year had a higher energy requirement than the first year. Among all major operations, fertilization requires higher energy in rice, wheat and the rice–wheat cropping system. The share of fertilizer application in total energy consumption is 54–62%, 66–75% and 59–68% in rice, wheat and RWCS, respectively. The fertilization (54–62%), land preparation (17–22%) and irrigation (8–10%) are the three major consumers of energy in rice. The energy required for nursery, seed and sowing accounts for 10–11% in PTR, 5–6% SRI and 6% in ARS. In wheat, 66–75% of the total energy was accounted for by fertilization. The contribution of land preparation to the total energy consumption was 16–17% in CDW and SWI, while it was zero in ZTW (Figures 1 and 2). The seed requirement was the lowest in SWI and therefore accounts for only 3–4% of total energy. The CDW and ZTW require 11% and 16% energy for seed. In the case of system energy inputs, fertilization, land preparation and irrigation accounts for 59–68%, 9–19% and 6–10% of total energy, respectively. Among all operations, the energy required for nursery, seed requirement, land preparation and fertilization varies across CEMs. The renewable energy (seed and labour) consumption in rice varies from 1257.0 to 1879.7 MJ ha^{-1}, while in wheat it varies from 1258.2 to 2516.6 MJ ha^{-1} (Table 4). The highest renewable energy consumption was observed in ARS and ZTW, while the highest non-renewable energy consumption was recorded in PTR and CDW. In all CEMs of rice, indirect energy accounts for 61.8 to 69.9% of total energy inputs and in wheat its share is 75.9 to 90.7%. In rice, both direct and indirect energy consumption was highest in PTR. In case of wheat, direct energy consumption was highest in SWI, while indirect energy use was highest in ZTW (Table 5). The application of microbial consortia and Zn fertilization require 20 and 101 MJ ha^{-1} energy, while the application of microbial consortia decreases energy requirements by 1964.5 MJ ha^{-1} over RDN (Tables 4 and 5). Among CEMs, PTR had the highest energy requirement and it was higher by 1222–1229 and 2043–2391 MJ ha^{-1}, respectively, than SRI and ARS. In wheat, ZTW reduces the energy requirement by 1655 and 684 MJ ha^{-1} over CDW and SWI. On the system basis, ARS-ZTW was found superior in saving energy.

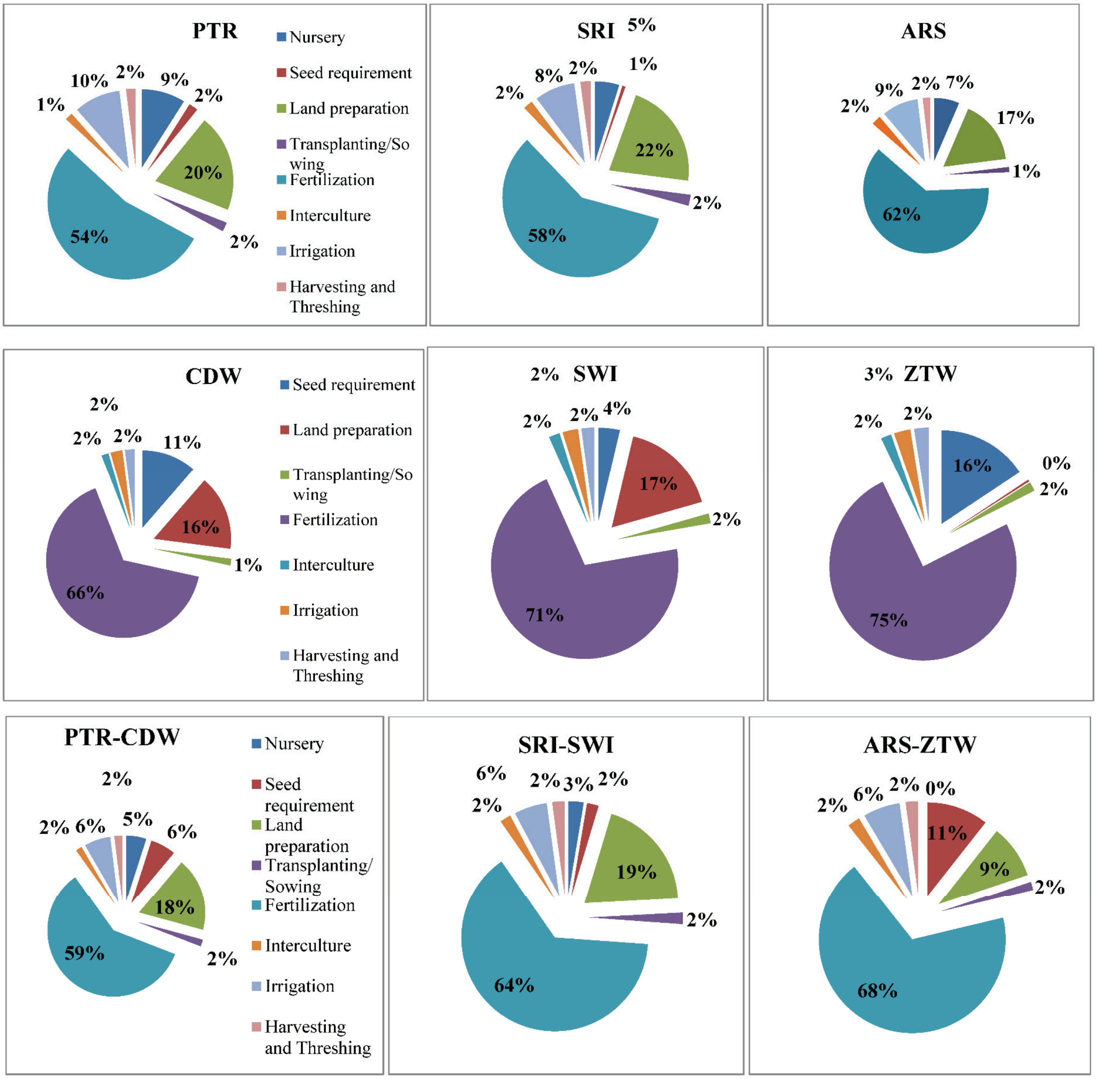


Figure 1. Effect of crop establishment methods on energy requirement for different inputs and operations in rice–wheat cropping system in 2013–2014.

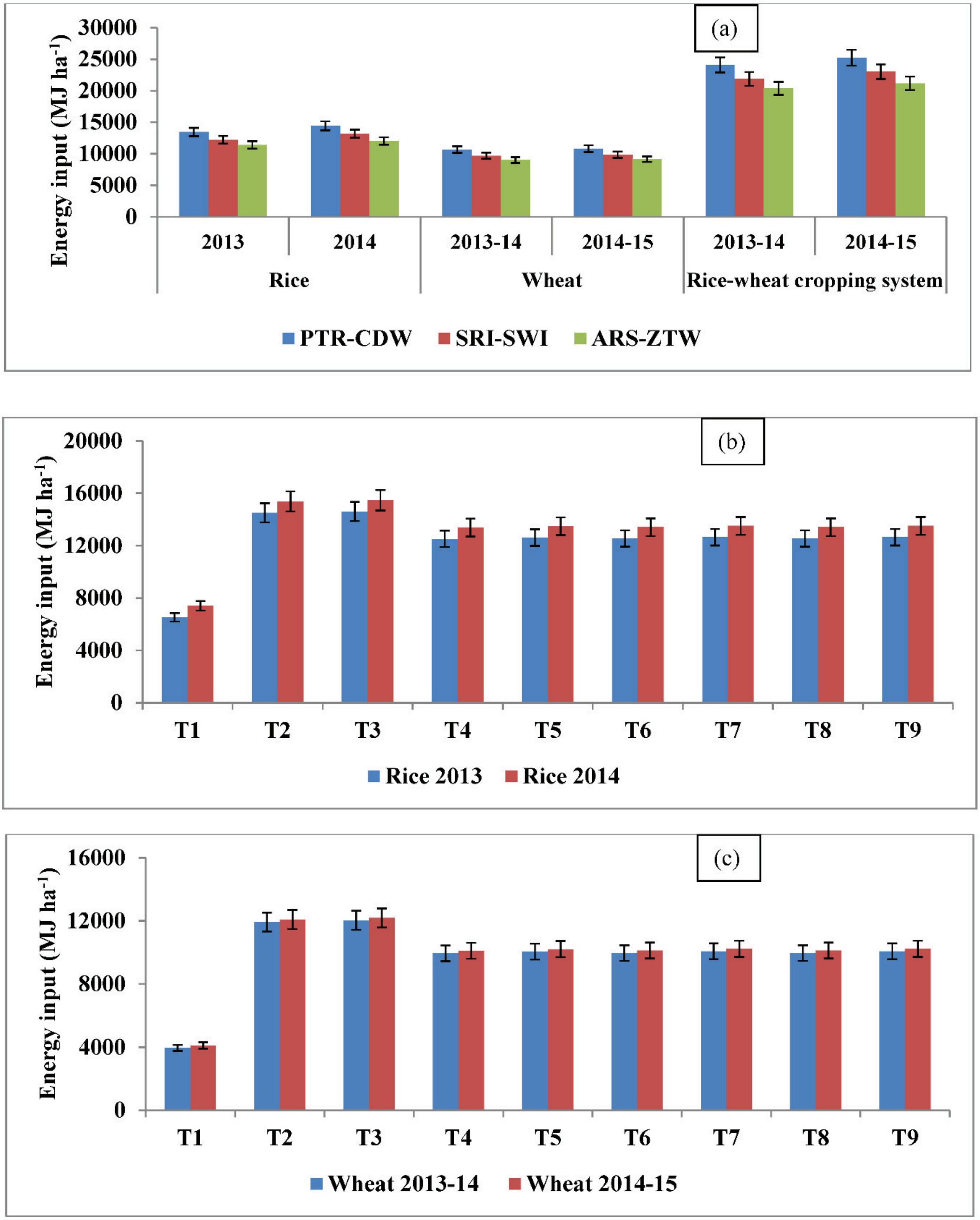


Figure 2. *Cont.*

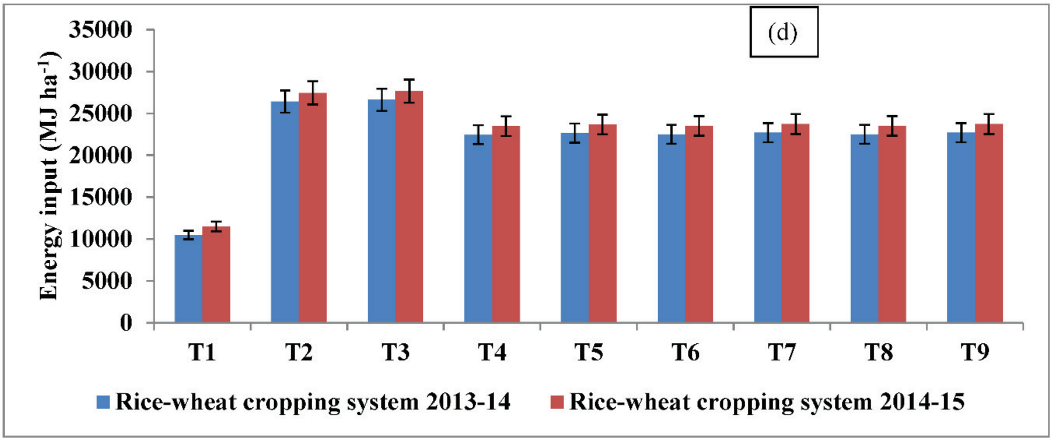


Figure 2. Effect of crop establishment methods (**a**) and nutrient management (**b**–**d**) on energy input requirement in rice–wheat cropping system. (T1: Control, T2: RDN, T3: RDN + Zn, T4: 75% RDN, T5: 75% RDN + Zn, T6: 75% RDN + MC1, T7: 75% RDN + MC1 + Zn, T8: 75% RDN + MC2 and T9: 75% RDN + MC2 + Zn). RDN Recommended dose of nutrients 120 kg N ha^{-1} and 25.8 kg P ha^{-1}; Zn: Soil applied with 5 kg Zn ha^{-1} through zinc sulphate heptahydrate; MC1: (*Anabaena* sp. (CR1) + *Providencia* sp. (PR3) consortia; MC2: *Anabaena-Pseudomonas* biofilmed formulations.

3.2. Energy Production

The PTR and SRI were found statistically superior to ARS in gross energy production in both years (Table 6). The net energy production in SRI was significantly higher over both PTR and ARS, while between PTR and ARS, PTR was found superior to ARS. The net energy production in SRI was higher by 1000 to 1500 MJ ha^{-1} over PTR and 4800 to 5100 MJ ha^{-1} over ARS. The lower net energy production in ARS was mainly due to lower yield. The saving in energy per tonne of rough rice produced in ARS was 401–492 and 86–167 MJ t^{-1} more than PTR and SRI. In the case of wheat, both gross and net energy production in ZTW were significantly higher than CDW and SWI. The increase in gross and net energy production in ZTW over CDW was 7500–8000 and 9200–9600 MJ ha^{-1} and similarly for ZTW versus SWI was 8200–8600 and 8900–9300 MJ ha^{-1}, respectively. The ZTW required the lowest amount of energy for production of a tonne of grain. The saving in energy per tonne of grain produced was 216–488 and 274–275 MJ ha^{-1} over CDW and SWI, respectively. The system gross energy output was highest in ARS–ZTW but remained on par with all other CEMs in the first year. During the second year, gross energy production in ARS–ZTW was significantly higher than SRI–SWI and remained on par with PTR–CDW. In regard to net energy production, ARS–ZTW was found superior to both PTR–CDW and SRI–SWI and increased net energy production by 5900 and 4100 MJ ha^{-1}. The energy required to produce a tonne of system yield varied between 2523 and 3039 MJ ha^{-1} and all three systems differed significantly, with ARS–ZTW found superior over the rest.

Table 4. Partitioning of energy inputs in different forms of energy in selected crop establishment methods of rice during first cycle of RWCS.

Particular	Direct Energy (MJ ha^{-1})				Total Direct Energy	Indirect Energy (MJ ha^{-1})				Total Indirect Energy (MJ ha^{-1})	Grand Total (MJ ha^{-1})
	Renewable	Non-Renewable				Renewable	Non-Renewable				
	Human Labour	Diesel	Electricity	Total		Seed	Fertilizers	Machinery	Total		
Energy requirement in puddled transplanted rice (PTR) for RDN + Zn											
Field preparation	139.2	3350.4	-	3350.4	3489.6	-	-	328.9	328.9	328.9	3818.5
Seed and sowing	341	-	-	-	341.0	294	-	-	-	294	635
Fertilization	41.2	-	-	-	41.2	-	9047	-	9047	9047	9088.2
Inter-cultural operation	270.5	-	-	-	270.5	-	-	-	-	-	270.5
Irrigation	173.5	-	1371.9	1371.9	1545.4	-	-	34.02	34.02	34.02	1579.4
Harvesting	305.8	-	-	-	305.8	-	-	-	-	-	305.8
Total	1271.2	3350.4	1371.9	4722.3	5993.5	294	9047	362.9	9409.9	9703.9	15,697.4
Energy requirement in system of rice intensification (SRI) for RDN + Zn											
Field preparation	100.9	3054.8	-	3054.8	3155.7	-	-	299.9	299.9	299.9	3455.6
Seed and sowing	341.0	-	-	-	341.0	88.2	-	-	-	88.2	429.2
Fertilization	41.2	-	-	-	41.2	-	8744	-	8744	8744	8785.2
Inter-cultural operation	270.5	-	-	-	270.5	-	-	-	-	-	270.5
Irrigation	109.4	-	1085.6	1085.6	1195.0	-	-	26.9	26.9	26.9	1221.9
Harvesting	305.8	-	-	-	305.8	-	-	-	-	-	305.8
Total	1168.8	3054.8	1085.6	4140.4	5309.2	88.2	8744	326.8	9070.8	9159	14,468.2
Energy requirement in aerobic rice system (ARS) for RDN + Zn											
Field preparation	101.9	1970.8	-	1970.8	2072.7	-	-	191.2	191.2	191.2	2263.9
Seed and sowing	176.4	-	-	-	176.4	882	-	-	-	882	1058.4
Fertilization	35.28	-	-	-	35.28	-	8441.0	-	8441.0	8441.0	8475.3
Inter-cultural operation	329.3	-	-	-	329.3	-	-	-	-	-	329.3

Table 4. *Cont.*

Particular	Direct Energy (MJ ha^{-1})				Total Direct Energy	Indirect Energy (MJ ha^{-1})				Total Indirect Energy (MJ ha^{-1})	Grand Total (MJ ha^{-1})
	Renewable	Non-Renewable				Renewable	Non-Renewable				
	Human Labour	Diesel	Electricity	Total		Seed	Fertilizers	Machinery	Total		
Irrigation	94.1	-	1145.3	1145.3	1239.4	-	-	28.4	28.4	28.4	1267.8
Harvesting	258.7	-	-	-	258.7	-	-	-	-	-	258.7
Total	995.7	1970.8	1145.3	3116.1	4111.8	882	8441	219.6	8660.6	9542.6	13,654.4

Table 5. Partitioning of energy inputs in different forms of energy in selected crop establishment methods of wheat during first cycle of RWCS.

Particular	Direct Energy				Total Direct Energy	Indirect Energy				Total Indirect Energy	Grand Total
	Renewable	Non-Renewable				Renewable	Non-Renewable				
	Human Labour	Diesel	Electricity	Total		Seed	Fertilizers	Machinery	Total		
Energy requirement in conventional drill-sown wheat (CDW) for RDN + Zn											
Field preparation	76.44	1773.8	-	1773.8	1850.2	-	-	169	169	169	2019.2
Seed and sowing	176.4	-	-	-	176.4	1470	-	-	-	1470	1646.4
Fertilization	35.3	-	-	-	35.3	-	8441	-	8441	8441	8476.3
Inter-cultural operation	188.2	-	-	-	188.2	-	-	-	-	-	188.2
Irrigation	23.5	-	286.3	286.3	309.8	-	-	7.1	7.1	7.1	316.9
Harvesting	258.7	-	-	-	258.7	-	-	-	-	-	258.7
Total	758.5	1773.8	286.3	2060.1	2818.6	1470	8441	176.1	8617.1	10,087.1	12,905.7

Table 5. *Cont.*

Particular	**Direct Energy**				**Total Direct Energy**	**Indirect Energy**				**Total Indirect Energy**	**Grand Total**
	Renewable	**Non-Renewable**				**Renewable**	**Non-Renewable**				
	Human Labour	**Diesel**	**Electricity**	**Total**		**Seed**	**Fertilizers**	**Machinery**	**Total**		
Energy requirement in system of wheat intensification (SWI) for RDN + Zn											
Field preparation	76.4	1773.7	-	1773.7	1850.2	-	-	169	169	169	2019.2
Seed and sowing	199.9	-	-	-	199.9	441	-	-	-	441	640.9
Fertilization	35.3	-	-	-	35.3	-	8441	-	8441	8441	8476.3
Inter-cultural operation	223.4	-	-	-	223.4	-	-	-	-	-	223.4
Irrigation	23.5	-	286.3	286.3	309.8	-	-	7.1	7.1	7.1	316.9
Harvesting	258.7	-	-	-	258.7	-	-	-	-	-	258.7
Total	817.2	1773.7	286.3	2060	2877.3	441	8441	176.1	8617.1	9058.1	11,935.4
Energy requirement in zero-tillage wheat (ZTW) for RDN + Zn											
Field preparation	35.3	-	-	-	35.3	-	-	-	-	-	35.3
Seed and sowing	176.4	-	-	-	176.4	1764	-	-	-	1764	1940.4
Fertilization	35.3	-	-	-	35.3	-	8441	-	-	8441	8476.3
Inter-cultural operation	199.9	-	-	-	199.9	-	-	-	-	-	199.9
Irrigation	23.5	-	286.3	286.3	309.8	-	-	7.1	7.1	7.1	316.9
Harvesting	282.2	-	-	-	282.2	-	-	-	-	-	282.8
Total	752.6	00	286.3	286.3	1038.9	1764	8441	7.1	8448.1	10,212.1	11,251.6

Table 6. Effect of crop establishment methods on energetic and protein yield of rice, wheat and rice–wheat cropping system.

Treatment	Gross Energy ($\times10^3$ MJ ha^{-1})		Net Energy ($\times10^3$ MJ ha^{-1})		Energy tonne^{-1} of Grain (MJ tonne^{-1})		Protein Yield (kg ha^{-1})	
	2013	2014	2013	2014	2013	2014	2013	2014
Rice								
Puddled transplanted rice (PTR)	151.2a	149.4a	137.8b	134.9b	3276a	3629a	246.5a	229.1a
System of rice intensification (SRI)	151.5a	149.6a	139.2a	136.4a	2961b	3304b	247.3a	229.6a
Aerobic rice system (ARS)	145.5b	143.7b	134.1c	131.6c	2875c	3137c	221.2b	206.0b
Wheat								
Conventional drill-sown wheat (CDW)	140.4b	142.7b	129.8b	131.9b	2421a	2497a	552.3b	535.1b
System of wheat intensification (SWI)	139.8b	142.0b	130.1b	132.2b	2208b	2281b	550.8b	533.1b
Zero-tillage wheat (ZTW)	148.4a	150.2b	139.4a	141.1a	1933c	2007c	639.1a	621.4a
Rice–wheat cropping system								
Puddled transplanted rice (PTR)–conventional drill-sown wheat (CDW)	291.7a	292.1a	267.6b	266.8b	2834a	3039a	798.8b	764.3b
System of rice intensification (SRI)–system of wheat intensification (SWI)	291.3a	291.6a	269.4b	268.6b	2573b	2773b	798.2b	762.7b
Aerobic rice system (ARS)–zero-tillage wheat (ZTW)	293.9a	293.9a	273.5a	272.7a	2366c	2523c	860.2a	827.3a

Within a column, means followed by the same letter are not significantly different at the 0.05 level of probability by the Duncan's multiple range test.

The gross energy production in rice was highest in RDN + Zn applied in PTR and found significantly superior over same treatments applied in SRI and ARS in both years (Table 7). Application of 75% RDN + MC1 + Zn and 75% RDN + MC2 + Zn in PTR and SRI remained on par with RDN and found significantly superior over same treatment applied in ARS in first year, while in second year only 75% RDN + MC2 + Zn in SRI was found on par with RDN. The net energy production was highest in 75% RDN + MC2 + Zn in SRI and found superior over same treatment applied in ARS in both years. The net energy production in 75% RDN + MC2 was higher by 900–1000 and 7300–8600 MJ ha^{-1} than RDN and 75% RDN (averaged over all CEMs). Application of MC1 increased net energy production by 6800–8300, 6900–8500 and 7100–8600 MJ ha^{-1}, respectively, in PTR, SRI and ARS. Similarly, increase in net energy production by MC2 was 7100–8400, 7000–8600 and 7500–8800 MJ ha^{-1}, respectively. The zinc fertilization significantly increased gross and net energy production in all CEMs and in all treatments. The increase in gross and net energy production due to Zn fertilization varied between 1600 and 7300 and 1400 and 7100 MJ ha^{-1}, respectively. The lowest amount of energy for production of one tonne of grain was in control. Among CEMs, control in ARS had significantly lower energy per tonne of rice grain produced. Application of MC1 lower energy required per tonne of grain produced by 167–233 MJ tonne^{-1} and MC2 by 183 to 234 MJ tonne^{-1} over 75% RDN.

Table 7. Effect of nutrient management options on energetic and protein yield of rice in different crop establishment methods.

Treatment	Gross Energy ($\times 10^3$ MJ ha^{-1})		Net Energy ($\times 10^3$ MJ ha^{-1})		Energy tonne^{-1} of Grain (MJ tonne^{-1})		Protein Yield (kg ha^{-1})	
	2013	2014	2013	2014	2013	2014	2013	2014
Puddled transplanted rice (PTR)								
Control	129.6j	127.0j	121.9l	118.4p	2395l	2842j	161.0j	144.2i
RDN *	154.3cd	152.0d	138.7efg	135.5ghij	3704a	4089a	258.2de	240.3d
RDN + Zn **	160.4a	159.3a	144.7ab	142.6a	3604ab	3897b	292.6a	274.9a
75% RDN	144.9h	144.0gh	131.3i	129.4m	3515bc	3825bc	215.5gh	202.7f
75% RDN + Zn	149.3f	147.2e	135.6h	132.5kl	3402cd	3783c	220.4fgh	204.5ef
75% RDN + MC1	153.2d	150.8d	139.6def	136.2fghi	3262efg	3642d	258.2de	235.3d
75% RDN + MC1 + Zn	157.8ab	156.3b	144.1abc	141.6abc	3167gh	3476e	283.3ab	267.6ab
75% RDN + MC2	153.3d	151.2d	139.7def	136.5efgh	3262efg	3623d	254.6de	235.1d
75% RDN + MC2 + Zn	158.1ab	156.6b	144.4ab	141.9ab	3174gh	3487e	274.6bcd	257.7bc
System of rice intensification (SRI)								
Control	131.4j	128.7j	125.0k	121.3o	1881m	2293k	172.6j	155.5i
RDN *	154.9cd	152.6d	140.6dc	137.3defg	3421cd	3797bc	259.8cde	241.8d
RDN + Zn **	156.5bc	155.4bc	142.0bcd	139.9bcd	3324def	3613d	290.7ab	273.1a
75% RDN	145.3gh	144.5fg	132.9i	131.1lm	3202fgh	3510e	208.8ghi	196.4fg
75% RDN + Zn	150.0ef	147.8e	137.5fgh	134.4hijk	3096hi	3467e	222.1fg	206.2ef
75% RDN + MC1	153.8cd	151.4d	141.4cde	138.0defg	2972j	3342f	255.6de	232.9d
75% RDN + MC1 + Zn	158.3ab	156.8b	145.8a	143.3a	2894jk	3199gh	276.4abc	257.3bc
75% RDN + MC2	153.9cd	151.8d	141.5cde	138.4def	2979ij	3333f	253.0de	233.6d
75% RDN + MC2 + Zn	159.0ab	157.4ab	146.5a	143.9a	2876jk	3181gh	286.9ab	269.5ab
Aerobic rice system (ARS)								
Control	122.0k	119.3k	116.4m	113.1q	1737n	2049l	154.1k	138.5j
RDN *	149.2f	146.9ef	135.7h	132.8kl	3341de	3637d	232.8f	216.1e
RDN + Zn **	154.6cd	153.5cd	140.9de	139.2cde	3269efg	3484e	261.9cde	245.5cd
75% RDN	139.1i	138.3i	127.6j	126.1n	3118h	3340f	193.0i	181.2h
75% RDN + Zn	143.8h	141.7h	132.2i	129.4m	2996ij	3282fg	204.2hi	189.1gh
75% RDN + MC1	147.8fg	145.4efg	136.2gh	133.2jkl	2900jk	3191gh	218.5fgh	198.1fg
75% RDN + MC1 + Zn	152.4de	150.9d	140.7de	138.6def	2810k	3037i	251.0de	243.3d
75% RDN + MC2	148.0f	145.9efg	136.4gh	133.6ijkl	2894jk	3168h	220.9fgh	203.3ef
75% RDN + MC2 + Zn	152.6de	151.1d	140.9de	138.7def	2814k	3045i	254.5de	238.5d
Nutrient management	*	*	*	*	*	*	*	*
Interaction	*	*	*	*	*	*	*	*

Within a column, means followed by the same letter are not significantly different at the 0.05 level of probability by the Duncan's multiple range test. "*": Indicates significant different of treatments the 0.05 level of probability by the Duncan's multiple range test; RDN *: Recommended dose of nutrients 120 kg N ha^{-1} and 25.8 kg P ha^{-1}; Zn **: Soil applied 5 kg Zn ha^{-1} through zinc sulphate heptahydrate; MC1: (*Anabaena* sp. (CR1) + *Providencia* sp. (PR3) consortia; MC2: *Anabaena-Pseudomonas* biofilmed formulations.

In wheat, the highest amount of gross energy production was recorded in RDN + Zn in ZTW and remained on par with 75% RDN + MC1 + Zn and 75% RDN + MC2 + Zn in ZTW

(Table 8). These three treatments were found significantly superior over same treatment applied in CDW and SWI except RDN in CDW. The net energy production in second year was 100 to 3500 MJ ha^{-1} higher than first year. The application of 75% RDN + MC2 + Zn had the highest net energy production. Application of MC1 and MC2 increases net energy production by 5500 to 6700 and 6800 to 7700 MJ ha^{-1}. Similarly increase in net energy production due to Zn fertilization was 1200 to 7900 MJ ha^{-1}. The energy per tonne of wheat grain produced varied between 786 and 2858 MJ tonne^{-1} in the first year and 853 and 2956 MJ tonne^{-1} in the second year. Application of microbial consortia significantly reduces energy required for production of one tonne of wheat grain, while Zn fertilization found statistically superior when applied with RDN in CDW during both the years and 75% RDN + MC1 in CDW and SWI in first year. The system gross and net energy production varied between 247.2 and 311.9 × 10^3 MJ ha^{-1} and 233.6 and 288.3 × 10^3 MJ ha^{-1} (Table 9). The highest gross and net energy production was found with RDN + Zn in ZTW and 75% RDN + MC2 + Zn in ZTW, respectively. The increase in system net returns due to microbial consortia and Zn fertilization was 12,900 to 16,100 and 4800 to 12,040 MJ ha^{-1}, respectively.

Table 8. Effect of nutrient management options on energetic and protein yield of wheat in different crop establishment methods.

Treatment	Gross Energy (×10^3 MJ ha^{-1})		Net Energy (×10^3 MJ ha^{-1})		Energy tonne^{-1} of Grain (MJ tonne^{-1})		Protein Yield (kg ha^{-1})	
	2013–2014	2014–2015	2013–2014	2014–2015	2013–2014	2014–2015	2013–2014	2014–2015
Conventional drill–wheat (CDW)								
Control	119.9l	120.2m	115.1i	115.2i	1358m	1439m	340.7h	312.6k
RDN *	143.8defgh	145.5defghi	131.0defg	132.5efgh	2858a	2956a	583.3de	561.5fg
RDN + Zn **	150.0abcde	151.4abcdef	137.1cde	138.3cde	2756b	2858b	647.8b	624.9bcd
75% RDN	134.3jk	138.0jkl	123.5gh	127.1h	2590c	2643d	498.5f	486.7i
75% RDN + Zn	135.6ijk	139.4ijkl	124.7fgh	128.3gh	2589c	2641d	503.3f	496.2hi
75% RDN + MC1	141.0fghijk	144.3fghij	130.1efg	133.3efgh	2469d	2528e	560.5e	546.0g
75% RDN + MC1 + Zn	148.7cdefg	150.2cdefg	137.8cde	139.2cde	2360efg	2449efg	633.4bc	612.0bcde
75% RDN + MC2	142.1efghij	145.3defghi	131.2defg	134.3defg	2450de	2510ef	565.8e	555.9g
75% RDN + MC2 + Zn	148.6cdefg	149.9cdef	137.6cde	138.8cde	2360efg	2453efg	637.8b	620.4bcd
System of wheat intensification (SWI)								
Control	123.8l	124.2m	119.9hi	120.2i	1028n	1103n	355.3h	326.0k
RDN *	143.0defghi	144.6efghij	131.2defg	132.7efgh	2656c	2750c	580.1e	557.9g
RDN + Zn **	148.5cdefg	149.7cdefgh	136.5cde	137.7cde	2575c	2675cd	642.8b	619.4bcd
75% RDN	133.3k	137.0kl	123.4gh	127.0h	2376ef	2428fg	497.0f	484.8i
75% RDN + Zn	135.4ijk	139.2ijkl	125.5fgh	129.1gh	2362efg	2413gh	505.0f	497.5hi
75% RDN + MC1	139.3hijk	142.6hijk	129.4efg	132.5efgh	2275gh	2333hij	554.7e	539.7gh
75% RDN + MC1 + Zn	147.2defgh	148.7defgh	137.3cde	138.6cde	2172i	2258j	628.9bcd	607.0bcdef
75% RDN + MC2	140.6ghijk	143.8ghijk	130.7defg	133.8defgh	2254hi	2312ij	560.8e	550.5g
75% RDN + MC2 + Zn	147.2defgh	148.5defgh	137.3cde	138.4cde	2170i	2260j	632.8bc	615.0bcde
Zero-tillage wheat (ZTW)								
Control	133.2k	133.3l	130.1efg	130.0fgh	786o	853o	436.3g	405.5j
RDN *	150.8abcd	152.1abcd	139.7abc	140.7bcd	2373ef	2466efg	666.1b	643.6bc
RDN + Zn **	157.3a	158.2a	146.1ab	146.8ab	2291fgh	2388ghi	737.2a	713.4a
75% RDN	141.5fghij	144.8defghij	132.3cdef	135.5cdefg	2082j	2139k	583.0de	571.3efg

Table 8. *Cont.*

Treatment	Gross Energy ($\times 10^3$ MJ ha^{-1})		Net Energy ($\times 10^3$ MJ ha^{-1})		Energy tonne^{-1} of Grain (MJ tonne^{-1})		Protein Yield (kg ha^{-1})	
	2013–2014	2014–2015	2013–2014	2014–2015	2013–2014	2014–2015	2013–2014	2014–2015
75% RDN + Zn	143.0defghi	146.4defghi	133.8cde	137.0cdef	2083j	2139k	589.2cde	582.5defg
75% RDN + MC1	148.2cdefg	151.1bcdefg	139.0bcd	141.7abc	1991k	2052l	642.6b	628.1bc
75% RDN + MC1 + Zn	155.6abc	156.7abc	146.3ab	147.2ab	1914kl	1998l	719.8a	697.7a
75% RDN + MC2	148.9bcdef	151.8abcde	139.7abc	142.4abc	1980kl	2042l	647.0b	637.3bc
75% RDN + MC2 + Zn	156.6ab	157.6ab	147.3a	148.1ab	1900l	1986l	730.8a	712.8a
LSD (*p*= 0.05)	4.05	3.58	4.05	3.58	48.4	45.4	47.6	45.5
Nutrient management	*	*	*	*	*	*	*	*
Interaction	*	*	*	*	*	*	*	*

Within a column, means followed by the same letter are not significantly different at the 0.05 level of probability by the Duncan's multiple range test. ″*″: Indicates significant different of treatments the 0.05 level of probability by the Duncan's multiple range test; RDN *: Recommended dose of nutrients 120 kg N ha^{-1} and 25.8 kg P ha^{-1}; Zn **: Soil applied with 5 kg Zn ha^{-1} through zinc sulphate heptahydrate; MC1: (*Anabaena* sp. (CR1) + *Providencia* sp. (PR3) consortia; MC2: *Anabaena-Pseudomonas* biofilmed formulations.

Table 9. Effect of nutrient management options on energetic and protein yield of rice–wheat cropping system in different crop establishment methods.

Treatment	Gross Energy ($\times 10^3$ MJ ha^{-1})		Net Energy ($\times 10^3$ MJ ha^{-1})		Energy tonne^{-1} of Grain (MJ tonne^{-1})		Protein Yield (kg ha^{-1})	
	2013	2014	2013	2014	2013	2014	2013–2014	2014–2015
PTR–CDW								
Control	249.5k	247.2g	237.0j	233.6l	1847k	2092o	501.7l	456.7m
RDN *	298.2cdef	297.5c	269.8def	268.0fghi	3268a	3500a	841.5fgh	801.8ghi
RDN + Zn **	310.4a	310.7a	281.8abc	280.9abc	3164b	3360b	940.5abcd	899.9bcd
75% RDN	279.2j	282.1f	254.7h	256.5j	3035c	3209cd	714.0j	689.5k
75% RDN + Zn	284.9ij	286.6f	260.3gh	260.8ij	2986c	3190cd	723.6ij	700.7jk
75% RDN + MC1	294.2fgh	295.1cd	269.7def	269.5efgh	2856de	3063ef	818.7ghi	781.3hi
75% RDN + MC1 + Zn	306.5abc	306.6a	281.8abc	280.8abc	2750efg	2945ghi	916.7cde	879.6def
75% RDN + MC2	295.4fg	296.5c	270.9def	270.9defgh	2844def	3043efg	820.3ghi	791.0ghi
75% RDN + MC2 + Zn	306.7abc	306.5a	282.0abc	280.7abc	2753efg	2952ghi	912.4cde	878.1def
SRI–SWI								
Control	255.2k	252.9g	244.9ij	241.5k	1433l	1662p	527.9l	481.5m
RDN *	297.9cdef	297.3c	271.7def	269.9defgh	3027c	3254c	839.9fhg	799.7ghi
RDN + Zn **	304.9abcde	305.1ab	278.5bcd	277.6bcd	2938cd	3131de	933.6bcd	892.5cde
75% RDN	278.6j	281.4f	256.3h	258.1j	2775efg	2947ghi	705.8j	681.2k
75% RDN + Zn	285.4hij	287.0ef	263.0fgh	263.4hij	2721g	2920hi	727.1ij	703.6jk
75% RDN + MC1	293.1fghi	294.0cde	270.8def	270.5defgh	2617h	2820j	810.3ghi	772.7hi
75% RDN + MC1 + Zn	305.5cdef	305.5ab	283.0abc	281.9ab	2522h	2714kl	905.4de	864.3def
75% RDN + MC2	294.5fg	295.6c	272.2def	272.2defg	2607h	2803jk	813.8ghi	784.1hi
75% RDN + MC2 + Zn	306.2abcd	305.9ab	283.7abc	282.3ab	2514hi	2708kl	919.7cde	884.5def

Table 9. *Cont.*

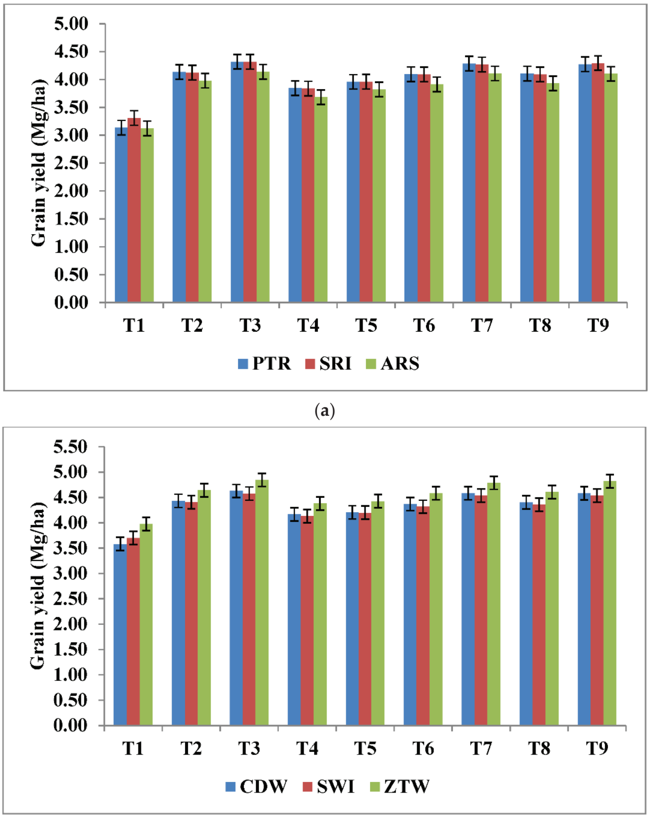
Treatment	Gross Energy ($\times 10^3$ MJ ha^{-1})		Net Energy ($\times 10^3$ MJ ha^{-1})		Energy tonne^{-1} of Grain (MJ tonne^{-1})		Protein Yield (kg ha^{-1})	
	2013	**2014**	**2013**	**2014**	**2013**	**2014**	**2013–2014**	**2014–2015**
ARS–ZTW								
Control	255.2k	252.6g	246.5i	243.1k	1207m	1376q	590.4k	544.1l
RDN *	300.1bcdef	299.0bc	275.4cde	273.5cdefg	2821efg	3005fgh	898.9def	859.7defg
RDN + Zn **	311.9a	311.7a	287.0ab	286.0a	2740fg	2894j	999.1a	958.9a
75% RDN	280.6j	283.1f	259.9gh	261.6ij	2556h	2687l	776.0i	752.5ij
75% RDN + Zn	286.9ghij	288.1def	265.9efg	266.4ghi	2509hi	2664m	793.4h	771.6hi
75% RDN + MC1	296.0def	296.5c	275.2cde	274.9bcdef	2413ij	2572n	861.1efg	826.2fgh
75% RDN + MC1 + Zn	308.0ab	307.6a	287.0ab	285.9a	2328j	2478n	970.9abc	941.0abc
75% RDN + MC2	296.9def	297.6c	276.2cd	276.1bcde	2404j	2557n	867.9efg	840.5efg
75% RDN + MC2 + Zn	309.2ab	308.7a	288.3a	286.9a	2320j	2473n	985.3ab	951.4ab
LSD (p = 0.05)	4.77	3.93	4.77	3.93	58.6	54.2	61.3	55.7
Nutrient management	*	*	*	*	*	*	*	*
Interaction	*	*	*	*	*	*	*	*

Within a column, means followed by the same letter are not significantly different at the 0.05 level of probability by the Duncan's multiple range test. "*": Indicates significant different of treatments the 0.05 level of probability by the Duncan's multiple range test; RDN *: Recommended dose of nutrients 120 kg N ha^{-1} and 25.8 kg P ha^{-1}; Zn **: Soil applied with 5 kg Zn ha^{-1} through zinc sulphate heptahydrate; MC1: (*Anabaena* sp. (CR1) + *Providencia* sp. (PR3) consortia; MC2: *Anabaena-Pseudomonas* biofilmed formulations.

3.3. Grain Yield, Protein Yield and Micronutrient Uptake

The grain yield was significantly affected at the individual crop level, while at system level it remained on par (Figure 3). Application of RDN + Zn recorded the highest yield in both crops, while the yield in 75% RDN + MC1 + Zn and 75% RDN + MC2 + Zn remained on par with RDN + Zn. The protein yield in wheat was higher than rice and this amount is 303 to 318 kg ha^{-1} in the first year and 304 to 315 kg ha^{-1} during the second year (Tables 6 and 7). The system protein yield varied between 456.7 and 999.1 kg ha^{-1}, respectively, with the highest and lowest in RDN + Zn in ARS–ZTW and control in CDW–PTR, respectively (Table 8). In both rice and wheat, CEMs differed significantly in protein production with the highest protein in SRI in rice and ZTW in wheat. The increase in protein yield in PTR over ARS was 25.2 to 23.2 kg ha^{-1}, while the same for ZTW over CDW and SWI was 86.2 to 88.3 kg ha^{-1}. The order of significance for the variation in system protein yield was RDN > microbial consortia > Zn fertilization > CEMs, while their contribution to protein yield was 112.7–326.3 kg ha^{-1}; 85.7–102.1 kg ha^{-1}, 16.1–105.1 kg ha^{-1} and 62–65 kg ha^{-1}, respectively.

The uptake of all studied micronutrients was affected significantly due to CEMs in both rice (white rice kernel) and wheat (whole grain) (Tables 10–12). In rice, PTR and SRI remained on par with each other and were found statistically superior to ARS for all micronutrients. In wheat, ZTW recorded significantly higher micronutrient uptake than both CDW and SWI. Among nutrient management treatments, the uptake of Zn and Fe was significantly affected due to all treatment variables, while for Mn and Cu, the uptake remained on par in all treatments except control. The highest uptake of Zn, Fe, Mn and Cu in rice was 45.42 g ha^{-1}, 235.0 g ha^{-1}, 24.78 g ha^{-1} and 19.66 g ha^{-1} (all in SRI), respectively. Similarly, for wheat it was 217.9 g ha^{-1} for Zn, 528.2 g ha^{-1} for Fe, 179.9 g ha^{-1} for Mn and 35.84 g ha^{-1} for Cu (all in ZTW), respectively.



(**a**)

(**b**)

Figure 3. Effect of crop establishment methods and nutrient management on grain yield of rice (**a**) and wheat (**b**) (pooled data over two years). (T1: Control, T2: RDN, T3: RDN + Zn, T4: 75% RDN, T5: 75% RDN + Zn, T6: 75% RDN + MC1, T7: 75% RDN + MC1 + Zn, T8: 75% RDN + MC2 and T9: 75% RDN + MC2 + Zn). RDN Recommended dose of nutrients 120 kg N ha^{-1} and 25.8 kg P ha^{-1}; Zn: Soil applied with 5 kg Zn ha^{-1} through zinc sulphate heptahydrate; MC1: (*Anabaena* sp. (CR1) + *Providencia* sp. (PR3) consortia; MC2: *Anabaena-Pseudomonas* biofilmed formulations.

Table 10. Effect of crop establishment methods on micronutrient uptake in rice and wheat.

Treatment	**Zn Uptake (g ha^{-1})**		**Fe Uptake (g ha^{-1})**		**Cu Uptake (g ha^{-1})**		**Mn Uptake (g ha^{-1})**	
	2013	**2014**	**2013**	**2014**	**2013**	**2014**	**2013**	**2014**
Rice								
Puddled transplanted rice (PTR)	38.3a	30.2a	206.0a	184.9a	17.0a	12.5a	21.4a	17.1a
System of rice intensification (SRI)	38.8a	30.3a	208.1a	187.8a	17.1a	12.8a	21.5a	17.2a
Aerobic rice system (ARS)	32.8b	25.8b	182.5b	165.3b	13.8b	10.1b	18.0a	15.1b
LSD (p = 0.05)	0.55	0.45	3.09	4.86	0.80	0.70	0.57	0.63
Wheat								
Conventional drill-sown wheat (CDW)	154.5b	138.1b	418.7b	400.6b	21.6b	18.5b	126.3b	115.2b
System of wheat intensification (SWI)	154.3b	138.7b	418.3b	401.2b	22.2b	18.9b	127.1b	116.3b
Zero-tillage wheat (ZTW)	183.6a	167.1a	475.8a	451.0a	31.1a	26.8a	155.1a	143.4a
LSD (p = 0.05)	1.96	1.82	8.93	7.74	1.88	1.67	5.07	4.90

Within a column, means followed by the same letter are not significantly different at the 0.05 level of probability by the Duncan's multiple range test.

Table 11. Effect of nutrient management options on micronutrient uptake in rice in different crop establishment methods.

Treatment	Zn Uptake (g ha^{-1})		Fe Uptake (g ha^{-1})		Cu Uptake (g ha^{-1})		Mn Uptake (g ha^{-1})	
	2013	**2014**	**2013**	**2014**	**2013**	**2014**	**2013–2014**	**2014–2015**
PTR–CDW								
Control	18.94k	15.15o	133.3kl	113.7n	5.70k	4.23j	9.18l	6.69j
RDN *	40.66cd	31.77ef	216.1cd	194.7def	18.35abc	13.49abcde	22.86abcde	18.29abcde
RDN + Zn **	42.22ab	35.96a	232.2a	210.5ab	19.89a	14.95a	25.03a	20.15a
75% RDN	34.22gh	27.05ij	189.6fgh	171.9hij	15.95efgh	12.04cdefg	20.26efghi	16.31efghi
75% RDN + Zn	35.67fg	27.72i	194.0fgh	175.0hi	16.19defgh	11.69efg	20.44defgh	16.21efghi
75% RDN + MC1	41.32c	32.61de	219.5bc	195.6def	18.97abc	13.91ab	23.51abcd	18.81abcd
75% RDN + MC1 + Zn	44.58ab	35.40ab	228.2ab	207.1abc	19.89a	14.57a	24.54ab	20.01ab
75% RDN + MC2	40.60c	32.20ef	217.0bcd	194.9def	18.49abc	13.56abcd	23.21abcd	18.44abcde
75% RDN + MC2 + Zn	43.55b	33.84cd	223.8abc	200.6cd	19.14ab	14.01ab	23.66abcd	18.98abcd
SRI–SWI								
Control	20.7k	16.44n	144.1k	124.5m	6.74k	5.16j	10.19l	7.45j
RDN *	41.11cd	31.85ef	219.4bc	199.1cd	18.26abcd	13.63abcd	22.95abcde	18.34abcde
RDN + Zn **	45.15ab	35.63a	231.9a	211.7a	19.66a	15.03a	24.78ab	19.93ab
75% RDN	33.45h	26.22jk	184.5hi	168.5ij	15.52ghi	11.95defg	19.69ghijk	15.84fghi
75% RDN + Zn	37.67e	29.31h	197.8efg	179.8gh	17.80abcdef	13.43abcde	22.08bcdefg	17.73bcdefg
75% RDN + MC1	41.03cd	32.09ef	218.8bc	196.3de	18.54abc	13.80abc	23.05abcd	18.40abcde
75% RDN + MC1 + Zn	43.92ab	34.15bc	224.1abc	201.8bcd	19.51a	14.27ab	24.06abcd	19.34abc
75% RDN + MC2	40.38c	31.72ef	217.1bcd	196.4de	18.04abcde	13.41abcde	22.76abcdef	18.02abcdef
75% RDN + MC2 + Zn	45.42a	35.00abc	235.0a	212.5a	19.54a	14.48a	24.28abc	19.43abc
ARS–ZTW								
Control	16.86l	13.54p	123.5l	106.1n	5.02k	3.65j	8.19l	6.49j
RDN *	35.76fg	27.84i	194.3fgh	176.1hi	15.72fgh	11.43fg	20.05fghij	16.79defgh
RDN + Zn **	39.41d	31.25efg	206.2de	188.0efg	16.86cdefg	12.56bcdef	21.67cdefg	18.23abcde
75% RDN	28.45j	22.73m	166.1j	151.5l	12.99j	9.64hi	16.99k	14.29i
75% RDN + Zn	31.01i	23.91lm	175.6ij	159.4kl	13.36j	9.39i	17.45jk	14.44i
75% RDN + MC1	32.44hi	25.24kl	183.6hi	164.2jk	13.58ij	9.49i	17.63ijk	14.55hi
75% RDN + MC1 + Zn	37.38ef	30.25gh	199.7ef	187.3fg	15.61fghi	11.36fgh	19.93ghij	17.36cdefg
75% RDN + MC2	33.61h	26.45ijk	187.2gh	168.9ij	14.48hij	10.33ghi	18.80hijk	15.52ghi
75% RDN + MC2 + Zn	39.74cd	30.84fg	206.0de	185.9g	17.25bcdefg	12.59bcdef	21.63cdefg	18.21abcde
LSD (*p* = 0.05)	1.84	1.45	11.4	8.7	2.13	1.83	2.73	2.32
Nutrient management treatments	*	*	*	*	*	*	*	*
Interactions	*	*	*	*	*	*	*	*

Within a column, means followed by the same letter are not significantly different at the 0.05 level of probability by the Duncan's multiple range test. "*": Indicates significant different of treatments at the 0.05 level of probability by the Duncan's multiple range test; RDN *: Recommended dose of nutrients 120 kg N ha^{-1} and 25.8 kg P ha^{-1}; Zn **: Soil applied with 5 kg Zn ha^{-1} through zinc sulphate heptahydrate; MC1: (*Anabaena* sp. (CR1) + *Providencia* sp. (PR3) consortia; MC2: *Anabaena-Pseudomonas* biofilmed formulations.

Table 12. Effect of nutrient management options on micronutrient uptake in wheat in different crop establishment methods.

Treatment	Zn Uptake (g ha^{-1})		Fe Uptake (g ha^{-1})		Cu Uptake (g ha^{-1})		Mn Uptake (g ha^{-1})	
	2013–2014	2014–2015	2013–2014	2014–2015	2013–2014	2014–2015	2013–2014	2014–2015
PTR–CDW								
Control	81.6k	70.5k	263.9k	249.0j	61.5j	55.4h	12.5j	10.4j
RDN *	163.0fgh	145.1g	442.3fg	422.4efgh	137.9fg	125.1cd	23.1fgh	19.9fgh
RDN + Zn **	186.1cde	163.4d	469.8de	446.1d	148.7ef	133.9cd	25.0def	21.5de
75% RDN	173.3i	127.8i	394.0b	381.4i	114.3h	106.8f	19.9hi	17.2gh
75% RDN + Zn	140.1i	128.9i	401.9h	386.7i	115.2h	107.3f	19.9hi	16.7hi
75% RDN + MC1	158.8gh	142.2gh	432.0g	416.8h	133.8g	122.8cd	22.7fgh	19.4fgh
75% RDN + MC1 + Zn	181.1e	160.2def	463.7ef	441.4de	144.4fg	130.1cd	24.3defgh	21.4def
75% RDN + MC2	160.1gh	143.7g	435.8g	420.7fgh	136.1g	123.7cd	22.9fgh	19.1fh
75% RDN + MC2 + Zn	182.8de	161.2de	465.1e	441.3de	145.3fg	131.6cd	24.2efgh	21.3defg
SRI–SWI								
Control	86.5k	75.6k	273.3j	258.9j	66.3j	60.4h	15.5i	13.0ij
RDN *	162.0fgh	145.1g	442.9fg	423.9efgh	137.9fg	125.6cd	23.0fgh	19.5fgh
RDN + Zn **	184.5de	162.8d	467.3e	444.7d	148.3ef	134.0c	25.0def	21.4def
75% RDN	136.8i	128.2i	394.7h	383.0i	114.8h	107.7f	20.2gh	17.4fgh
75% RDN + Zn	142.7i	132.3i	403.8h	389.5i	118.7h	111.1ef	22.6fgh	19.3fgh
75% RDN + MC1	156.9h	141.3gh	429.1g	414.8h	133.2g	122.5de	22.4fgh	18.9fgh
75% RDN + MC1 + Zn	179.9e	160.1def	458.8e	437.7defg	144.5fg	130.6cd	24.7defg	21.5ef
75% RDN + MC2	158.3gh	142.9g	433.6g	419.5gh	135.4g	123.4cd	22.5fgh	18.5fgh
75% RDN + MC2 + Zn	180.8e	160.3def	461.6ef	438.8def	144.5fg	131.3cd	23.7fgh	20.5efgh
ARS–ZTW								
Control	111.6j	99.8j	320.8i	299.8i	90.5i	83.4g	23.2fgh	19.7fhg
RDN *	192.6c	174.6c	501.3bc	474.5bc	167.5bcd	154.1ab	33.4ab	28.9ab
RDN + Zn **	217.9a	194.8a	528.2a	497.2a	179.9a	164.3a	35.7a	30.9a
75% RDN	164.4fg	155.4f	450.5efg	431.8defgh	141.4fg	133.7cd	28.9bcd	25.1bcd
75% RDN + Zn	167.3f	156.4ef	458.8f	437.2defg	142.2fg	134.0c	28.6cde	24.3cde
75% RDN + MC1	185.2de	163.7d	490.3cd	468.5c	160.2cde	148.7b	29.8bc	25.3bcd
75% RDN + MC1 + Zn	209.9b	188.9b	515.8ab	486.6abc	172.8abc	157.8ab	32.9abc	28.8ab
75% RDN + MC2	187.9cd	171.6c	493.4c	471.8c	164.0cd	151.1b	31.6abc	26.6bc
75% RDN + MC2 + Zn	216.1ab	194.1ab	522.9ab	491.8ab	177.8ab	163.3a	35.8a	31.6a
Nutrient management treatments	*	*	*	*	*	*	*	*
Interactions	*	*	*	*	*	*	*	*

Within a column, means followed by the same letter are not significantly different at the 0.05 level of probability by the Duncan's multiple range test. "*": Indicates significant different of treatments at the 0.05 level of probability by the Duncan's multiple range test; RDN *: Recommended dose of nutrients 120 kg N ha^{-1} and 25.8 kg P ha^{-1}; Zn **: Soil applied with 5 kg Zn ha^{-1} through zinc sulphate heptahydrate; MC1: (*Anabaena* sp. (CR1) + *Providencia* sp. (PR3) consortia; MC2: *Anabaena-Pseudomonas* biofilmed formulations.

4. Discussion

4.1. Energy Input and Type of Energy

The study of energy input is important in rice and wheat at the individual crop level as well as system level due to significant variations in cultivation practices which include CEMs, nutrient management and soil hydrological regimes across a region. The faster adoption of CEMs such as ZTW [41], which is reported to reduce the energy expenditure on tillage, promotion of consortia-based microbial inoculations for nutrient endowments in crops [42,43], thereby reducing the total nutrient applied and increasing the use of micronutrients due to crop response [44,45], was evaluated for biological parameters and economic scale, while their evaluation in terms of energy requirement carries significant importance considering their share in total energy consumption in the crop production process (Figure 1).

In our study, CEMs, rate of N and P application, Zn fertilization and microbial inoculation significantly affected the energetics of RWCS. The higher energy requirement in rice than wheat was contributed by the field preparation, nursery and higher number of irrigations [5,17]. The variation in energy inputs across CEMs in rice was governed by nursery, puddling, seed and sowing and number of irrigations, while in wheat tillage, seed rate and weeding operation contributed to the variation in energy input, with highest contribution coming from tillage. The highest share of fertilization to total energy consumption [18,46] was due to the energy equivalent for N (60.6 MJ kg^{-1}), P_2O_5 (11.1 MJ kg^{-1}) and K_2O (6.7 MJ kg^{-1}) and the higher quantity (90–120 kg N, 44.67–59.1 kg P_2O_5 and 60 kg K_2O) applied, while higher energy equivalents for tractor-operated machinery and diesel increased the share of field preparation in total energy input. As the share of fertilizer in energy consumption is higher in wheat, the increase in energy efficiency by using microbial consortia will be more profitable for wheat. The variation in energy requirement due to irrigation was contributed by rice alone as the irrigation requirement of all CEMs in wheat remained the same. The saving in energy by changing CEM from PTR to ARS was 563.7 MJ ha^{-1} (2%). At the same time, this contribution was less if calculated based on monetary terms at the farmer field level which might be due to the subsidized rate of electricity and very low irrigation charges. At the system level, the share of irrigation in total energy input remained the same (6%) even though the difference in energy consumption in irrigation among CEMs is 326.6 MJ ha^{-1}. The reduction in energy requirement by changing CEM was reported by [17,47].

The ARS and ZTW were found to be better as they use higher renewable energy than PTR and CDW. The use of higher seed rate and absence of puddling and tillage in ARS and ZTW were the important reasons for higher renewable energy consumption. At the same time, total energy input was also lower in ARS and ZTW which makes them energy-efficient. Both methods were also recommended on the issue of water shortage [48,49] and timely planting along with energy efficiency [7]. Among nutrient management treatments, the use of microbial inoculations reduces the share of non-renewable energy; therefore, treatment with 75% RDN + MC1 or MC2 increases the share of renewable energy in crop production.

The variation in gross energy production arose due to yield superiority of PTR [50] and SRI [51] over ARS in rice and ZTW [52] over CDW and SWI in wheat. The higher gross energy than ARS and lower energy input than PTR make SRI significantly superior in net energy production. The variation across CEMs in energy input and net energy production [53,54] was also reported. We found that in rice and wheat, the variations in energy input and gross energy production contribute equally towards the variation in net energy production among CEMs, while at the system level, the variation in input has the highest contribution to the increase in net energy production.

4.2. Energy Production

Among the nutrient management options, gross and net energy output was affected significantly by the rate of N and P application, Zn fertilization and microbial inoculation. The rate of N and P application had the highest contribution to variation in energy production, while Zn fertilization had the lowest contribution to energy production. The highest gross energy in RDN + Zn was the outcome of highest yield, while the highest net energy production in 75% RDN + Zn + MC1 or MC2 was due to reduction in cost of cultivation on 25% of N and P fertilizer. The difference in energy input across CEMs had a higher contribution to the variation in net energy production than gross energy production. The variation in energy input across CEM was 6.53 × 10^3 to 15.47 × 10^3 MJ ha^{-1} for rice, 3.95 to 12.19 MJ ha^{-1} for wheat and 10.49 to 27.66 MJ ha^{-1} for RWCS, while variation in gross energy production was 125.0–157.1, 125.7–153.1 and 250.9–309.2 MJ ha^{-1} for rice, wheat and RWCS, respectively.

The nutrient application through chemical fertilizers is the single most important source of nutrients. Its importance has increased over the years due to increasing nutrient deficiency [55,56], response to fertilization and use of high-yielding nutrient responsive varieties. In terms of energy, fertilizer contributes 59–64% to total energy input in RWCS and the cost of chemical fertilizer is also going to increase in future on account of the increasing cost of fertilizer production, depletion in natural reserves and increasing demand. The rice and wheat together contribute 61% (17.67 million tonnes) to total fertilizer consumption in India. Considering this, complimentary options such as use of microbial inoculations with partial replacement of chemical fertilizers will help in making the RWCS more energy-efficient.

4.3. Grain Yield, Protein Yield and Micronutrient Uptake

The calculations of nutritional status of staple crops are essential considering the shifting of focus of India from food security to nutritional security [57,58]. Protein energy malnutrition (PEM) ranks first among the major nutrition-related disorders in India [21]. As both rice and wheat are the staple crops catering the protein need of the majority of the population (especially BLP where PEM is a severe problem), the calculation of their protein yield will be more focused than just the calculation of yield. In our experiment, the variation in protein yield was accounted due to the variation in grain yield of rice and wheat and the factor used for calculation converting nitrogen content to protein. The yield variations in rice recorded due to better crop establishments leading to superior growth and yield attributes due to transplanting in both PTR and SRI and less weed menace due to puddling than ARS. The variation in yields response by different CEMs was reported by [59,60], while variation in weed dynamics across CEMs [61] and weed problem in aerobic rice [62] was also reported. This significantly higher yield variation across CEMs nullified the effect of factor used for calculation of protein yield which is higher in rice (5.95) than wheat (5.70).

Another health-related risk is micronutrient deficiency also called as hidden hunger [63]. The need and significant of micronutrient application for enhancing yield [20] as well as increasing grain micronutrient concentration and uptake was reported [64], while their uptake variation across the CEMs with use of different microbial inoculations is meagre and studied in this investigation. The uptake of all studied micronutrients was higher in wheat. Along with uptake, concentration dilution by dry matter production and presence of anti-nutritional factors (phytate) [65] are the other factors deciding the nutritional status of food grains. The higher micronutrient uptake in PTR and SRI signifies the role of puddling in enhancing the uptake of micronutrients [66], while significantly higher micronutrient uptake in ZTW is the indication of the superior performance of ZTW arose due to residual effect of previous season rice (ARS) and better root growth leading higher forage area due to less physical constraints for root growth (non-puddled ARS). The uptake of Zn and Fe in both rice and wheat was significantly affected by application of microbial inoculations, RDN and Zn fertilization. This indicates ability of above-mentioned factors in amending the micronutrient uptake in rice. The variation in micronutrient uptake across the CEMs was explained by changes in hydrological regimes across CEMs in rice and residual effect as well as soil physical constraints in wheat.

5. Conclusions

The crop establishment methods (CEMs) differ significantly in energy input and output along with protein and micronutrient uptake in both years of study. The gross and net energy production was highest in ARS–ZTW which was 293.9 × 10^3 MJ ha^{-1} and 273.5–267.6 × 10^3 MJ ha^{-1}, respectively. The protein yield increase in ARS–ZTW was 61.5–62 kg ha^{-1} in the first year and 86.2–88.3 kg ha^{-1} in the second year over other CEMs, respectively,, while it reduced the energy required for the production of one tonne of system yield by 206 and 250 MJ $tonne^{-1}$ over PTR–CDW in the first and second year, while the same for SRI–SWI was 467 and 517 MJ $tonne^{-1}$, respectively, for the first and second year. The application of 75% RDN with microbial consortia and Zn showed promise in enhancing net and gross energy production over all other combinations. This signifies their role of microbial consortia in energy efficiency and nutrient security of RWCS. The future research may focus on evaluation and standardization of microbial consortia in other crops and cropping systems under diverse ecologies. Furthermore, understanding the physiological and biochemical processes or mechanisms which are affected by the microbial consortia in rice and wheat can be an innovative line of research work. Besides this, the energy inputs and output and energy efficiency need to be studied for the increased level of mechanization in crop production as the lack of labour availability and higher wage rate in the future will increase mechanization in crop production.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/su14105986/s1. Table S1: Mean weekly meteorological data during the rice-growing season in 2013 and 2014; Table S2: Mean weekly meteorological data during the wheat growing season in 2013–14 and 2014–15.

Author Contributions: Conceptualization, Y.S.S. and A.A.S.; methodology, A.A.S., Y.S.S. and R.P.; software, R.S.B. and A.A.S.; validation, A.A.S., Y.S.S. and D.K.; formal analysis, Y.S.S., A.A.S. and R.S.B.; investigation, Y.S.S. and A.A.S.; resources, Y.S.S., D.K., R.P. and A.A.S.; data curation, R.S.B.; writing—original draft preparation, A.A.S.; writing—review and editing, Y.S.S., R.P. and R.S.B.; visualization, A.A.S. and Y.S.S.; supervision, Y.S.S.; project administration, Y.S.S.; funding acquisition, Y.S.S. and A.A.S. All authors have read and agreed to the published version of the manuscript.

Funding: The research was part of PhD research work conducted at Indian Council of Agricultural Research (ICAR)-Indian Agricultural Research Institute (IARI), New Delhi (India). This research received no external funding, except the financial support as Senior Research Fellowship (SRF) from ICAR for living expenses and facilities from ICAR-Indian Agricultural Research Institute.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors duly acknowledge ICAR-Indian Agricultural Research Institute, New Delhi (India) for providing financial support as Senior Research Fellowship. Our sincere thanks are also to Head and Professor, Division of Agronomy, ICAR-Indian Agricultural Research Institute, New Delhi, India for providing facilities required for the field experiment.

Conflicts of Interest: The authors declare no competing interest.

References

1. Rampal, P. An analysis of protein consumption in India through plant and animal sources. *Food Nutr. Bull.* **2016**, *39*, 564–580. [CrossRef] [PubMed]
2. Sautter, C.; Poletti, S.; Zhang, P.; Gruissem, W. Biofortification of essential nutritional compounds and trace elements in rice and cassava. *Proc. Nutr. Soc.* **2006**, *65*, 153–159. [CrossRef] [PubMed]
3. Annual Report 2020–21, Department of Agriculture, Cooperation & Farmers' Welfare, Ministry of Agriculture & Farmers' Welfare, Government of India, Krishi Bhawan, New Delhi. 2022. Available online: https://agricoop.nic.in/sites/default/files/Web%20copy%20of%20AR%20%28Eng%29_7.pdf (accessed on 26 January 2022).
4. Agricultural Statistics at a Glance 2020, Department of Agriculture, Co-Operation and Farmers' Welfare, Ministry of Agriculture and Farmers' Welfare, Government of India. 2020, p. 147. Available online: eands.dacnet.nic.in/ (accessed on 4 February 2022).
5. Bohra, J.S.; Kumar, R. Effect of crop establishment methods on productivity, profitability and energetic of rice (*Oryza sativa*)-wheat (*Triticum aestivum*) system. *Indian J. Agric. Sci.* **2015**, *85*, 217–223.
6. Gathala, M.K.; Laing, A.M.; Tiwari, T.P.; Timsina, J.; Islam, S.; Bhattacharya, P.M.; Dhar, T.; Ghosh, A.; Sinha, A.K.; Chowdhury, A.K.; et al. Energy efficient, sustainable crop production practices benefit for small holder farmers and the environment across three countries in the eastern Gangetic plains, South Asia. *J. Clean. Prod.* **2020**, *246*, 118982. [CrossRef]
7. Erenstein, O.U.; Farooq, R.K.; Malik, M.; Sharif. On-farm impacts of zero tillage wheat in South Asia's rice-wheat systems. *Field Crops Res.* **2008**, *105*, 240–252. [CrossRef]
8. Prasad, R.; Shivay, Y.S.; Kumar, D. *Textbook of Field Crops Production—Foodgrain Crops*; Prasad, R., Ed.; Directorate of Knowledge Management in Agriculture, Indian Council of Agricultural Research, Krishi Anuasandhan Bhavan I: Pusa, New Delhi, India, 2012; Volume I, pp. 1–65.

9. Thakur, A.K.; Rath, S.; Mandal, K.G. Differential responses of system of rice intensification and conventional flooded rice management methods to applications of nitrogen fertilizer. *Plant Soil* **2013**, *370*, 59–71. [CrossRef]
10. Gathala, M.K.; Ladha, J.K.; Kumar, V.; Saharawat, Y.S.; Kumar, V.; Sharma, P.K.; Sharma, S.; Pathak, H. Tillage and crop establishment affects sustainability of south Asian rice-wheat system. *Agron. J.* **2011**, *103*, 961–971. [CrossRef]
11. Jat, A.L.; Srivastava, V.K.; Singh, R.K. Effect of crop establishment methods and integrated nutrient management on productivity of hybrid rice (*Oryza sativa*) and wheat (*Triticum aestivum*) cropping system. *Indian J. Agron.* **2015**, *60*, 341–346.
12. Erenstein, O.; Laxmi, V. Zero tillage impacts in India's rice-wheat system: A review. *Soil Tillage Res.* **2008**, *100*, 1–14. [CrossRef]
13. Bruckner, T.; Bashmakov, I.A.; Mulugetta, Y.; Chum, H.; de la Vega Navarro, A.; Edmonds, J.; Faaij, A.; Fungtammasan, B.; Garg, A.; Hertwich, E.; et al. Energy Systems. In *Climate Change 2014: Mitigation of Climate Change. Contribution of Working Group III to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change*; Edenhofer, O., Pichs-Madruga, R., Sokona, Y., Farahani, E., Kadner, S., Seyboth, K., Adler, A., Baum, I., Brunner, S., Eickemeier, P., Eds.; Cambridge University Press: Cambridge, UK; New York, NY, USA, 2014; Available online: https://www.ipcc.ch/site/assets/uploads/2018/02/ipcc_wg3_ar5_chapter7.pdf (accessed on 4 February 2022).
14. Gangwar, K.S.; Chaudhary, V.P.; Gangwar, B.; Pandey, D.K.; Singh, K.K. Mechanization practices in rice-wheat cropping systems in upper Indo-Gangetic plains India. *Agric. Mech. Asia Afr. Lat. Am.* **2012**, *43*, 66–74.
15. Borah, N.; Dutta, S.; Barua, P. Soil resource management under diversified rice-based cropping systems and increasing farm mechanization in Assam. *Proceedings* **2017**, 225–232.
16. Devasenapathy, P.; Senthilkumar, G.; Shanmugam, P.M. Energy management in crop production. *Indian J. Agron.* **2009**, *54*, 80–90.
17. Kumar, V.; Saharawat, Y.S.; Gathala, M.K.; Jat, A.S.; Singh, S.K.; Chaudhary, N.; Jat, M.L. Effect of different tillage and seeding methods on energy use efficiency and productivity of wheat in the Indo-Gangetic Plains. *Field Crop Res.* **2013**, *142*, 1–8. [CrossRef]
18. Singh, R.J.; Ghosh, B.N.; Sharma, N.K.; Patra, S.; Dadhwal, K.S.; Mishra, P.K. Energy budgeting and energy synthesis of rainfed maize-wheat rotation system with different soil amendment applications. *Ecol. Indic.* **2016**, *61*, 753–765. [CrossRef]
19. Cakmak, I. Enrichment of cereal grains with zinc: Agronomic or genetic biofortification? *Plant Soil* **2008**, *302*, 1–17. [CrossRef]
20. Prasad, R.; Shivay, Y.S.; Kumar, D. Agronomic biofortification of cereals grains with iron and zinc. In *Advances in Agronomy*; Elsevier Inc.: Amsterdam, The Netherlands, 2014; pp. 55–91.
21. Bhutia, D.T. Protein energy malnutrition in India: The plight of our under five children. *J. Fam. Med. Fam. Care* **2014**, *3*, 63–67. [CrossRef]
22. Alsulami, S.; Bodhini, D.; Sudha, V.; Shanthi Rani, C.S.; Pradeepa, R.; Anjana, R.M.; Radha, V.; Lovegrove, J.A.; Gayathri, R.; Mohan, V.; et al. Lower dietary intake of plant protein is associated with genetic risk of diabetes-related traits in urban Asian Indian adults. *Nutrients* **2021**, *13*, 3064. [CrossRef]
23. Bouyoucos, G. Hydrometer method for making particle size analysis of soils. *J. Agron.* **1962**, *54*, 464–465. [CrossRef]
24. Walkley, A.J.; Black, I.K. An examination of the Degtjareff method for determination of soil organic matter and a proposed modification of the chromic acid titration method. *Soil Sci.* **1934**, *37*, 29–38. [CrossRef]
25. Subbiah, B.V.; Asija, G.L. A rapid procedure for assessment of available nitrogen in rice soils. *Curr. Sci.* **1956**, *25*, 259–260.
26. Jackson, M.L. *Soil Chemical Analysis*; Prentice Hall of India Pvt Ltd.: New Delhi, India, 1973.
27. Hanway, J.J.; Heidel, H. *Soil Analysis Methods as Used in IOWA State College Soil Testing Laboratory*; Iowa State College of Agriculture: Ames, IA, USA, 1952; Volume 57, p. 131.
28. Lindsay, W.L.; Norvell, W.A. Development of DTPA soil test for zinc, iron, manganese and copper. *Soil Sci. Soc. Am. J.* **1978**, *42*, 421–428. [CrossRef]
29. Piper, C.S. *Soil and Plant Analysis*; The University of Adelaide: Adelaide, SA, Australia, 1950; p. 286.
30. Uphoff, N.; Randriamiharisoa, R. Reducing water use in irrigated rice production with the Madagascar System of Rice Intensification (SRI). In *Water-Wise Rice Production, Proceedings of the a Thematic Workshop on Water-Wise Rice Production, IRRI Headquarters, Los Banos, Philippines, 8–11 April 2002*; Bouman, B.A.M., Hengsdijk, H., Hardy, B., Bindraban, P.S., Toung, T.P., Ladha, J.K., Eds.; International Rice Research Institute: Los Banos, Philippines, 2003.
31. Dobermann, A. A critical assessment of the system of rice intensification. *Agric. Syst.* **2004**, *79*, 261–281. [CrossRef]
32. Uphoff, N.; Kassam, A.; Thakur, A. Challenges of increasing water saving and water productivity in the rice sector: Introduction to the system of rice intensification (SRI) and this issue. *Taiwan Water Conserv.* **2013**, *61*, 1–13.
33. Prasad, R. Aerobic rice systems. *Adv. Agron.* **2011**, *111*, 208–233.
34. Dhar, S.; Barah, B.C.; Vyas, A.K.; Uphoff, N.T. Comparing of system of wheat intensification (SWI) with standard recommended practices in north western plain zone of India. *Arch. Agron. Soil Sci.* **2016**, *62*, 994–1006. [CrossRef]
35. Rana, L.; Banerjee, H.; Ray, K.; Sarkar, S. System of wheat intensification- A new approach for enhancing wheat yield in small farming system. *J. Appl. Natur. Sci.* **2017**, *9*, 1453–1464. [CrossRef]
36. Adhikari, P.; Araya, H.; Aruna, G.; Balamatti, A.; Banerjee, S.; Baskaran, P.; Barah, B.C.; Behera, D.; Berhe, T.; Boruah, P.; et al. System of crop intensification for more productive, resource-conserving, climate-resilient, and sustainable agriculture: Experience with diverse crops in varying agroecologies. *Int. J. Agric. Sustain.* **2018**, *16*, 1–28. [CrossRef]
37. Prasanna, R.; Joshi, M.; Rana, A.; Shivay, Y.S.; Nain, L. Influence of co-inoculation of bacteria-cyanobacteria on crop yield and C–N sequestration in soil under rice crop. *World J. Microbiol. Biotech.* **2012**, *28*, 1223–1235. [CrossRef]
38. Nain, L.; Rana, A.; Joshi, M.; Jadhav, S.D.; Kumar, D.; Shivay, Y.S.; Paul, S.; Prasanna, R. Evaluation of synergistic effects of bacterial and cyanobacterial strains as bio-fertilizers for wheat. *Plant Soil* **2010**, *331*, 217–230. [CrossRef]

39. Piper, C.S. *Soil and Plant Analysis*; Hans Publishers: Bombay, India, 1966; pp. 197–201.
40. Prasad, R.; Shivay, Y.S.; Kumar, D.; Sharma, S.N. *Learning by Doing Exercises in Soil Fertility (A Practical Manual for Soil Fertility)*; Division of Agronomy, Indian Agricultural Research Institute: New Delhi, India, 2006; p. 68.
41. Keil, A.; D'souza, A.; McDonald, A. Zero-tillage is a proven technology for sustainable wheat intensification in the Eastern Indo-Gangetic Plains: What determines farmer awareness and adoption? *Food Secur.* **2017**, *9*, 723–743. [CrossRef]
42. Jha, M.; Chourasia, S.; Sinha, S. Microbial consortia for sustainable rice production. *Agroecol. Sustain. Food Syst.* **2013**, *37*, 340–362. [CrossRef]
43. Ghoghari, N.; Bharwad, K.; Champaneria, A.; Rajkumar, S. Microbial consortia for augmentation of plant growth–Revisiting the promising approach towards sustainable agriculture. In *New and Future Developments in Microbial Biotechnology and Bioengineering*; Elsevier publication: Amsterdam, The Netherlands, 2022; pp. 231–256. [CrossRef]
44. Shivay, Y.S.; Kumar, D.; Prasad, R. Effect of zinc enriched urea on productivity, zinc uptake and efficiency of an aromatic rice-wheat cropping system. *Nutr. Cycl. Agroecosyst.* **2008**, *81*, 229–243. [CrossRef]
45. Shivay, Y.S.; Prasad, R.; Rahal, A. Relative efficiency of zinc oxide and zinc sulphate-enriched urea for spring wheat. *Nutr. Cycl. Agroecosyst.* **2008**, *82*, 259–264. [CrossRef]
46. Biswas, B.; Ghosh, D.C.; Dasgupta, M.K.; Trivedi, N.; Timsina, J.; Dobermann, A. Integrated assessment of cropping systems in the Eastern Indo-Gangetic plain. *Field Crop Res.* **2006**, *99*, 35–47. [CrossRef]
47. Parihar, C.M.; Jat, S.L.; Singh, A.K.; Majumdar, K.; Jat, M.L.; Saharawat, Y.S.; Pradhan, S.; Kuri, B.E. Bio-energy, water-use efficiency and economics of maize-wheat-mungbean system under precision-conservation agriculture in semi-arid agro-ecosystem. *Energy* **2017**, *119*, 245–256. [CrossRef]
48. Govindarasu, R.; Paramasivam, K.; Nadaradjan, S.; Shashidhara, N.; Vengatesh, M. Aerobic rice: A production system for water scarceness. *AE Int. J. Sci. Tech.* **2015**, *3*, 1–4.
49. Peng, Z.; Wang, L.; Xie, J.; Li, L.; Coulter, J.A.; Zhang, R.; Luo, Z.; Kholova, J.; Choudhary, S. Conservation tillage increases water use efficiency of spring wheat by optimizing water transfer in a semi-arid environment. *Agronomy* **2019**, *9*, 583. [CrossRef]
50. Xu, L.; Li, X.; Wang, X.; Xiong, D.; Wang, F. Comparing the grain yields of direct-seeded and transplanted rice: A meta-analysis. *Agronomy* **2019**, *9*, 767. [CrossRef]
51. Thakur, A.K.; Mohanty, R.K.; Patil, D.U.; Kumar, A. Impact of water management on yield and water productivity with system of rice intensification (SRI) and conventional transplanting system in rice. *Paddy Water Environ.* **2014**, *12*, 413–424. [CrossRef]
52. Keil, A.; D'souza, A.; McDonald, A. Zero-tillage as a pathway for sustainable wheat intensification in the Eastern Indo-Gangetic Plains: Does it work in farmers' fields? *Food Secur.* **2015**, *7*, 983–1001. [CrossRef]
53. Saharawat, Y.S.; Singh, B.; Malik, R.K.; Ladha, J.K.; Gathala, M.; Jat, M.L.; Kumar, V. Evaluation of alternative tillage and crop establishment methods in a rice-wheat rotation in North Western IGP. *Field Crop Res.* **2010**, *116*, 260–267. [CrossRef]
54. Nandan, R.; Poonia, S.P.; Singh, S.S.; Nath, C.P.; Kumar, V.; Malik, R.K.; McDonald, A.; Hazra, K.K. Potential of conservation agriculture modules for energy conservation and sustainability of rice-based production systems of Indo-Gangetic Plain region. *Environ. Sci. Pollut. Res.* **2021**, *28*, 246–261. [CrossRef] [PubMed]
55. Rattan, R.K.; Kumar, M.; Narwal, R.P.; Singh, A.P. Soil health and nutritional security—Micronutrients. In *Soil Science in the Service of Nation, Proceedings of the Platinum Jubilee Symposium, New Delhi, India, 22–25 December 2009*; Goswami, N.N., Bijay-Singh, Pal, D.K., Rao, D.L.N., Rattan, R.K., Eds.; Indian Society of Soil Science: New Delhi, India, 2009; pp. 249–265.
56. Rattan, R.K. *Soil—A Precious Resource*; ISSS Newsletter No. 33–35; Indian Society of Soil Science: New Delhi, India, 2013.
57. Yadava, D.K.; Choudhury, P.R.; Firoz, H.; Kumar, D. *Biofortified Varieties: Sustainable Way to Alleviate Malnutrition*; Indian Council of Agricultural Research: New Delhi, India, 2017; p. 19.
58. Yadava, D.K.; Choudhury, P.R.; Hossain, F.; Kumar, D.; Mohapatra, T. *Biofortified Varieties: Sustainable Way to Alleviate Malnutrition*, 3rd ed.; Indian Council of Agricultural Research: New Delhi, India, 2020; p. 86.
59. Ishfaq, M.; Akbar, N.; Anjum, S.A.; Anwar-Ijl-Haq, M. Growth, yield and water productivity of dry direct seeded rice and transplanted aromatic rice under different irrigation management regimes. *J. Integr. Agric.* **2020**, *19*, 2656–2673. [CrossRef]
60. Singh, M.; Kumar, M.; Kumar, V.; Solanki, I.S.; McDonald, A.J.; Kumar, A.; Poonia, S.P.; Kumar, V.; Ajay, A.; Kumar, A.; et al. Inter-comparison of crop establishment methods for improving yield and profitability in the rice-wheat system of Eastern India. *Field Crop Res.* **2020**, *250*, 107776. [CrossRef]
61. Jehangir, I.A.; Hussain, A.; Sofi, N.R.; Wani, S.H.; Ali, O.M.; Abdel Latef, A.A.H.; Raja, W.; Bhat, M.A. Crop establishment methods and weed management practices affect grain yield and weed dynamics in temperate rice. *Agronomy* **2021**, *11*, 2137. [CrossRef]
62. Singh, V.P.; Singh, A.P.; Dhyani, V.C.; Banga, A.; Kumar, K.; Satyawali; Bisht, N. Weed management in direct seeded rice. *Indian J. Weed Sci.* **2016**, *48*, 233–246. [CrossRef]
63. Bamji, M.S.; Nair, K.M. Food-based approach to combat micronutrient deficiencies. *Proc. Indian Natl. Sci. Acad.* **2016**, *82*, 1529–1540. [CrossRef]
64. Prasad, R.; Shivay, Y.S. Agronomic biofortification of plant foods with minerals, vitamins and metabolites with chemical fertilizers and liming. *J. Plant Nutr.* **2020**, *43*, 1534–1554. [CrossRef]
65. Lonnerdal, B. Dietary factors influencing zinc absorption. *J. Nutr.* **2000**, *130*, 1378S–1383S. [CrossRef]
66. Sharma, P.K.; Ladha, J.K.; Bhushan, L. Soil physical effects of puddling in rice-wheat cropping systems. In *Improving the Productivity and Sustainability of Rice-Wheat Systems: Issues and Impacts*; Ladha, J.K., Hill, J.E., Duxbury, J.M., Gupta, R.K., Buresh, R.J., Eds.; ASA Special Publications: Madison, WI, USA, 2003; Volume 65. [CrossRef]

energies

MDPI

Article

Deep-Learning-Based Adaptive Model for Solar Forecasting Using Clustering

Sourav Malakar [1,*,†], Saptarsi Goswami [2,†], Bhaswati Ganguli [3], Amlan Chakrabarti [1], Sugata Sen Roy [3], K. Boopathi [4] and A. G. Rangaraj [4]

[1] A.K. Choudhury School of Information Technology, University of Calcutta, Kolkata 700073, India; achakra12@yahoo.com
[2] Bangabasi Morning College, University of Calcutta, Kolkata 700073, India; sgakc@caluniv.ac.in
[3] Department of Statistics, University of Calcutta, Kolkata 700073, India; bgstat@gmail.com (B.G.); sugatasr@gmail.com (S.S.R.)
[4] National Institute of Wind Energy (NIWE), The Ministry of New and Renewable Energy, Government of India, New Delhi 110003, India; boopathi.niwe@nic.in (K.B.); rangaraj.niwe@nic.in (A.G.R.)
* Correspondence: sourav.xaviers@gmail.com
† These authors contributed equally to this work.

Abstract: Accurate short-term solar forecasting is challenging due to weather uncertainties associated with cloud movements. Typically, a solar station comprises a single prediction model irrespective of time and cloud condition, which often results in suboptimal performance. In the proposed model, different categories of cloud movement are discovered using K-medoid clustering. To ensure broader variation in cloud movements, neighboring stations were also used that were selected using a dynamic time warping (DTW)-based similarity score. Next, cluster-specific models were constructed. At the prediction time, the current weather condition is first matched with the different weather groups found through clustering, and a cluster-specific model is subsequently chosen. As a result, multiple models are dynamically used for a particular day and solar station, which improves performance over a single site-specific model. The proposed model achieved 19.74% and 59% less normalized root mean square error (NRMSE) and mean rank compared to the benchmarks, respectively, and was validated for nine solar stations across two regions and three climatic zones of India.

Keywords: clearness index forecasting; cloud cover; clustering; DTW

Citation: Malakar, S.; Goswami, S.; Ganguli, B.; Chakrabarti, A.; Roy, S.S.; Boopathi, K.; Rangaraj, A.G. Deep-Learning-Based Adaptive Model for Solar Forecasting Using Clustering. *Energies* **2022**, *15*, 3568. https://doi.org/10.3390/en15103568

Academic Editors: Luis Hernández-Callejo, Sergio Nesmachnow and Sara Gallardo Saavedra

Received: 24 March 2022
Accepted: 19 April 2022
Published: 13 May 2022

1. Introduction

Solar power is one of the viable alternatives to fossil-fuel-generated power, which causes serious environmental damage [1]. In terms of total energy consumption, India is ranked third after China and the United States [2], and has a target of producing 57% of total electricity capacity from renewable sources by 2027 [3]. In this paper, we developed a novel method for the short-term (some hours ahead) [4] forecasting of the clearness index (Kt) (defined as the ratio of global horizontal irradiance (GHI) to extraterrestrial irradiance) [5–8] while accounting for unpredictable weather conditions, focusing on variability in cloud cover [9–12]. Cloud variability leads to highly localized solar prediction, as a single model is unable to provide accurate forecasts under different weather conditions [13,14].

Long short-term memory (LSTM) [15] is one of the most popular deep-learning algorithms, mainly used to handle sequential data, and it can preserve knowledge by passing through the subsequent time steps of a time series [16]. In [17], the authors developed a site-specific univariate LSTM for the hourly forecasting of photovoltaic power output. In [18], the authors compared the performance of several alternative models for forecasting clear-sky GHI. These included gated recurrent units (GRUs), LSTM, recurrent neural networks (RNNs), feed-forward neural networks (FFNNs), and support vector regression (SVR). GRU and LSTM outperformed the other models in terms of root mean square error

(RMSE). In [19], the authors proposed an hour-ahead solar power forecasting model based on RNN-LSTM for three different solar plants. In [9], LSTM and GRU dominated over artificial neural networks (ANNs), FFNNs, SVR, random forest regressor (RFR), and multilayer perceptron (MLP) in solar forecasting. The above discussion suggests that the authors used a single model to forecast solar irradiation for a particular day and did not consider cloud cover at the time of forecasting. In [20], the authors designed a forecasting model for one-day ahead hourly prediction using LSTM. The authors reported that the algorithm performed effectively under fully or partially cloudy conditions. In [21], the authors proposed a one-hour-ahead hybrid solar forecasting model using traditional machine-learning models such as random forest (RF), gradient boosting (GB), support vector machines (SVMs), and ANNs. The RF model showed the best forecasting accuracy for the spring and autumn seasons, while the SVR model performed best for the winter and summer seasons. In [22], the authors evaluated 68 machine-learning models for 3 sky conditions, 7 locations, and 5 climate zones in the continental United States. No universal model exists, and specific models for each sky and climate condition are recommended. Hence, it is well-established that a single site-specific forecasting model is unable to produce consistent forecasting performance in all cloud conditions and seasons [20–22].

Typically, a site-specific model is built for solar-energy prediction, and multiple models are built for different seasons. However, even within the same day, there can be fluctuations due to variability in cloud cover [23]. So, a single model gives a very high error in terms of NRMSE. The error is further pronounced for time windows with high cloud variability. The authors in [22], found that a specific forecasting model showed very high error in NRMSE in overcast cloud conditions in comparison with clear-sky conditions on a particular day. They also stated that forecasting performance significantly changed with the change in cloud conditions on a particular day. The authors in [24], stated that LSTM outperformed other predictive models in short-term solar forecasting. Nevertheless, its ability to predict cloudy days with low solar irradiance is significantly reduced. This serves as a motivation to implement an adaptive model. Table 1 summarizes the forecasting error of LSTM for nine solar stations across three climatic zones of India. Solar stations are described in Section 3.1. Figure 1 depicts the deviation of NRMSE in high- and low-variability cloud-cover conditions in comparison with overall NRMSE. In high cloud-cover variability, forecasting error was significantly higher compared to overall NRMSE. This signifies that if cloud variability ever increases too much, site-specific LSTM cannot handle such a situation very well. Another motivation is provided by the parity plot in Figure 2, which shows forecast and actual clearness indices for three solar stations separately for high and low cloud-cover variability conditions. Forecasts were more accurate under low-variability cloud cover conditions than those for high-variability cloud conditions.

In this paper, we propose a novel short-term (2 h ahead) solar forecasting approach [25] that uses clustering on the basis of cloud parameters as a preprocessing step, and subsequently uses LSTM that is cluster-specific for the forecasting clearness index. Specifically,

- For each forecasting site, the nearest three neighboring stations were selected on the basis of DTW similarity scores [26].
- A global dataset was created by combining some derived features of cloud cover and clearness indices of each station and those of its neighbors. The derived features were obtained following [23].
- The entire day was divided into time windows. For clustering, the K-medoids [27] algorithm was applied on those time windows.
- A separate LSTM model was trained for each cluster that represented different cloud conditions.

The major contributions of the paper are as follows:

- An adaptive forecasting model (CB-LSTM) is proposed that can apply multiple models for a site on the basis of existing weather conditions.
- A global dataset was created on the basis of derived cloud-related information that is used to cluster a day into different weather types.

- The proposed model showed promising forecasting performance compared to benchmark models such as convolutional neural network (CNN)-LSTM and nonclustering-based site-specific LSTMs. The model achieved less forecasting error for solar stations having significant solar variability.
- Performance (measured in terms of forecasting accuracy) was validated for nine solar stations from three climatic zones in India. To our knowledge, this is the first time that such an approach was applied to data from the Indian subcontinent.

The rest of the paper is organized as follows. Section 2 provides the background on various deep-learning model architectures. Section 3 presents the proposed method, and Section 4 presents details on its forecasting performance. The paper is concluded with a discussion in Section 5.

Table 1. Forecasting performance (NRMSE in %) of LSTM for high- and low-cloud-variability cloud cover.

Stations	High Cloud-Cover Variability	Low Cloud-Cover Variability	Overall NRMSE
Composite climate zone			
Bhainsdehi	44.72	17.71	29.85
Begamganj	39.03	20.80	27.98
Dindori	41.16	14.90	25.73
Hot and dry climate zone			
Tiruchirappalli	59.06	44.02	47.57
Idukki	87.99	47.70	61.13
Madurai	47.89	44.29	45.65
Warm and humid climate zone			
Khaga	34.23	17.42	23.63
Vaibhavwadi	92.22	38.22	57.31
Osmanabad	71.72	33.58	47.83

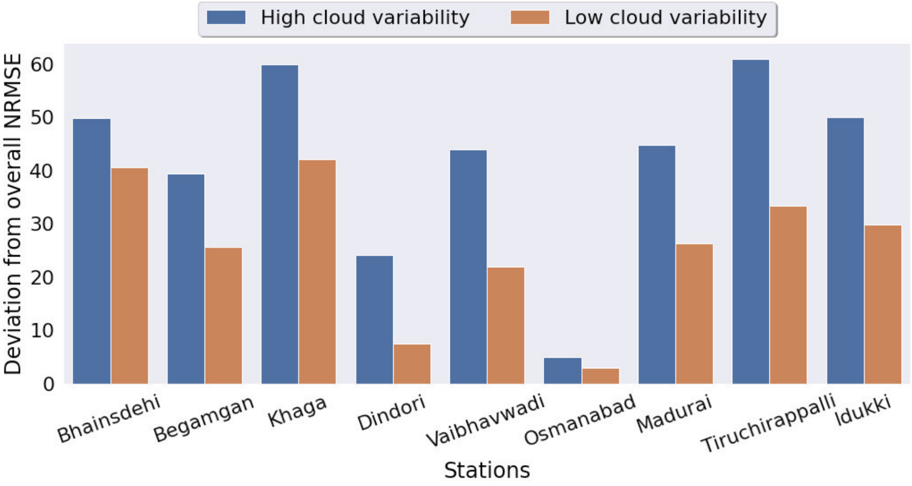
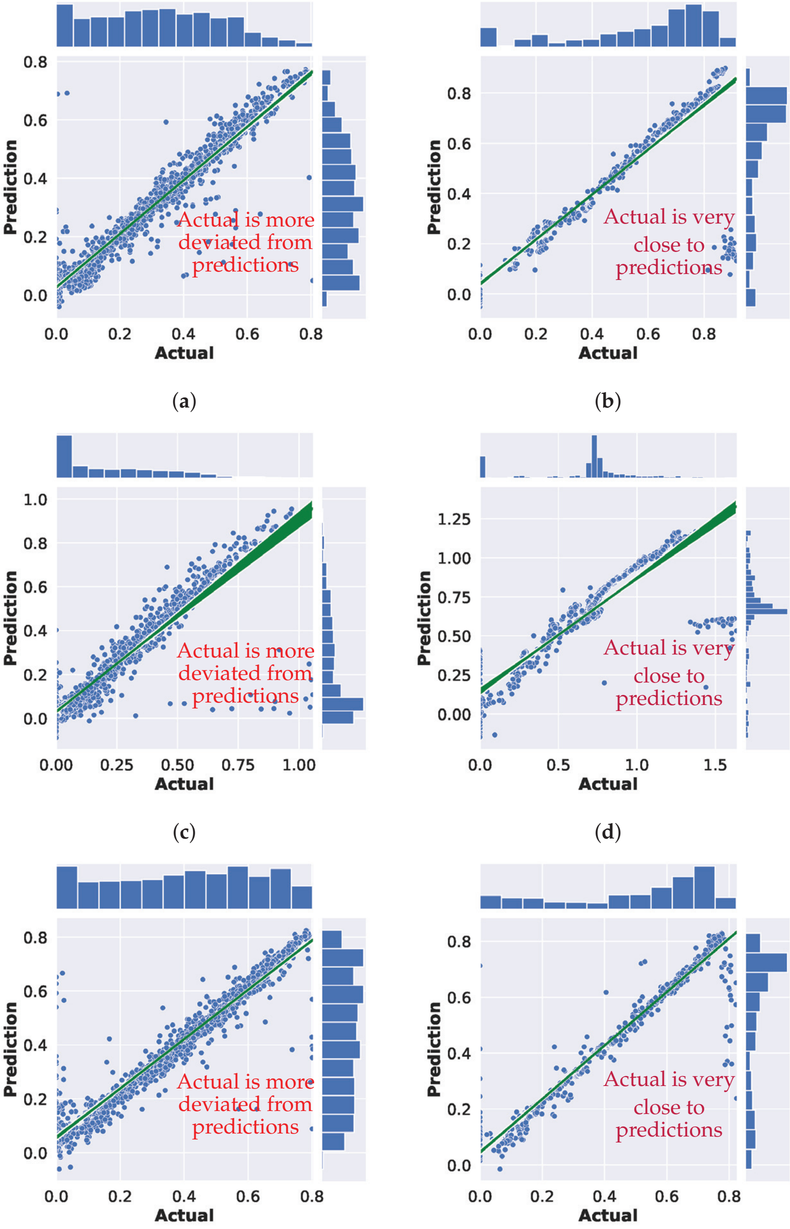


Figure 1. Deviation of NRMSE (%) in high and low cloud-cover variability conditions compared to overall NRMSE (%).



(**a**)

(**b**)

(**c**)

(**d**)

(**e**)

(**f**)

Figure 2. Parity plot showing forecast and actual clearness indices for three solar stations for high and low cloud-cover variability conditions. (**a**) Bhainsdehi (high cloud-cover variability); (**b**) Bhainsdehi (low cloud-cover variability); (**c**) Osmanabad (high cloud-cover variability); (**d**) Osmanabad (low cloud-cover variability); (**e**) Khaga (high cloud-cover variability); (**f**) Khaga (low cloud-cover variability).

2. Deep-Learning Models

ANN in the learning phase is unable to utilize information learned from the past time steps while processing the current time step, which is the major drawback of traditional neural networks. An RNN can solve this problem and is one of the deep-learning models designed to handle sequential data. To preserve information, it recursively transfers learning from previous time steps of the network to the current time step. However, it is susceptible to the vanishing gradient problem. As a result, it is unable to remember long-term dependencies.

LSTMs are a special type of RNN that is especially designed to learn both long- and short-term dependencies [15]. Compared to a traditional neural network, LSTM units encompass a ‘memory cell’ that can retain and maintain information for long periods of time [28]. Figure 3 is a schematic diagram of an LSTM cell. A set of gates are used to customize the hidden states. Three different gates are used, representing input, forget, and output. The functionality of each gate is summarized as follows.

- **Forget gate [29]** $(f_t) = \sigma(w_f[h_{t-1}, x_t] + b_f)$**:** On the basis of certain conditions such as x_t, h_{t-1}, and a sigmoid layer, a forget gate produces either 0 or 1. If 1, memory information is preserved; otherwise, it is discarded.
- **Input gate [29]** $(i_t) = \sigma(w_i[h_{t-1}, x_t] + b_i)$**:** helps in deciding which values from the input are used for the current memory state.
- **Cell state [29]** $(c_t) = tanh(w_c[h_{t-1}, x_t] + b_c)$**:** new cell state c_t is the summation of $c_{t-1} * f_t$, and $\widehat{c_t} * i_t$. $c_{t-1} * f_t$ decides the fraction of the old cell state that is discarded, and the amount of new information that is added is decided through $\widehat{c_t} * i_t$.
- **Output gate [29]** $(o_t) = \sigma(w_o[h_{t-1}, x_t] + b_o)$**:** decides what to output on the basis of the current input and previous hidden state.
- **Hidden state [29]** $(h_t) = o_t * tanh(c_t)$**:** current hidden state is computed by multiplying the output gate by the current cell state using the tanh function.

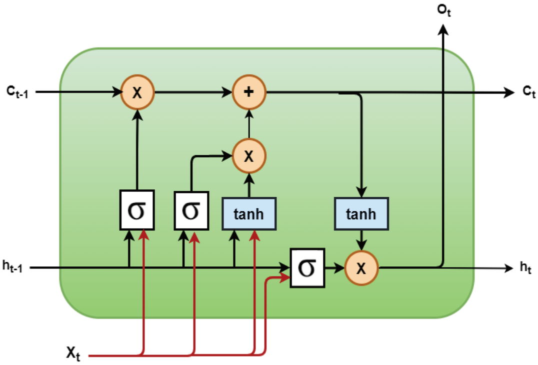


Figure 3. Schematic diagram of an LSTM cell.

Here, w_f, w_i, w_c and w_o are weight matrices. b_f, b_i, b_c and b_o are biases for individual gates. σ indicates a sigmoid activation function. * stands for element-wise multiplication, and + implies element-wise addition.

3. Materials and Methods

3.1. Dataset Description and Preprocessing

We merged two sources of data for this analysis. The first was obtained from Modern-Era Retrospective analysis for Research and Applications, Version 2 (MERRA-2) [30] satellite. This provides information on PM2.5, surface wind speed (m/s), surface air temperature (k), total cloud area fraction, dew point temperature at 2 m (k), 2 m eastward wind (m/s), and 2 m northward wind (m/s). The second set of data was extracted from from the National Renewable Energy Laboratory (NREL) [31] for 2013. This contains information on DHI, DNI, GHI, clear-sky DHI, clear-sky DNI, clear -sky GHI, and solar zenith angle. The clearness index at time t (denoted by Kt) was calculated on the basis of GHI values. These two datasets were merged on the basis of latitude and longitude. For each location (unique combination of latitude and longitude), a 10 km radius was used for the merge. Table 2 describes the nine solar stations studied in this paper.

Table 2. Dataset description.

Latitude and Longitude	Region	Climatic Zone	Location
23.50 and 78.75	Inland	Composite	Begamganj (Madhya Pradesh)
22.50 and 81.25	Inland	Composite	Dindori (Madhya Pradesh)
25.50 and 81.25	Inland	Composite	Khaga (Uttar Pradesh)
21.50 and 77.50	Inland	Hot and dry	Bhainsdehi (Madhya Pradesh)
16.50 and 73.75	Coastal	Hot and dry	Vaibhavwadi (Maharashtra)
18.00 and 76.25	Coastal	Hot and dry	Osmanabad (Maharashtra)
10.00 and 78.125	Inland	Warm and humid	Madurai (Tamil Nadu)
11.00 and 78.75	Coastal	Warm and humid	Tiruchirappalli (Tamil Nadu)
10.00 and 76.875	Coastal	Warm and humid	Idukki (Kerala)

After collecting solar data (time series) from the PV module, they were stored in a database, and a series of standard preprocessing steps were applied.

- Night hours (8 p.m. to 7 a.m.) are removed.The resolution of the collected data was 15 min.
- Next, a standard sliding-window approach was applied to the time-series data to convert them into a suitable representation (supervised) for deep-learning models. Figure 4 depicts the generic approach of a sliding window with input and output window sizes n and m. The input window covered n past observations, such as $\{X_1, X_2, X_3, \ldots, X_n\}$, and used to predict the next m observations as $\{X_{n+1}, \ldots, X_{n+m}\}$. After that, the input window is shifted one position to the right as $\{X_2, X_3, X_4, \ldots, X_{n+1}\}$, and $\{X_{n+2}, \ldots, X_{n+m+1}\}$ are the new input and output sequences. This process continues until no data points of the time series are left.

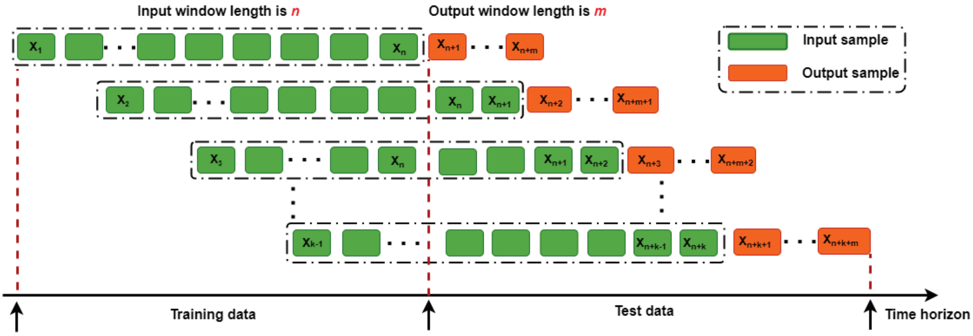


Figure 4. Sliding-window approach.

3.2. Locally and Remotely Derived Variables for Clustering

Variables that were obtained on the basis of the meteorological parameters of the forecast site are called locally derived variables, while variables acquired depending upon the neighboring forecasting sites are called remotely derived variables. Derived variables were obtained following [23].

Neighboring solar stations relative to the forecast site were selected as follows:

- A radius of 120 km was used to select the solar stations around the forecasting site.
- DTW [32] was used as the similarity measure of clearness index (Kt).

- Three solar stations with the best similarity score were selected and used for clustering. Table 3 lists all locally and remotely derived variables.

Table 3. Variables used for K-medioid clustering.

Predictors	Equation	Description
Locally derived variables		
kt_{trend}	$\text{KtPrev15} - \text{KtPrev30}$	It was calculated to capturethe most recent trend
Kt temporal variability (Stdev 1-h)	$\sigma(\text{KtPrev15}, \text{KtPrev30}, \text{KtPrev45}, \text{KtPrev60})$	Computed by taking the standard deviation of the four subsequent observations in an hour
Kt Slope (1-h)	$\beta_0 + \beta_1\text{KtPrev15} + \beta_2\text{KtPrev30} + \beta_3\text{KtPrev45} + \beta_4\text{KtPrev60}$	It was calculated by fitting a linear equation through four consecutive observations of the clearness index.
Remotely derived variables		
KtPrev15 nearby mean	$\mu(\text{KtPrev15}_{Source1}, \text{KtPrev15}_{Source2}, \text{KtPrev15}_{Source3})$	It was computed by taking the mean of the clearness index of three neighboring sites.
KtPrev15 nearby std	$\sigma(\text{KtPrev15}_{Source1}, \text{KtPrev15}_{Source2}, \text{KtPrev15}_{Source3})$	It was computed by taking the standard deviation of the clearness index of three neighboring sites.
Cloud-cover variability (Stdev)	$\sigma(\text{CV}_{Source1}, \text{CV}_{Source2}, \text{CV}_{Source3})$	It is the standard deviation of the cloud cover of neighboring solar stations.
Cloud Cover Squired	$(\mu(\text{CV}_{Source1}, \text{CV}_{Source2}, \text{CV}_{Source3}))^2$	It is the squared value of the mean of cloud covers of three neighboring sites.

3.3. Multivariate LSTM (M-LSTM)

Figure 5b describes the architecture of the multivariate LSTM model, summarized as follows:

- It is a site-specific model where predictors are directly used from the forecasting site. The model uses additional information on predictors such as temperature, dew point, wind speed, and cloud cover.
- A stateful LSTM was used for maintaining inter- and intrabatch dependency. The input layer of an LSTM consists of 5 input features of 16 time steps (4 h) each. Two hidden layers and a tanh activation were used.

3.4. Spatiotemporal LSTM (ST-LSTM)

This implementation led to a spatiotemporal model, as information from the forecast site is used together with information from neighboring sites.

Figure 5a shows the architecture in detail and can be summarized as follows:

- A spatiotemporal dataset was created by combining information on meteorological parameters, including dew point, temperature, wind speed, and cloud cover from the three neighboring sites and from the forecast site.
- Stateful LSTM was used. The input layer consisted of 20 input features of 16 time steps each. Two hidden layers were used with a dropout rate of 20%. As hidden-layer activation, tanh was used.
- A spatiotemporal dataset was used to train the LSTM and forecast clearness index for three different times of the day.

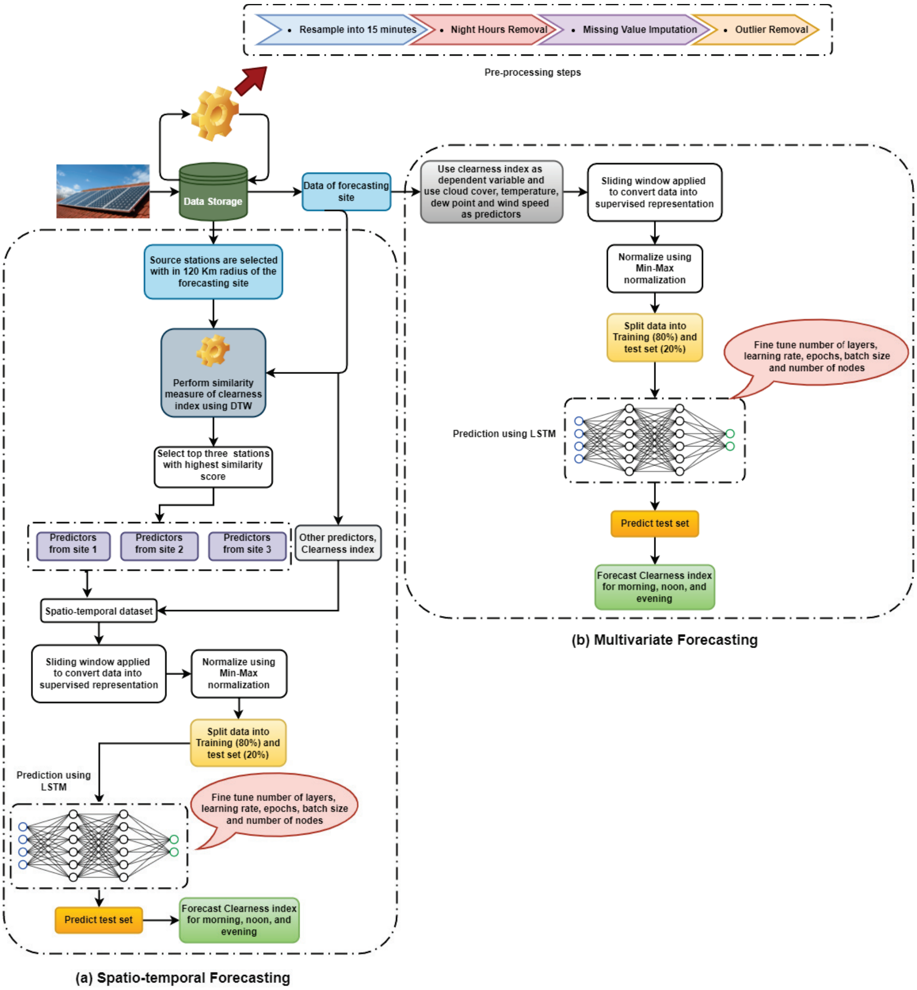

Resample into 15 minutes
Night Hours Removal
Missing Value Imputation
Outlier Removal
Pre-processing steps
Data Storage
Data of forecasting site
Source stations are selected with in 120 Km radius of the forecasting site
Perform similarity measure of clearness index using DTW
Select top three stations with highest similarity score
Predictors from site 1
Predictors from site 2
Predictors from site 3
Other predictors, Clearness index
Spatio-temporal dataset
Sliding window applied to convert data into supervised representation
Normalize using Min-Max normalization
Split data into Training (80%) and test set (20%)
Prediction using LSTM
Fine tune number of layers, learning rate, epochs, batch size and number of nodes
Predict test set
Forecast Clearness index for morning, noon, and evening
(a) Spatio-temporal Forecasting
Use clearness index as dependent variable and use cloud cover, temperature, dew point and wind speed as predictors
Sliding window applied to convert data into supervised representation
Normalize using Min-Max normalization
Split data into Training (80%) and test set (20%)
Fine tune number of layers, learning rate, epochs, batch size and number of nodes
Prediction using LSTM
Predict test set
Forecast Clearness index for morning, noon, and evening
(b) Multivariate Forecasting

Figure 5. Architecture of—(**a**) spatiotemporal and (**b**) multivariate forecasting models.

3.5. Clustering-Based ANN (CB-ANN) and LSTM (CB-LSTM)

CB-ANN was designed following [23]. CB-LSTM is a global forecasting model, designed using K-medoid clustering followed by LSTM. Meteorological parameters collected from the neighboring sites together with clearness index values from the forecast site were not directly used as predictors. As mentioned in Section 3.2, derived features were extracted for Kt from the forecast site and cloud cover information from neighboring sites.

Figure 6 shows a cluster-specific feature identification to understand the important features of a cluster. For each forecasting site, a spatiotemporal dataset was created that was split into training (80%) and test sets (20%). A global dataset was created by combining the training sets of all the forecast sites, and normalized using the min–max normalizer [33]. Next, the optimal number of clusters was determined on the basis of the elbow–silhouette [34] method. The K-medoids algorithm is used to cluster the time windows in the dataset. As the input attributes were related to cloud formation, the clusters intuitively represent different cloud types. As a result, the dataset was split into k clusters, where each cluster center was represented by a medoid.

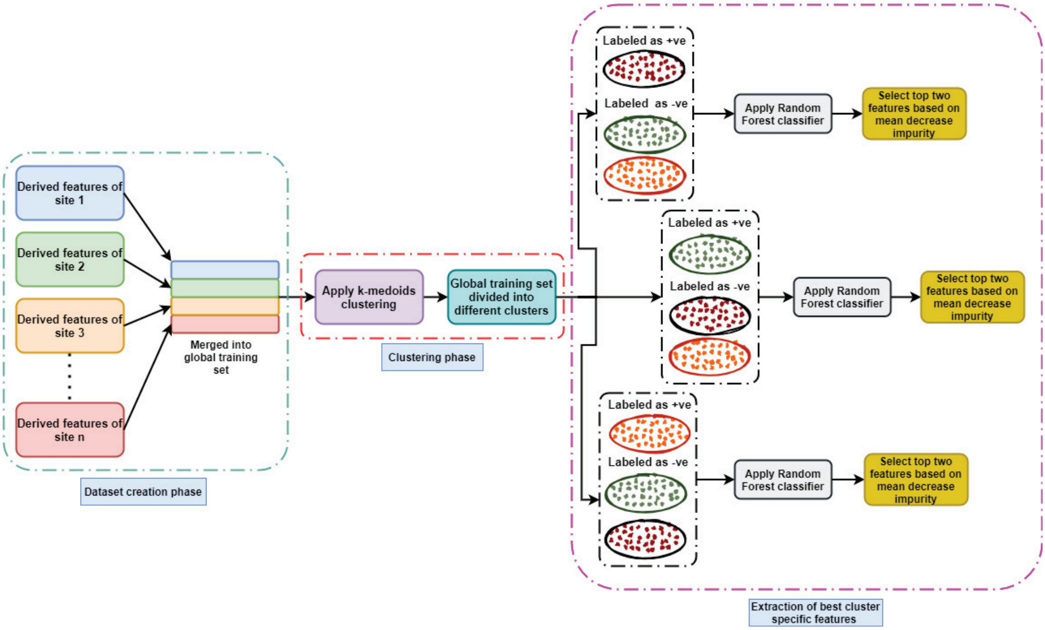

Derived features of site 1
Derived features of site 2
Derived features of site 3
Derived features of site n
Merged into global training set
Dataset creation phase
Apply k-medoids clustering
Global training set divided into different clusters
Clustering phase
Labeled as +ve
Labeled as -ve
Apply Random Forest classifier
Select top two features based on mean decrease impurity
Extraction of best cluster specific features

Figure 6. Cluster-specific best feature identification strategy.

The proposed approach is presented in Figure 7 and is described as follows:

- For each cluster, separate stateless LSTM was built. The network was implemented using the keras package of version 2.7.0 in Python.The input layer of the LSTM consisted of 13 input features of 1 time step each. In the hidden layers, hyperbolic tangent activation function (tanh) was used. After each hidden layer, a batch-normalization [35] layer was used to transform inputs into a mean of 0 and a standard deviation of 1. We used a dropout and L2 regularization [36] to protect the network from overfitting. The network weights were initialized using the Xavier uniform initializer [37]. The output layer consisted of 8 nodes with linear activation and was used to forecast Kt for the next two hours. Figure 8 shows the network configuration of CB-LSTM together with input and output features. For obtaining the best performance of CB-LSTM, specific hyperparameters such as number of layers, number of nodes in each layer, batch size, number of epochs, dropout, and learning rate were optimized. The tree-structured Parzen estimator (TPE) [38] algorithm was used for optimization. Table 4 presents the hyperparameter settings. Table 5 shows the optimal hyperparameter settings for the proposed approach (CB-LSTM) in Section 4.
- Overall forecasting accuracy was computed for each site using the weighted average of the generated accuracy by the different LSTM models. Forecasting accuracy was separately computed for three times of the day.

Table 4. Hyperparameters to optimize.

Model	Hyperparameter	Value
CB-LSTM	Number of layers	1, 2, 3
	Nodes in layers	25, 50, 75, 100
	Learning rate	0.1, 0.01, 0.001
	Batch size	1, 10, 20, 50, 100
	Epoch	25, 50, 100, 150, 200
	Dropout	0.05, 0.1, 0.2

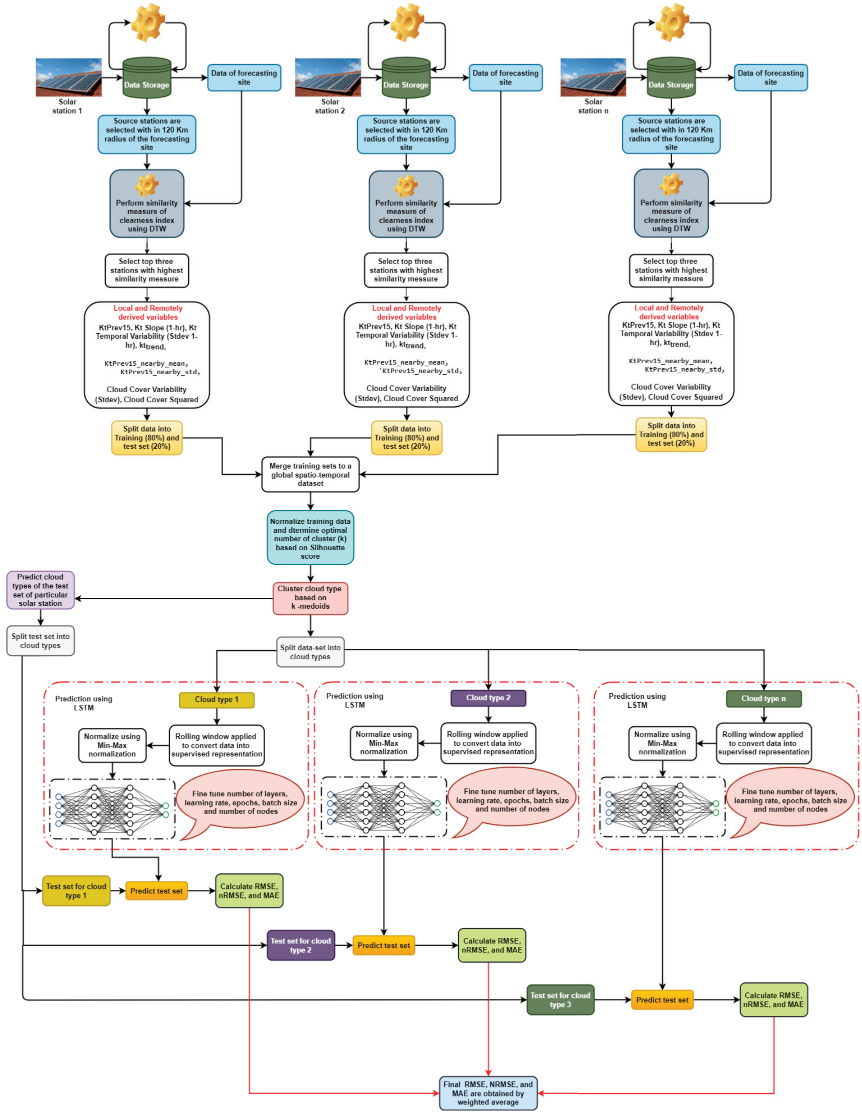

Solar station 1
Data Storage
Data of forecasting site
Source stations are selected with in 120 Km radius of the forecasting site
Perform similarity measure of clearness index using DTW
Select top three stations with highest similarity messure
Local and Remotely derived variables
KtPrev15, Kt Slope (1-hr), Kt Temporal Variability (Stdev 1-hr), kt_{trend},
KtPrev15_nearby_mean, KtPrev15_nearby_std,
Cloud Cover Variability (Stdev), Cloud Cover Squared
Split data into Training (80%) and test set (20%)
Solar station 2
Data Storage
Data of forecasting site
Source stations are selected with in 120 Km radius of the forecasting site
Perform similarity measure of clearness index using DTW
Select top three stations with highest similarity messure
Local and Remotely derived variables
KtPrev15, Kt Slope (1-hr), Kt Temporal Variability (Stdev 1-hr), kt_{trend},
KtPrev15_nearby_mean, 'KtPrev15_nearby_std,
Cloud Cover Variability (Stdev), Cloud Cover Squared
Split data into Training (80%) and test set (20%)
Solar station n
Data Storage
Data of forecasting site
Source stations are selected with in 120 Km radius of the forecasting site
Perform similarity measure of clearness index using DTW
Select top three stations with highest similarity messure
Local and Remotely derived variables
KtPrev15, Kt Slope (1-hr), Kt Temporal Variability (Stdev 1-hr), kt_{trend},
KtPrev15_nearby_mean, KtPrev15_nearby_std,
Cloud Cover Variability (Stdev), Cloud Cover Squared
Split data into Training (80%) and test set (20%)
Merge training sets to a global spatio-temporal dataset
Normalize training data and dtermine optimal number of cluster (k) based on Silhouette score
Predict cloud types of the test set of particular solar station
Cluster cloud type based on k -medoids
Split test set into cloud types
Split data-set into cloud types
Prediction using LSTM
Cloud type 1
Normalize using Min-Max normalization
Rolling window applied to convert data into supervised representation
Fine tune number of layers, learning rate, epochs, batch size and number of nodes
Prediction using LSTM
Cloud type 2
Normalize using Min-Max normalization
Rolling window applied to convert data into supervised representation
Fine tune number of layers, learning rate, epochs, batch size and number of nodes
Prediction using LSTM
Cloud type n
Normalize using Min-Max normalization
Rolling window applied to convert data into supervised representation
Fine tune number of layers, learning rate, epochs, batch size and number of nodes
Test set for cloud type 1
Predict test set
Calculate RMSE, nRMSE, and MAE
Test set for cloud type 2
Predict test set
Calculate RMSE, nRMSE, and MAE
Test set for cloud type 3
Predict test set
Calculate RMSE, nRMSE, and MAE
Final RMSE, NRMSE, and MAE are obtained by weighted average

Figure 7. Architecture of clustering-based forecasting model.

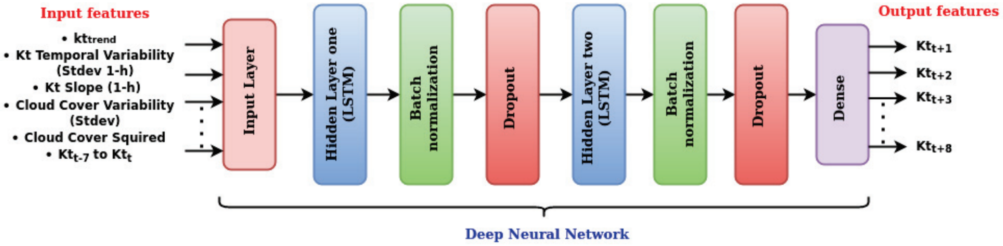


Figure 8. Network configuration of CB-LSTM.

Table 5. Optimal hyperparameter settings for CB-LSTM in different cloud conditions.

Hyperparameters	Broken	Clear/Sunny	Broken/Overcast
Number of hidden layers	2	2	2
Nodes in hidden layer one	100	25	100
Nodes in hidden layer two	100	25	100
Learning rate	0.001	0.001	0.001
Batch size	1	1	1
Epochs	200	200	100
Dropout	0.2	0.2	0.2
Model parameters	127,608	9408	127,608

3.6. Evaluation

Forecasting performance was evaluated using three performance evaluation metrics, namely, mean absolute error (MAE) [39], RMSE [40], and NRMSE [39]. They are defined as follows:

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|\hat{y}_i - y_i| \tag{1}$$

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}{N}} \tag{2}$$

$$\text{NRMSE} = \frac{\sqrt{\frac{\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}{N}}}{\mu} \tag{3}$$

$$\text{NRMSE}(\%) = \frac{\sqrt{\frac{\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}{N}}}{\mu} * 100\% \tag{4}$$

4. Result and Discussion

Figure 9 shows the average value and standard deviation of the clearness index for all forecasting sites. Greater variability was observed for Idukki, Vaibhavwadi, Tiruchirappalli, and Osmanabad. Khaga had the lowest variability.

Figure 10 shows cluster-specific features, and K-medoids led to three clusters. Features were selected on the basis of the reduction in impurity scores. For Cluster 0, the mean decrease in impurity was highest for features Kt slope (1-h) and KtPrev15. For Cluster 1, cloud cover squared and KtPrev15-nearby-mean were the most important features. For Cluster 2, cloud cover squared and Kt Slope (1-h) were the most important features. To understand the cloud type of each cluster, we calculated the percentage of cloud-cover information falling in each cluster, and this is illustrated in Figure 11. For Cluster 0, the majority of observations belonged to the broken-cloud type. In Cluster 1, the majority of observations belonged to the clear/sunny-sky type. For Cluster 2, the total numbers of observations in the broken and bvercast cloud types were relatively similar.

Table 5 shows the optimal hyperparameters of the proposed approach (CB-LSTM) for three different cloud conditions of broken, clear/sunny, and broken/overcast. Complex cloud conditions (broken or overcast) require more hidden nodes and parameters to produce good forecasting. Nevertheless, model complexity is less in clear/sunny sky conditions.

Table 6 provides information on the climatic zone-specific forecasting performance of ST-LSTM and a comparison with M-LSTM. For the composite climatic zone, ST-LSTM achieved 5.96%, 3.71%, and 8.80% less RMSE, NRMSE, and MAE, respectively, than the M-ULSTM did. For the hot and dry climatic zone, the corresponding values were 1.58%, 1.44%, and 0.25% respectively. The biggest gain was in the warm and humid climatic zone, with corresponding percentages at 8.65%, 8.34%, and 11.55% respectively.

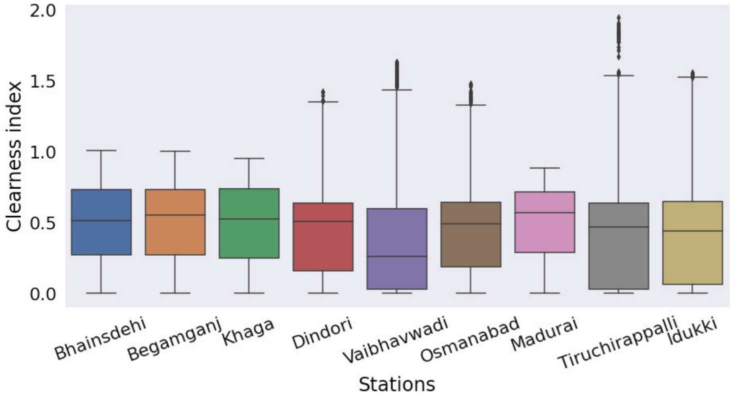


Figure 9. Average value and variability of clearness index of forecasting sites.

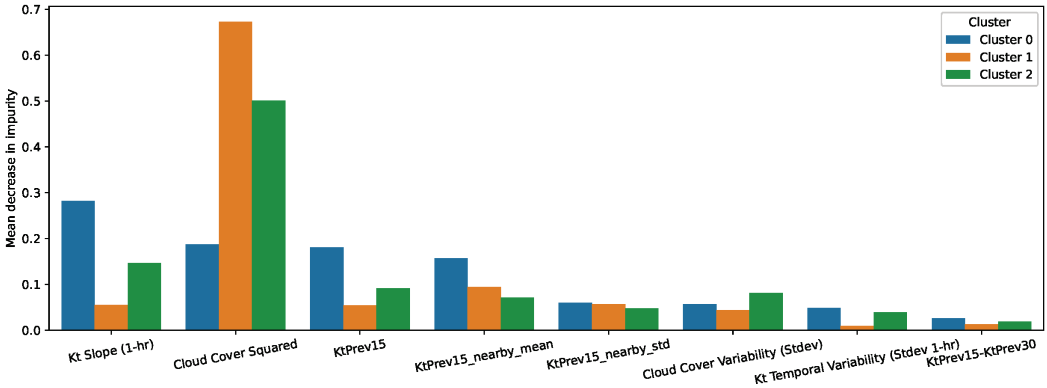


Figure 10. Cluster-specific best features in terms of mean decrease in impurity.

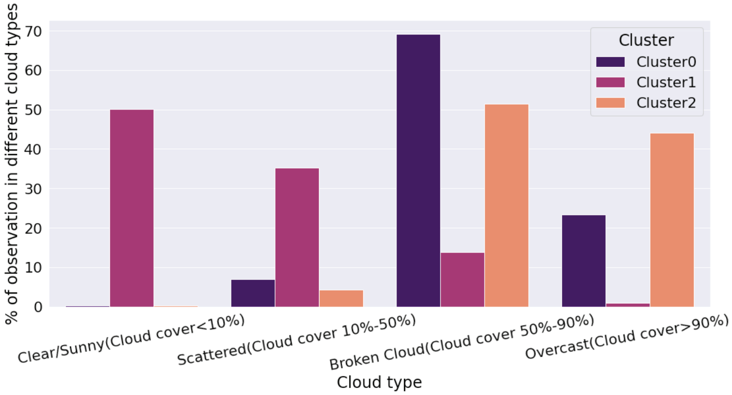


Figure 11. Understanding cloud patterns via cluster-specific distribution of cloud type.

Table 6. Forecasting performance of spatial LSTM compared to univariate and multivariate LSTMs.

		M-LSTM			ST-LSTM	
Forecasting Sites	**RMSE**	**NRMSE**	**MAE**	**RMSE**	**NRMSE**	**MAE**
			Composite			
Bhainsdehi	0.1429	0.2550	0.0870	0.1408	0.2515	0.0812
Begamganj	0.2211	0.3568	0.0952	0.2090	0.3410	0.0960
Dindori	0.2430	0.3541	0.1258	0.2164	0.3352	0.1358
			Hot and dry			
Tiruchirappalli	0.2859	0.5426	0.1894	0.2854	0.5419	0.1888
Idukki	0.3087	0.6243	0.1957	0.2969	0.6037	0.1837
Madurai	0.2725	0.5497	0.1746	0.2704	0.5447	0.1845
			Warm and humid			
Khaga	0.2253	0.3888	0.1049	0.1890	0.3300	0.1113
Vaibhavwadi	0.2911	0.4708	0.1821	0.2726	0.4423	0.1831
Osmanabad	0.3001	0.4541	0.2188	0.2896	0.4408	0.1941

Table 7 demonstrates the superiority of CB-LSTM over CB-ANN and ST-LSTM in terms of RMSE, NRMSE, and MAE. For the composite climatic zone, CB-LSTM outperformed CB-ANN by 27.16%, 29.49%, and 38.86% in terms of RMSE, NRMSE, and MAE, respectively. For the hot and dry climatic zone, percentages were -5.28%, 8.03%, and 8.85%, respectively. For the warm and humid climatic zone, CB-LSTM achieved 9.80%, 22.04%, and 19.94% less RMSE, NRMSE, and MAE, respectively, as compared to CB-ANN. ST-LSTM was dominated by CB-LSTM in the composite climatic zone by 33.77%, 28.49%, and 19.64% in terms of RMSE, NRMSE, and MAE, respectively. In the hot and dry climatic zone, CB-LSTM led to reductions of 35.37%, 35.26%, and 34.74% in RMSE, NRMSE, and MAE, respectively, as compared to ST-LSTM. For the warm and humid climatic zone, CB-LSTM led to corresponding reductions of 27.65%, 17.78%, and 25.34%, respectively.

The largest gain was observed in the composite climatic zone compared to CB-ANN in terms of RMSE, NRMSE, and MAE. On the other hand, compared to CB-LSTM, the greatest gain was seen in the hot and dry climatic zone. Thus, CB-LSTM led to less forecasting error than that of the M-LSTM and ST-LSTM. CB-LSTM dominated both M-LSTM and ST-LSTM at each of the three different times of day in terms of NRMSE.

Table 7. Forecasting performance of CB-LSTM compared to multivariate and spatiotemporal LSTM.

		CB-LSTM			CB-ANN			ST-LSTM	
Forecasting Sites	**RMSE**	**NRMSE**	**MAE**	**RMSE**	**NRMSE**	**MAE**	**RMSE**	**NRMSE**	**MAE**
				Composite					
Bhainsdehi	0.1096	0.1936	0.0677	0.1569	0.2762	0.1243	0.1408	0.2515	0.0812
Begamganj	0.1148	0.2016	0.0751	0.1664	0.3118	0.1384	0.2090	0.3410	0.0960
Dindori	0.1426	0.2530	0.1079	0.1790	0.3440	0.1445	0.2164	0.3352	0.1358
				Hot and dry					
Tiruchirappalli	0.1605	0.2903	0.1094	0.1617	0.3414	0.1173	0.2854	0.5419	0.1888
Idukki	0.2118	0.4934	0.1339	0.2001	0.5288	0.1376	0.2969	0.6037	0.1837
Madurai	0.1793	0.3208	0.1193	0.1619	0.3288	0.1447	0.2704	0.5447	0.1845
				Warm and humid					
Khaga	0.1483	0.2641	0.1139	0.1623	0.3195	0.1258	0.1890	0.3300	0.1113
Vaibhavwadi	0.2009	0.3599	0.1200	0.2109	0.4980	0.1475	0.2726	0.4423	0.1831
Osmanabad	0.1879	0.3757	0.1089	0.2238	0.4760	0.1595	0.2896	0.4408	0.1941

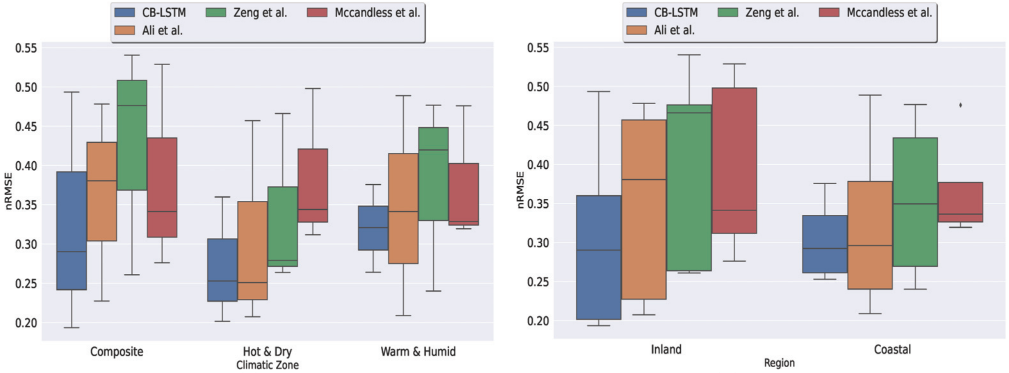
Table 8 illustrates the climatic-zone-specific forecasting superiority of CB-LSTM compared to three benchmark models [21,23,41]. In the hot and dry climatic zone, CB-LSTM achieved maximal gain with 8.86% and 26.81% lower NRMSE compared to [21,41]. On the other hand, in the composite climatic zone, the best NRMSE was 30.56% compared to [23].

Table 8. Forecasting performance of CB-LSTM compared with benchmark models in terms of NRMSE (%).

Forecasting Sites	CB-LSTM	[21]	[41]	[23]
Composite				
Bhainsdehi	19.36%	22.75%	26.09%	27.62%
Begamganj	20.16%	20.74%	26.39%	31.18%
Dindori	25.30%	25.09%	27.93%	34.40%
Hot and dry				
Tiruchirappalli	29.03%	38.06%	47.62%	34.14%
Idukki	49.34%	49.82%	54.04%	52.88%
Madurai	32.08%	34.14%	47.67%	32.88%
Warm and humid				
Khaga	26.41%	20.90%	24.02%	31.95%
Vaibhavwadi	35.99%	45.70%	46.61%	49.80%
Osmanabad	37.57%	48.89%	41.98%	47.60%

Figure 12a shows climatic-zone-specific variability in predictions of CB-LSTM in terms of NRMSE. clustering-based ANN [23] for the composite and hot and dry climatic zone, and RF-SVR [21] for the warm and humid climatic zone were the worst-performing models. For all climatic zones, CB-LSTM achieves the least prediction error.

Figure 12b shows region-specific variability in predictions of CB-LSTM in terms of NRMSE. CB-LSTM had the least prediction error in both inland and coastal regions.



(**a**) (**b**)

Figure 12. (**a**) Climatic-zone-specific variability in predictions; (**b**) region-specific variability in predictions. The symbol "†" indicates an outlier.

Table 9 shows a comparison of the overall forecasting performance of CB-LSTM to that of three benchmark models in terms of NRMSE and mean rank. CB-LSTM showed the lowest overall NRMSE and mean rank.

Table 9. Overall forecasting performance of CB-LSTM compared to that of benchmarks.

Models	Overall NRMSE	Mean Rank
[41]	0.3803	3.11
[21]	0.3378	2.11
[23]	0.3805	3.33
CB-LSTM	0.3058	1.44

5. Conclusions

CB-LSTM achieved better forecasting performance than that of M-LSTM and ST-LSTM for all climatic zones and regions. In terms of RMSE, MAE, and NRMSE, CB-LSTM dominated M-LSTM and ST-LSTM by 32.07%, 26.50%, 30.59%, 32.26%, 26.57%, and 27.18%.

CB-LSTM also outperformed three benchmark models [21,23,41] by 10.46%, 24.42%, and 24.36% in terms of overall NRMSE. CB-LSTM achieved the best mean rank compared to all the benchmark models. This holds for all the climatic zones and regions compared to the three benchmark models.

Thus, the performance of CB-LSTM was robust under differing conditions. We also obtained insights into the common nature of cloud patterns in India, as the clustering algorithm indicated relevant features about cloud patterns that could lead to improved forecasts.

The proposed model helps grid operators in better distributing power across the national grid. A future goal is to validate our model for more locations across other climatic zones, seasons, and topographical regions of India.

Author Contributions: Conceptualization, S.M. and S.G.; methodology, S.M.; software, S.M.; validation, S.M., S.G., B.G., A.C., S.S.R., K.B. and A.G.R.; formal analysis, S.M.; investigation, S.M.; resources, S.M.; data curation, S.M., K.B. and A.G.R.; writing—original draft preparation, S.M.; writing—review and editing, S.M., S.G., B.G. and A.C.; visualization, S.M.; supervision, S.G., B.G. and A.C.; project administration, S.G., B.G. and A.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Acknowledgments: This article resulted from Indo-USA collaborative project LISA 2020 between the University of Calcutta, India, and the University of Colorado, USA, and the research was supported by the National Institute of Wind Energy (NIWE) and the Technical Education Quality Improvement Programme (TEQIP).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Elizabeth Michael, N.; Mishra, M.; Hasan, S.; Al-Durra, A. Short-Term Solar Power Predicting Model Based on Multi-Step CNN Stacked LSTM Technique. *Energies* **2022**, *15*, 2150. [CrossRef]
2. Dudley, B. BP statistical review of world energy. In *BP Statistical Review*; BP p.l.c.: London, UK, 2018; Volume 6, p. 00116.
3. Safi, M. India plans nearly 60% of electricity capacity from non-fossil fuels by 2027. *The Guardian*, 21 December 2016; p. 22.
4. Gensler, A.; Henze, J.; Sick, B.; Raabe, N. Deep Learning for solar power forecasting—An approach using AutoEncoder and LSTM Neural Networks. In Proceedings of the 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Budapest, Hungary, 9–12 October 2016; pp. 002858–002865.
5. Akarslan, E.; Hocaoglu, F.O. A novel adaptive approach for hourly solar radiation forecasting. *Renew. Energy* **2016**, *87*, 628–633. [CrossRef]
6. Kumar, R.; Umanand, L. Estimation of global radiation using clearness index model for sizing photovoltaic system. *Renew. Energy* **2005**, *30*, 2221–2233. [CrossRef]
7. Liu, B.; Jordan, R. Daily insolation on surfaces tilted towards equator. *ASHRAE J.* **1961**, *10*, 5047843.
8. Perez, R.; Ineichen, P.; Seals, R.; Zelenka, A. Making full use of the clearness index for parameterizing hourly insolation conditions. *Sol. Energy* **1990**, *45*, 111–114. [CrossRef]
9. Rajagukguk, R.A.; Ramadhan, R.A.; Lee, H.J. A Review on Deep Learning Models for Forecasting Time Series Data of Solar Irradiance and Photovoltaic Power. *Energies* **2020**, *13*, 6623. [CrossRef]
10. Inman, R.H.; Pedro, H.T.; Coimbra, C.F. Solar forecasting methods for renewable energy integration. *Prog. Energy Combust. Sci.* **2013**, *39*, 535–576. [CrossRef]
11. Yang, D.; Jirutitijaroen, P.; Walsh, W.M. Hourly solar irradiance time series forecasting using cloud cover index. *Sol. Energy* **2012**, *86*, 3531–3543. [CrossRef]

12. Sanjari, M.J.; Gooi, H. Probabilistic forecast of PV power generation based on higher order Markov chain. *IEEE Trans. Power Syst.* **2016**, *32*, 2942–2952. [CrossRef]
13. Feng, C.; Cui, M.; Hodge, B.M.; Lu, S.; Hamann, H.F.; Zhang, J. Unsupervised clustering-based short-term solar forecasting. *IEEE Trans. Sustain. Energy* **2018**, *10*, 2174–2185. [CrossRef]
14. Bandara, K.; Bergmeir, C.; Smyl, S. Forecasting across time series databases using recurrent neural networks on groups of similar series: A clustering approach. *Expert Syst. Appl.* **2020**, *140*, 112896. [CrossRef]
15. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
16. Wang, H.; Li, M.; Yue, X. IncLSTM: Incremental Ensemble LSTM Model towards Time Series Data. *Comput. Electr. Eng.* **2021**, *92*, 107156. [CrossRef]
17. Abdel-Nasser, M.; Mahmoud, K. Accurate photovoltaic power forecasting models using deep LSTM-RNN. *Neural Comput. Appl.* **2019**, *31*, 2727–2740. [CrossRef]
18. Aslam, M.; Lee, J.M.; Kim, H.S.; Lee, S.J.; Hong, S. Deep learning models for long-term solar radiation forecasting considering microgrid installation: A comparative study. *Energies* **2020**, *13*, 147. [CrossRef]
19. Akhter, M.N.; Mekhilef, S.; Mokhlis, H.; Almohaimeed, Z.M.; Muhammad, M.A.; Khairuddin, A.S.M.; Akram, R.; Hussain, M.M. An Hour-Ahead PV Power Forecasting Method Based on an RNN-LSTM Model for Three Different PV Plants. *Energies* **2022**, *15*, 2243. [CrossRef]
20. Qing, X.; Niu, Y. Hourly day-ahead solar irradiance prediction using weather forecasts by LSTM. *Energy* **2018**, *148*, 461–468. [CrossRef]
21. Ali-Ou-Salah, H.; Oukarfi, B.; Bahani, K.; Moujabbir, M. A New Hybrid Model for Hourly Solar Radiation Forecasting Using Daily Classification Technique and Machine Learning Algorithms. *Math. Probl. Eng.* **2021**, *2021*, 6692626. [CrossRef]
22. Yagli, G.M.; Yang, D.; Srinivasan, D. Automatic hourly solar forecasting using machine learning models. *Renew. Sustain. Energy Rev.* **2019**, *105*, 487–498. [CrossRef]
23. McCandless, T.; Haupt, S.; Young, G. A regime-dependent artificial neural network technique for short-range solar irradiance forecasting. *Renew. Energy* **2016**, *89*, 351–359. [CrossRef]
24. Yu, Y.; Cao, J.; Zhu, J. An LSTM short-term solar irradiance forecasting under complicated weather conditions. *IEEE Access* **2019**, *7*, 145651–145666. [CrossRef]
25. Voyant, C.; Notton, G.; Kalogirou, S.; Nivet, M.L.; Paoli, C.; Motte, F.; Fouilloy, A. Machine learning methods for solar radiation forecasting: A review. *Renew. Energy* **2017**, *105*, 569–582. [CrossRef]
26. Berndt, D.J.; Clifford, J. Using dynamic time warping to find patterns in time series. In Proceedings of the KDD Workshop, Seattle, WA, USA, 31 July 1994; Volume 10, pp. 359–370.
27. Park, H.S.; Jun, C.H. A simple and fast algorithm for K-medoids clustering. *Expert Syst. Appl.* **2009**, *36*, 3336–3341. [CrossRef]
28. Sharma, N.; Mangla, M.; Yadav, S.; Goyal, N.; Singh, A.; Verma, S.; Saber, T. A sequential ensemble model for photovoltaic power forecasting. *Comput. Electr. Eng.* **2021**, *96*, 107484. [CrossRef]
29. Greff, K.; Srivastava, R.K.; Koutník, J.; Steunebrink, B.R.; Schmidhuber, J. LSTM: A search space odyssey. *IEEE Trans. Neural Netw. Learn. Syst.* **2016**, *28*, 2222–2232. [CrossRef]
30. Gelaro, R.; McCarty, W.; Suárez, M.J.; Todling, R.; Molod, A.; Takacs, L.; Randles, C.A.; Darmenov, A.; Bosilovich, M.G.; Reichle, R.; et al. The modern-era retrospective analysis for research and applications, version 2 (MERRA-2). *J. Clim.* **2017**, *30*, 5419–5454. [CrossRef]
31. Marion, B.; Kroposki, B.; Emery, K.; Del Cueto, J.; Myers, D.; Osterwald, C. *Validation of a Photovoltaic Module Energy Ratings Procedure at NREL*; Technical Report; National Renewable Energy Lab.: Golden, CO, USA, 1999. [CrossRef]
32. Müller, M. Dynamic time warping. In *Information Retrieval for Music and Motion*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 69–84. [CrossRef]
33. Patro, S.; Sahu, K.K. Normalization: A preprocessing stage. *arXiv* **2015**, arXiv:1503.0646.
34. Krishna, T.S.; Babu, A.Y.; Kumar, R.K. Determination of optimal clusters for a Non-hierarchical clustering paradigm K-Means algorithm. In *Proceedings of International Conference on Computational Intelligence and Data Engineering*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 301–316. [CrossRef]
35. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the International Conference on Machine Learning, Lille, France, 7–9 July 2015; pp. 448–456.
36. Cortes, C.; Mohri, M.; Rostamizadeh, A. L2 regularization for learning kernels. *arXiv* **2012**, arXiv:1205.2653.
37. Glorot, X.; Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, Sardinia, Italy, 13–15 May 2010; pp. 249–256.
38. Bergstra, J.; Bardenet, R.; Bengio, Y.; Kégl, B. Algorithms for hyper-parameter optimization. In Proceedings of the 25th Annual Conference on Neural Information Processing Systems (NIPS 2011), Granada, Spain, 12–17 December 2011, Volume 24.
39. Shcherbakov, M.V.; Brebels, A.; Shcherbakova, N.L.; Tyukov, A.P.; Janovsky, T.A.; Kamaev, V.A. A survey of forecast error measures. *World Appl. Sci. J.* **2013**, *24*, 171–176. [CrossRef]
40. Chai, T.; Draxler, R.R. Root mean square error (RMSE) or mean absolute error (MAE)?—Arguments against avoiding RMSE in the literature. *Geosci. Model Dev.* **2014**, *7*, 1247–1250. [CrossRef]
41. Zang, H.; Liu, L.; Sun, L.; Cheng, L.; Wei, Z.; Sun, G. Short-term global horizontal irradiance forecasting based on a hybrid CNN-LSTM model with spatiotemporal correlations. *Renew. Energy* **2020**, *160*, 26–41. [CrossRef]

sustainability

MDPI

Article

Analysis of the Integration of Drift Detection Methods in Learning Algorithms for Electrical Consumption Forecasting in Smart Buildings

Deyslen Mariano-Hernández [1,2,*], Luis Hernández-Callejo [2,*], Martín Solís [3], Angel Zorita-Lamadrid [4], Oscar Duque-Pérez [4], Luis Gonzalez-Morales [5], Felix Santos García [6], Alvaro Jaramillo-Duque [7], Adalberto Ospino-Castro [8], Victor Alonso-Gómez [9] and Hugo J. Bello [10]

[1] Área de Ingeniería, Instituto Tecnológico de Santo Domingo, Santo Domingo 10602, Dominican Republic
[2] ADIRE-ITAP, Departamento Ingeniería Agrícola y Forestal, Universidad de Valladolid, 42004 Soria, Spain
[3] Tecnológico de Costa Rica, Cartago 30101, Costa Rica; marsolis@itcr.ac.cr
[4] ADIRE-ITAP, Departamento de Ingeniería Eléctrica, Universidad de Valladolid, 47002 Valladolid, Spain; zorita@eii.uva.es (A.Z.-L.); oscar.duque@eii.uva.es (O.D.-P.)
[5] Departamento de Ingeniería Eléctrica, Electrónica y Telecomunicaciones–DEET, Facultad de Ingeniería, Universidad de Cuenca, Cuenca 010107, Ecuador; luis.gonzalez@ucuenca.edu.ec
[6] Área de Ciencias Básicas y Ambientales, Instituto Tecnológico de Santo Domingo, Santo Domingo 10602, Dominican Republic; felix.santos@intec.edu.do
[7] GIMEL, Departamento de Ingeniería Eléctrica, Universidad de Antioquia, Medellín 050010, Colombia; alvaro.jaramillod@udea.edu.co
[8] Facultad de Ingeniería, Universidad de la Costa, Barranquilla 080002, Colombia; aospino8@cuc.edu.co
[9] Departamento de Física, Universidad de Valladolid, 47011 Valladolid, Spain; victor.alonso.gomez@uva.es
[10] Departamento de Matemática Aplicada, Universidad de Valladolid, 47002 Valladolid, Spain; hugojose.bello@uva.es
* Correspondence: deyslen.mariano@intec.edu.do (D.M.-H.); luis.hernandez.callejo@uva.es (L.H.-C.); Tel.: +1-809-949-1227 (D.M.-H.); +34-975-129-418 (L.H.-C.)

Citation: Mariano-Hernández, D.; Hernández-Callejo, L.; Solís, M.; Zorita-Lamadrid, A.; Duque-Pérez, O.; Gonzalez-Morales, L.; García, F.S.; Jaramillo-Duque, A.; Ospino-Castro, A.; Alonso-Gómez, V.; et al. Analysis of the Integration of Drift Detection Methods in Learning Algorithms for Electrical Consumption Forecasting in Smart Buildings. *Sustainability* **2022**, *14*, 5857. https://doi.org/10.3390/su14105857

Academic Editor: Antonio Caggiano

Received: 6 April 2022
Accepted: 10 May 2022
Published: 12 May 2022

Abstract: Buildings are currently among the largest consumers of electrical energy with considerable increases in CO_2 emissions in recent years. Although there have been notable advances in energy efficiency, buildings still have great untapped savings potential. Within demand-side management, some tools have helped improve electricity consumption, such as energy forecast models. However, because most forecasting models are not focused on updating based on the changing nature of buildings, they do not help exploit the savings potential of buildings. Considering the aforementioned, the objective of this article is to analyze the integration of methods that can help forecasting models to better adapt to the changes that occur in the behavior of buildings, ensuring that these can be used as tools to enhance savings in buildings. For this study, active and passive change detection methods were considered to be integrators in the decision tree and deep learning models. The results show that constant retraining for the decision tree models, integrating change detection methods, helped them to better adapt to changes in the whole building's electrical consumption. However, for deep learning models, this was not the case, as constant retraining with small volumes of data only worsened their performance. These results may lead to the option of using tree decision models in buildings where electricity consumption is constantly changing.

Keywords: drift detection; electrical consumption forecasting; energy forecasting; machine learning; smart buildings

1. Introduction

Buildings presently produce up to 40% of worldwide energy consumption and 30% of carbon dioxide emissions, numbers which are constantly increasing due to urbanization [1]. Additionally, considering the long life expectancy of buildings, it is assessed that 85–95% of buildings that exist today will still be utilized in 2050 [2]. Hence, changes in energy utilization on buildings are inclined to intensely affect current society, including major economic and environmental changes such as climate change and global warming [3,4]. Buildings are becoming substantially more complex and sophisticated. They integrate conventional energy services systems, on-site energy generation systems, and charging systems [5]. For this reason, energy management is becoming fundamental for buildings around the world, and energy forecasting is essential as an initial step to establish an energy management system [6]. The forecasting of building energy utilization supports smart building performance through low energy and control procedures [7].

In recent times, because of their important application in various fields including electric energy consumption in buildings, data-driven models such as machine- and deep learning-based approaches have become exceptionally well known [8] and are being utilized to improve forecast accuracy [9]. In real life, electrical consumption forecasting models should regularly be made online in real-time. An online setting brings extra challenges since there could be an anticipation of changes to the information distribution over the long haul [10]. However, traditional electric energy forecasting models are normally trained once and not re-trained again with new data, thus missing out on the new information that new data can provide [11]. When this situation happens, it can lead to incorrect forecasting [12].

Recognizing change points and incorporating these uncertain change points into electric energy forecasting models is one of the most difficult tasks [13]. The unexpected changes in the data distribution over time, are known as concept drift [14]. Concept drift has been perceived as the root cause of decreased effectiveness in data-driven decision support systems [15]. Based on how the data change, concept drift can be separated into different kinds: sudden, gradual, recurring, and incremental [16]. Sudden drift happens when the data change quickly and without variation. Whenever the data begin changing in class distribution, this is defined as gradual drift. Recurring drifts happen when the data change for a moment and then return sooner or later. Incremental drift occurs when the data continuously change over the long run [17].

To address those different situations in forecasting models, two main strategies have been used: active and passive methods. For active methods, a model is equipped with a change detection strategy and re-trained when a trigger has been flagged. Nonetheless, in passive methods, algorithms are re-trained at regular intervals regardless of whether a change has occurred or not [18]. There has been a very important effort investigating concept drift in regression tasks (see Table 1) that have focused on load forecasting in houses [19,20], energy consumption in smart grids [21], electricity supply and demand [22], total reactive power [23], energy production for a wind farm [24], power generation in a photovoltaic plant [25], and electricity price [26,27]. However, there have not been many works in real cases where concept drift techniques are used to maintain or improve the results of machine learning techniques in smart buildings. Therefore, this paper's objective is to provide a novel analysis of the integration of drift detection methods in decision trees and deep learning algorithms for whole building electricity consumption forecasting in smart buildings.

Given the above, the main contributions of this paper in this field of research could be summarized as follows:

- Integration of drift detection methods to a multi-step forecasting strategy that forecasts the next 24 h from any hour of the day.
- An analysis of the integration of drift detection methods in decision trees and deep learning algorithms for forecasting the electricity consumption of the entire building.
- Comparison analysis between active and passive drift detection methods for building electricity consumption forecasting in smart buildings.

Table 1. Summary of literature review, their contributions, and their limitations.

Ref.	Contributions	Limitations
[19]	A proposed approach for load forecasting where the model is persistently refreshed as new information shows up.	The tuning module could utilize a more modern approach to following precision patterns.
[20]	Proposed online ensemble methods for load forecasting under the concept of drift.	The research did not evaluate concept drift or the performance during the drifting duration.
[21]	Proposed a model that helps to identify anomalies using paired learners.	Delay of a few hours between the anomaly and its detection.
[22]	Analyzed different drift detection methods for data streams in smart city applications.	Absence of accessible or reusable benchmark datasets in the literature to completely compare the outcomes.
[23]	Proposed an unsupervised drift detection approach capable of analyzing streaming data in a smart grid.	The approach was not compared with a deep learning algorithm that incorporates drift detection methods.
[24]	Suggested a drift detection approach based on the analysis of the change caused by new information using extreme learning machines.	Need for an automatic setting of the parameters for the proposed drift detection approach.
[25]	Implemented a segmentation of time series based on stationarity using drift detection methods.	The approach needs to have previous knowledge about the time series cyclical behaviors.
[26]	Proposed a passive drift detection approach using Robust Soft and Generalized Learning Vector Quantization.	The proposed method was compared with drift detection algorithms without optimized hyperparameters.
[27]	Proposed an improvement for the Robust Soft Learning Vector Quantization algorithm to be used in drift detection.	The proposed approach method performs better in synthetic concept drift streams but not in real-world streams.
[28]	Proposed an approach based on random trees algorithm to deal with changes using drift detection methods.	The proposed approach discards the previous anomaly instead of updating the detection model.

2. Methodology and Approach

The use of drift detection methods is well known, however, the integration of these methods into a multi-step forecasting strategy to predict continuous hourly electricity energy consumption in the entire building turns out to be a novel topic.

Therefore, this section describes data preprocessing, forecasting algorithms, drift detection methods, and performance metrics used in this article. Section 2.1 provides information on how the datasets from the two buildings used to train the learning algorithms were made. Section 2.2 presents the approach and the learning algorithms used to forecast the electrical consumption in buildings. Section 2.3 describes the drift detection methods and their incorporation into the learning algorithms. Section 2.4 explains the metrics used for evaluating the performance of learning algorithms. A summary of the methodology used is shown in Figure 1.

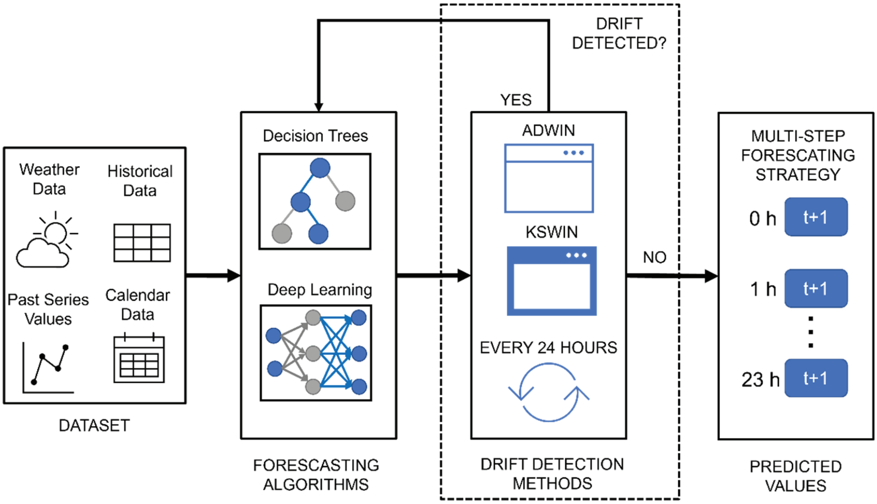


Figure 1. Methodology used for the analysis of the integration of drift detection methods.

2.1. Datasets Construction

For this research, the data from two buildings located on the campus of the University of Valladolid were used. These data were obtained through smart meters installed in each of the buildings at their electrical power transformer stations, which record the active energy consumed (kWh) of the entire building in intervals of 15 min from 2016 to 2019. At the time of analyzing the data, some missing records were found, because these missing records did not exceed 0.5% of the total value of the data and were not found consecutively, a line interpolation technique was applied to complete these missing records. After completing the missing data, since it was desired to forecast the electricity consumption per hour, the data were conditioned to have the consumption per hour for each building.

Based on previous studies [29–33] where it has been proven that the use of weather, calendar variables, and past values data can help improve the training of learning algorithms, these were included in the datasets. To obtain the past values data, the autocorrelation and partial autocorrelation of the energy consumption variable were analyzed, resulting in a significant autocorrelation up to lag 25. For calendar variables, the timestamps of the historical data were used to obtain the variables of the hour, day, month, and year. Additionally, a variable was added to indicate when it is a working day or not, this variable was made based on the annual calendar of the university. The weather variables that were used were those that are related to the comfort of the occupants inside the building, such as relative humidity, precipitation, minimum temperature, average temperature, maximum temperature, heating degree days, cooling degree days, and all-sky surface longwave downward irradiance. The weather data were obtained from the NASA Langley Research Center (LaRC) POWER Project funded through the NASA Earth Science/Applied Science Program (https://power.larc.nasa.gov/, accessed on 16 March 2022).

2.2. Approach and Forecasting Algorithms

For the electricity consumption forecast, a multi-step forecasting strategy was used, which in this case can predict electricity consumption for the next 24 h from one hour. The advantage of this strategy is that it allows electricity consumption forecasting from any hour of the day, the disadvantage is that it is necessary to prepare the dataset with past values data so that this information can be used by the learning algorithms to forecast the multiple hours more accurately.

Based on studies where decision tree [34–37] and deep learning algorithms [38–41] obtained good results in forecasting electrical consumption in buildings, two decision trees, and two deep learning algorithms were selected. From the decision tree algorithms, Random Forest (RF) and eXtreme Gradient Boosting (XGBoost) were selected, while from the deep learning algorithms, Convolutional Neural Network (CNN), and Temporal Convolutional Network (TCN) was chosen. The architectures of the learning algorithms used are shown in Figure 2.

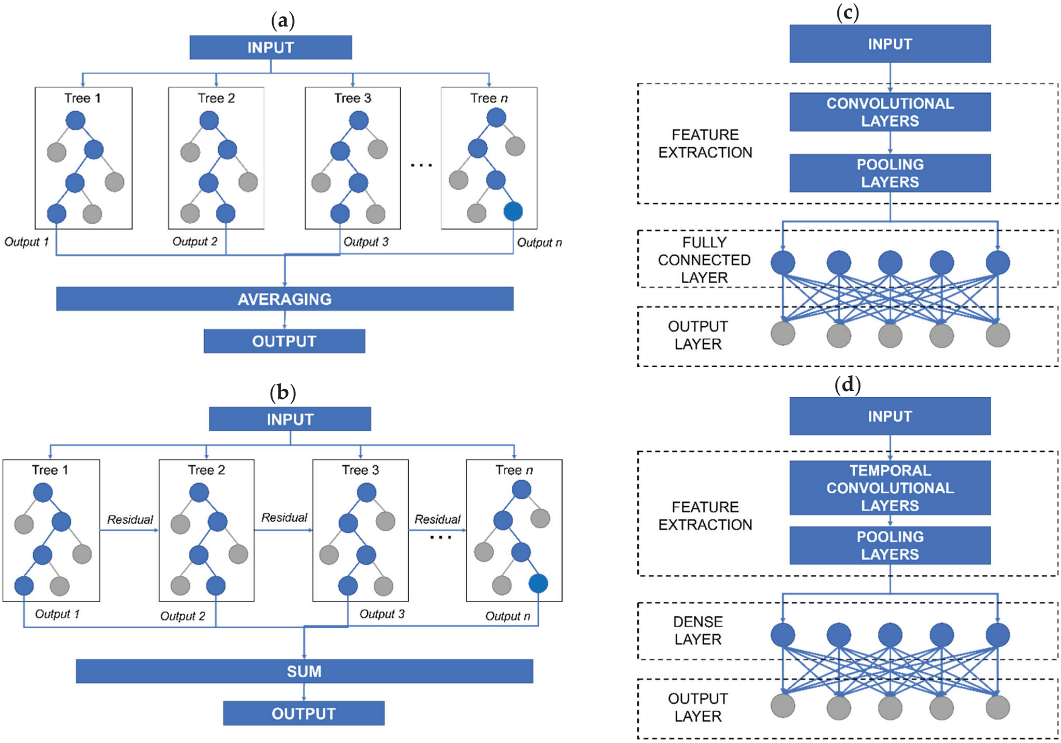


Figure 2. (**a**) Basic RF architecture. (**b**) Basic XGBoost architecture. (**c**) Basic CNN architecture. (**d**) Basic TCN architecture.

The algorithms used were programmed in Python using the Scikit-learn, XGBoost, Keras, and TensorFlow libraries. To obtain the best combination of hyperparameters and architecture for the algorithms, backtesting with sliding windows was used. The backtesting with sliding windows procedure consisted of keeping the same training size and sliding a data window to create five different training tests (see Figure 3). For this case, the data from 2016 to 2017 were used for the training set, while the data from 2018 were used for the validation sample. Once the best architecture and parameters were defined through backtesting, the model was adjusted with data from 2016 to 2018, leaving 2019 as the testing set. The best combinations of parameters obtained in the backtesting process are shown in Table 2. The parameters that do not appear in the table are absent because their default values were used.

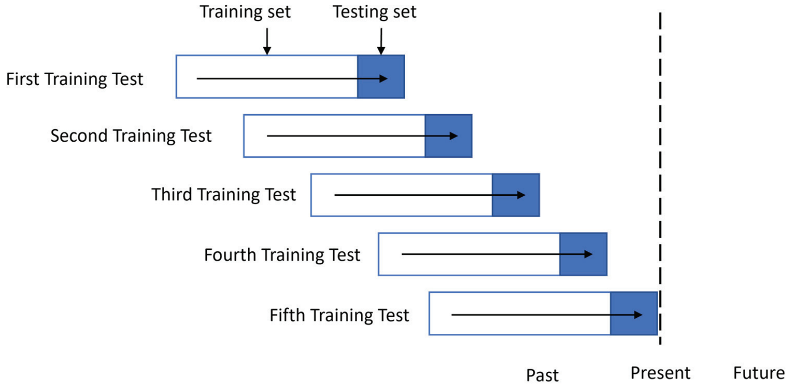


Figure 3. Backtesting with sliding windows procedure.

Table 2. Best combinations of parameters obtained through backtesting procedure.

Algorithms	Hyperparameter
Random Forest	max_depth = 45; n_estimators = 200; min_samples_leaf = 1
eXtreme Gradient Boosting	n_estimators = 50; eta = 0.1; max_depth = 5; colsample_bytree = 0.8; subsample = 0.8; gamma = 1
Convolutional Neural Network	filters = 64; kernel_size = 2; batch size = 1; activation function = linear; optimizer = adam; learning rate = 0.001; maxpooling1D (pool_size = 2); loss function = mean squared error
Temporal Convolutional Network	filters = 200; kernel_size = 4; batch size = 1; dilations = [1, 2, 4, 8, 16, 32]; activation function = linear

2.3. Drift Detection Methods

Since the selected algorithms are not capable of detecting changes in the data distribution, two well-known active drift detection methods (DDM), Adaptive Window (ADWIN) and Kolmogorov–Smirnov Window (KSWIN) [28] were incorporated into them. These methods were selected because the training uses the latest batch of data with the latest training instances and the size of the window is generally determined by the user.

ADWIN accurately keeps a variable-length window of late values; to such an extent that it holds that there has not been a change in the data distribution. This window is additionally isolated into two sub-windows (W_0, W_1) used to decide whether a change has occurred. ADWIN contrasts the median of W_0 and W_1 to affirm that they coincide with a similar distribution. Concept drift is identified assuming the distribution correspondence does not hold anymore. After recognizing a drift, W_0 is changed by W_1 and a new W_1 is introduced. ADWIN utilizes a certainty value $\delta \in (0, 1)$ to decide whether the two sub-windows coincide with a similar dispersion [42].

KSWIN is a drift detection method based on the Kolmogorov–Smirnov (KS) measurable test. KS-test is a measurable test without really any suspicion of basic information appropriation. KSWIN keeps a sliding window Ψ of fixed size n (window_size). The last r (stat_size) tests of Ψ are accepted to address the last idea considered as R. From the main $n - r$ examples of Ψ, r tests are consistently drawn, addressing an approximated last concept W. The KS-test is performed on the windows R also W, of a similar size. KS-test looks at the distance of the observational aggregate data distribution $dist(R, W)$ [27].

A sudden change is distinguished by KSWIN if:

$$dist(R, W) > \sqrt{-\frac{ln\alpha}{r}} \tag{1}$$

where α is the probability for the test statistic of the KS-test, and r is the size of the statistic window.

The reason for using methods based on window size was because the training utilizes the last batch of data with the last training set. The window of fixed size approach is the least complex rule and the window size is usually decided by the user. By having data on the time size of the change, a window of the fixed size approach is a valuable decision [11].

2.4. Performace Metrics

To analyze the integration of the DDM, in addition to using active methods, it was proposed to use a passive method, which consisted of retraining the algorithms every 24 h regardless of whether there was a change in the data distribution. These methods were compared in each of the algorithms using performance metrics, mean absolute percentage error (MAPE), mean absolute error (MAE), root mean square error (RMSE), and coefficient of determination (R^2) were used.

MAPE shows the measure of the precision of the estimated values comparative with the real values (in a percentage) [43], which is determined according to Equation (2).

$$\text{MAPE} \ = \frac{\sum_{i\,=\,1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right|}{n} \times 100\% \tag{2}$$

MAE is utilized to assess how close estimates or expectations are to the real results. It is determined by averaging the absolute differences between the expected values and the real values [44], as shown in Equation (3).

$$\text{MAE} \ = \frac{\sum_{i\,=\,1}^{n}|y_i - \hat{y}_i|}{n} \tag{3}$$

RMSE evaluates the differences between the real values and estimated values [45], which is determined according to Equation (4):

$$\text{RMSE} \ = \sqrt{\frac{\sum_{i\,=\,1}^{n}(y_i - \hat{y}_i)^2}{n}} \tag{4}$$

R^2 is a statistical measure of the variance between estimated values acquired by the model and real values (level of direct relationship among anticipated and estimated values) [46], which is determined according to Equation (5).

$$\text{R}^2 \ = 1 - \frac{\sum_{i\,=\,1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i\,=\,1}^{n}(y_i - \overline{y}_i)^2} \tag{5}$$

where y_i is the expected value, $\hat{y}_i$ is the real value, $\overline{y}_i$ is the average value, and n is the total number of estimations.

The reason why these metrics were chosen was to have an overview of the performance of the models. In the case of the MAPE, it was chosen because it is easy to understand since it presents percentage values, but due to its limitations, it was decided to accompany it with the MAE, which shows how much inaccuracy is expected from the forecast on average, helping to determine which models are better. However, because the MAE can have difficulty distinguishing large from small errors, it was combined with the RMSE to be on the safe side. As for R^2, it was selected to know how the data fit the models.

3. Experimentation Setup

Two buildings with a continental Mediterranean climate were selected for testing. These buildings have a lighting and air conditioning control system, as well as an energy monitoring system to provide a balance between the comfort of the occupants and the consumption of electrical energy. The first building corresponds to the Faculty of Science of the University of Valladolid located at coordinates 41.663411°, −4.705539°, which is dedicated to administrative offices, while the second building corresponds to the Faculty of Economics located at coordinates 41.658586°, −4.710667°, which is dedicated to teaching activities. These buildings were selected due to their different behavior in electricity consumption during the selected years. In case of Building 1, it has had changes in consumption only in specific periods, while Building 2 has had a decrease in energy consumption gradually each year because energy efficiency improvements were made, and solar panels were integrated into the building (see Figure 4). The energy source used for Building 1 comes from the electrical grid, while for Building 2, the energy source comes from the electrical grid and photovoltaic panels.

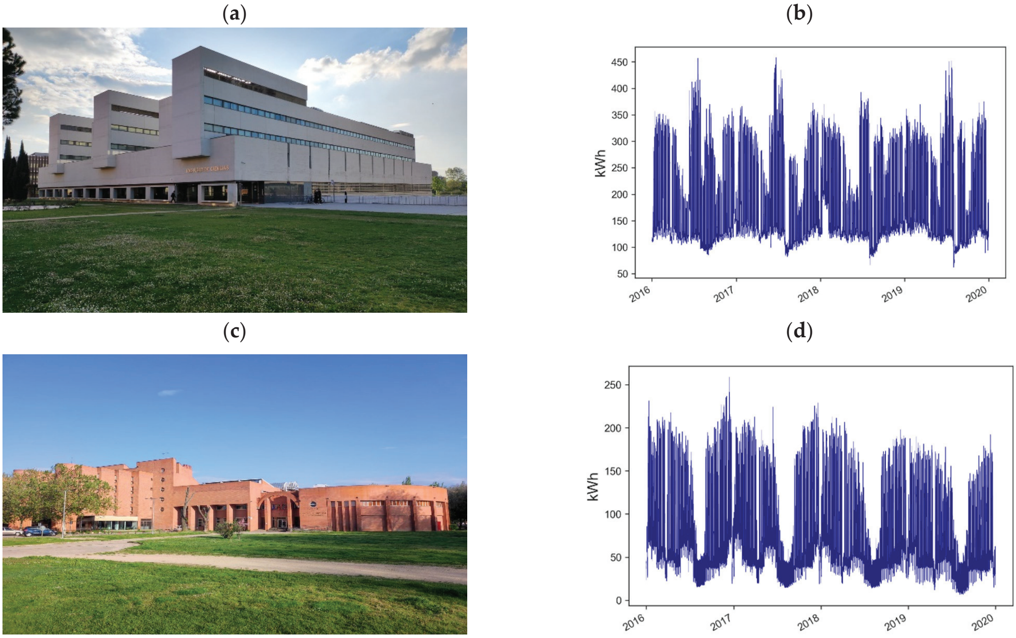


Figure 4. (**a**) View of Building 1. (**b**) Hourly electricity consumption for Building 1. (**c**) View of Building 2. (**d**) Hourly electricity consumption for Building 2.

The records of the electrical consumption that were used to test the proposed method were from 2016 to 2019. For the training stage, the years 2016 to 2018 were used, while for the test stage the year 2019 was used. To evaluate the learning algorithms with the DDM, two Python scripts were developed, one for the decision tree algorithms and the other for the deep learning algorithms. Two functions were created in the scripts, the first for updating the algorithms with a passive method and the second for updating with the active methods. In the passive method, the algorithms were retrained every 24 h over a period of one year, while in the active methods, the algorithms were retrained every time a change in the data distribution was detected for the same period. It should be noted that to apply the ADWIN and KSWIN methods to the models, the scikit-multiflow library was used.

For this study, the active methods take the first three years of the dataset as a reference and compare it with the new data. If a change is detected, the model is retrained. The way the model is retrained depends on the type. For decision trees, the model is built from scratch while for deep learning, transfer learning was used, to reduce training time. The transfer learning was carried out by freezing the layers except for the last two, which were updated every time the detection method indicated that it was required to retrain the model.

4. Results and Discussion

4.1. Decision Trees Models Evaluation

After integrating the active and passive DDM with decision tree models, the results obtained for Building 1 (see Table 3) show that the models with DDM obtained better performance for both algorithms than the model without DDM. Likewise, it is highlighted that the passive method used for training presents better results than the active methods.

Table 3. Decision tree model results for Building 1.

		RF				XGBOOST			
Method	**ND**	**MAPE (%)**	**MAE (kWh)**	**RMSE (kWh)**	**R^2**	**MAPE (%)**	**MAE (kWh)**	**RMSE (kWh)**	**R^2**
Wo/DDM	n/a	9.23	16.24	29.48	0.827	8.81	15.01	27.16	0.853
ADWIN	10	8.95	15.68	28.61	0.837	8.69	14.84	26.89	0.856
KSWIN	111	8.53	14.98	27.78	0.846	8.56	14.63	26.62	0.859
24 H	365	8.46	14.83	27.59	0.848	8.51	14.57	26.59	0.859

Wo/DDM = without drift detection method, ND = numbers of detections, n/a = not applicable.

Table 4 shows the results in Building 2 where it is observed that, like Building 1, the models with DDM present better performance for both algorithms than the model without DDM. However, if we focus on the RMSE and R^2 metrics, the passive method does not clearly show that it obtains better performance than the KSWIN method in the case of XGBoost.

Table 4. Decision tree model results for Building 2.

		RF				XGBOOST			
Method	**ND**	**MAPE (%)**	**MAE (kWh)**	**RMSE (kWh)**	**R^2**	**MAPE (%)**	**MAE (kWh)**	**RMSE (kWh)**	**R^2**
Wo/DDM	n/a	19.47	9.08	14.95	0.861	17.78	8.17	13.97	0.878
ADWIN	15	17.61	8.51	14.42	0.870	16.96	7.94	13.73	0.882
KSWIN	108	16.44	7.91	13.89	0.880	16.68	7.78	13.54	0.886
24H	365	16.14	7.83	13.87	0.880	16.55	7.77	13.57	0.885

Wo/DDM = without drift detection method, ND = numbers of detections, n/a = not applicable.

The findings show that the decision tree algorithms certainly benefited from the integration of the DDM, showing improvement in the results. When analyzing the detection number, which corresponds to the number of sudden changes detected by the DDM, it could be concluded that for active methods a higher number of detections, which in our case would be the same as the retraining number, could lead to better results. However, when we compare the passive method with the KSWIN method, it can be seen that the results are very approximate but in the case of the KSWIN method, the number of retraining is less than 50% of the retraining performed by the passive method.

Even though the passive method has shown better performance, it cannot be affirmed with certainty that it would be better to use it since it assumes that the data distribution undergoes daily changes, which would not necessarily be true since it could be the case that the behavior of the occupants or energy savings measures causes changes in electricity consumption in periods greater than 24 h and the model is being retrained at a time when it is not necessary.

4.2. Deep Learning Models Evaluation

After integrating the active and passive DDM with deep learning models, the results obtained for Building 1 (see Table 5) show that for the TCN, the model without DDM obtains better performance than the models with DDM. However, in the case of CNN, it is observed that the model without DDM obtains better performance than the active methods but not better than the passive method if we focus on the RMSE and R^2 metrics.

Table 5. Deep learning model results for Building 1.

		CNN				TCN			
Method	**ND**	**MAPE (%)**	**MAE (kWh)**	**RMSE (kWh)**	**R^2**	**MAPE (%)**	**MAE (kWh)**	**RMSE (kWh)**	**R^2**
Wo/DDM	n/a	9.40	17.14	30.78	0.811	9.03	15.88	29.42	0.828
ADWIN	10	12.51	20.74	32.21	0.793	10.89	18.9	33.28	0.780
KSWIN	111	12.35	20.45	31.96	0.797	10.11	17.68	32.01	0.796
24H	365	10.93	18.56	30.75	0.812	10.15	17.41	30.97	0.809

Wo/DDM = without drift detection method, ND = numbers of detections, n/a = not applicable.

Table 6 shows the results in Building 2 where it is observed that, like Building 1, the TCN obtains better performance without DDM. However, for CNN, if we focus on the RMSE and R^2 metrics, the KSWIN method obtained better performance than the model without DDM.

Table 6. Deep learning model results for Building 2.

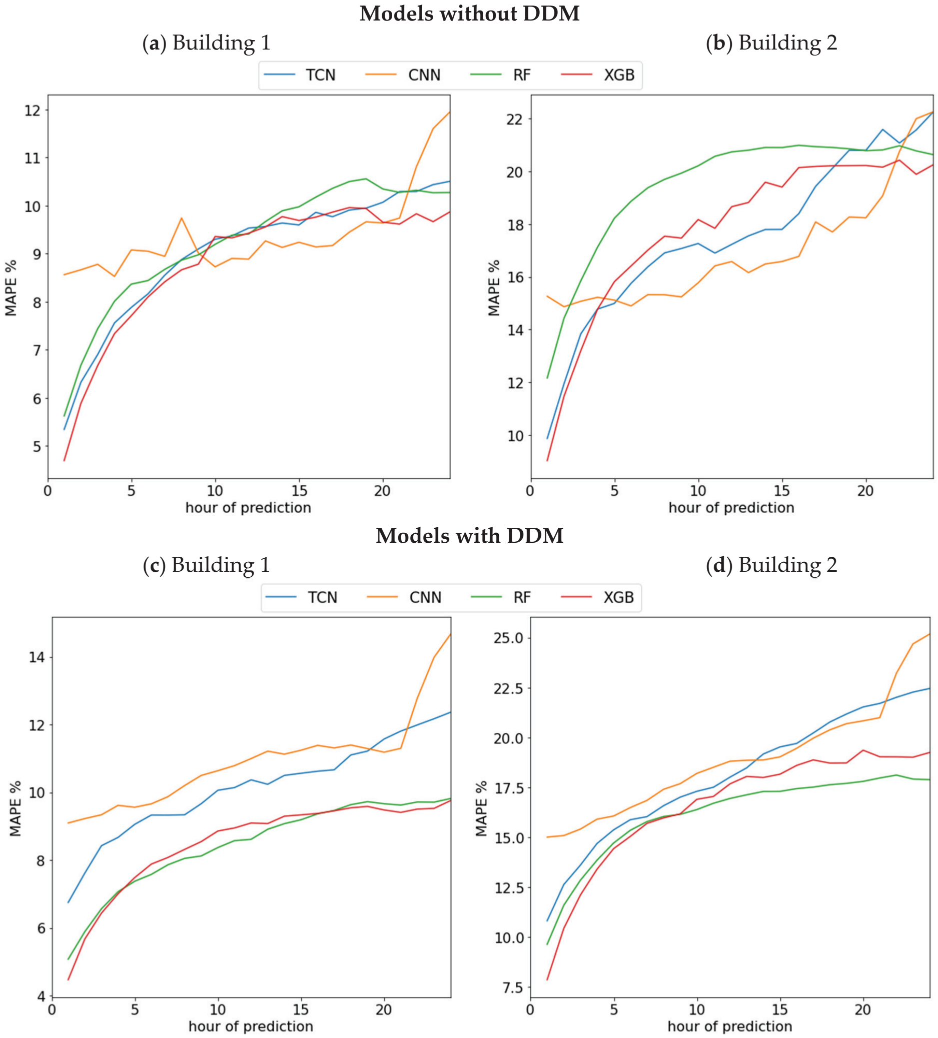
		CNN				TCN			
Method	**ND**	**MAPE (%)**	**MAE (kWh)**	**RMSE (kWh)**	**R^2**	**MAPE (%)**	**MAE (kWh)**	**RMSE (kWh)**	**R^2**
Wo/DDM	n/a	16.97	9.62	17.41	0.811	17.58	8.98	15.85	0.843
ADWIN	15	21.49	11.39	18.57	0.785	19.18	9.66	17.01	0.819
KSWIN	108	19.67	10.18	16.95	0.821	17.38	8.93	16.24	0.835
24H	365	18.89	10.10	17.14	0.817	18.09	9.17	16.39	0.832

Wo/DDM = without drift detection method, ND = numbers of detections, n/a = not applicable.

For the deep learning models, the findings show that the ADWIN method, which performs the smallest amount of retraining, presents the worst performance of the active methods, while the passive method presents the better performance. However, in general, the model without DDM obtains better performance except in the RMSE and R^2 metrics for CNN with DDM. Which would suggest that the type of change in the data distribution is not abrupt enough to require the retraining of the deep learning models.

This behavior in the performance of the deep learning models would make us question the need for retraining in this case, but if we compare the outcomes of the decision tree models versus the deep learning models, it can be seen that, in the case of Building 2 where the deep learning models without DDM have better performance than the decision tree models without DDM when DDM is applied, decision tree models perform better than deep learning models without DDM.

Figure 5 shows the average error of the forecast algorithms by hours of the electrical consumption of the entire building from the first hour that is forecast for each algorithm. As can be seen, when we analyze the average error per hour in each of the buildings, we realize that the decision tree models, when integrating the DDMs, improve their performance in each of the hours, however, this is not the case for deep learning models.



Models without DDM

(**a**) Building 1

(**b**) Building 2

Models with DDM

(**c**) Building 1

(**d**) Building 2

Figure 5. (**a**) Performance of forecasting algorithms without DDM by hours in Building 1. (**b**) Performance of forecasting algorithms without DDM by hours in Building 2. (**c**) Performance of forecasting algorithms with DDM by hours in Building 1. (**d**) Performance of forecasting algorithms with DDM by hours in Building 2.

The results show that the proposed method can be applied to maintain or even improve the performance of learning algorithms in situations where there are constant changes in the behavior of electrical consumption in buildings. A limitation is the drift detection methods that were integrated. In the case of ADWIN, only the confidence value parameter was allowed to be modified, while in the case of KSWIN an inappropriate modification of the values of the size of windows would cause the method to not detect sudden changes in the distribution data.

5. Conclusions

In this paper, the integration of drift detection methods is analyzed in models for electricity consumption forecasting in buildings so that these models can adapt to the changing behavior that has been occurring in buildings due to energy-saving measures. Two active methods and one passive method were proposed to be integrated with the decision tree and deep learning models to know when the models should be retrained according to changes in the data distribution. The passive method consisted of retraining the models every 24 h assuming that the models should be constantly updated, while the active methods were ADWIN and KSWIN, which are based on a variable-length window approach.

The main conclusion that can be learned from this study, after analyzing the results, is that in the case of decision tree models, the incorporation of DDM not only allows them to keep up to date with changes in the data distribution but also improves their accuracy. Being the best case RF, without DDM obtained a MAPE of 9.23% for Building 1 and 19.47% for Building 2 while with the passive DDM it obtained a MAPE of 8.46% for building 1 and 16.14% for Building 2. However, in the case of deep learning models, the incorporation of DDM did not turn out to be as favorable as decision tree models. With the CNN being the worst case, without DDM an MAPE of 9.40% was obtained for Building 1 and 16.97% for Building 2 while with the passive DDM it obtained an MAPE of 10.93% for building 1 and 18.89% for Building 2. We can deduce from this that in the case of deep learning models, constantly updating them with small volumes of data would only worsen their performance. In cases such as Building 2 with sudden changes in load curves due to improvements, the model becomes inefficient, because deep learning models cannot adapt with small data to constant changes in the short term.

Considering the results obtained in the deep learning models, for future lines of research it would be necessary to focus on how it would be possible to adapt the deep learning models to constant changes within the electrical consumption forecasting in buildings to avoid model obsolescence.

Author Contributions: Conceptualization, D.M.-H., M.S. and L.H.-C.; methodology, D.M.-H. and L.H.-C.; software, D.M.-H., M.S. and V.A.-G.; validation, A.Z.-L., O.D.-P., L.G.-M. and F.S.G.; data analysis, D.M.-H., M.S., V.A.-G. and H.J.B.; investigation, D.M.-H.; data curation, D.M.-H. and M.S.; writing—original draft preparation, D.M.-H.; writing—review and editing, A.Z.-L., O.D.-P., A.O.-C. and A.J.-D.; visualization, D.M.-H. and M.S.; supervision, L.H.-C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors acknowledge the University of Valladolid and the Instituto Tecnológico de Santo Domingo for their support in this research, which is the result of a co-supervised doctoral thesis.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. IEA Tracking Buildings. 2021. Available online: https://www.iea.org/reports/tracking-buildings-2021 (accessed on 16 March 2022).
2. Cholewa, T.; Siuta-Olcha, A.; Smolarz, A.; Muryjas, P.; Wolszczak, P.; Guz, Ł.; Bocian, M.; Balaras, C.A. An easy and widely applicable forecast control for heating systems in existing and new buildings: First field experiences. *J. Clean. Prod.* **2022**, *352*, 131605. [CrossRef]
3. Devagiri, V.M.; Boeva, V.; Abghari, S.; Basiri, F.; Lavesson, N. Multi-view data analysis techniques for monitoring smart building systems. *Sensors* **2021**, *21*, 6775. [CrossRef] [PubMed]

4. Izidio, D.M.; de Mattos Neto, P.S.; Barbosa, L.; de Oliveira, J.F.; Marinho, M.H.D.N.; Rissi, G.F. Evolutionary hybrid system for energy consumption forecasting for smart meters. *Energies* **2021**, *14*, 1794. [CrossRef]
5. Hong, T.; Wang, Z.; Luo, X.; Zhang, W. State-of-the-art on research and applications of machine learning in the building life cycle. *Energy Build.* **2020**, *212*, 109831. [CrossRef]
6. Kim, J.Y.; Cho, S.B. Electric energy consumption prediction by deep learning with state explainable autoencoder. *Energies* **2019**, *12*, 739. [CrossRef]
7. Zeng, A.; Ho, H.; Yu, Y. Prediction of building electricity usage using Gaussian Process Regression. *J. Build. Eng.* **2020**, *28*, 101054. [CrossRef]
8. Xu, W.; Peng, H.; Zeng, X.; Zhou, F.; Tian, X.; Peng, X. A hybrid modelling method for time series forecasting based on a linear regression model and deep learning. *Appl. Intell.* **2019**, *49*, 3002–3015. [CrossRef]
9. Cholewa, T.; Siuta-Olcha, A.; Smolarz, A.; Muryjas, P.; Wolszczak, P.; Anasiewicz, R.; Balaras, C.A. A simple building energy model in form of an equivalent outdoor temperature. *Energy Build.* **2021**, *236*, 110766. [CrossRef]
10. Žliobaitė, I.; Pechenizkiy, M.; Gama, J. *An Overview of Concept Drift Applications BT—Big Data Analysis: New Algorithms for a New Society*; Japkowicz, N., Stefanowski, J., Eds.; Springer International Publishing: Cham, Switzerland, 2016; pp. 91–114, ISBN 978-3-319-26989-4.
11. Iwashita, A.S.; Papa, J.P. An Overview on Concept Drift Learning. *IEEE Access* **2019**, *7*, 1532–1547. [CrossRef]
12. Baier, L.; Kühl, N.; Satzger, G.; Hofmann, M.; Mohr, M. Handling concept drifts in regression problems—the error intersection approach. In *WI2020 Zentrale Tracks*; GITO Verlag: Berlin, Germany, 2020; pp. 210–224.
13. Kahraman, A.; Kantardzic, M.; Kahraman, M.; Kotan, M. A data-driven multi-regime approach for predicting energy consumption. *Energies* **2021**, *14*, 6763. [CrossRef]
14. Webb, G.I.; Lee, L.K.; Goethals, B.; Petitjean, F. Analyzing concept drift and shift from sample data. *Data Min. Knowl. Discov.* **2018**, *32*, 1179–1199. [CrossRef]
15. Lu, J.; Liu, A.; Dong, F.; Gu, F.; Gama, J.; Zhang, G. Learning under Concept Drift: A Review. *IEEE Trans. Knowl. Data Eng.* **2019**, *31*, 2346–2363. [CrossRef]
16. Brzezinski, D.; Stefanowski, J. Reacting to different types of concept drift: The accuracy updated ensemble algorithm. *IEEE Trans. Neural Netw. Learn. Syst.* **2014**, *25*, 81–94. [CrossRef]
17. Wadewale, K.; Desai, S.; Tennant, M.; Stahl, F.; Rana, O.; Gomes, J.B.; Thakre, A.A.; Redes, E.M.; Padmalatha, E.; Rani, P.; et al. Survey on Method of Drift Detection and Classification for time varying data set. *Comput. Biol. Med.* **2016**, *32*, 1–7.
18. Khezri, S.; Tanha, J.; Ahmadi, A.; Sharifi, A. A novel semi-supervised ensemble algorithm using a performance-based selection metric to non-stationary data streams. *Neurocomputing* **2021**, *442*, 125–145. [CrossRef]
19. Fekri, M.N.; Patel, H.; Grolinger, K.; Sharma, V. Deep learning for load forecasting with smart meter data: Online Adaptive Recurrent Neural Network. *Appl. Energy* **2021**, *282*, 116177. [CrossRef]
20. Jagait, R.K.; Fekri, M.N.; Grolinger, K.; Mir, S. Load Forecasting Under Concept Drift: Online Ensemble Learning With Recurrent Neural Network and ARIMA. *IEEE Access* **2021**, *9*, 98992–99008. [CrossRef]
21. Fenza, G.; Gallo, M.; Loia, V. Drift-aware methodology for anomaly detection in smart grid. *IEEE Access* **2019**, *7*, 9645–9657. [CrossRef]
22. Mehmood, H.; Kostakos, P.; Cortes, M.; Anagnostopoulos, T.; Pirttikangas, S.; Gilman, E. Concept drift adaptation techniques in distributed environment for real-world data streams. *Smart Cities* **2021**, *4*, 349–371. [CrossRef]
23. Ceci, M.; Corizzo, R.; Japkowicz, N.; Mignone, P.; Pio, G. ECHAD: Embedding-Based Change Detection from Multivariate Time Series in Smart Grids. *IEEE Access* **2020**, *8*, 156053–156066. [CrossRef]
24. Yang, Z.; Al-Dahidi, S.; Baraldi, P.; Zio, E.; Montelatici, L. A Novel Concept Drift Detection Method for Incremental Learning in Nonstationary Environments. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *31*, 309–320. [CrossRef] [PubMed]
25. Silva, R.P.; Zarpelão, B.B.; Cano, A.; Barbon Junior, S. Time series segmentation based on stationarity analysis to improve new samples prediction. *Sensors* **2021**, *21*, 7333. [CrossRef] [PubMed]
26. Heusinger, M.; Raab, C.; Schleif, F.M. Passive concept drift handling via variations of learning vector quantization. *Neural Comput. Appl.* **2022**, *34*, 89–100. [CrossRef]
27. Raab, C.; Heusinger, M.; Schleif, F.M. Reactive Soft Prototype Computing for Concept Drift Streams. *Neurocomputing* **2020**, *416*, 340–351. [CrossRef]
28. Togbe, M.U.; Chabchoub, Y.; Boly, A.; Barry, M.; Chiky, R.; Bahri, M. Anomalies detection using isolation in concept-drifting data streams. *Computers* **2021**, *10*, 13. [CrossRef]
29. Moon, J.; Park, S.; Rho, S.; Hwang, E. A comparative analysis of artificial neural network architectures for building energy consumption forecasting. *Int. J. Distrib. Sens. Netw.* **2019**, *15*, 155014771987761. [CrossRef]
30. Kiprijanovska, I.; Stankoski, S.; Ilievski, I.; Jovanovski, S.; Gams, M.; Gjoreski, H. HousEEC: Day-Ahead Household Electrical Energy Consumption Forecasting Using Deep Learning. *Energies* **2020**, *13*, 2672. [CrossRef]
31. Zor, K.; Çelik, Ö.; Timur, O.; Teke, A. Short-term building electrical energy consumption forecasting by employing gene expression programming and GMDH networks. *Energies* **2020**, *13*, 1102. [CrossRef]
32. Li, Z.; Friedrich, D.; Harrison, G.P. Demand Forecasting for a Mixed-Use Building Using Agent-Schedule Information with a Data-Driven Model. *Energies* **2020**, *13*, 780. [CrossRef]

33. Culaba, A.B.; Del Rosario, A.J.R.; Ubando, A.T.; Chang, J.-S. Machine learning-based energy consumption clustering and forecasting for mixed-use buildings. *Int. J. Energy Res.* **2020**, *44*, 9659–9673. [CrossRef]
34. Wang, Z.; Wang, Y.; Zeng, R.; Srinivasan, R.S.; Ahrentzen, S. Random Forest based hourly building energy prediction. *Energy Build.* **2018**, *171*, 11–25. [CrossRef]
35. Sauer, J.; Mariani, V.C.; dos Santos Coelho, L.; Ribeiro, M.H.D.M.; Rampazzo, M. Extreme gradient boosting model based on improved Jaya optimizer applied to forecasting energy consumption in residential buildings. *Evol. Syst.* **2021**, 1–12. [CrossRef]
36. Bassi, A.; Shenoy, A.; Sharma, A.; Sigurdson, H.; Glossop, C.; Chan, J.H. Building energy consumption forecasting: A comparison of gradient boosting models. In Proceedings of the 12th International Conference on Advances in Information Technology, Bangkok, Thailand, 29 June–1 July 2021. [CrossRef]
37. Mariano-Hernández, D.; Hernández-Callejo, L.; Solís, M.; Zorita-Lamadrid, A.; Duque-Perez, O.; Gonzalez-Morales, L.; Santos-García, F. A Data-Driven Forecasting Strategy to Predict Continuous Hourly Energy Demand in Smart Buildings. *Appl. Sci.* **2021**, *11*, 7886. [CrossRef]
38. Olu-Ajayi, R.; Alaka, H.; Sulaimon, I.; Sunmola, F.; Ajayi, S. Building energy consumption prediction for residential buildings using deep learning and other machine learning techniques. *J. Build. Eng.* **2022**, *45*, 103406. [CrossRef]
39. Lemos, V.H.B.; Almeida, J.D.S.; Paiva, A.C.; Junior, G.B.; Silva, A.C.; Neto, S.M.B.; Lima, A.C.M.; Cipriano, C.L.S.; Fernandes, E.C.; Silva, M.I.A. Temporal convolutional network applied for forecasting individual monthly electric energy consumption. In Proceedings of the 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Toronto, ON, Canada, 11–14 October 2020; pp. 2002–2007.
40. Bendaoud, N.M.M.; Farah, N. Using deep learning for short-term load forecasting. *Neural Comput. Appl.* **2020**, *32*, 15029–15041. [CrossRef]
41. Gao, Y.; Ruan, Y.; Fang, C.; Yin, S. Deep learning and transfer learning models of energy consumption forecasting for a building with poor information data. *Energy Build.* **2020**, *223*, 110156. [CrossRef]
42. Bifet, A.; Gavaldà, R. Learning from time-changing data with adaptive windowing. In Proceedings of the 7th SIAM International Conference on Data Mining, Minneapolis, MN, USA, 26–28 April 2007; pp. 443–448.
43. Moon, J.; Kim, Y.; Son, M.; Hwang, E. Hybrid Short-Term Load Forecasting Scheme Using Random Forest and Multilayer Perceptron. *Energies* **2018**, *11*, 3283. [CrossRef]
44. Khosravani, H.; Castilla, M.; Berenguel, M.; Ruano, A.; Ferreira, P. A Comparison of Energy Consumption Prediction Models Based on Neural Networks of a Bioclimatic Building. *Energies* **2016**, *9*, 57. [CrossRef]
45. Ali, U.; Shamsi, M.H.; Bohacek, M.; Hoare, C.; Purcell, K.; Mangina, E.; O'Donnell, J. A data-driven approach to optimize urban scale energy retrofit decisions for residential buildings. *Appl. Energy* **2020**, *267*, 114861. [CrossRef]
46. Anđelković, A.S.; Bajatović, D. Integration of weather forecast and artificial intelligence for a short-term city-scale natural gas consumption prediction. *J. Clean. Prod.* **2020**, *266*, 122096. [CrossRef]

energies

MDPI

Article

Exploratory Weather Data Analysis for Electricity Load Forecasting Using SVM and GRNN, Case Study in Bali, Indonesia

Siti Aisyah [1], Arionmaro Asi Simaremare [1], Didit Adytia [2,*], Indra A. Aditya [1] and Andry Alamsyah [2]

[1] Generation Division, PLN Research Institute, Jakarta 12760, Indonesia; siti.aisyah@pln.co.id (S.A.); arionmaro@pln.co.id (A.A.S.); indra.aditya@pln.co.id (I.A.A.)
[2] School of Computing, Telkom University, Bandung 40257, Indonesia; andrya@telkomuniversity.ac.id
* Correspondence: adytia@telkomuniversity.ac.id

Abstract: Accurate forecasting of electricity load is essential for electricity companies, primarily for planning electricity generators. Overestimated or underestimated forecasting value may lead to inefficiency of electricity generator or electricity deficiency in the electricity grid system. Parameters that may affect electricity demand are the weather conditions at the location of the electricity system. In this paper, we investigate possible weather parameters that affect electricity load. As a case study, we choose an area with an isolated electricity system, i.e., Bali Island, in Indonesia. We calculate correlations of various weather parameters with electricity load in Bali during the period 2018–2019. We use two machine learning models to design an electricity load forecasting system, i.e., the Generalized Regression Neural Network (GRNN) and Support Vector Machine (SVM), using features from various weather parameters. We design scenarios that add one-by-one weather parameters to investigate which weather parameters affect the electricity load. The results show that the weather parameter with the highest correlation value with the electricity load in Bali is the temperature, which is then followed by sun radiation and wind speed parameter. We obtain the best prediction with GRNN and SVR with a correlation coefficient value of 0.95 and 0.965, respectively.

Keywords: electricity load; forecasting; weather; GRNN; SVM

Citation: Aisyah, S.; Simaremare, A.A.; Adytia, D.; Aditya, I.A.; Alamsyah, A. Exploratory Weather Data Analysis for Electricity Load Forecasting Using SVM and GRNN, Case Study in Bali, Indonesia. *Energies* **2022**, *15*, 3566. https://doi.org/10.3390/en15103566

Academic Editors: Luis Hernández-Callejo, Sergio Nesmachnow and Sara Gallardo Saavedra

Received: 30 March 2022
Accepted: 4 May 2022
Published: 12 May 2022

1. Introduction

Electricity has become a vital part of the life of modern society nowadays. It is said that electricity access is an essential factor to enable the economic growth of a country or region [1]. Many studies also imply that the interruption of electricity supply has a severe impact on business and residential customers [2–4], where total electricity blackout can cost up to billions of dollars of economic activity [5]. These emphasize the importance of reliable and stable electricity supply to our current society.

One of the critical tasks in securing the electricity system's reliability is maintaining the balance between electricity supply and demand. In current large power systems, the task is done by adjusting the power generated from generation units in the systems to a forecasted system electricity demand. Failure to do this correctly may cause the instability of the power system or even a blackout. On the other hand, low accuracy of electricity demand forecasting may also cause inefficient and costly operation of the generation units caused by the requirements of higher capacity of spinning reserve generators and lower efficiency of thermal generators [6]. The latter may also lead to higher carbon emissions which contribute to global temperature rises or global warming [7]. Inevitably, the accuracy of electricity demand forecasting is paramount in electric power system planning and operation.

There are two approaches for estimating energy use: statistical techniques and artificial intelligence [8]. In recent years, artificial intelligence has accelerated, with one of its applications being to improve the control of the current generation system. Predicting electrical loads for energy consumption is no longer a novel concept, as it can be accomplished through machine learning to predict future energy consumption points [9]. Numerous studies have been conducted because it is critical to understand the prediction of electrical energy consumption. For example, in 2018, Li and Zhang completed a short-term forecasting of electricity consumption in Shanghai by using grey prediction model [10]. Tian et al. predicted short-term electrical energy consumption using a combination model between STL (Seasonal and Trend decomposition using Loess) and GRU in the same year. They made predictions for the next 3 to 10 days using a combination model between STL (Seasonal and Trend decomposition using Loess) and GRU. When compared to GRU and SVM, GRU produces better results [11]. Hamdoun et al. projected electrical energy by comparing two different approaches, namely statistics and machine learning, to see more accuracy. They found that the prediction model based on machine learning produced the best results and had the lowest error rate among the findings they obtained [12]. Using a combination of the FPA (Flower Pollination Algorithm) model to optimize the Feedforward Neural Network (FNN), Zhao et al. made a short-term prediction of electricity consumption in 2020, then compared it with the SVR and RBFN models. The FPA-FNN model produced good results, with MAPE values of 1.41 percent and RMSE [13]. The Nonlinear Autoregressive (NARM) model was used to predict the electricity load for the next month for the energy management system in 2019. Ahmad and Chen then compared the NARM model to the Random Forest model and the linear model using stepwise regression in the case of ISO New England using the results obtained. They discovered that when compared to the other two models, the NARM model produced the best results [14].

Several studies have shown that weather parameters can affect the electricity load and need to be incorporated in power system planning and demand forecasting [15], both for short-term and long-term system planning [16]. Some studies evaluate the effect of weather parameters on the electricity system at regional and country levels, such as Algeria [17] and Turkey [18]. Other studies evaluated at a lower level, such as building electricity demand or residential house electricity consumption [19,20].

Aisyah and Simaremare investigate the correlation between weather parameters and electricity load in Bali by using three different weather source data, i.e., GFS, ERA5, and observation data from AWS (Automatic Weather Station) BMKG [21]. They conclude that three weather parameters are highly correlated with electricity load in Bali, i.e., temperature, wind speed, and solar radiation. This paper investigates which weather parameters affect electricity consumption in an isolated area by calculating the correlation coefficient with electricity load data. Bali has a significant increase in electricity consumption, and the Island does not have conventional resources [22], so it is crucial to estimate the electricity load for the future. That is by investigating which weather parameters affect most the electricity consumption. Additionally, to our best knowledge, no published research yet on the machine learning area was conducted for the electricity load forecasting in Bali. Thus, we chose Bali Island in Indonesia as a case study. Moreover, we also developed electricity load forecasting using two machine learning models: the Support Vector Regression (SVR) and the Generalized Regression Neural Network (GRNN), with weather parameter data and consumer characteristics as input for the machine learning models.

The SVM is one of the machine learning models that is usually used to solve regression and classification problems. It performs efficiently for time series prediction, especially for seasonal data [23]. Moreover, the SVM also effectively prevents overfitting problems by implementing Structural Risk Minimization (SRM) [24]. GRNN is simple to train and gives a satisfactory prediction, modeling, mapping, and interpolation [25,26]. It also performs efficiently for continuous data [27]. It has a higher learning speed than RBF [28]. To determine which weather parameters have the most significant effects on the electrical load, we create scenarios by gradually increasing the number of weather parameters used as features. Moreover, we also add scenarios in which moving average (MA) of electricity load data is used as a feature for the machine learning models. The innovations in this paper are as follows: firstly, we introduce a technique for feature selection from weather parameters, in which the selected features are used as the inputs to design machine-learning-based electricity forecasting. In [9,29], deep networks are used to make an electricity forecasting model, but they did not make feature selection for weather parameters. Secondly, weather parameters are used as features, but we also consider moving average (MA) data (daily, weekly, and monthly MA) as input for the machine-learning-based electricity load forecasting.

The content of this paper is as follows. Section 2 discusses electricity load data and some weather parameters in Bali and two machine learning models used. We discuss exploratory data analysis between weather parameters and electricity load data in Section 3. It is then followed by descriptions of obtained results and some discussions in Section 4. We conclude the paper in the final section.

2. Materials and Methods

2.1. Electricity Load Data

This study was conducted in a case study location with an isolated power grid system, i.e., Bali Island, located in Indonesia. Bali's power is provided by external electricity producers from East Java Province and domestic electricity generators within the island. All of the power generated in Bali is utilized solely inside the island's boundaries. As indicated in Figure 1, we are using two-year electricity load data, i.e., 2018–2019. As seen in Figure 1, the electricity demand in Bali follows a consistent pattern throughout the year, with peak demand occurring during January through May and September through November and peak demand occurring during June through August. According to Figure 1, there are anomalies in the power load statistics for both 2018 and 2019, namely, 17 March 2018, and 7 March 2019, which are both Nyepi Days in Bali, during which people in Bali refrain from engaging in any activity, indoor or outdoor, on those days. There was also an electricity interruption on 5 September 2018, which resulted in a total outage of electricity over the whole Bali islands. The daily averaged electrical load and the daily trend is also depicted in Figure 1, which was derived through linear regression. We may also assume from this trend line that the power demand increased from 2018 to 2019. On average, 0.123 megawatts (MW) per day are added to the daily trend line in 2018, and 0.162 MW per day is added to the daily trend line in 2019.

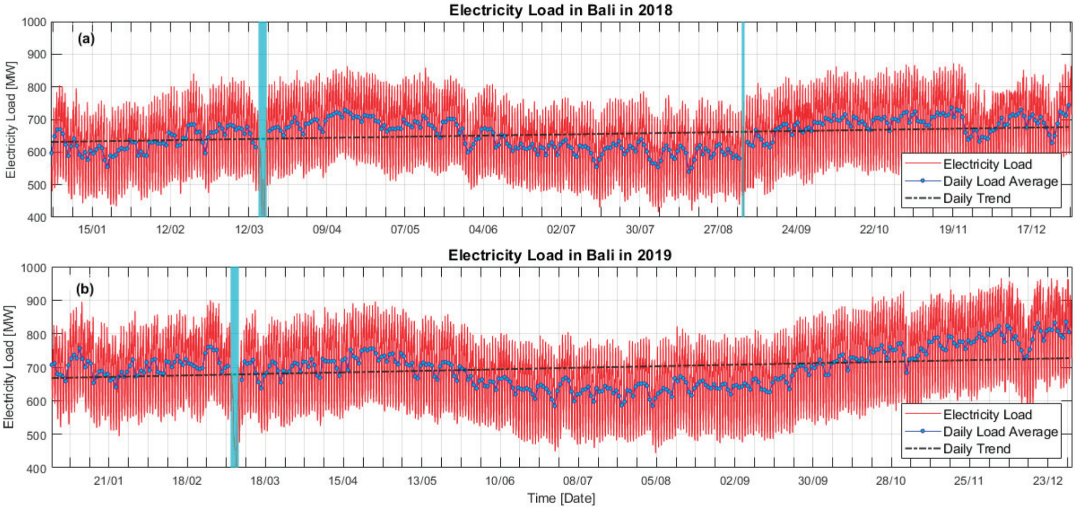


Figure 1. Electricity load data in Bali Island during 2018 (**a**) and 2019 (**b**). Hourly, daily, and daily trend electricity load are denoted by red line, blue line with circle, and black dot-dashed line, respectively.

Not only is it vital to examine the yearly trend, but it is also critical to evaluate the daily and weekly variations of the power load in the system under investigation. Figure 2 depicts the weekly and daily variance in electricity demand in Bali for the year 2019. In terms of weekly variation, we can see in the left-hand portion of Figure 2 that the characteristics of electricity demand in Bali remain relatively constant during the weekdays. In contrast, on weekends, there is a slight decrease in electricity demand on Saturday and a slight decrease in electricity demand on Sunday. These features are unsurprising given that most people do not work on Sundays, resulting in decreased electricity demand. The lowest power consumption in Bali is at 4:00 a.m. local time, as seen in the right portion of Figure 2. Still, the highest electrical demand is between 8:00 and 10:00 a.m. local time, when people begin their activities during the day. During the lunch hour, between 12:00 and 01:00 p.m., when most individuals take their lunch break, there was a modest decrease in electricity demand. The most significant demand for power happened between 07:00 and 08:00 p.m. when individuals ate their dinner. These hourly and daily characteristics are crucial to consider when constructing an electrical load forecasting system. In the following subsection, we will discuss the weather data used in this work.

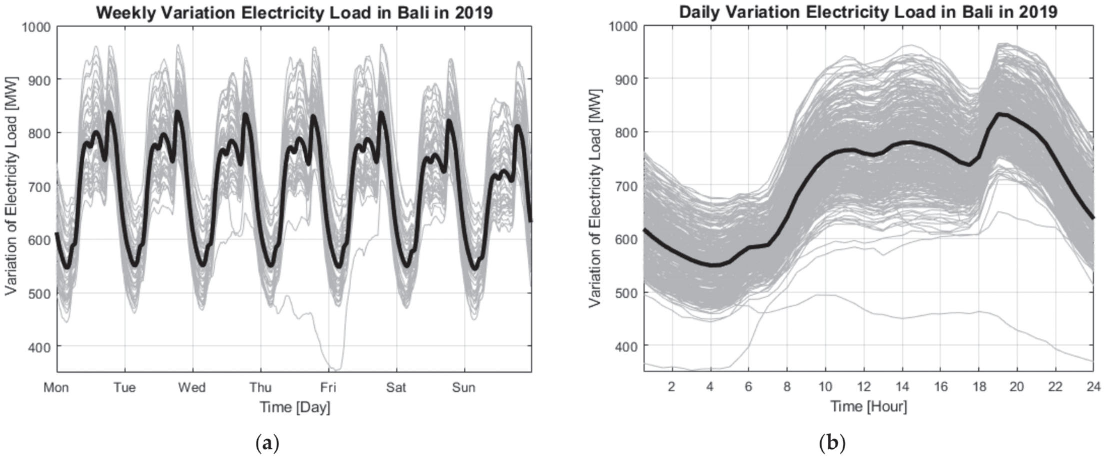


Figure 2. Variations of electricity load in Bali during 2019: (**a**) Weekly variation; (**b**) Daily variation. The solid black lines denote the mean value, whereas the gray lines denote the variations.

2.2. Weather Data

Primary weather data collected from field observations are the most optimal weather data to explain real-world weather conditions. While this fundamental data is somewhat inexpensive due to the requirement of a real-time measuring device, it is also highly costly. Furthermore, to use the observation data as a component of the energy load forecasting system, the observation data must be delivered to the forecasting system continuously, which necessitates the usage of a reliable measuring instrument. For this work, we will employ reanalysis weather data instead of real-time data as input for a machine learning model to be used as a feature in the electricity load forecasting system. This study uses the reanalysis weather data from the European Centre for Medium-Range Weather Forecasts, often known as the ECMWF, collected from the ERA5 model [30]. Since 1979, hourly weather data has been available, with spatial resolution varying between 0.25° and 0.75°. Weather parameters such as temperature, solar radiation, wind speed, rainfall rate, pressure, and relative humidity are investigated in this study.

To determine the quality of ERA5 weather parameter data, we compared the reanalysis data with observation data collected on Bali Island using an Automatic Weather Station (AWS) that has a temporal grid of 20 min. The AWS is positioned at latitude and longitude

115.167° E and 8.75° S. This study employs the most recent reanalysis ERA5 data from the nearest accessible grid to the AWS site, located at 115.00° E and 8.50° S, as shown in Figure 3. Indeed, the locations are quite a distance apart from one to another. Nonetheless, as seen in Figure 4, we compare many weather parameters from the ERA5 with the observed AWS data to identify any differences. We examine four meteorological factors: rainfall rate, solar radiation, temperature, and wind speed in Figure 4 during June 2019.

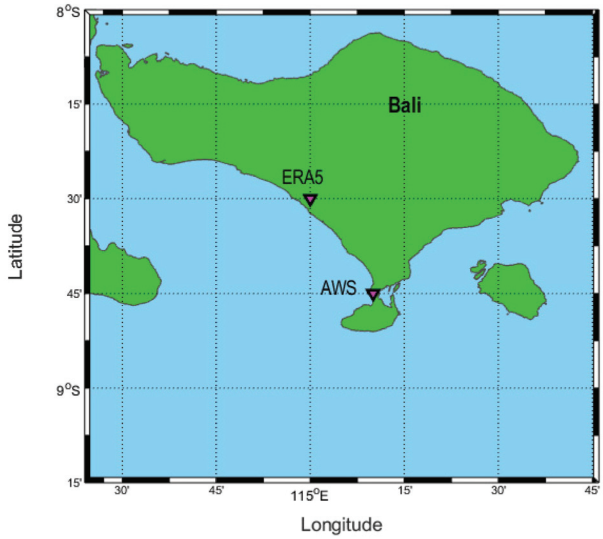
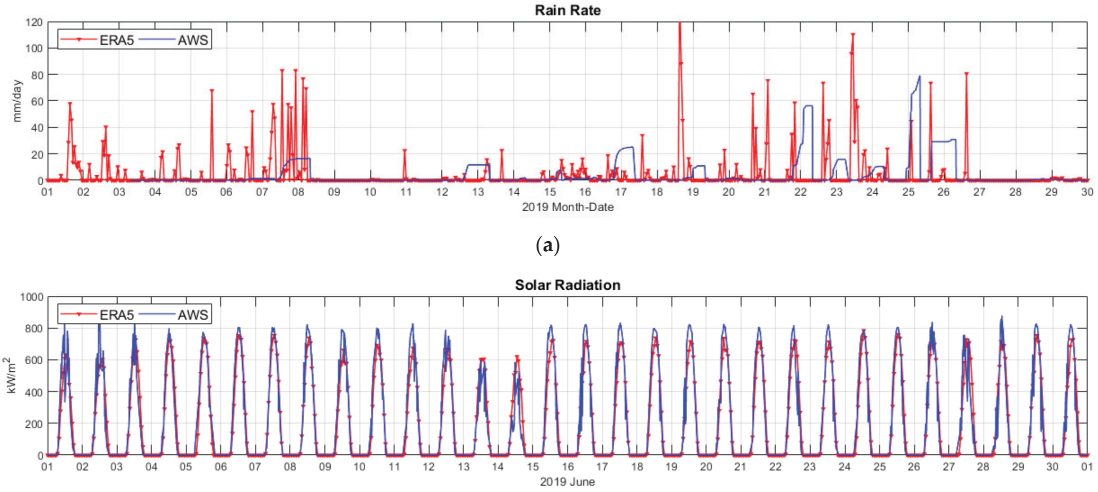


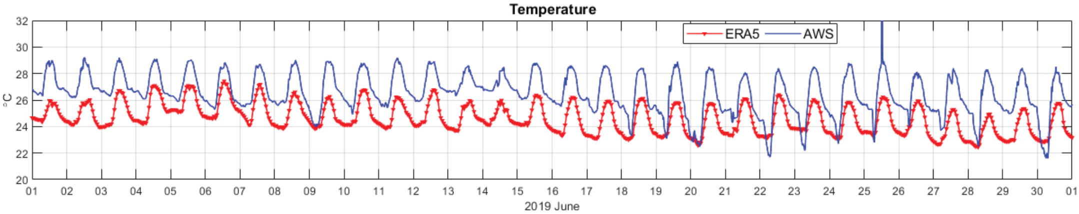
Figure 3. Location of Automatic Weather Station (AWS) in Ngurah Rai, Bali, and location of point for ERA5 data, in Bali Island, Indonesia.



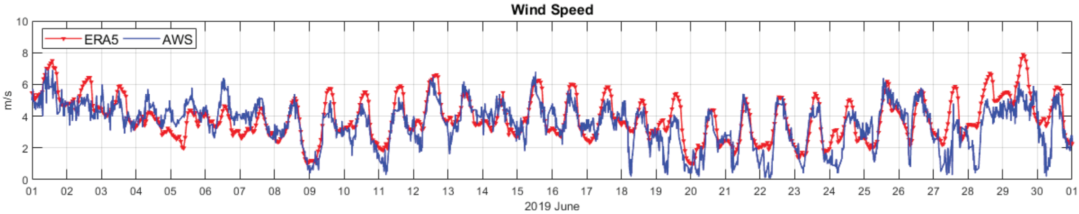
(**a**)

(**b**)

Figure 4. *Cont.*



(**c**)



(**d**)

Figure 4. Comparison of weather data from ERA5-ECMWF (red line with triangle) and Automatic Weather Station or AWS (blue line) for: (**a**) Rainfall Rate; (**b**) Solar Radiation; (**c**) Temperature; (**d**) Wind speed.

As shown in Figure 4, the solar radiation and wind speed, in particular, show a relatively similar trend between ERA5 and the observation data from AWS. In contrast, the other two parameters, i.e., the temperature and the rainfall rate, show a similar trend but with a different magnitude between ERA5 and the observation data from AWS. Because a significant distance separates the ERA5 point and the AWS point locations, this disparity might be caused by differences in local temperature and rainfall rates that are potentially highly different. The ERA5 data offers a good representation of the trend of meteorological parameters for Bali Island when compared to other data sources.

2.3. Methods

This paper has two main steps to design an electricity load forecasting system: (1) Exploratory data process to investigate correlations between weather parameters and electricity load; (2) Design a machine-learning-based model for electricity load forecasting using the best weather features obtained from step (1). For electricity load forecasting, two machine learning methods were utilized, namely, the Generalized Regression Neural Network (GRNN) and the Support Vector Regression (SVR) techniques (SVR). In the following subsections, we briefly describe these two methods.

2.3.1. Generalized Regression Neural Network

Donald F. Specht initially presented the General Regression Neural Network (GRNN) in 1991 [25], which is a deformation version of the radial basis function (RBF) neural network [28]. In comparison to RBF, GRNN improves at approximation and learning speed [31]. Its functioning is based on nonlinear or kernel regression, which implies that the result is dependent on the input. GRNNs may be utilized for prediction, modeling, mapping, and interpolation, as well as serving as controllers [25].

The GRNN architecture, as seen in Figure 5, is composed of four layers: the input layer, the pattern layer, the summation layer, and the output layer. The input layer takes and stores the input data $X_i = [x_1, x_2, \ldots, x_n]$. The number of neurons in a network is proportional to the amount of data input. The input layer's result is then transmitted to the pattern layer. The pattern layer is nonlinear, and its neurons can retain information about the interaction between the input neurons and the pattern layer [31]. A pattern based on the Gaussian function Pi can be expressed as follows

$$P_i = exp\left[-\frac{(X - X_i)^T (X - X_i)}{2\sigma^2}\right] (i = 1, 2, \ldots, n) \tag{1}$$

where σ is the smoothing or spreading parameter. The input variable is denoted by X, whereas x_i denotes a more precise training sample from neuron i in the pattern layer.

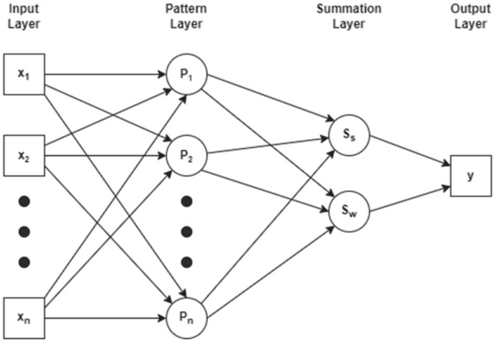


Figure 5. The architecture of General Regression Neural Network.

Following the pattern, the summation layer performs two distinct computations referred to as numerators and denominators. The first kind is used to determine the number of weighted outputs from the pattern layer, whereas the second type is used to determine the number of unweighted outputs from the pattern layer [26]. The pattern layer's purpose is as follows:

$$S_s = \sum_{i=1} P_i, (i = 1, 2, \ldots, n) \tag{2}$$

$$S_w = \sum_{i=1} w_i P_i, (i = 1, 2, \ldots, n) \tag{3}$$

where S_s is the denominator, S_w is the numerator, and w_i is the weight of the pattern neuron i connected to the summation layer.

The last layer is the output layer, the results of which are produced by dividing the neuron numerator S_s by the neuron denominator S_w. The output layer performs the following calculations:

$$y = \frac{S_w}{S_s} \tag{4}$$

In comparison to other approaches, the primary advantage of GRNN is that it is simple to train and requires only one independent parameter [26]. GRNN does not require recurrent training and may be trained in a short period of time. While this is a disadvantage of GRNN over other algorithms, it does need significant processing to analyze new points. These shortcomings, however, can be solved by adopting the clustering version of GRNN or by executing computations using an embedded parallel structure and building a semi-conductor chip [25].

2.3.2. Support Vector Machine

Vapnik et al. pioneered the Support Vector Machine (SVM) in 1999 [32]. SVM is a classification and regression technique used in machine learning [23]. This technique is more effective when used in conjunction with Structural Risk Minimization (SRM) than when used in conjunction with Empirical Risk Minimization (ERM) [24]. Support Vector

Regression is a machine learning model that allows for trade-offs between minimizing empirical errors and the complexity of the resultant fitted function, hence lowering the danger of overfitting [33]. SVR employs a soft margin approach to achieve the highest degree of generalization; the regression issue is handled using an alternate loss function and two slack variables [24]. As follows is the definition of the nonlinear regression problem using the SVR model.

$$y = f(x) = \omega \cdot \psi(x) + b \tag{5}$$

where ω is a weighted vector, b is a constant bias, and $\psi(x)$ is the feature space mapping function. The following minimization procedure is used to obtain the coefficients of ω and b:

$$Minimize\ \frac{1}{2}\|w^2\| + C\frac{1}{N}\sum_{i=1}^{N}(\xi_i + \xi_i{}^*) \tag{6}$$

$$Subject\ to \begin{cases} y_i - (w,\ x_i + b) \ \geq\ \varepsilon + \xi_i \\ (w,\ x_i) + b - y_i \ \leq\ \varepsilon + \xi_i{}^* \\ \xi_i,\ \xi_i{}^* \ \geq\ 0 \end{cases} \tag{7}$$

where the parameters C and ε are model-defined. C evaluates the trade-off between empirical risk and smoothness, whereas $\frac{1}{2}\|w^2\|$ quantifies the function's smoothness. ξ and ξ^* are positive slack variables that indicate the difference between the actual and corresponding limit values in the approximation function's ε-tube model.

Following the application of the Lagrangian multiplier and optimization of the conditions, the nonlinear regression function $f(x)$ is as follows.

$$f(x) = \sum_{i=1}^{N}(\delta_i - \delta_i^*)\ K\big(x_i,\ x_j\big) + b \tag{8}$$

where $K\big(x_i,\ x_j\big)$ is a kernel function that describes the inner product in D-dimensional feature space [34], and δ_i and δ_i^* are Lagrangian multipliers.

The GRNN and SVR method were utilized for designing machine-learning-based model for electricity load forecasting with weather parameters are features input. In the next section, we perform exploratory data to calculate correlations between weather parameters with electricity load in Bali.

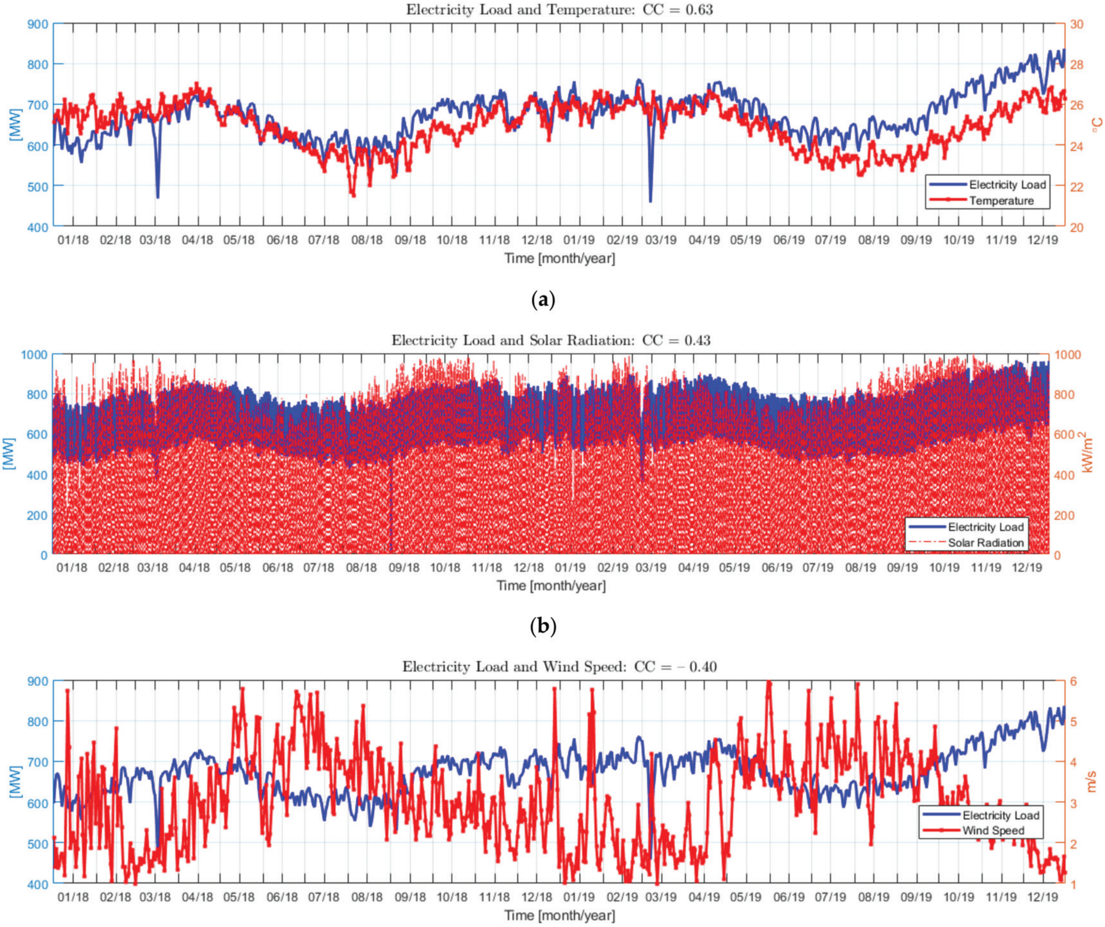
3. Exploratory Data Analysis

The relationship between weather data parameters with electricity load in Bali is investigated in this section by calculating how correlate these parameters with each other. To calculate correlation between two variables, we employ the so-called correlation coefficient (CC), which is utilized to show how close a relationship between two variables' data is to one another, especially for the trend of these variables. The formula for the correlation coefficient is defined as follows:

$$CC = \frac{cov(X,Y)}{\sigma_x \sigma_y} \tag{9}$$

where X and Y are variables that being compared, $cov(X,Y)$ denotes the covariance between two variables, and σ_x and σ_y denotes the standard deviation of data X and Y, respectively. In this paper, we use Formula (9) to calculate the correlation between electricity load with weather parameters, such as 2 m temperature, net solar radiation, wind speed, rainfall rate, pressure, and relative humidity.

Figure 6 compares electricity load data in Bali Island during 2019 with weather parameters such as temperature, solar radiation, and wind speed, whereas Figure 7 shows comparisons for rain rate, pressure, and relative humidity. In Figures 6 and 7, the electricity load data is denoted as blue lines with the left-hand side y-axis, whereas weather parameters are red lines with the right-hand side y-axis. As shown in Figure 6, we can directly notice that the temperature and solar radiation have a very similar trend with the electricity load in Bali, which indicates these two weather parameters have a high (positive) correlation with electricity load in Bali. For the wind speed, as shown in the lower part of Figure 6, the trend of electricity load is in the opposite direction, indicating that the wind speed and electricity load have a negative correlation.



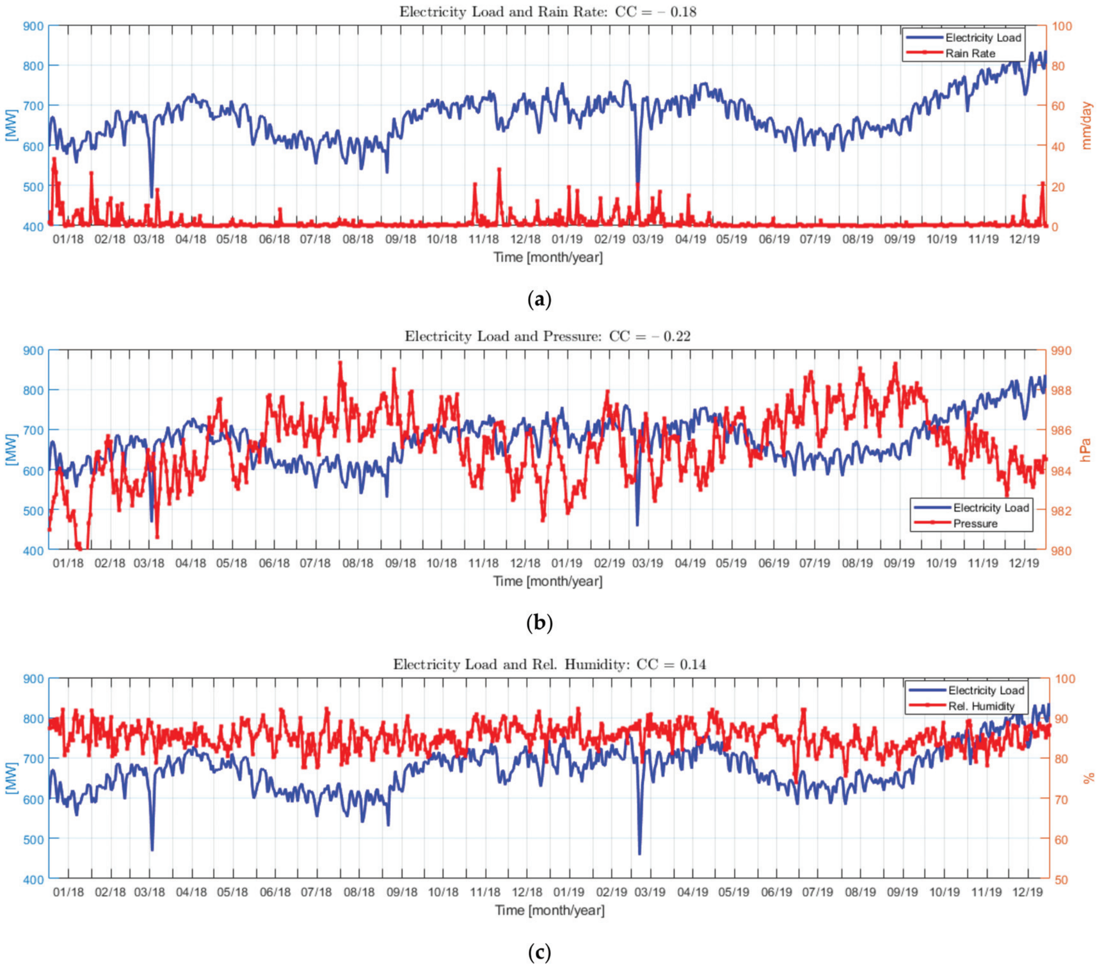
(**a**)

(**b**)

(**c**)

Figure 6. Plots of electricity load in Bali during 2019 in comparison with weather parameters; (**a**) temperature; (**b**) solar radiation; (**c**) wind speed. The magnitude of electricity load belongs to left *y*-axis, whereas the magnitude of weather parameters is in the right *y*-axis.

In Figure 7, we can see lower correlations between the rainfall rate with electricity load. In contrast, for the pressure, we can also see a negative correlation with electricity load, as with the wind parameter. The trend of the relative humidity parameter with the electricity load is not very clear, which indicates a low correlation value. Table 1 shows correlation coefficient (CC) values between each weather parameter in Figures 6 and 7 with electricity load in Bali. As shown qualitatively in Figure 6, the most correlated weather parameter with the electricity load is the 2 m temperature and is followed by the net solar radiation with CC values of 0.63 and 0.43, respectively. As also noticed in Figure 6, the wind parameter negatively correlates with the electricity load, with a CC value of −0.40, which is relatively high. Other weather parameters such as rainfall rate, pressure, and relative humidity have lower CC values, i.e., −0.18, −0.22, and 0.14, respectively. Based on this exploratory data, we can conclude that three weather parameters have a high correlation with the electricity load in Bali island, i.e., 2 m temperature, net solar radiation, and wind speed. These parameters will be used as features for machine learning models, which will be discussed in the next section.



(**a**)

(**b**)

(**c**)

Figure 7. As in Figure 4, for other weather parameters; (**a**) rainfall rate; (**b**) pressure; (**c**) relative humidity.

Table 1. Correlation Coefficient (CC) between electricity load and various weather parameters.

Weather Parameter	CC
2 m Temperature	0.63
Net Solar Radiation	0.43
Wind Speed	−0.40
Rainfall Rate	−0.18
Pressure	−0.22
Relative Humidity	0.14

4. Prediction of Electricity Load

As discussed in the previous section, we have investigated correlations between various weather parameters with the electricity load in Bali. The three most correlated weather parameters, i.e., 2 m temperature, net solar radiation, and wind speed, with CC values varying from −0.40 to 0.63. Three other parameters have lower CC values. This section explores possible designs for feature input of a machine-learning-based model for the electricity load forecasting system. Firstly, we investigate which weather parameters will give the best configuration for feature input for machine learning models. Secondly, we also investigate scenarios to improve prediction results by adding moving average information as an additional input for machine learning prediction.

4.1. Prediction Using Weather Data

This subsection proposes multiple scenarios for feature input to design a machine-learning-based electricity load forecasting system. We design scenarios that add one-by-one weather parameters, from high to low CC value, as feature input for two machine learning models, i.e., the GRNN and SVR. Besides weather parameters, customer characteristics also significantly affect electricity load consumption, as shown in Figure 2. We include two characteristics of electricity customers in Bali island, i.e., hourly and daily characteristics, illustrated in Figure 2. The hourly characteristics are represented as values from 1 to 24 that represent hours, whereas for the daily characteristics, there are values from 1 to 7 that represent day number. These two customer characteristics are included as scenario-1 in Table 2. For other scenarios, i.e., scenarios 2 to 6, we added one-by-one weather parameters, from high to low correlated weather parameters, as shown in Table 2.

Table 2. Scenarios to investigate effects of each weather parameter as feature input for the machine learning.

Scenario	**Feature**	
	User Behavior	**Weather Parameter**
1	Hourly Characteristics	-
	Daily Characteristics	
2	Hourly Characteristics	2 m Temperature
	Daily Characteristics	
3	Hourly Characteristics	2 m Temperature
	Daily Characteristics	Net Solar Radiation
4	Hourly Characteristics	2 m Temperature
	Daily Characteristics	Net Solar Radiation
		Wind Speed
5	Hourly Characteristics	2 m Temperature
	Daily Characteristics	Net Solar Radiation
		Wind Speed
		Rainfall Rate
6	Hourly Characteristics	2 m Temperature
	Daily Characteristics	Net Solar Radiation
		Wind Speed
		Rainfall Rate
		Pressure

For training data for the machine learning models, we use one year data, i.e., during 2018, to forecast 1-month electricity load data, i.e., January 2019. Using features configuration scenarios as shown in Table 1, we perform electricity load forecasting using the GRNN model, as shown qualitatively in Figure 8. Here, we can see qualitatively that scenario-2 in Figure 8b. gives the best prediction compared to other scenarios. The scenario-2 consisted of hourly and daily characteristics with 2 m temperature as input for the machine learning model. Adding additional weather parameters features such as scenario-3 to -6 results in worse prediction performances, as shown qualitatively in Figure 8c–f.

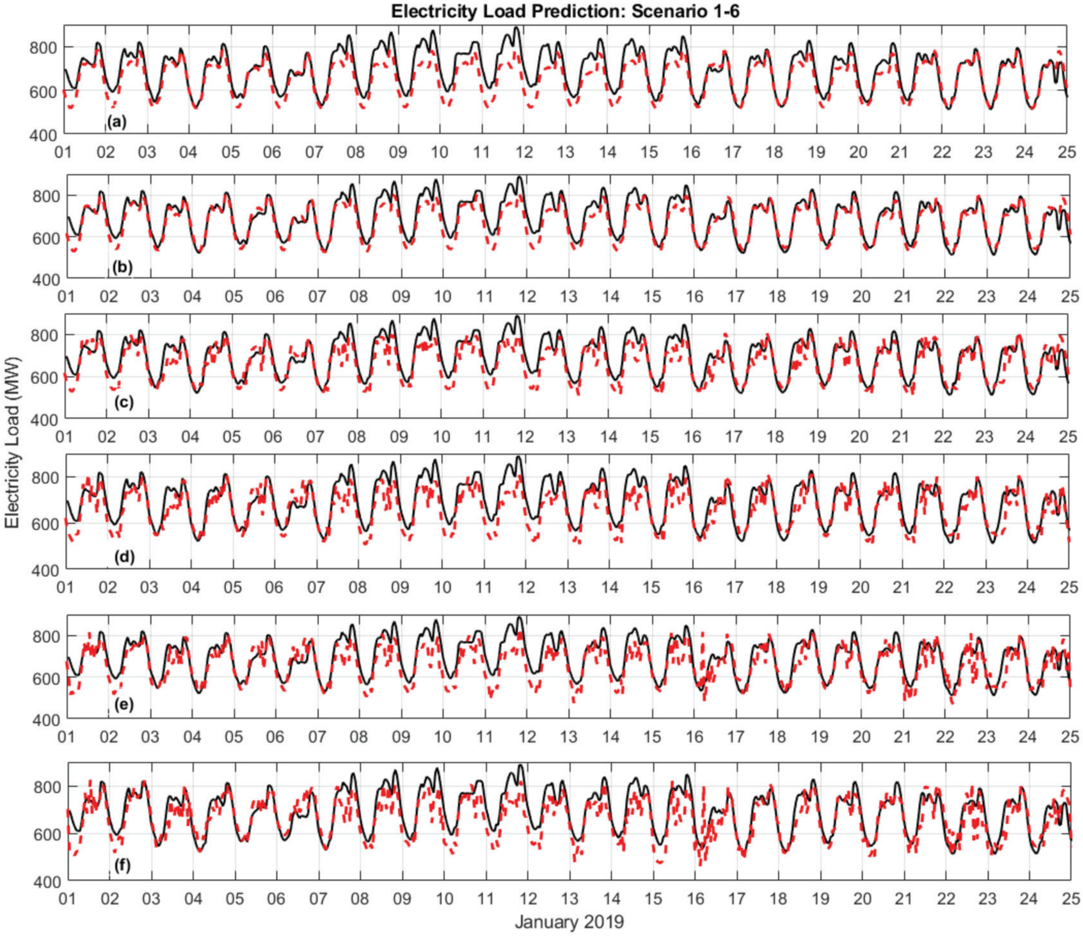


Figure 8. Comparison between electricity load data (solid black line) during the period 1–25 January 2019, with results of prediction by using GRNN model (dashed red line) with various feature scenarios; (**a**) scenario-1; (**b**) scenario-2; (**c**) scenario-3; (**d**) scenario-4; (**e**) scenario-5; and (**f**) scenario-6.

We also optimize the GRNN and SVR model parameter settings to give the best prediction. For the GRNN, there is only one parameter to be optimized, i.e., the "spread" parameter. The spread parameter is optimized by varying its value, as shown in Table 3. Table 3 shows results of various values of spread parameter of GRNN model for predicting scenario-2. From this table, the spread value of 0.50 gives the best performance. We also optimized parameter settings in the SVR model. The best result is obtained with radial basis function kernel, regularization parameter C value of 100, kernel coefficient γ of 2, ϵ value in SVR model is 0.1, and polynomial degree 3.

Table 3. Results of various value of parameter Spread in the GRNN model for scenario-2.

Spread	CC	RMSE
1.25	0.917	46.35
1.00	0.926	44.36
0.75	0.933	42.68
0.50	0.937	41.72

Not only using the GRNN, we also perform prediction by using the SVR model, in which results of prediction by using two models are summarized in Table 4. Here, the best scenario for the GRNN model is obtained by scenario-2, which results in a CC value of 0.937 and a root mean square error (RMSE) value of 41.72. For the SVR model, the best scenario is obtained by scenario-3, i.e., with weather parameter temperature and net solar radiation, resulting in a CC value of 0.934 and an RMSE value of 48.88. Note that the RMSE value of the best scenario obtained by using the GRNN model is lower than the SVR model. It is also the same with the CC value; the GRNN model gives slightly better performance than the SVR model.

Table 4. Results of prediction by using GRNN and SVR model with various weather parameter scenarios, as described in Table 2.

Scenario	GRNN		SVR	
	CC	RMSE	CC	RMSE
1	0.886	53.87	0.877	62.21
2	0.937	41.72	0.929	49.88
3	0.897	50.79	0.934	48.88
4	0.894	52.44	0.917	53.44
5	0.884	54.62	0.906	55.43
6	0.879	53.61	0.876	59.51

4.2. Prediction Using Moving Average Data

We also explore the possibility of improving the accuracy of the machine-learning-based electricity forecasting system by adding another feature configuration. In this subsection, we experiment with scenarios when additional features are added into machine learning, i.e., moving average (MA) data of the electricity load data. The moving average data is the electricity load that is averaged with a specific time frame range. It is possible to obtain this MA data in the implementation of the electricity load forecasting as long as realization (observation) data of electricity load can be accessed directly and fed into the machine learning forecasting system.

This subsection added three scenarios of moving average (MA) data, i.e., monthly, weekly, and daily moving average data. Monthly moving average data means that averaged electricity load data is calculated with a time frame of one month from the time series of historical electricity load data. The MA information is fed into the machine learning forecasting system. To compare how effective the addition of MA data was into the machine learning model, we performed electricity load prediction using the GRNN and SVR model with scenario-2, as shown in the previous subsection. The scenario-2 is added with monthly, weekly, and daily MA as new scenarios. Figure 9 shows the results of each scenario with MA data. From Figure 9, the scenario with monthly MA data results in worse performance than the scenario without MA.

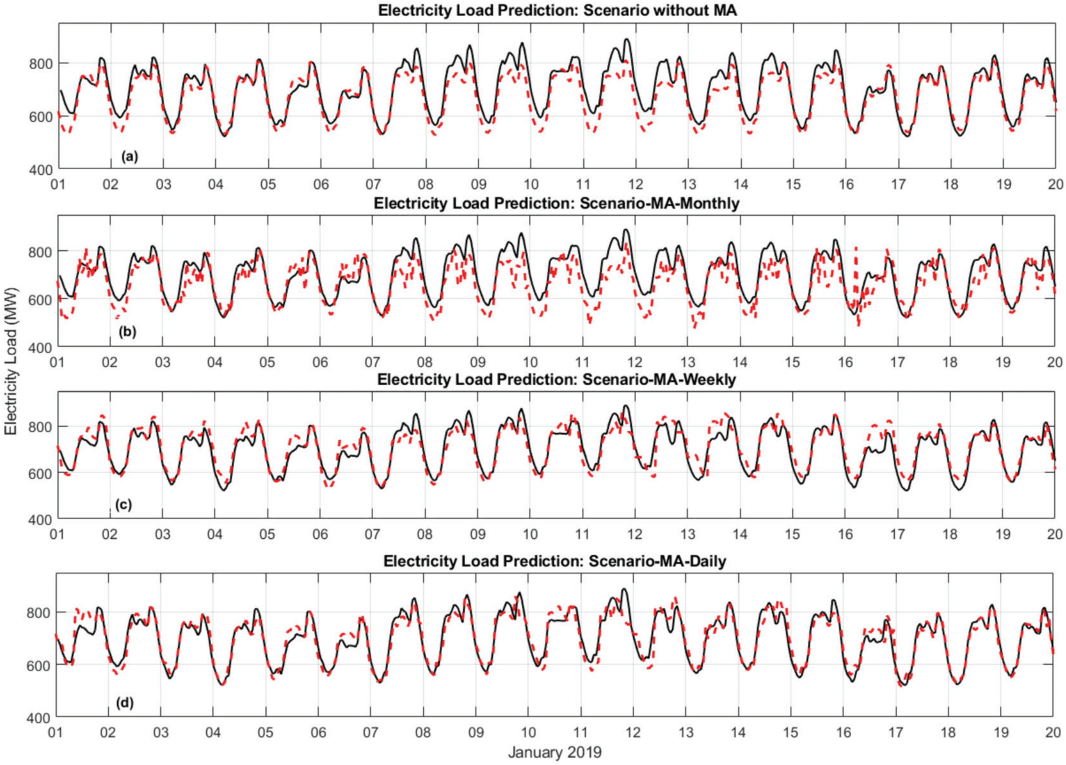


Figure 9. Comparison between electricity load data (solid black line) during the period 1–20 January 2019, with results of prediction by using GRNN model (dashed red line) with various feature moving averaged (M.A.) scenarios; (**a**) scenario without M.A.; (**b**) scenario with MA-Monthly; (**c**) scenario with MA-Weekly; (**d**) scenario with MA-Daily.

On the other hand, better performance is achieved by scenarios with weekly and daily MA data. Quantitatively, each scenario's performance is summarized in Table 5 for both using GRNN and SVR model. The best performance scenario for both GRNN and SVR is the scenario with MA-daily; for the GRNN model, the best scenario gives a CC value of 0.956 and RMSE value of 28.82, whereas for the SVR model, it gives a CC value of 0.965, and RMSE value of 44.40. Note that the SVR model gives slightly better performance in terms of CC value than the GRNN model results but gives a worse performance in terms of RMSE value. Overall, the GRNN model gives better results than the SVR model.

Table 5. Results of prediction by using GRNN and SVR model with various scenario with Moving Average (M.A.) values; Monthly, Weekly, and Daily.

Scenario	GRNN		SVR	
	CC	RMSE	CC	RMSE
Without MA	0.937	41.72	0.929	49.88
MA-Monthly	0.884	54.62	0.931	47.88
MA-Weekly	0.916	40.27	0.943	46.77
MA-Daily	0.956	28.82	0.965	44.40

We compare prediction results using a scenario with MA-daily in Figure 10 for both GRNN and SVR models. Qualitatively, the GRNN gives better prediction, especially vertical direction errors, confirmed by RMSE values as in Table 5.

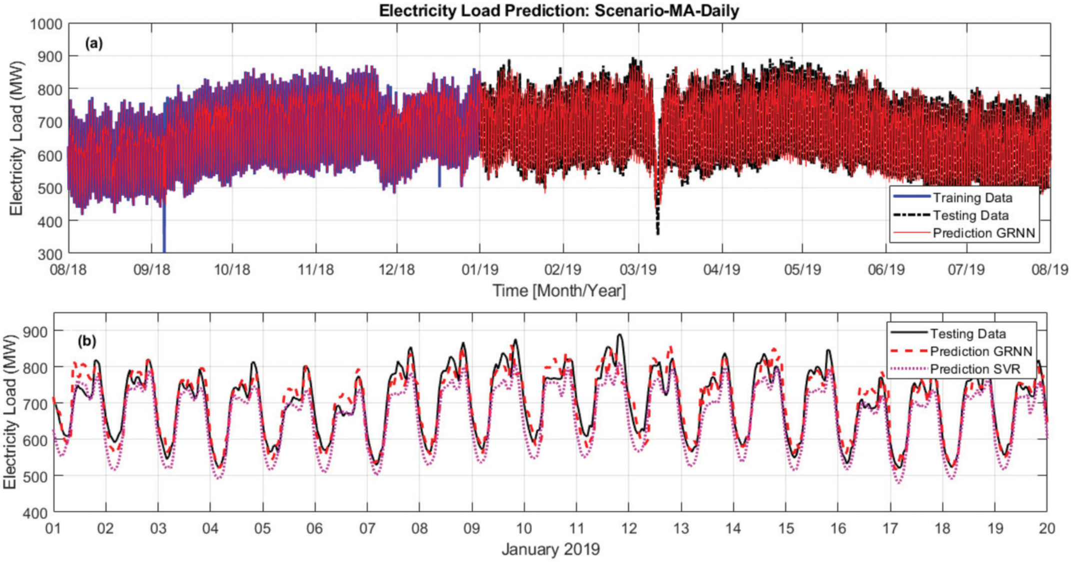


Figure 10. Comparison between electricity load testing data (solid black line) with results of prediction by using GRNN model (dashed red line), and SVR (dotted magenta line) for Scenario with MA-Daily; (**a**) during the period 1 January–1 August 2019; (**b**) during the period 1–20 January 2019.

5. Conclusions

This paper aims to design a machine-learning-based electricity load forecasting system. We investigate two primary studies, i.e., exploratory data, to investigate the correlation between weather parameters and electricity load data and feature selection optimization for the machine learning forecasting model. This paper uses a statistical method, i.e., the correlation coefficient (CC), to select highly correlated weather parameters with the electricity load data. The results of this step are used as input for the machine-learning-based electricity forecasting model, which is not considered a statistical method. However, our results show that this feature selection step significantly affects the machine learning prediction accuracy. We found that this statistically based feature selection improves the accuracy of the machine learning model.

Results from exploratory data conclude that three weather parameters highly correlated with the electricity load in Bali islands, i.e., 2 m temperature, net solar radiation, and wind speed. Other weather parameters, such as rainfall rate, pressure, and relative humidity, are less correlated. To investigate the effects of weather parameters as feature input for the machine learning model, we perform scenarios in which we added one-by-one weather parameters, from high to low correlated weather parameters. For the GRNN model, the best performance scenario is achieved for the featured scenario only with 2 m temperature, a CC value of 0.937, and an RMSE value of 41.72. On the other hand, the best performance scenario for the SVR model is a feature scenario of 2 m temperature and net solar radiation, resulting in a CC value of 0.934 and an RMSE value of 48.88. Predicting using the GRNN is better than the SVR, especially in terms of correlation coefficient (CC) value and RMSE value, as shown in scenario-2 in Section 4.1. This result can be related to the fact that the GRNN only has one parameter to be optimized, i.e., the spread parameter. In contrast, there are more parameters to be optimized in the SVR model, i.e., type of kernel function, regularization parameter, kernel coefficient, polynomial degree, etc. Therefore, optimizing the GRNN is more straightforward than the SVR. Moreover, the GRNN is a model with strong nonlinear mapping capabilities suitable for solving the electricity load forecasting problem with weather parameter features, as in this paper.

To improve the performance of the prediction, we also investigate an option to add another feature to the machine learning forecasting model, i.e., we add the moving average (MA) of historical electricity load data itself to the machine learning. There are three scenarios of moving average data that we investigated, i.e., monthly, weekly, and daily moving average data. Scenario with the additional feature of MA-monthly data gives worse performance than scenario without MA-monthly data. The other two scenarios, i.e., MA-weekly and MA-daily, give better performance than without MA data. The best performance scenario is achieved with MA-daily data; the GRNN model gives the CC value of 0.956, RMSE of 28.82, and the SVR model gives the CC value of 0.965 and RMSE value of 44.40. In conclusion, the GRNN model performs better than the SVR model regarding the RMSE value. The inclusion of moving average electricity load data is possible when the forecasting system can obtain near real-time realization (observation) data of electricity load.

For future research direction, there are several points that can be investigated further. Firstly, to further improve the accuracy of the electricity load prediction, more advanced machine learning models can be investigated, i.e., deep learning models. Secondly, in an area that is connected with multiple electricity grid systems, the correlation between weather parameters and electricity load can be low. Therefore, a new technique for feature selection is needed to design electricity load forecasting for this type of area.

Author Contributions: Conceptualization, S.A., D.A. and A.A.; methodology, S.A. and A.A.S.; software, I.A.A.; validation, D.A. and A.A.; formal analysis, S.A. and D.A.; investigation, I.A.A.; resources A.A.S.; data curation, I.A.A.; writing—original draft preparation, S.A.; writing—review and editing, D.A. and A.A.; visualization, I.A.A.; supervision, D.A.; project administration, S.A.; funding acquisition, I.A.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by PT PLN (Persero) Puslitbang Ketenagalistrikan with contract number: 0020.Pj/LIT.00.02/C0000000/2021.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Burke, P.J.; Stern, D.I.; Bruns, S.B. The Impact of Electricity on Economic Development: A Macroeconomic Perspective. *Int. Rev. Environ. Resour. Econ.* **2018**, *12*, 85–127. [CrossRef]
2. Wu, K.Y.; Huang, Y.H.; Wu, J.H. Impact of electricity shortages during energy transitions in Taiwan. *Energy* **2018**, *151*, 622–632. [CrossRef]
3. Nduhuura, P.; Garschagen, M.; Zerga, A. Impacts of electricity outages in urban households in developing countries: A case of Accra, Ghana. *Energies* **2021**, *14*, 3676. [CrossRef]
4. Falentina, A.T.; Resosudarmo, B.P. The impact of blackouts on the performance of micro and small enterprises: Evidence from Indonesia. *World Dev.* **2019**, *124*, 104635. [CrossRef]
5. Koks, E.; Pant, R.; Thacker, S.; Hall, J.W. Understanding business disruption and economic losses due to electricity failures and flooding. *Int. J. Disaster Risk Sci.* **2019**, *10*, 421–438. [CrossRef]
6. Zhang, J.; Wang, Y.; Hug, G. Cost-oriented load forecasting. *Electr. Power Syst. Res.* **2022**, *205*, 107723. [CrossRef]
7. Jorgenson, A.; Longhofer, W.; Grant, D. Disproportionality in power plants' carbon emissions: A cross-national study. *Sci. Rep.* **2016**, *6*, 28661. [CrossRef]
8. Salam, A.; El Hibaoui, A. Energy consumption prediction model with deep inception residual network inspiration and LSTM. *Math. Comput. Simul.* **2021**, *190*, 97–109. [CrossRef]
9. Wahab, A.; Tahir, M.A.; Iqbal, N.; Ul-Hasan, A.; Shafait, F.; Kazmi, S.M.R. A Novel Technique for Short-Term Load Forecasting Using Sequential Models and Feature Engineering. *IEEE Access* **2021**, *9*, 96221–96232. [CrossRef]
10. Li, K.; Zhang, T. Forecasting electricity consumption using an improved grey prediction model. *Information* **2018**, *9*, 204.

11. Tian, Y.J.; Zhou, S.J.; Wen, M.; Li, J.G. A Short-Term Electricity Forecasting Scheme Based on Combined GRU Model with STL Decomposition. *IOP Conf. Ser. Earth Environ. Sci.* **2021**, *701*, 012008. [CrossRef]
12. Hamdoun, H.; Sagheer, A.; Youness, H. Energy time series forecasting-analytical and empirical assessment of conventional and machine learning models. *J. Intell. Fuzzy Syst.* **2021**, *40*, 12477–12505. [CrossRef]
13. Zhao, W.; Liu, X.; Li, C.; Zhang, F.; Wang, Q. Short-term Load Forecasting of PowerSystem Based on Improved Feedforward Neural Network. *J. Phys. Conf. Ser.* **2020**, *1549*, 052095. [CrossRef]
14. Ahmad, T.; Chen, H. Nonlinear autoregressive and random forest approaches to forecasting electricity load for utility energy management systems. *Sustain. Cities Soc.* **2019**, *45*, 460–473. [CrossRef]
15. Kumar, R.; Rachunok, B.; Maia-Silva, D.; Nateghi, R. Asymmetrical response of California electricity demand to summer-time temperature variation. *Sci. Rep.* **2020**, *10*, 1–9. [CrossRef]
16. Cassarino, T.G.; Sharp, E.; Barrett, M. The impact of social and weather drivers on the historical electricity demand in Europe. *Appl. Energy* **2018**, *229*, 176–185. [CrossRef]
17. Chabouni, N.; Belarbi, Y.; Benhassine, W. Electricity load dynamics, temperature and seasonality Nexus in Algeria. *Energy* **2020**, *200*, 117513. [CrossRef]
18. Bozkurt, Ö.Ö.; Biricik, G.; Tayşi, Z.C. Artificial neural network and SARIMA based models for power load forecasting in Turkish electricity market. *PLoS ONE* **2017**, *12*, e0175915. [CrossRef]
19. Li, M.; Allinson, D.; He, M. Seasonal variation in household electricity demand: A comparison of monitored and synthetic daily load profiles. *Energy Build.* **2018**, *179*, 292–300. [CrossRef]
20. Sukarno, I.; Matsumoto, H.; Susanti, L. Household lifestyle effect on residential electrical energy consumption in Indonesia: On-site measurement methods. *Urban Clim.* **2017**, *20*, 20–32. [CrossRef]
21. Aisyah, S.; Arionmaro, A.S. Correlation between Weather Variables and Electricity Demand. *IOP Conf. Ser. Earth Environ. Sci.* **2021**, *927*, 012015. [CrossRef]
22. Kumara, I.N.S.; Ariastina, W.G.; Sukerayasa, I.W.; Giriantari, I.A.D. On the potential and progress of renewable electricity generation in Bali. In Proceedings of the 2014 6th International Conference on Information Technology and Electrical Engineering (ICITEE), Yogyakarta, Indonesia, 7–8 October 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 1–6.
23. Li, G.; Li, Y.; Roozitalab, F. Midterm load forecasting: A multistep approach based on phase space reconstruction and support vector machine. *IEEE Syst. J.* **2020**, *14*, 4967–4977. [CrossRef]
24. Shukla, D.; Jaiswal, S.; Babu, V.P.; Singh, S.P. Near Real Time Load Forecasting in Power System. In Proceedings of the 2020 21st National Power Systems Conference (NPSC), Gandhinagar, India, 17–19 December 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1–6.
25. Specht, D.F. A general regression neural network. *IEEE Trans. Neural Netw.* **1991**, *2*, 568–576. [CrossRef] [PubMed]
26. Kim, B.; Lee, D.W.; Park, K.Y.; Choi, S.R.; Choi, S. Prediction of plasma etching using a randomized generalized regression neural network. *Vacuum* **2004**, *76*, 37–43. [CrossRef]
27. Eskidere, Ö.; Ertaş, F.; Hanilçi, C. A comparison of regression methods for remote tracking of Parkinson's disease progression. *Expert Syst. Appl.* **2012**, *39*, 5523–5528. [CrossRef]
28. Hu, R.; Wen, S.; Zeng, Z.; Huang, T. A short-term power load forecasting model based on the generalized regression neural network with decreasing step fruit fly optimization algorithm. *Neurocomputing* **2017**, *221*, 24–31. [CrossRef]
29. Han, L.; Peng, Y.; Li, Y.; Yong, B.; Zhou, Q.; Shu, L. Enhanced deep networks for short-term and medium-term load forecasting. *IEEE Access* **2018**, *7*, 4045–4055. [CrossRef]
30. Hersbach, H.; Bell, B.; Berrisford, P.; Hirahara, S.; Horányi, A.; Muñoz-Sabater, J.; Nicolas, J.; Peubey, C.; Radu, R.; Schepers, D.; et al. The ERA5 global reanalysis. *Q. J. R. Meteorol. Soc.* **2020**, *146*, 1999–2049. [CrossRef]
31. Liu, J.; Bao, W.; Shi, L.; Zuo, B.; Gao, W. General regression neural network for prediction of sound absorption coefficients of sandwich structure nonwoven absorbers. *Appl. Acoust.* **2014**, *76*, 128–137. [CrossRef]
32. Vapnik, V. *The Nature of Statistical Learning Theory*; Springer Science & Business Media: Berlin, Germany, 1999.
33. Smola, A.J.; Schölkopf, B. A tutorial on support vector regression. *Stat. Comput.* **2004**, *14*, 199–222. [CrossRef]
34. Al-Musaylh, M.S.; Deo, R.C.; Adamowski, J.F.; Li, Y. Short-term electricity demand forecasting with MARS, SVR and ARIMA models using aggregated demand data in Queensland, Australia. *Adv. Eng. Inform.* **2018**, *35*, 1–16. [CrossRef]

sustainability

MDPI

Article

A Novel Transfer Learning Method Based on Conditional Variational Generative Adversarial Networks for Fault Diagnosis of Wind Turbine Gearboxes under Variable Working Conditions

Xiaobo Liu, Haifei Ma and Yibing Liu *

Key Laboratory of Power Station Energy Transfer Conversion and System, North China Electric Power University, Ministry of Education, Beijing 102206, China; liuxiaobo116@163.com (X.L.); mahaifeii@163.com (H.M.)
* Correspondence: lyb@ncepu.edu.cn

Abstract: The rapid development of artificial intelligence offers more opportunities for intelligent mechanical diagnosis. Recently, due to various reasons such as difficulty in obtaining fault data and random changes in operating conditions, deep transfer learning has achieved great attention in solving mechanical fault diagnoses. In order to solve the problems of variable working conditions and data imbalance, a novel transfer learning method based on conditional variational generative adversarial networks (CVAE-GAN) is proposed to realize the fault diagnosis of wind turbine test bed data. Specifically, frequency spectra are employed as model signals, then the improved CVAE-GAN are implemented to generate missing data for other operating conditions. In order to reduce the difference in distribution between the source and target domains, the maximum mean difference (MMD) is used in the model to constrain the training of the target domain generation model. The generated data is used to supplement the missing sample data for fault classification. The verification results confirm that the proposed method is a promising tool that can obtain higher diagnosis efficiency. The feature embedding is visualized by t-distributed stochastic neighbor embedding (t-SNE) to test the effectiveness of the proposed model.

Keywords: conditional variational generative adversarial networks; transfer learning; wind turbines; variable working conditions

Citation: Liu, X.; Ma, H.; Liu, Y. A Novel Transfer Learning Method Based on Conditional Variational Generative Adversarial Networks for Fault Diagnosis of Wind Turbine Gearboxes under Variable Working Conditions. *Sustainability* **2022**, *14*, 5441. https://doi.org/10.3390/su14095441

Academic Editors: Luis Hernández-Callejo, Sergio Nesmachnow and Sara Gallardo Saavedra

Received: 26 March 2022
Accepted: 29 April 2022
Published: 30 April 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

1. Introduction

Fault diagnosis of wind turbines plays an important role in equipment health management. Recently, deep learning (DL) has become a promising method in intelligent fault diagnosis. DL methods usually follow two principles: (1) the dataset should be large and well labeled and (2) the training and testing datasets are subject to the same distribution. However, in reality, wind turbines often face the problems of working condition variation, sample imbalance, and few fault samples, which brings challenges for deep learning to achieve wind turbine fault diagnosis. Compared with DL, transfer learning (TL) allows different probability distributions of samples between source and target domains. This means that a new but related task in the target domain can be effectively addressed by the learned knowledge from the source domain.

TL-based models have been employed for intelligent fault diagnosis under different working conditions. Li et al. proposed a novel weighted adversarial transfer network (WATN) for partial domain fault diagnosis [1]. Huang et al. proposed a deep adversarial capsule network (DACN) to embed multi-domain generalization into the intelligent compound fault diagnosis [2]. Li et al. proposed a two-stage transfer adversarial network (TSTAN) for multiple new faults detection of rotating machinery [3]. Chen et al. proposed a transferable convolutional neural network to improve the learning of target tasks [4].

Li et al. proposed a method named deep adversarial transfer learning network (DATLN) for new emerging fault detection [5]. Li et al. proposed a data-driven fault feature separation method (DFSM) that can eliminate the working condition features from all the information and employ the rest of the fault information for diagnosis [6]. Qian et al. proposed a method called improved joint distribution adaptation (IJDA) to align both the marginal and conditional distributions of datasets more comprehensively [7]. Guo et al. proposed a deep convolutional transfer learning network (DCTLN), which consists of condition recognition and domain adaptation, for intelligent fault diagnosis of machines with unlabeled data [8]. Yang et al. proposed a feature-based transfer neural network (FTNN) to identify the health states of real-case machines with the help of the diagnosis knowledge from laboratory machines [9].

Domain adaptive (DA) technology plays an important role in transfer learning. Maximum mean discrepancy (MMD) is commonly used to measure the distribution discrepancy of the transferable features [10]. The MMD-based domain adaptive technology has been widely used to accomplish transfer learning tasks in the fields of computers [11,12]. The key to domain adaptation is to find a way to decrease the distribution divergence between different domains. Feature matching and instance reweighting are the main learning strategies for DA research. Zhang et al. applied the maximum variance discrepancy (MVD) for combining with the maximum mean discrepancy (MMD) for the feature matching [13]. Zhang et al. proposed a novel geodesic flow kernel-based domain adaptation approach for intelligent fault diagnosis under varying working conditions [14]. An et al. proposed a novel adaptive cross-domain feature extraction (ACFE) method that can automatically extract similar features between different feature spaces [15]. Qian et al. proposed a novel distribution discrepancy evaluating method called auto-balanced high-order Kullback–Leibler (AHKL) divergence for DA [16]. Based on polynomial kernel-induced MMD (PK-MMD), Yang et al. proposed a model that was constructed to reuse diagnosis knowledge from one machine to another [17].

However, an important problem in TL-based fault diagnosis methods is that target domain mechanical fault datasets are always highly imbalanced with abundant normal condition mechanical samples but a paucity of samples from rare fault conditions. The generative adversarial network (GAN) [18] uses the adversarial principle of generator and discriminator to enhance the diversity of data and provides the possibility to solve the above problems. Zheng et al. proposed a dual discriminator conditional generative adversarial network to enhance the accuracy of imbalance fault diagnosis [19]. Wang et al. implemented a Wasserstein generative adversarial network (WGAN) to generate simulated signals based on a labeled dataset [20]. There has been a proliferation of adversarial models presented by GAN, such as AnoGANs [21], GANormaly [22], etc. GAN has been developed in the field of fault diagnosis and anomaly detection [23–26]. Auto-encoder (AE) is another way of generating samples. AE has now developed numerous variants, e.g., variational AE (VAE) [27], adversarial AE (AAE) [28], etc.

The problem of missing data from wind turbines can be effectively solved by GAN and AE. Qu et al. proposed a data imputation method with multiple optimizations based on generative adversarial networks (GANs) for wind turbines [29]. Guo et al. proposed improved adversarial learning to generate fault features for the fault diagnosis of a wind turbine gearbox with unbalanced fault classes [30]. Jiang et al. proposed an improved oversampling algorithm to generate and develop a balanced dataset based on the imbalanced dataset of unfixed-length [31]. Jing et al. proposed an improved context encoder network (ICE) for missing wind speed data reconstruction [32]. In the literature [33], an improved auto-encoder (AE) network with a transfer layer was designed to eliminate the effect of SCADA data in the ambiguous status and enhance the reliability of a training dataset.

However, the samples generated by AE are often very fuzzy because there is no advanced discriminant network, and GAN has problems such as unstable training and mode collapse. Therefore, the two are combined to generate data to achieve better results, such as VAE-GAN [34], etc. Bao et al. proposed CVAE-GAN [35], which takes labels as conditional inputs to the model to generate images of specified classification and produced relatively good images in all categories.

Gearboxes are important components for power transmission and speed regulation in mechanical equipment. In wind turbines, the downtime and power loss caused by the failure of gearbox components is the highest among all components. Wind turbine gearboxes operate under variable conditions for long periods of time. Due to the difficulty in obtaining operating data for different operating conditions, the diagnostic accuracy can be low when only data from a single operating condition is used to train the neural network for fault diagnosis. By generating data for unknown operating conditions through GAN and solving the problem of data imbalance, the fault diagnosis accuracy of wind turbine gearboxes can be effectively improved.

In this paper, we proposed a model named transfer learning based on conditional variational generative adversarial networks (TL-CVAE-GAN). An improved CVAE-GAN is used for transfer learning to achieve the generation of unknown samples for wind turbine transmission platforms in different conditions and solve the classification problem of variable conditions data. The known data are used to train CVAE-GAN1, and then the MMD between the known and unknown conditions is calculated. The MMD is added to the loss of CVAE-GAN2, which is an unknown generator, to achieve the generator's domain migration. The problem of data imbalance for wind turbine gearboxes is solved by generating missing data for unknown working conditions via CVAE-GAN2. The raw data and generated data are fed into the classifier to train the model for classification.

The rest of this paper is organized as follows. Section 2 introduces the basic concepts of DA and CVAE-GAN. In Section 3, a novel fault diagnosis model named transfer learning based on conditional variational generative adversarial networks (TL-CVAE-GAN) for a wind turbines testbench is proposed. In Section 4, the wind turbine testbench datasets are input into the proposed model for training and testing, and the results are analyzed. Section 5 presents the conclusion.

2. Conditional Variational Generative Adversarial Networks and Domain Adaptive Technology

2.1. Conditional Variational Generative Adversarial Networks (CVAE-GAN)

The model structure is shown in Figure 1 and includes four parts: encoder network, E, generator network, G, discriminator network, D, and classifier network, C.

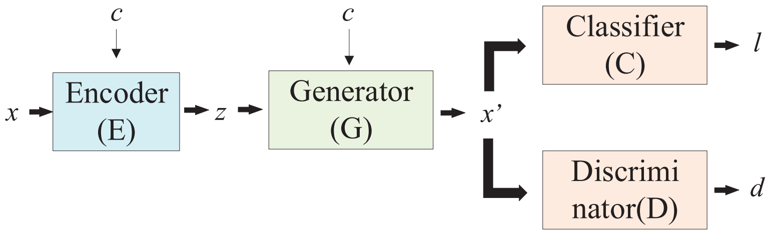


Figure 1. Model structure of CVAE-GAN.

The encoder network, E, maps a sample, x, to a potential representation, z, via a learnable distribution, $P(z\,|\,x,c)$, with c denoting the class of the data. Bounds on the prior $P(z)$ and the recommended distribution are reduced using KL loss:

$$L_{KL} = \frac{1}{2}(-\log\sigma^2 + \mu^2 + \sigma^2 - 1) \quad (1)$$

where μ and σ are the mean and covariance of the output of the potential vector from encoder network E.

The generative network, G, generates the data, x', by sampling from the learnable distribution, $P(x' \mid z,c)$. The functions of G and D are the same as GAN. The network, G, attempts to learn the distribution of the real data by means of gradients from the discriminator network, D, which is able to distinguish between true/false samples. The loss function of the discriminator network, D, is:

$$L_D = -E_{x\sim p_r}[\log D(x)] - E_{z\sim p_z}[\log(1 - D(G(z)))] \tag{2}$$

where x is the input data and z is the potential vector from encoder network, E.

The generator uses an average feature matching the objective function. This objective function requires the feature centers of the synthetic samples to match the feature centers of the real samples. The generator, G, tries to minimize the loss function as:

$$L_{GD} = \frac{1}{2}\left\|E_{x\sim p_r} f_D(x) - E_{z\sim p_z} f_D(G(z))\right\|_2^2 \tag{3}$$

where $f_D(x)$ denotes the features in the middle layer of the discriminator, D.

The generating network, G, uses the average feature to match the objective function. Let the network, G, attempt to minimize:

$$L_{GC} = \frac{1}{2}\sum_c\left\|E_{x\sim p_r} f_C(x) - E_{z\sim p_z} f_C(G(z,c))\right\|_2^2 \tag{4}$$

where $f_C(x)$ denotes the intermediate layer outputs of the classifier and c denotes the label of the input data, x.

Then, an $L2$ reconstruction loss and pairwise feature matching-based loss are added to x and x':

$$L_G = \frac{1}{2}(\|x - x'\|_2^2 + \|f_D(x) - f_D(x')\|_2^2 + \|f_C(x) - f_C(x')\|_2^2) \tag{5}$$

where x is the input data and x' is the generated data from the generator, G.

Network C takes x' as input and outputs a k-dimensional vector, which is then converted to probability-like values using the softmax function. Each port of the output represents the posterior probability, $P(c \mid x')$. In the training phase, network, C, attempts to minimize the softmax loss. The function of the classifier network, C, is to measure the posterior of $P(c \mid x')$:

$$L_C = -E_{x\sim p_r}[\log P(\, c|x')] \tag{6}$$

The total loss function is:

$$L = L_{KL} + L_G + L_{GD} + L_{GC} + L_D + L_C \tag{7}$$

L_{KL} is only relevant to the encoder network, E, indicating whether the distribution of potential vectors is as expected. L_G, L_{GD}, and L_{GC} are relevant to the generator network, G, indicating whether the synthetic sample is the same as the input training samples, the real sample, and other samples in the same category, respectively. L_C is relevant to the classifier network, C, indicating how well the network is used to classify different categories of samples; L_D is relevant to the discriminator network, D, indicating how well the network is able to distinguish between real/synthetic samples. All these objective functions are complementary to each other and ultimately lead to optimal results for the algorithm.

2.2. Domain Adaptive Technology (DA)

As shown in Figure 2, domain adaptation is used to map data features from different domains to the same feature space, so that other domain data can be used to enhance the target domain training. There are two fundamental concepts in domain adaptation: the source domain and the target domain. The source domain, $D_S = \{X_S, P(X_S)\}$, is rich in supervised learning information. The target domain, $D_T = \{X_T, P(X_T)\}$, represents the domain in which the test set is located, usually without labels or with only a few labels. Source and target domains are often the same type of task but are distributed differently.

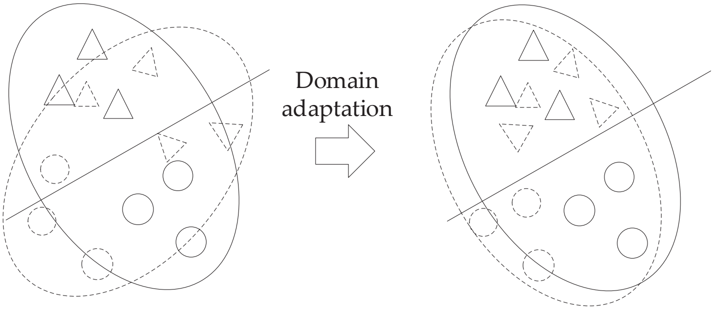


Figure 2. Domain adaptation.

Common domain adaptation methods include:

(1) Sample adaptation: resampling samples in the source domain so that their distribution converges with the target domain distribution.
(2) Feature adaptation: projecting the source and target domains into a common feature subspace.
(3) Model adaption: modification of the source domain error function.

Domain loss is calculated using the maximum mean difference (MMD). To be specific, the transferable features are first mapped into reproduced kernel Hilbert space (RKHS), in which the mean distance between them is viewed as the metric to their distribution discrepancy:

$$MMD(X_S, X_T) = \left\| \frac{1}{|X_S|} \sum \phi(x_s) - \frac{1}{|X_T|} \sum \phi(x_t) \right\| \tag{8}$$

where ϕ is a mapping function, X_S is the source data, and X_T is the target data.

3. Transfer Learning Based on Conditional Variational Generative Adversarial Networks (TL-CVAE-GAN)

In this paper, we proposed a model named transfer learning based conditional variational generative adversarial networks (TL-CVAE-GAN) for fault diagnosis of wind turbine transmission platform datasets under different conditions. An improved CVAE-GAN is used for transfer learning to achieve the generation of unknown samples in different conditions.

As shown in Table 1, the data in this paper include X_{S1}, X_{S2}, X_{T1}, and X_{T2}. X_S is the source domain data and X_T is the target domain data. X_{S1} and X_{T1} are in operating speed 1, X_{S2} and X_{T2} are in operating speed 2. In this paper, X_{T2} is unknown.

Table 1. Variables and conditions of the data.

Domain	Data	Work Condition	Known or Not
Source domain	X_{S1} X_{S2}	$Speed_1$ $Speed_2$	Data available Data available
Target domain	X_{T1} X_{T2}	$Speed_1$ $Speed_2$	Data available Data not available

The model structure of TL-CVAE-GAN is shown in Figure 3. A generative model CVAE-GAN1 is trained to generate $X_{S2'}$ from X_{S1}, and another generative model CVAE-GAN2 is trained to generate $X_{T2'}$ from X_{T1}. The structure of the neural network model is the same for CVAE-GAN1 and CVAE-GAN2, both containing: an encoder, E, a decoder, De, a generator, G, and a discriminator, D.

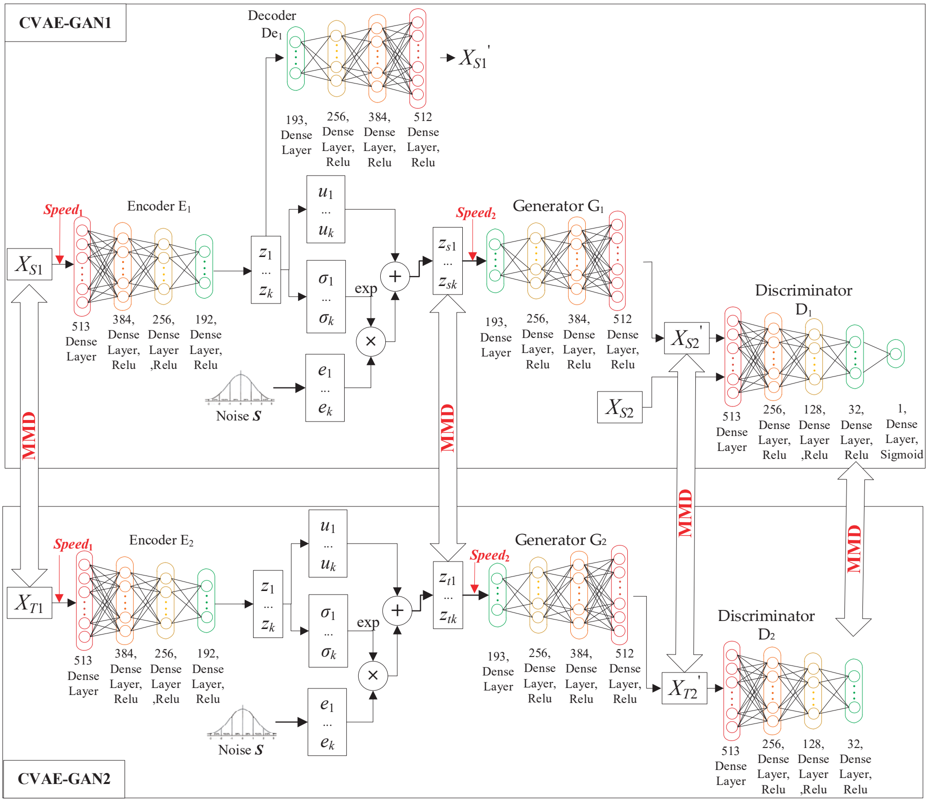


Figure 3. Model structure of TL-CVAE-GAN.

Assume that X_1 is the data corresponding to $Speed_1$ and X_2 is the data corresponding to $Speed_2$. The data, X_1, is fed into the encoder, E, together with $Speed_1$ for dimensionality reduction to obtain an intermediate code, z, with the $Speed_1$ information removed; z is input to the decoder, De, for reconstruction, and the mean, u, and variance, σ, are additionally obtained from z. The variance, σ, is multiplied by the random noise, e, plus the mean, u, to obtain z_s; z_s is fed into the generator, G, together with $Speed_2$ to produce $X_{2'}$. The real X_2 is fed into the discriminator, D, together with the generated $X_{2'}$ for discrimination. The accuracy of the generated model is improved by confronting the generator with the discriminator.

The MMD of X_{S1} to X_{T1} is solved for domain adaptation. The MMD is added to the loss function of the generator CVAE-GAN2 for backpropagation to generate $X_{T2'}$.

The TL-CVAE-GAN model is divided into the following steps:

In the first step, update the parameters of CVAE-GAN1 and generate $X_{S2'}$. The source data X_{S1} and its corresponding rotational speed are input into encoder E to obtain the intermediate key feature, z, with the rotational speed information removed. The mean value, u, and variance, σ, are obtained from z, then a new sample, z_s, is formed by u, σ, and the noise e; z is fed into the decoder De for reconstruction and z_s and the corresponding speed,

speed$_2$, are input to the generator, G, to generate $X_{S2'}$. The discriminator, D, discriminates between the generated data, $X_{S2'}$, and the real data, X_{S2}. The loss of CVAE-GAN1 is:

$$Loss_1 = Loss_{VAE} + Loss_{GAN} \tag{9}$$

$$Loss_{VAE} = \|X_{S1} - X_{S1}'\|_2^2 + \frac{1}{2}(-\log\sigma^2 + \mu^2 + \sigma^2 - 1) \tag{10}$$

$$Loss_{GAN} = -E_{x\sim p_r}[\log D(X_{S2})] - E_{z\sim p_z}[\log(1 - D(G(z_s, speed_2)))] \tag{11}$$

where X_{S1} is the input source data, $X_{S1'}$ is the generated data from X_{S1}, μ and σ are the mean and covariance of the output of the potential vector from the encoder network, E, X_{S2} is another input source data that the operating conditions are different from X_{S1}, *speed*$_2$ is the operating speed of X_{S2}, and z_s is the potential vector of source data from encoder network, E.

In the second step, update the parameters of CVAE-GAN2. The MMD between CVAE-GAN1 and CVAE-GAN2 is calculated. The MMD is added to the loss of CVAE-GAN2 to achieve the generator's domain migration. The loss is as follows:

$$Loss_2 = MMD(X_{S1}, X_{T1}) + MMD(z_s, z_t) + MMD(X_{S2}', X_{T2}') + MMD(f_D'(X_{S2}'), f_D'(X_{T2}')) \tag{12}$$

where $f_D'(x)$ denotes the features in the penultimate layer of the discriminator, D. X_{T1} and X_{T2} are the target data and z_s and z_t are the potential vector of source data and target data from encoder network E.

In the last step, repeat steps 1 and steps 2 for 5 training cycles to generate the unknown data, $X_{T2'}$.

The model structure of the classifier is shown in Figure 4. The existing data, X_{S1}, X_{S2}, X_{T1}, and the generated data, $X_{T2'}$, are together input into the classifier for training.

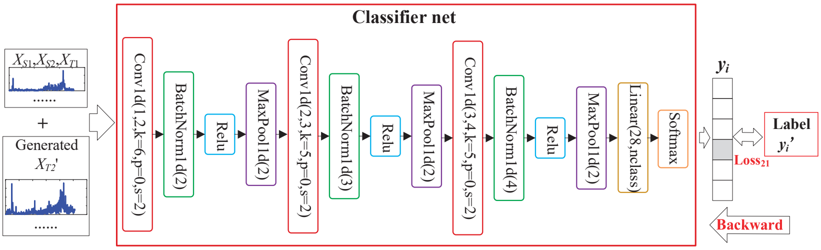


Figure 4. Model structure of the classifier.

The classifier model contains three convolution layers, three BatchNorm1d, three MaxPool1d, and one fully-connected layer. The number of neurons in each layer is shown in Figure 4. The activation function for the last layer is Softmax and the activation functions for the other layers are Relu. All the data goes through the fast Fourier transform, and is then fed into the model. The classifier is updated by:

$$Loss_{21} = \sum_{x_i, y_i \sim T_i} [y_i \log y_i' + (1 - y_i)\log(1 - y_i')] \tag{13}$$

where y_i is the real label of the data and y_i' is the output of the classifier model.

The feature embedding is visualized by t-SNE to test the effectiveness of the proposed model. The complete algorithm flow is shown in Algorithm 1.

Algorithm 1. TL-CVAE-GAN

Input: Input training data, $Tr = \{(X_{S1}, Y_{S1}), (X_{S2}, Y_{S2}), (X_{T1}, Y_{T1})\}$, testing data, Te, classified model, f_C. In the CVAE-GAN1 model: encoder network, f_{E1}, decoder network, f_{DE1}, generator network, f_{G1}, and discriminator network, f_{D1}. In the CVAE-GAN2 model: encoder network, f_{E2}, generator network, f_{G2}, discriminator network, f_{D2}.The learning rate, lr.

```
########################Cycle 5 times ####################
For f from 0 to 4:
########################train CVAE-GAN1 model ####################
    For each training epoch, do:
        For each batch, do:
            $z_i = f_{E1}(x_{s1i}, Speed_1)$, $x_{s1i}' = f_{DE1}(z_i)$, the mean value, $u_{si}$, and variance, $\sigma_{si}$, are
          obtained from $z_i$, sample $e$ from the random noise $S$. $z_{si} = u_{si} + \sigma_{si} * e$,
          $x_{s2i}' = f_{G1}(z_{si}, Speed_2)$, $d_{s2i}' = f_{D1}(x_{s2i}')$, $d_{s2i} = f_{D1}(x_{s2i})$
        Backward propagation by Equation (9).
    end
    save CVAE-GAN1 model
#################### train CVAE-GAN2 model use MMD ######################
    download CVAE-GAN1 model. Use the parameters of the CVAE-GAN1 model as the
initial parameters of CVAE-GAN2.
    For each training, do:
        For each batch, do:
            $z_i = f_{E2}(x_{t1i})$, $z_{ti} = u_{ti} + \sigma_{ti} * e$, $x_{t2i}' = f_{G2}(z_{ti})$,
        Backward propagation by Equation (12).
    end
    save CVAE-GAN2 model
    $lr = lr/2$
    if $f > 0$:
            download the CVAE-GAN2 model. Use the parameters of the CVAE-GAN2
        model as the initial parameters of CVAE-GAN1.
end
########### train classifier net use $Tr$ and the generate data $X_{T2'}$ #################
########the input data is X = {$(X_{S1}, Y_{S1})$, $(X_{S2}, Y_{S2})$, $(X_{T1}, Y_{T1})$, $(X_{T2'}, Y_{T2})$}###########
For each training, do:
    For each batch, do:
        $y_i' = f_C(x_i)$
    Backward propagation by Equation (13).
end
###################### testing results and t-SNE ########################
For the test set, calculate $c_{Ti} = f_C\ (Te_i)$, calculate the accuracy, and draw the t-SNE diagram.
```

Output: testing results.

4. Case Analysis

In this section, the data of the wind turbine transmission platform are used to verify our model. The wind turbine transmission platform is shown in Figure 5. It consists of a drive motor, a stator gearbox, a planetary gearbox, and a load device to simulate the vibration state under various gear faults.

The number of teeth of each gear in the drive system is shown in Figure 6. The stator gearbox consists of four gears in a two-stage drive with three shafts. The fault occurred in the intermediate shaft gear. Piezoelectric sensors are placed on the bearing seat at the right end of the intermediate shaft. This paper simulates the multiple faults of a wind turbine gearbox under variable operating conditions. Six fault modes in the stator gearbox are adopted, including normal, cracked, chipped, missing teeth, wear, and eccentricity. The data available is shown in Table 2. The data consists of six categories, with one health category and five fault categories. The first three categories contain data for four operating speeds (38 Hz, 40 Hz, 43 Hz, 45 Hz) and the last three categories only have data for 43 Hz and 45 Hz. The speed is of the driver motor. The data is sampled at a frequency of 8192 Hz; 256 data are available in each category for each working condition.

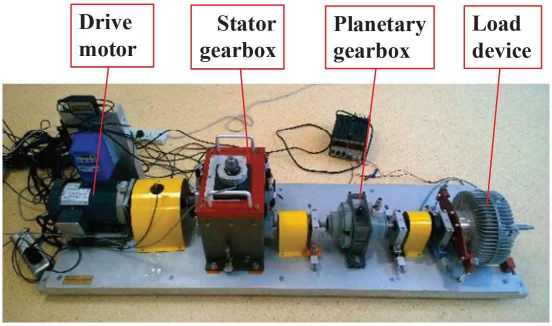


Figure 5. Transmission platform of wind turbine.

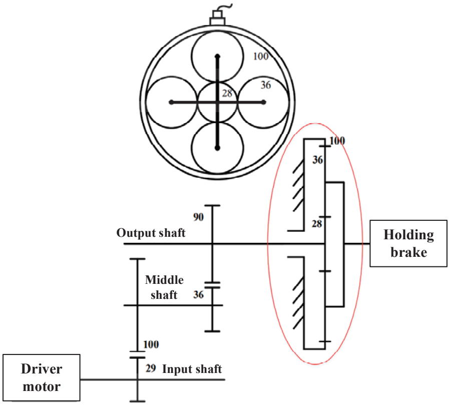


Figure 6. General structure of the gear system.

Table 2. Fault description of wind turbine transmission platform.

Fault Modes	Label	Speed (Hz)	Sampling Frequency	Number of Dataset
Normal	0	38, 40, 43, 45	8192 Hz	256 × 4
Cracked	1	38, 40, 43, 45	8192 Hz	256 × 4
Chipped	2	38, 40, 43, 45	8192 Hz	256 × 4
Missing	3	43, 45	8192 Hz	256 × 2
Wear	4	43, 45	8192 Hz	256 × 2
Eccentricity	5	43, 45	8192 Hz	256 × 2

The data description of the training dataset and testing dataset is shown in Table 3. There are 256 data in each category for each speed. For each class of data under each speed, the first 160 are taken as the training set and all data are testing data. The trained percentages are 62.5%. This case addresses the problem of unbalanced data from the wind turbine transmission platform, generating missing data and improving diagnostic accuracy. Therefore, in this case, the data in categories 3, 4, and 5 where the speed is 38 Hz and 40 Hz are set missing and are not included in the training set.

Table 3. Data description of the training dataset and testing dataset.

	Data	Label	Speed (Hz)	Number of Training Dataset	Number of Testing Dataset
Source domain	X_{S1}	0, 1, 2	43, 45	160 × 3 × 2	256 × 3 × 2
	X_{S2}	0, 1, 2	38, 40	160 × 3 × 2	256 × 3 × 2
Target domain	X_{T1}	3, 4, 5	43, 45	160 × 3 × 2	256 × 3 × 2
	X_{T2}	3, 4, 5	38, 40	0	256 × 3 × 2

For the TL-CVAE-GAN and classifier model, the update function is Adam, the training epochs for the update are 400, and the batch size is 32.

Figures 7–9 show the missing data, $X_{T2'}$, generated by the generator CVAE-GAN2. It can be seen that the generator effectively generates data for the unknown operating conditions (38 Hz, 40 Hz).

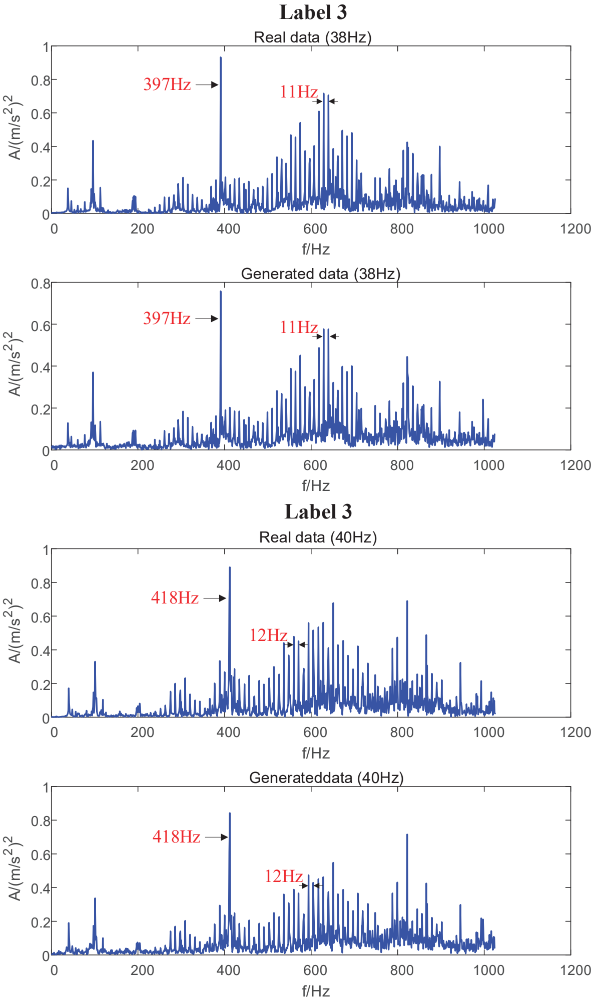


Figure 7. The generated data and its corresponding real data for fault 3 at 38 Hz and 40 Hz.

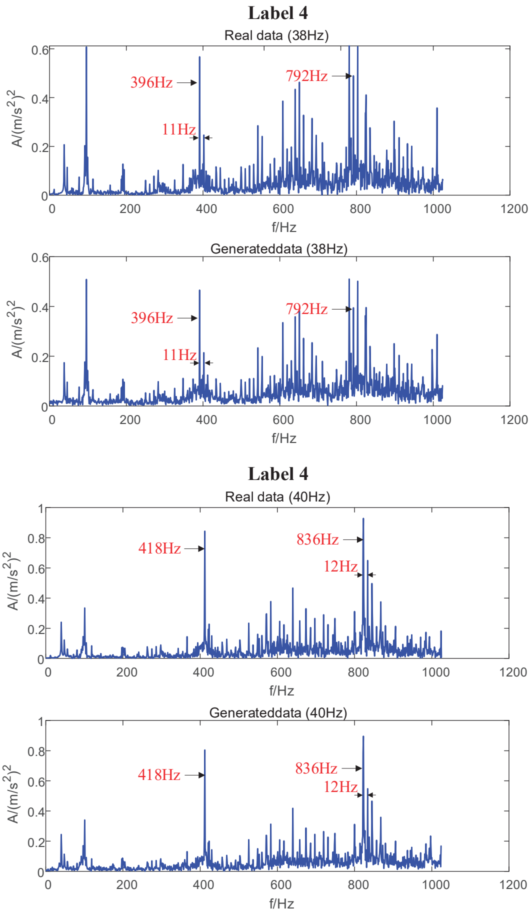


Figure 8. The generated data and its corresponding real data for fault 4 at 38 Hz and 40 Hz.

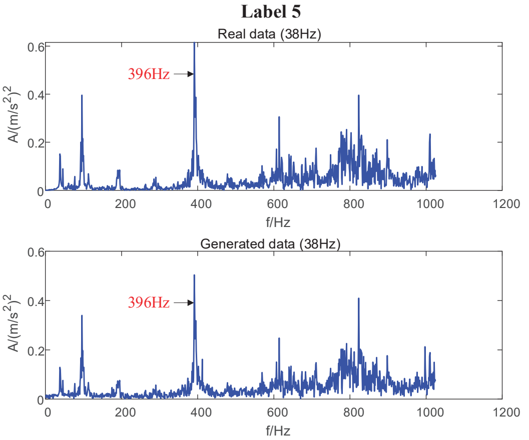


Figure 9. *Cont*.

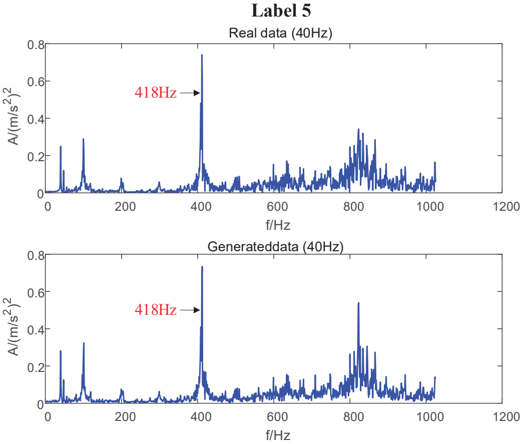


Figure 9. The generated data and its corresponding real data for fault 5 at 38 Hz and 40 Hz.

In this case, it is the pinion of the intermediate shaft that has failed. Therefore, the rotational frequency is given in Equation (14) and the meshing frequency is given in Equation (15).

$$f_r = speed \times 29/100 \quad (14)$$

$$f_m = f_r \times 36 \quad (15)$$

When the operating condition is 38 Hz, the rotational frequency is 397 Hz and the meshing frequency is 11 Hz. When the operating condition is 40 Hz, the rotational frequency is 418 Hz and the meshing frequency is 12 Hz. The rotational and meshing frequency characteristics are evident in both the real data and the generated data. At the same time, there are differences in the frequency spectrum of missing, wear, and eccentric faults.

Figure 7 shows a missing fault. When a gear has a broken tooth, there is a strong shock at the broken tooth for every week the gear rotates, so there are distinct rotational and meshing frequencies present in the frequency spectrum. It is clearly modulated by the rotational frequency throughout the frequency band. The edge band is characterized by a large number of edge frequencies, a wide range, and a uniform and relatively flat distribution. It can be seen that the generated data effectively exhibits these characteristics.

Figure 8 shows a wear fault. The gears are uniformly worn, with a high amplitude sideband at the engagement frequency and its harmonics. The amplitude of the higher harmonics of the meshing frequency is large. In this data, the wear is more severe and the amplitude of the second harmonic has exceeded the amplitude of the fundamental wave of the meshing frequency. It can be seen that the generated data effectively exhibits these characteristics.

Figure 9 shows an eccentric fault. This data has only eccentricity, no faulty gears, so there are no sidebands at the meshing frequency. It can be seen that the generated data effectively exhibits these characteristics.

The generated data for the unknown working conditions are trained together with the known data for the classifier. We compared the classification accuracy of the trained model using only the training set and the training set with the generated unknown data. For better comparison, the same classifier, the same number of training epochs, and the same learning rate were used for both cases. The obtained fault classification accuracy and t-SNE is shown in Figure 10. The comparison of classification accuracy with and without the addition of generated data is shown in Table 4. It can be seen that after the data generated by TL-CVAE-GAN with unknown working conditions were added to the training set, the test accuracy of the trained classifier was improved by 21.3%.

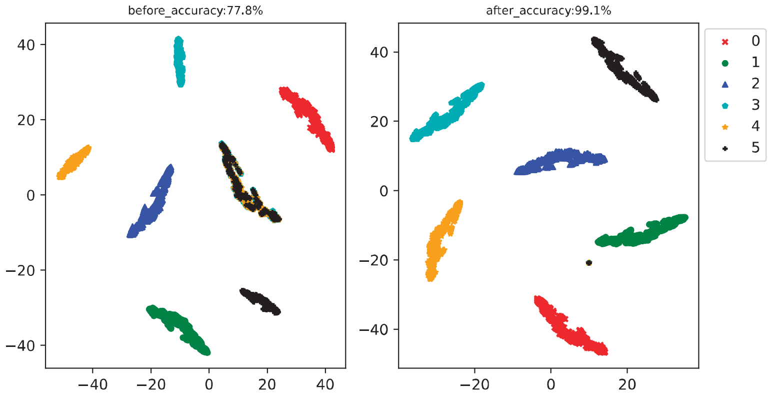


Figure 10. The t-SNE of the classify using only the training set and the training set with the generated unknown data.

Table 4. Comparison of classification accuracy with and without the addition of generated data.

	Only the Training Set Trains the Classifier	Training Set and Generated Data together to Train the Classifier	Improved
Classification accuracy	77.8%	99.1%	21.3%

5. Conclusions

Fault diagnosis of wind turbines plays an important role in improving the reliability of wind turbines. However, the operating conditions of wind turbines vary randomly, and data on different operating conditions are not easily available.

In this paper, the wind turbine transmission platform data is supplemented by the generation of data for unknown operating conditions, which in turn improves the classification accuracy. The proposed TL-CVAE-GAN model combines the better performance of CVAE-GAN in generating samples with the idea of domain adaptive migration. It achieves the generation of unknown samples for wind turbine transmission platforms in different conditions and solves the classification problem of variable conditions data. Work conditions are input to the model as conditions, and the generation of data in different work conditions between similar classes is achieved by domain migration. The known data are used to train CVAE-GAN1. In CVAE-GAN, the known working conditions are fed into the encoder as conditional information to obtain the intermediate key information for the removal of the working conditions. The intermediate key information and the unknown conditions are fed together into the generator to generate the same class of data for the unknown conditions. The generation can be improved by confronting the generator with the discriminator.

The MMD between the known and unknown conditions is then calculated. The MMD is added to the loss of CVAE-GAN2, which is an unknown generator, to achieve the generator's domain migration. The problem of data imbalance for wind turbine gearboxes is solved by generating missing data for unknown working conditions via CVAE-GAN2. The raw data and generated data are fed into the classifier to train the model for classification.

The results show that the proposed model, TL-CVAE-GAN, effectively generates data for unknown working conditions. After the generated data of unknown operating conditions were added to the training set as a supplement, the test accuracy of the trained classifier was improved by 21.3%, effectively improving the fault diagnosis accuracy under-sample imbalance. The model can better solve the problem of fault diagnosis of wind turbines with variable operating conditions.

Author Contributions: Writing—original draft, X.L.; Writing—review & editing, H.M. and Y.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Li, W.; Chen, Z.; He, G. A novel weighted adversarial transfer network for partial domain fault diagnosis of machinery. *IEEE Trans. Ind. Inform.* **2020**, *17*, 1753–1762. [CrossRef]
2. Huang, R.; Li, J.; Liao, Y.; Chen, J.; Wang, Z.; Liu, W. Deep adversarial capsule network for compound fault diagnosis of machinery toward multidomain generalization Task. *IEEE Trans. Instrum. Meas.* **2020**, *70*, 3506311. [CrossRef]
3. Li, J.; Huang, R.; He, G.; Liao, Y.; Wang, Z.; Li, W. A two-stage transfer adversarial network for intelligent fault diagnosis of rotating machinery with multiple new faults. *IEEE ASME Trans. Mechatron.* **2020**, *26*, 1591–1601. [CrossRef]
4. Chen, Z.; Gryllias, K.; Li, W. Intelligent fault diagnosis for rotary machinery using transferable convolutional neural network. *IEEE Trans. Ind. Inform.* **2020**, *16*, 339–349. [CrossRef]
5. Li, J.; Huang, R.; He, G.; Wang, S.; Li, G.; Li, W. A deep adversarial transfer learning network for machinery emerging fault detection. *IEEE Sens. J.* **2020**, *20*, 8413–8422. [CrossRef]
6. Li, S.; An, Z.; Lu, J. A novel data-driven fault feature separation method and its application on intelligent fault diagnosis under variable working conditions. *IEEE Access* **2020**, *8*, 113702–113712. [CrossRef]
7. Qian, W.; Li, S.; Yi, P.; Zhang, K. A novel transfer learning method for robust fault diagnosis of rotating machines under variable working conditions. *Measurement* **2019**, *138*, 514–525. [CrossRef]
8. Guo, L.; Lei, Y.; Xing, S.; Yan, T.; Li, N. Deep convolutional transfer learning network: A new method for intelligent fault diagnosis of machines with unlabeled data. *IEEE Trans. Ind. Electron.* **2019**, *66*, 7316–7325. [CrossRef]
9. Yang, B.; Lei, Y.; Jia, F.; Xing, S. An intelligent fault diagnosis approach based on transfer learning from laboratory bearings to locomotive bearings. *Mech. Syst. Signal Process.* **2019**, *122*, 692–706. [CrossRef]
10. Gretton, A.; Borgwardt, K.M.; Rasch, M.J.; Schölkopf, B.; Smola, A. A kernel two-sample test. *Mach. Learn. Res.* **2012**, *13*, 723–773.
11. Venkateswara, H.; Chakraborty, S.; Panchanathan, S. Deep-learning systems for domain adaptation in computer vision learning transferable feature representations. *IEEE Signal Process.* **2017**, *34*, 117–129. [CrossRef]
12. Sun, S.; Zhang, B.; Xie, L.; Zhang, Y. An unsupervised deep domain adaptation approach for robust speech recognition. *Neurocomputing* **2017**, *257*, 79–87. [CrossRef]
13. Zhang, Z.; Chen, H.; Li, S.; An, Z. Unsupervised domain adaptation via enhanced transfer joint matching for bearing fault diagnosis. *Measurement* **2020**, *165*, 108071. [CrossRef]
14. Zhang, Z.; Chen, H.; Li, S.; An, Z.; Wang, J. A novel geodesic flow kernel based domain adaptation approach for intelligent fault diagnosis under varying working conditions. *Neurocomputing* **2019**, *376*, 54–64. [CrossRef]
15. An, Z.; Li, S.; Jiang, X.; Xin, Y.; Wang, J. Adaptive cross-domain feature extraction method and its application on machinery intelligent fault diagnosis under different working conditions. *IEEE Access* **2019**, *8*, 535–546. [CrossRef]
16. Qian, W.; Li, S.; Jiang, X. Deep transfer network for rotating machine fault analysis. *Pattern Recognit.* **2019**, *96*, 106993. [CrossRef]
17. Yang, B.; Lei, Y.; Jia, F.; Li, N.; Du, Z. A polynomial kernel induced distance metric to improve deep transfer learning for fault diagnosis of machines. *IEEE Trans. Ind. Electron.* **2019**, *67*, 9747–9757. [CrossRef]
18. Goodfellow, I.J.; Pouget-Abadie, J.; Mirza, M.; Xu, B. Generative adversarial nets. In Proceedings of the International Conference on Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014; pp. 1–9.
19. Zheng, T.; Song, L.; Wang, J.; Teng, W.; Xu, X. Data synthesis using dual discriminator conditional generative adversarial networks for imbalanced fault diagnosis of rolling bearings. *Measurement* **2020**, *158*, 107741. [CrossRef]
20. Wang, J.; Li, S.; Han, B.; An, Z.; Bao, H.; Ji, S. Generalization of deep neural networks for imbalanced fault classification of machinery using generative adversarial networks. *IEEE Access* **2019**, *7*, 111168–111180. [CrossRef]
21. Schlegl, T.; Seebck, P.; Waldstein, S.M.; Schmidt-Erfurth, U.; Langs, G. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. *Comput. Vis. Pattern Recognit.* **2017**, *10265*, 146–157.
22. Akcay, S.; Atapour-Abarghouei, A.; Breckon, T.P. GANomaly: Semi-supervised anomaly detection via adversarial training. *Comput. Vis. Pattern Recognit.* **2018**, *11363*, 622–637.
23. Guo, Q.; Li, Y.; Song, Y.; Wang, D.; Chen, W. Intelligent fault diagnosis method based on full 1-D convolutional generative adversarial network. *IEEE Trans. Ind. Inform.* **2019**, *16*, 2044–2053. [CrossRef]
24. Lyu, Y.; Han, Z.; Zhong, J.; Li, C.; Liu, Z. A generic anomaly detection of catenary support components based on generative adversarial networks. *IEEE Trans. Instrum. Meas.* **2020**, *69*, 2439–2448. [CrossRef]
25. Yan, K.; Su, J.; Huang, J.; Mo, Y. Chiller fault diagnosis based on VAE-enabled generative adversarial networks. *IEEE Trans. Autom. Sci. Eng.* **2020**, *19*, 387–395. [CrossRef]
26. Liu, S.; Jiang, H.; Wu, Z.; Li, X. Data synthesis using deep feature enhanced generative adversarial networks for rolling bearing imbalanced fault diagnosis. *Mech. Syst. Signal Process.* **2022**, *163*, 108139. [CrossRef]

27. Kingma, D.P.; Welling, M. Auto-encoding variational bayes. *arXiv* **2014**, arXiv:1312.6114.
28. Makhzani, A.; Shlens, J.; Jaitly, N.; Goodfellow, I.; Frey, B. Adversarial autoencoders. *arXiv* **2015**, arXiv:1511.05644.
29. Qu, F.; Liu, J.; Ma, Y.; Zang, D.; Fu, M. A novel wind turbine data imputation method with multiple optimizations based on GANs. *Mech. Syst. Signal Process.* **2020**, *139*, 106610. [CrossRef]
30. Guo, Z.; Pu, Z.; Du, W.; Wang, H.; Li, C. Improved adversarial learning for fault feature generation of wind turbine gearbox. *Renew. Energy* **2022**, *185*, 255–266. [CrossRef]
31. Jiang, N.; Li, N. A wind turbine frequent principal fault detection and localization approach with imbalanced data using an improved synthetic oversampling technique. *Int. J. Electr. Power Energy Syst.* **2021**, *126*, 106595. [CrossRef]
32. Jing, B.; Pei, Y.; Qian, Z.; Wang, A.; Zhu, S.; An, J. Missing wind speed data reconstruction with improved context encoder network. *Energy Rep.* **2022**, *8*, 3386–3394. [CrossRef]
33. Wang, A.; Qian, Z.; Pei, Y.; Jing, B. A de-ambiguous condition monitoring scheme for wind turbines using least squares generative adversarial networks. *Renew. Energy* **2022**, *185*, 267–279. [CrossRef]
34. Yu, X.; Zhang, X.; Cao, Y.; Xia, M. VAEGAN: A collaborative filtering framework based on adversarial variational autoencoders. In Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI-19), Macao, China, 10–16 August 2019; pp. 4206–4212.
35. Bao, J.; Chen, D.; Wen, F.; Li, H.; Hua, G. CVAE-GAN: Fine-grained image generation through asymmetric training. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2745–2754.

applied sciences

MDPI

Article

Comparison of Feedforward Perceptron Network with LSTM for Solar Cell Radiation Prediction

Tugba Ozdemir [1,2,*], Fatma Taher [3], Babajide O. Ayinde [2], Jacek M. Zurada [2,4] and Ozge Tuzun Ozmen [1,5]

[1] Department of Physics, Faculty of Arts and Sciences, Duzce University, Konuralp Yerleskesi, Duzce 81620, Turkey; ozge.ozmen@bakircay.edu.tr
[2] Electrical and Computer Engineering, University of Louisville, Louisville, KY 40292, USA; babajide.ayinde@echonous.com (B.O.A.); jacek.zurada@louisville.edu (J.M.Z.)
[3] Department of Computing & Applied Technology & Assistant Dean for Research and Out Reach in the College of Technological Innovation, Zayed University, Dubai 19282, United Arab Emirates; fatma.taher@zu.ac.ae
[4] Information Technology Institute, University of Social Science, 90-113 Łódz, Poland
[5] Department of Fundamental Sciences, İzmir Bakırçay University, Izmir 35665, Turkey
* Correspondence: tugbaozdemir238@gmail.com

Citation: Ozdemir, T.; Taher, F.; Ayinde, B.O.; Zurada, J.M.; Tuzun Ozmen, O. Comparison of Feedforward Perceptron Network with LSTM for Solar Cell Radiation Prediction. *Appl. Sci.* **2022**, *12*, 4463. https://doi.org/10.3390/app12094463

Academic Editor: Luis Hernández-Callejo

Received: 9 January 2022
Accepted: 21 March 2022
Published: 28 April 2022

Abstract: Intermittency of electrical power in developing countries, as well as some European countries such as Turkey, can be eluded by taking advantage of solar energy. Correct prediction of solar radiation constitutes a very important step to take advantage of PV solar panels. We propose an experimental study to predict the amount of solar radiation using a classical artificial neural network (ANN) and deep learning methods. PV panel and solar radiation data were collected at Duzce University in Turkey. Moreover, we included meteorological data collected from the Meteorological Ministry of Turkey in Duzce. Data were collected on a daily basis with a 5-min interval. Data were cleaned and preprocessed to train long-short-term memory (LSTM) and ANN models to predict the solar radiation amount of one day ahead. Models were evaluated using coefficient of determination (R^2), mean square error (MSE), root mean squared error (RMSE), mean absolute error (MAE), and mean biased error (MBE). LSTM outperformed ANN with R^2, MSE, RMSE, MAE, and MBE of 0.93, 0.008, 0.089, 0.17, and 0.09, respectively. Moreover, we compared our results with two similar studies in the literature. The proposed study paves the way for utilizing renewable energy by leveraging the usage of PV panels.

Keywords: renewable energy; solar energy; artificial neural network; deep learning; LSTM; radiation prediction

1. Introduction

1.1. Background

In recent years, the role of energy in the life standard of human beings has been vitally important [1–3]. As the human population increases, energy demands increase exponentially [2–5]. Researchers demonstrate that the energy demand is anticipated to be approximately 1.5–3 times by 2050 [2,6,7]. Given that fact, we can anticipate that fossil fuels such as petroleum, natural gas, and coal, which are the traditional energy sources, will be depleted very soon. One more reason to switch to renewable energy is how harmful the fossil fuels are to the environment [4,8]. It should be emphasized that consumption of energy from fossil fuels is increasing CO_2 (carbon dioxide) and greenhouse gas (GHG) emissions all over the world [6,9]. Increasing GHGs cause a rising atmospheric temperature of the Earth's surface [7–13]. With this concern, renewable energy has come into question for the last century [2–5,7–13].

Alternatively, solar energy, which is among renewable energy sources, is abundant and environmentally friendly, and photovoltaic (PV) technology has provided development and discovery for both rural and urban choices on a global scale [5,9–17]. The history of modern PV energy is based on Alexandre Becquerel's 1839 observation of the photoelectric effect [13–17]. However, after the 1990s, studies on PV energy rapidly improved [5,18]. In addition, annual PV solar energy exceeded that of wind power for the first time and reached about 70 GW, and was even 50% higher than in the previous year [18]. The global solar PV capacity reached at least 303 GW (48% compared to 2015) at the end of 2016 [18,19]. Furthermore, reports from the world's solar photovoltaic electricity supplies anticipate that PV technologies will increase to 345 GW and 1081 GW by 2020 and 2030, respectively [1,5,12,19].

The rapid expansion of PV systems does not only provide economic benefits to the electrical systems but also contributes to the reduction of global heating problems [19]. Although a solar PV system can operate by itself, a grid-connected system is required in order to reliably evaluate the electricity generation system [20,21]. Nonetheless, the instability of weather conditions and solar radiation lead to the instability of the power produced by PV panels, which causes a lot of problems in the control and operation of grid-connected PV panels [22–24]. To solve the instability problems, researchers have been developing methods to predict the output power of PV panels based on historical data and meteorological data [25,26]. Recently, artificial neural networks (ANNs) have been used to improve the prediction power of PV panels' output. ANNs have been utilized to solve further problems such as estimating radiation amount, solar power, and ambient temperature parameters [26,27]. ANNs have been applied for the modeling, identification, optimization, prediction, and control of complex systems. Hence, several studies report using ANNs in solar radiation modeling and prediction. Most of those studies utilized the geographical coordinate and meteorological data such as relative humidity, air temperature, pressure, sunshine duration, etc. as an input to the ANNs for estimating of global solar radiation [26,27]. In the following subsection, we go through some of the relevant literature to demonstrate the attempts to predict solar radiation using machine learning.

1.2. Literature Review

Table 1 covers the literature review section of this paper. In the following table, we mention the authors, cities at which the data was collected, the research aim, date when the data was acquired, the models utilized for achieving the research aim, and last but not least, the performance of each model.

Table 1. Literature review.

Authors and Reference	Case Study	Research Objective	Data	Models Used	Performance of Models
A. Mellit et al. [28]	Trieste, Italy	Estimate the amount of solar radiation for 24h using grid-connected photovoltaic plants (GCPV).	From July 1st 2008 to May 23rd 2009 for solar radiation, from November 23rd 2009 to January 24th 2010 for air temperature data.	ANN	The correlation coefficient was 98–99% for sunny days and 94–96% for cloudy days.
C. Voyant et al. [29]	Mediterranean Sea: Ajaccio, Bastia, Montpellier, Marseille, and Nice	Estimate the global solar radiation with two models.	Data on an hourly basis from October 2002 to December 2008 and from French meteorological organization.	• ARMA/ANN hybrid model, • the numerical weather prediction model (ALADIN).	• The nRMSE for hybrid model MLP/ARMA was 14.9% compared to 26.2% for the naïve persistence predictor.
A. Sozen et al. [30]	17 different cities in Turkey	• Estimate the solar potential based on geographic coordinates meteorological data (and the corresponding month) as inputs to the network.	The data were collected from 17 meteorological stations between 2000 and 2002.	ANNs	• MAPE (mean absolute percentage error) was found to be less than 6.735%. • R^2 was found to be about 99.893% for the testing stations.
A. Mellit et al. [31]	In Tahifet, south Algeria	They presented an application of an RNN-based approach to estimate the daily electricity generation of a photovoltaic power system (PVPS).	The measured weather data and the output of electrical signals (voltage and current) were recorded at the PVPS station in Algeria from 1992 to 1997.	ANN and RNN	• MAPE was lower than 5.5%. • The correlation coefficient ranged between 95 and 97%.
J. M. S. de Araujo [32]	Gifu, Japan	For hourly solar radiation prediction.	• The dataset from the NOMADS website. • Three years' data of solar radiation from 1st January 2014 to 31st December 2016 for LSTM.	LSTMWRF (weather research and forecasting)	• LSTM algorithm was 310 W m^{-2} higher compared to 210 W m^{-2} from the WRF model. • The error of WRF was 19% lower compared to 28% of LSTM for the nRMSE error metric.

Table 1. *Cont.*

Authors and Reference	Case Study	Research Objective	Data	Models Used	Performance of Models
A. Alzahrani et al. [33]	Canada	Estimate solar irradiance using a deep neural network.	The data were recorded for four days, from Canada's natural sources.	• Deep recurrent neural networks (LSTM) • Support vector regression (SVR) • Feedforward neural networks (FNN)	RMSE: • LSTM = 0.086 • SVR = 0.11 • FNN = 0.16 MBE: • LSTM = 0.004 • SVR = 0.0042 • FNN = 0.005
A. Rai et al. [34]	Three different geographical regions in different climatic zones.	For midterm solar radiation estimation.	The data came from three different geographical regions in different climatic zones between 2014 and 2015 years.	• A convolution neural network (CNN) • Bi-direction long-short-term memory (BiLSTM)-based hybrid deep learning (DL) model.	For CNN-BiLSTM • R^2 = 0.924 • MAE = 0.0397
J. H. Yousif et al. [35]	Many different locations around the world.	Some different ANN techniques to estimate the photovoltaic thermal (PV/T) energy.	Data were taken from 2008–2017 for locations with different latitudes and climates.	Some models: • Bayesian neural network (BNN) • RNN • Generalized feed-forward (GFF) • MLP • LSTM	They gave error results such as MAPE, MSE, RMSE, MBE, MPE, and R^2.
Y. Jung et al. [36]	South Korea	To predict the amount of PV solar power.	The data were obtained from 164 PV plants for 63 months.	RNN-LSTM	• RMSE = 7.416% • MAPE = 10.805% for the testing data
M. Mishra et al. [37]	Urbana Champaign, Illinois	To forecast a short-term solar power using various time intervals (1 day, 15 days, 30 days, 60 days ahead forecasting).	The datasets from February 2016–August 2017 and September 2017–October 2017.	• Wavelet transform (WT)-based DLM • LSTM-Dropout • Linear regression (LR), • Some other models	They gave error results such as RMSE, MAE, MAPE, and R^2.
S. Ghimire et al. [38]	Australia	Propose a convolutional long-short-term memory (CLSTM) neural network hybrid model to predict half-hourly global solar radiation (GSR).	Data from 1 January 2006 to 31 August 2018.	Some models: • Convolutional neural networks (CNN) • LSTM • Gated recurrent unit (GRU).	• Relative root mean square error (≈1.515%) • Mean absolute percentage error (≈4.672%) • Absolute percentage bias (≈1.233%)
D. Lee et al. [39]	Gumi city in South Korea	Build three different deep learning models to predict the solar power output of PV panels.	Data were a PV power output dataset for 39 months (from 1 June 2013 to 31 August 2016) from a PV operator located in Gumi city in South Korea.	• ANN • DNN LSTM	LSTM-based model performs better by more than 50% compared to the conventional statistical models in terms of mean absolute error.
Z. Pang et al. [40]	Tuscaloosa, Alabama, United States	Create two models using a shallow ANN and an RNN to estimate the solar radiation.	The data utilized wereonly meteorologicaldata from a localweather station in Tuscaloosa, Alabama, United States	• A shallow ANN an RNN	They gave error results of RMSE and NMBE for both models.

1.3. The Proposed Study

Based on the aforementioned literature review, we found that data from PV panels and/or meteorological data are utilized to predict solar radiations. The highest achievable results were found by deep learning techniques [28,31,36–44]. Therefore, we designed our experiment based on shallow and deep learning models. The motivation behind the proposed study was the irregularity of energy delivery in Duzce city in Turkey, which may exist in similar cities around the world. We utilized both PV historical data, which was collected from the city of Duzce in Turkey for the period between 2014 to 2018, as well as the daily meteorological data for the same period. In the proposed study, we compared between a deep ANN and an LSTM model in terms of predicting the solar radiation in the city of Duzce in Turkey on daily basis. We performed hyperparameter optimization at predefined hyperparameter values for both the networks, ANN and LSTM. Selecting a deep learning architecture to perform an accurate prediction of the solar radiation amount is crucial for the system operators to reduce costs and uncertainties [17,41–44]. The main contributions of the proposed work can be summarized as: (i) conducting a comparison between the performance of the most common deep learning models in the literature, (ii) building an LSTM to accurately predict the solar radiation at the city of Duzce in Turkey with the potential to be generalized to more cities around the world, and (iii) conducting a comparison between our results in terms of the coefficient of determination (R^2), root mean squared error (RMSE), mean biased error (MBE), and mean absolute error (MAE).

2. Materials and Methods

2.1. Dataset

The solar data, which were utilized in the current study, were collected from three different types of grid-connected PV panels. The PV panels were installed on the top roof of the University of Duzce Scientific and Technological Research and Application Center (DUBIT) by Duzce University Clean Energy Resources Application and Research Center (DÜTEM) in 2013 in Turkey. The geographic location of the center panel is 40°54′14.7″ N and 31°10′56.7″ E. Figure 1 shows the three different PV solar panels of schemas in DUBIT in Duzce University in Duzce.

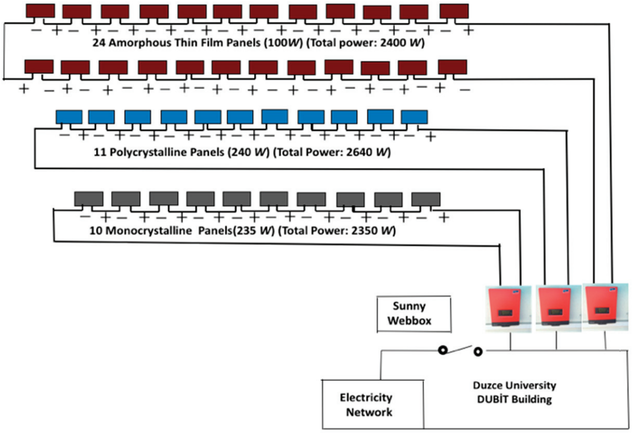

24 Amorphous Thin Film Panels (100*W*) (Total power: 2400 *W*)
11 Polycrystalline Panels (240 *W*) (Total Power: 2640 *W*)
10 Monocrystalline Panels(235 *W*) (Total Power: 2350 *W*)
Sunny Webbox
Electricity Network
Duzce University DUBİT Building

Figure 1. Three different PV solar panels of schemas in DUBIT in Duzce.

As shown in Figure 1, the first type of panels used (P1) is an amorphous thin film silicon panel. A single P1 panel has the power of 100 W. In the proposed study, 24 P1 panels were utilized. The 24 P1 panels were structured in the form of a matrix with two rows and twelve columns (2 × 12). The total output power generated by the (P1) panels matrix equals 2400 W. The second type of panel (P2) is a polycrystalline silicon panel. A single P2 has a solar panel power of 240 W. Eleven P2 panels were utilized in the current study. The 11 (P2) panels were placed as a single row. That row produces a total power output of 2400 W. The third type of panel (P3) is a monocrystalline silicon panel. P3 produces a solar panel power of 235 W. Ten P3 panels were placed in a single row. Those have a total power output 2350 W. That system of panels (P1, P2, and P3) has been recording data every 5 min since 2013. Output power is recorded for each panel. Average temperature, radiation amount, and average atmospheric temperature were recorded for all panels.

Table 2 demonstrates an example of the recorded data recorded from P1, P2, and P3. Therefore, for every day, there are 288 rows of data and 6 columns (3 columns denote the output power for each panel type (kWh), 1 column denotes average atmospheric temperature ($\hat{A}$ °C), 1 column denotes radiation amount (W/m^2), and 1 column denotes panel temperature ($\hat{A}$ °C). Rows are indexed with the time of acquisition. Moreover, meteorological data were recorded on daily basis. Thus, for every 288 rows of panels' data, there is a corresponding row of meteorological data. Meteorological data acquired were as follow: daily average relative humidity, daily sunshine time, and daily average cloudiness. Meteorological data were recorded by the Ministry of Metrology in Turkey. The rationale behind using the meteorological data is to include any factor that might be affecting the radiation amount detected by the panels. Some of the meteorological data is presented in Table 3.

Table 2. Data from PV panels recorded every five minutes in DUBIT.

Dates dd.MM.yyyy HH:mm	P_1 Amorphous Thin-Film Silicon (kWh)	P_2 Polycrystalline Silicon (kWh)	P_3 Monocrystalline Silicon (kWh)	Average Atmospheric Temperature (Â °C)	Radiation Amounts (W/m^2)	Panels Temperature (Â °C)
01.01.2014 11:50	454.81	600.56	613.59	7.40	55.00	7.70
01.01.2014 11:55	454.82	600.57	613.60	7.40	56.00	7.70
01.01.2014 12:00	454.83	600.58	613.61	7.40	56.00	7.70
01.01.2014 12:05	454.84	600.58	613.62	7.40	54.00	7.60
01.01.2014 12:10	454.84	600.59	613.62	7.50	53.00	7.60
01.01.2014 12:15	454.85	600.60	613.63	7.50	53.00	7.70
01.01.2014 12:20	454.86	600.61	613.64	7.50	56.00	7.70
01.01.2014 12:25	454.87	600.62	613.65	7.60	56.00	7.80
01.01.2014 12:30	454.88	600.62	613.65	7.50	56.00	7.80

Table 3. An example of daily average cloudiness from meteorological data in the Turkish State Meteorological Service (the numbers indicate rate of average cloudiness).

	Months											
Days	**Jan.**	**Feb.**	**Mar.**	**Apr.**	**May**	**Jun.**	**Jul.**	**Aug.**	**Sept.**	**Oct.**	**Nov.**	**Dec.**
1	8.0	7.0	6.4	1.5	4.6	6.2	4.4	2.5	5.0	1.2	7.0	7.7
2	7.0	1.1	6.9	4.4	5.2	5.8	4.1	5.2	3.8	2.9	7.0	7.0
3	7.0	0.0	5.9	4.3	4.2	6.0	5.3	6.8	5.4	5.7	3.7	6.5
4	6.4	3.1	6.7	2.7	6.1	6.0	6.1	5.2	4.7	7.0	0.8	6.3
5	4.6	0.7	4.6	5.7	6.3	7.0	4.0	4.3	6.8	5.5	1.9	7.2
6	5.4	0.9	4.3	5.8	6.8	7.0	0.6	1.9	6.3	6.7	0.8	6.4
7	3.6	0.2	3.8	5.3	5.4	6.3	1.1	6.1	3.8	3.8	0.5	6.2
8	0.0	0.6	7.9	6.8	6.7	6.3	1.6	5.6	5.4	3.0	3.0	6.4
9	7.6	5.7	8.0	0.8	6.8	4.9	0.0	5.8	4.6	6.6	5.4	6.8
10	8.6	6.4	8.0	1.2	6.6	3.0	1.9	2.7	5.0	4.8	5.9	6.8

Table 3 shows an example of 10 days' data of the meteorological data that were acquired from the Meteorological Ministry of Turkey in 2014 in Duzce, Turkey. Rows represents days, columns represent months, and the values in each cell represent the average cloudiness on that day in that month and recorded average daily cloudiness for 12 months of the year. Similar tables are given for the other meteorological data. Meteorological data corresponding to 4 years from 2014 to 2018 were utilized in the current study [45].

Data were cleaned by removing rows with missing values, then all the data were aggregated in a single table containing the meteorological data along with the panels' data. Python 3.7 and pandas were utilized for data cleaning and manipulation.

2.2. *Deep Neural Network Approaches*

In this section, we present the shallow ANN and deep ANN architectures used for forecasting of solar radiation amount as an output, including conventional multi-layer deep ANN, sequential model, recurrent neural network, and long-short-term memory.

2.2.1. Conventional Deep ANN/Multilayer Perceptron (MLP)

Multilayer perceptron was introduced by Rosenblatt in 1958 as the basic type of neural network and consists of a number of perceptron [46–59]. There is an input layer to receive the data and there is an output layer that determines and predicts the output value in multilayer perceptron. Between the input and output layers, there is a selected number of hidden layers, which is the main processing engine of MLP [42–46,56–59].

As shown in Figure 2, MLP is a simple neural network. Equation (1) is used to calculate the output of a single perceptron or neuron [46,59].

$$output = f\left(\sum_{i}^{inputs}(x_i.w_i + b_i)\right), \tag{1}$$

where x_i is the input of the neuron, w_i is the weight on each connection to the neuron, b_i is the bias, and $f(.)$ is the activation function, for instance, the tanh activation function [16,47,59].

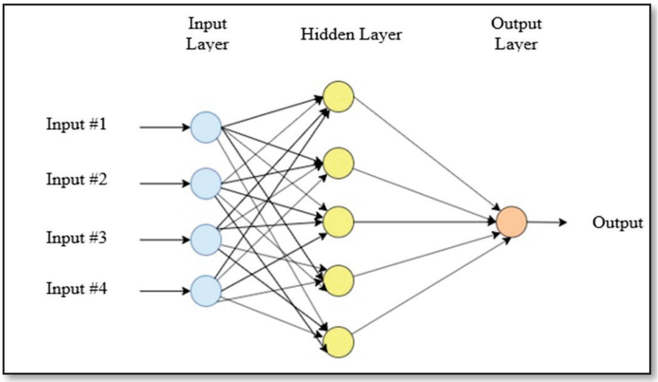


Figure 2. Multilayer perceptron network with single output.

2.2.2. Recurrent Neural Network (RNN)

RNNs are conventional neural networks consisting of one or more feedback loops [47]. RNNs have the ability to utilize their input memory to process entries [48]. In conventional neural networks, all inputs and outputs are considered to be independent of each other. This means that the output is not fed back to the network as an input; however, in the case of RNNs, output can be fed again with the input to be considered in future decisions [47,48]. RNNs' basic architecture is shown in Figure 3.

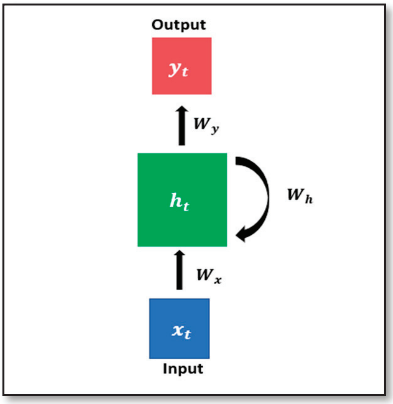


Figure 3. Basic recurrent neural network (RNN).

In Figure 3, the RNN consists of input (x_t), hidden state (h_t), and outputs (y_t). W_x, W_y, and W_h are weight matrices. The most important part of RNN is the hidden state (h_t), which is a vector that can also have an arbitrary dimension [48].

$$h_t = F_w\ (h_{t-1},\ x_t), \tag{2}$$

$$h_t = tanh\left(W_h h_{(t-1)}\ + W_x x_t\right), \tag{3}$$

$$y_t = W_y h_t, \tag{4}$$

Figure 3 also shows the relationship between functions in RNN. In the functions, $h_{(t-1)}$ of previous hidden state contains information from the previous time step; F_w is an activation function as shown Equation (3) [47,48].

2.2.3. Long-Short-Term Memory (LSTM)

LSTM is based on the RNN architecture. It is a model designed to expand the RNN memory [45,46]. This memory has the ability to store information over an arbitrary length of time. There are three gates, which are the input, output, and forget gate, to control the information flow into and out of the neuron's memory [48–51]. Those three gates get the same input as the input neuron. Furthermore, each gate possesses an activation function [41,48,52].

Figure 4 shows the figuration of LSTM at time t. Mathematically, LSTM can be described using the following functions [50–58].

$$f_t = g\left(W_f\, x_t + U_f\, h_{t-1} + b_f\right), \tag{5}$$

$$i_t = g\,(W_i\, x_t + U_i\, h_{t-1} + b_i), \tag{6}$$

$$k_t = tanh\,(W_k\, x_t + U_k\, h_{t-1} + b_k), \tag{7}$$

$$c_t = f_t\, c_{t-1} + i_t\, k_t, \tag{8}$$

$$o_t = g\,(W_o\, x_t + U_o\, h_{t-1} + b_0), \tag{9}$$

$$h_t = o_t\, tanh\,(c_t), \tag{10}$$

where x_t is the input vector at time t and g is an activation function (sigmoid, tanh, or ReLU). W and U are weight matrices, and b is the bias vector. h_t and c_t are output and cell state vector at time t. f_t has been used for remembering old information and it has been used for getting new information [38,49,50,52].

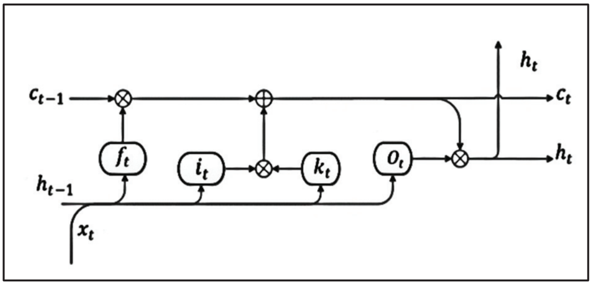


Figure 4. Long-short-term memory (LSTM) at time t.

2.3. Activation Functions

Activation functions are used to add the non-linearity behavior of the ANN [53–56]. Without the activation function, the output of each layer of the ANN would just be the output of a linear model with number of parameters equal to the number of the neurons in each layer [54,55]. Consequently, activation functions increase the overall performance of the ANN and add a nonlinear behavior to it, depending on the behavior of the activation function itself. Thus, if activation functions are not applied on the ANN, the ANN usually has limited performance and acts as a linear regression model [54,55,57,59].

Figure 5 shows the basic structure of the activation function, where x = inputs, w = weights, $f(\Sigma)$ = activation functions, and y = outputs [54,55].

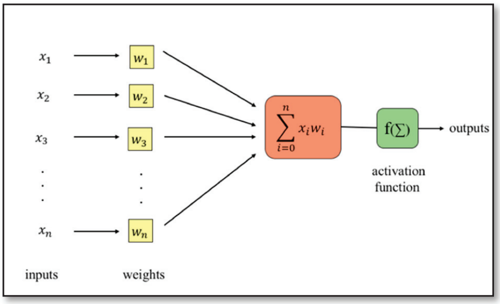


Figure 5. Structure of neural networks with activation function.

The most common activation functions are hyperbolic tangent function, sigmoid function, linear function, ReLU (rectified linear unit) function, leaky-ReLU function, softmax function, and swish (a self-gated) function [55,58,59].

In this work, we use the hyperbolic tangent function (*tanh*) as the activation function for our proposed ANN and DNN models. The tanh function is used for the input and hidden layers.

In hyperbolic tangent function $(-1, 1)$

$$f(x) = tanh(x) = \frac{(e^x - e^{-x}\ )}{(e^x + e^{-x}\ )}\ , \tag{11}$$

However, the rectified linear unit (ReLU) activation function is utilized in the output layer to provide a non-negative solar radiation predictive value [54–59].

ReLU (rectified linear unit) function $[0, \infty)$

$$f(x) = \begin{cases} 0\ for\ x < 0 \\ x\ for\ x \geq 0 \end{cases}, \tag{12}$$

3. Experimental Design

3.1. Dataset Description

Panels data and meteorological data of the full dataset were used in this study as detailed in the methodology. Panels log their power reading during the daytime, i.e., sunrise to sunset. From the sunset to sunrise, the panel does not provide any information about their output power; however, we still have data about their average temperature. Therefore, to maximize the information in our data, we filtered out the period between sunset to sunrise which varies between winter and summer. We added the meteorological data, which consisted of cloudiness, relative humidity, and sun time, to the panels' data. Meteorological data were recorded as one sample per day while panels' data were recorded every 5 min. Therefore, we created three new columns for every panel's data file and assigned to those columns the meteorological values for that day by repeating it n times where n is the number of rows/entries in that panel's data file. In this way, we have built a connection between solar data and meteorological data. We used the same epochs numbers and batch size for two models owing to the comparison.

Deep ANN and LSTM were utilized in this study to predict the daily solar radiation. Inputs were amorphous silicon PV panel in kWh, mono silicon PV panel in kWh, poly silicon PV panel in kWh, average atmospheric temperature in °C, average panel temperature in °C, daily average cloudiness, daily average relative humidity (%), and daily sunshine time in hours, and the output was the predicted radiation amount (W/m^2).

Four years' worth of data were utilized in the proposed study. The data were split into 3 years for training and 1 year for testing. Results in terms of mean square error (MSE) were computed for each model. The training set was split further into a training and validation set for both models. Eventually, the two trained models were evaluated using the testing set. The assumption was that the trained model which has been trained on the 3 years of data can then be used to perform the prediction throughout the 4th year with the same range of error. For both models, we concatenated the daily meteorological data to the data acquired from the PV panels. In order to preserve the data acquired by the panel, each daily metrological value was repeated in the rows corresponding to that day. We trained both models using the aggregated data by averaging every 12 rows (=60 min) and predicting the following 48 row's solar radiation (=predicting the solar radiation after 48 h). The data were then normalized between 0 and 1, and the normalized data were used for the learning process.

3.2. Description of the ANN Model

As shown in Figure 6, we created an ANN model for prediction of radiation amounts. Deep ANN was utilized with a structure of 1 input layer of size 8, 2 hidden layers each of size 50, and a single output layer with size 1. Therefore, we utilized only one row to predict the following 48th row.

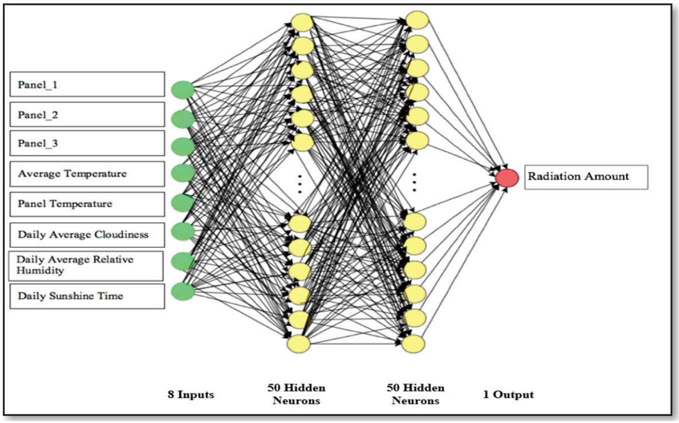


Figure 6. ANN model.

For the model in Figure 6, the tanh function was used as the activation function for the proposed model. Hyperparameter optimization using random grid search was performed on the batch size and the learning rate. Stochastic gradient descent (SGD) was utilized as the optimizing parameter for the deep ANN. The hyperparameters' ranges are specified in the following Table 4.

Table 4. Hyperparameters of the ANN model.

Epochs	500
Batch size	16, 32, 64, 128, 256, 512, 1024
Learning Rate (LR)	[0.0005, 0.05] with step 0.005

Table 4 demonstrates the hyperparameters of the ANN model.

3.3. Description of the LSTM Model

LSTM is an advanced RNN used to specify which feature should be memorized or forgotten when the network is being trained. Therefore, given a sufficient history of features and solar radiation, the LSTM can determine the required history for each feature to provide an accurate solar radiation estimation.

In the proposed LSTM/DNN model, we allowed the LSTM to access up to 30 h in the past in order to predict the solar radiation after 48 h. Hyperparameters optimization via grid search was performed on the hyperparameters shown in Table 5.

Table 5. Hyper parameters of the LSTM model.

Epochs	500
Batch size	16, 32, 64, 128, 256, 512, 1024
n	1, 2, 5, 10, 15, 20, 30 h

The loss function of LSTM was the mean squared error (MSE) and the model was implemented by Keras.

3.4. Error Measures

The performance of the reference methods and the different approaches were evaluated with five different error measures for ANNs. The equation shows the mean-square error (MSE) [50,59].

$$\text{MSE}\left(x', x\right) = \frac{1}{N}\sum_{n=1}^{N}\left(x'_n - x_n\right)^2 \ , \tag{13}$$

In the equations, x is the measured power time series, $x^{2\prime}$ is the predicted power time series, and N denotes the number of samples of the time series [47,48,59].

4. Results and Discussion

The training and texting process is shown in Figure 7. Loss function with 500 epochs, batch size of 256 was used for this ANN model.

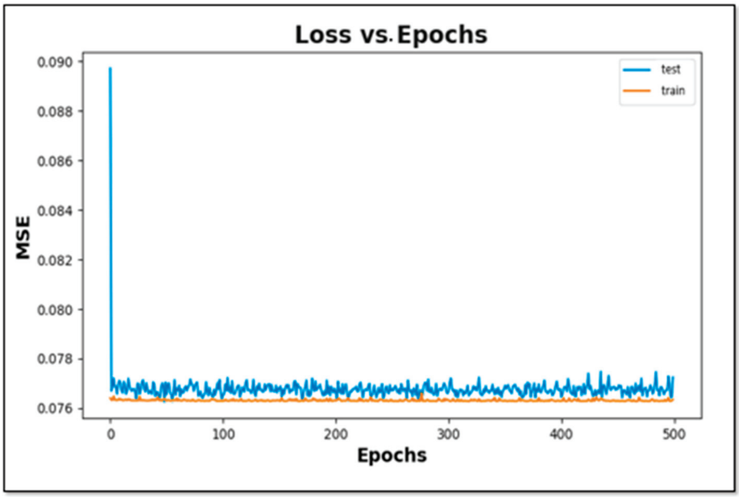


Figure 7. Loss function with 500 epochs for ANN model, batch size of 256.

As shown in Figure 7, the minimum training MSE and minimum testing MSE were 0.0762 and 0.0775, respectively, for the ANN model. The optimum parameters selected for the ANN model were batch size 256 and learning rate 0.01.

Figure 8 shows the graph of training and texting of data with the LSTM model for 500 epochs.

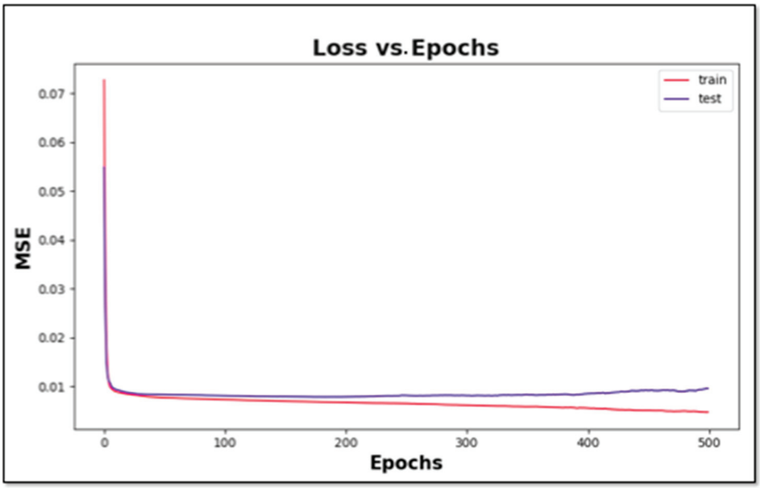


Figure 8. The graph of training and testing of data with LSTM model for 500 epochs.

As demonstrated in Figure 8 for the LSTM model, the minimum training MSE and minimum testing MSE were 0.0049 and 0.0080, respectively. The optimum parameters selected for the ANN model were batch size 256 and past days of 15 h.

Table 6 compares the results obtained by using the ANN model and the LSTM model. Using MSE to calculate the error/loss of the two models, it was found that LSTM improves the results about 18 times in case of training, and about 9 times in case of testing. Since LSTM successfully outperformed ANN by utilizing the data from the previous 15 h, LSTM was the chosen model to test on the 4th year testing data.

Table 6. Error comparison of models after 500 epochs.

Method	MSE	
ANN Model	Minimum training loss	0.0762
	Minimum testing loss	0.0775
LSTM (Deep Learning)	Minimum training loss	0.0049
	Minimum testing loss	0.0080

Figure 9 shows a sample of the prediction performed using the LSTM trained model on 175 days of the 4th year assigned for testing the trained model. Number of days are shown on the horizontal axis versus the normalized solar radiation on the vertical axis. We calculated the coefficient of determination, R^2, along with the MSE for the testing results. R^2 was found to be 0.9365 and MSE was 0.01. In order to demonstrate the significance of our results, we compared our results to similar work in the literature by M. Mishra et al. [37] and U. Agbulut et al. [60]. Moreover, U. Agbulut et al. [60] predicted the solar radiation by using deep learning models for four different cities in Turkey. We averaged the values of their metric scores over the four cities to compare with ours. On the other hand, M. Mishra et al. [37] utilized wavelet transformation on the historical PV solar output at the University of Illinois in Urbana Champaign along with the meteorological data to train LSTM model to perform daily predictions. The authors compared the performance of different ML models to LSTM. Similar to our findings, LSTM outperformed the other models. They trained the models using 18 months of data and tested with one month. It is worth noting that they were performing hourly predictions for 1 day ahead.

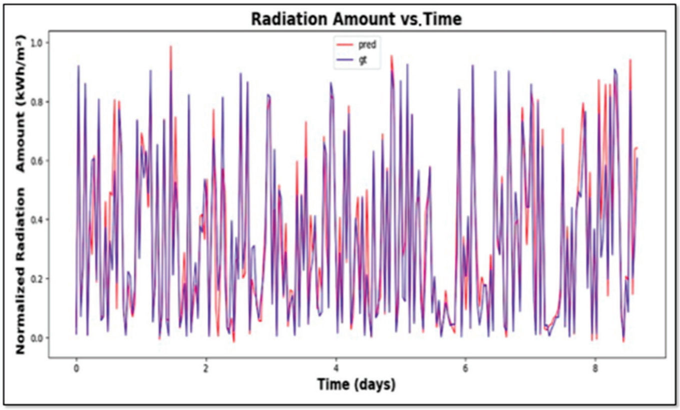


Figure 9. Radiation of 175 days (testing days) as LSTM predicted (blue) versus the ground truth as measured (red).

Table 7 demonstrates the performance of our model with respect to other similar models in the literature. Although we outperformed the other models in terms of R^2, M. Mishra et al. [37] achieved better results in terms of RMSE and MAE. We claim that this higher performance is due to the fact that they performed hourly prediction of the one day ahead and not the whole day. Moreover, we tested on a whole year of data and not only one month.

Table 7. Comparison for ANN.

Metric	[38]	[58]	Proposed Method
R^2	0.426	0.916	0.93
RMSE	0.011	2.138	0.089
MBE	NA	0.3874	0.009
MAE	0.074	1.781	0.17

5. Conclusions and Limitations

In this study, we collected data from three different types of solar panels for the city of Duzce in Turkey and trained an ANN and an LSTM to accurately predict the solar radiation using PV historical data as well as meteorological data. Data were collected for the years between 2014 and 2018 on a daily basis with a 5-min interval. The first model was an ANN model which is frequently used for solar prediction according to the literature. The second model was LSTM which is based on RNNs and is getting more utilization in time series forecasting studies. In the proposed study, we demonstrate the feasibility of accurately predicting solar radiation after 24 h if 15 h of PV historical data along with one previous day of meteorological data are provided to the LSTM. The ability of the LSTM to utilize the historical values of the features allows it to outperform other deep learning models in time series applications. Moreover, we conducted a comparison between our results and similar work in the literature in terms of many error metrics.

Two main limitations of the proposed study would be training the models on data collected solely from the city of Duzce in Turkey. For future work, we plan to collect data from different places in Turkey, or around the globe if possible, to study the generalizability of a trained LSTM model to be used as a prediction tool for solar radiation in different locations. We are aware of the fact that the weather in Duzce is stable most of the time and it perhaps assisted in creating a very accurate model; thus, we are planning to acquire data from places where the weather is more turbulent.

Author Contributions: Conceptualization, T.O. and B.O.A.; methodology, T.O.; software, T.O. and B.O.A.; validation, J.M.Z. and O.T.O.; formal analysis, J.M.Z. and O.T.O.; investigation, T.O.; resources, T.O.; data curation, T.O. and B.O.A.; writing—original draft preparation, T.O.; writing—review and editing, T.O., F.T. and J.M.Z.; visualization, T.O.; supervision, J.M.Z. and O.T.O.; project administration, T.O.; funding acquisition, The Scientific and Technological Research Council of Turkey (TUBITAK). All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by The Scientific and Technological Research Council of Turkey (TUBITAK), grant number 1059B141800505.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data utilized in this study were obtained from the Duzce University using the aforementioned described solar panels. Those solar panels were installed in the Duzce University for scientific research in 2013. Data were recorded and saved in the Duzce University database. Moreover, meteorological data were obtained via a protocol between the physics department, Duzce University, and the Ministry of Metrology in Turkey. Data are available upon request.

Acknowledgments: This study was supported by 1059B141800505 from The Scientific and Technological Research Council of Turkey (TUBITAK).

Conflicts of Interest: There is no conflict of interest.

References

1. Husain, A.A.; Hasan, W.Z.W.; Shafie, S.; Hamidon, M.N.; Pandey, S.S. A review of transparent solar photovoltaic technologies. *Renew. Sustain. Energy Rev.* **2018**, *94*, 779–791. [CrossRef]
2. Shahsavari, A.; Akbari, M. Potential of solar energy in developing countries for reducing energy-related emissions. *Renew. Sustain. Energy Rev.* **2018**, *90*, 275–291. [CrossRef]
3. Wang, Z.; Song, H.; Liu, H.; Ye, J. Coupling of solar energy and thermal energy for carbon dioxide reduction: Status and prospects. *Angew. Chem. Int. Ed.* **2020**, *59*, 8016–8035. [CrossRef]
4. Zhang, Y.; Jing, R.; Yanru, P.; Wang, P. Solar energy potential assessment: A framework to integrate geographic, technological, and economic indices for a potential analysis. *Renew. Energy* **2020**, *149*, 577–586. [CrossRef]
5. Rabaia, M.K.H.; Abdelkareem, M.A.; Sayed, E.T.; Elsaid, K.; Chae, K.; Wilberforce, T.; Olabi, A.G. Environmental impacts of solar energy systems: A review. *Sci. Total Environ.* **2021**, *754*, 141989. [CrossRef]
6. Li, J.; Huang, J. The expansion of China's solar energy: Challenges and policy options. *Renew. Sustain. Energy Rev.* **2020**, *132*, 110002. [CrossRef]
7. Grubler, A.; Wilson, C.; Bento, N.; Boza-Kiss, B.; Krey, V.; McCollum, D.L.; Rao, N.D.; Riahi, K.; Rogelj, J.; De Stercke, S.; et al. A low energy demand scenario for meeting the 1.5 C target and sustainable development goals without negative emission technologies. *Nat. Energy* **2018**, *3*, 515–527. [CrossRef]
8. Shamshirband, S.; Rabczuk, T.; Chau, K.W. A survey of deep learning techniques: Application in wind and solar energy resources. *IEEE Access* **2019**, *7*, 164650–164666. [CrossRef]
9. Bildirici, M.E.; Gökmenoğlu, S.M. Environmental pollution, hydropower energy consumption and economic growth: Evidence from G7 countries. *Renew. Sustain. Energy Rev.* **2017**, *75*, 68–85. [CrossRef]
10. Franco, S.; Mandla, V.R.; Rao, K.R.M. Urbanization, energy consumption and emissions in the Indian context A review. *Renew. Sustain. Energy Rev.* **2017**, *71*, 898–907. [CrossRef]
11. Jebli, M.B.; Youssef, S.B. The role of renewable energy and agriculture in reducing CO_2 emissions: Evidence for North Africa countries. *Ecol. Indic.* **2017**, *74*, 295–301. [CrossRef]
12. Nabavi-Pelesaraei, A.; Hosseinzadeh-Bandbafha, H.; Qasemi-Kordkheili, P.; Kouchaki-Penchah, H.; Riahi-Dorcheh, F. Applying optimization techniques to improve of energy efficiency and GHG (greenhouse gas) emissions of wheat production. *Energy* **2016**, *103*, 672–678. [CrossRef]
13. Jebli, I.; Belouadha, F.Z.; Kabbaj, M.I.; Tilioua, A. Prediction of solar energy guided by pearson correlation using machine learning. *Energy* **2021**, *224*, 120109. [CrossRef]
14. Smil, V. *Energy and Civilization: A History*; MIT Press: Cambridge, MA, USA, 2018.
15. Shezan, S.; Julai, S.; Kibria, M.; Ullah, K.; Saidur, R.; Chong, W.T.; Akikur, R. Performance analysis of an off-grid wind-PV (photovoltaic)-diesel-battery hybrid energy system feasible for remote areas. *J. Clean. Prod.* **2016**, *125*, 121–132. [CrossRef]
16. Antonanzas, J.; Osorio, N.; Escobar, R.; Urraca, R.; Martinez-de-Pison, F.J.; Antonanzas-Torres, F. Review of photovoltaic power forecasting. *Sol. Energy* **2016**, *136*, 78–111. [CrossRef]
17. Ahmad, M.J.; Tiwari, G.N. Solar radiation models—A review. *Int. J. Energy Res.* **2011**, *35*, 271–290. [CrossRef]
18. Pandey, A.K.; Tyagi, V.; Selvaraj, J.A.; Rahim, N.; Tyagi, S. Recent advances in solar photovoltaic systems for emerging trends and advanced applications. *Renew. Sustain. Energy Rev.* **2016**, *53*, 859–884. [CrossRef]

19. Wang, F.; Mi, Z.; Su, S.; Zhao, H. Short-Term Solar Irradiance Forecasting Model Based on Artificial Neural Network Using Statistical Feature Parameters. *Energies* **2012**, *5*, 1355–1370. [CrossRef]
20. Su, Y.; Chan, L.-C.; Shu, L.; Tsui, K.-L. Real-time prediction models for output power and efficiency of grid-connected solar photovoltaic systems. *Appl. Energy* **2012**, *93*, 319–326. [CrossRef]
21. Malvoni, M.; Leggieri, A.; Maggiotto, G.; Congedo, P.; de Giorgi, M.G. Long term performance, losses and efficiency analysis of a 960 kW P photovoltaic system in the Mediterranean climate. *Energy Convers. Manag.* **2017**, *145*, 169–181. [CrossRef]
22. Ghazi, S.; Ip, K. The effect of weather conditions on the efficiency of PV panels in the southeast of UK. *Renew. Energy* **2014**, *69*, 50–59. [CrossRef]
23. Salimi, H.; Ashtiani, H.A.D.; Lavasani, A.M.; Fazaeli, R. Experimental analysis and modeling of weather condition effects on photovoltaic systems' performance: Tehran case study. *Energy Sources Part A Recovery Util. Environ. Eff.* **2020**, *7*, 1–13. [CrossRef]
24. Hachicha, A.A.; Al-Sawafta, I.; Said, Z. Impact of dust on the performance of solar photovoltaic (PV) systems under United Arab Emirates weather conditions. *Renew. Energy* **2019**, *141*, 287–297. [CrossRef]
25. Mellit, A.; Pavan, A.M.; Lughi, V. Short-term forecasting of power production in a large-scale photovoltaic plant. *Sol. Energy* **2014**, *105*, 401–413. [CrossRef]
26. Ogliari, E.; Dolara, A.; Manzolini, G.; Leva, S. Physical and hybrid methods comparison for the day ahead PV output power forecast. *Renew. Energy* **2017**, *113*, 11–21. [CrossRef]
27. Benghanem, M.; Mellit, A.; Alamri, S. ANN-based modelling and estimation of daily global solar radiation data: A case study. *Energy Convers. Manag.* **2009**, *50*, 1644–1655. [CrossRef]
28. Mellit, A.; Pavan, A.M. A 24-h forecast of solar irradiance using artificial neural network: Application for performance prediction of a grid-connected PV plant at Trieste, Italy. *Sol. Energy* **2010**, *84*, 807–821. [CrossRef]
29. Voyant, C.; Muselli, M.; Paoli, C.; Nivet, M.-L. Numerical weather prediction (NWP) and hybrid ARMA/ANN model to predict global radiation. *Energy* **2012**, *39*, 341–355. [CrossRef]
30. Sözen, A.; Arcaklioğlu, E.; Özalp, M. Estimation of solar potential in Turkey by artificial neural networks using meteorological and geographical data. *Energy Convers. Manag.* **2004**, *45*, 3033–3052. [CrossRef]
31. Mellit, A.; Shaari, S. Recurrent neural network-based forecasting of the daily electricity generation of a Photovoltaic power system. *Ecol. Veh. Renew. Energy (EVER)* **2009**, 26–29. Available online: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.533.9274&rep=rep1&type=pdf (accessed on 1 September 2019).
32. de Araujo, J.M.S. Performance comparison of solar radiation forecasting between WRF and LSTM in Gifu, Japan. *Environ. Res. Commun.* **2020**, *2*, 045002. [CrossRef]
33. Alzahrani, A.; Shamsi, P.; Dagli, C.; Ferdowsi, M. Solar Irradiance Forecasting Using Deep Neural Networks. *Procedia Comput. Sci.* **2017**, *114*, 304–313. [CrossRef]
34. Rai, A.; Shrivastava, A.; Jana, K.C. A CNN-BiLSTM based deep learning model for mid-term solar radiation prediction. *Int. Trans. Electr. Energy Syst.* **2020**, *31*, e12664. [CrossRef]
35. Yousif, J.H.; Kazem, H.A.; Alattar, N.N.; Elhassan, I.I. A comparison study based on artificial neural network for assessing PV/T solar energy production. *Case Stud. Therm. Eng.* **2019**, *13*, 100407. [CrossRef]
36. Jung, Y.; Jung, J.; Kim, B.; Han, S.U. Long short-term memory recurrent neural network for model in temporal patterns in long-term power forecasting for solar PV facilities: Case study of South Korea. *J. Clean. Prod.* **2020**, *250*, 119476. [CrossRef]
37. Mishra, M.; Dash, P.B.; Nayak, J.; Naik, B.; Swain, S.K. Deep learning and wavelet transform integrated approach for short-term solar PV power prediction. *Measurement* **2020**, *166*, 108250. [CrossRef]
38. Ghimire, S.; Deo, R.C.; Raj, N.; Mi, J. Deep solar radiation forecasting with convolutional neural network and long short-term memory network algorithms. *Appl. Energy* **2019**, *253*, 113541. [CrossRef]
39. Lee, D.; Kim, K. Recurrent Neural Network-Based Hourly Prediction of Photovoltaic Power Output Using Meteorological Information. *Energies* **2019**, *12*, 215. [CrossRef]
40. Pang, Z.; Niu, F.; O'Neill, Z. Solar radiation prediction using recurrent neural network and artificial neural network: A case study with comparisons. *Renew. Energy* **2020**, *156*, 279–289. [CrossRef]
41. Gensler, A.; Henze, J.; Sick, B.; Raabe, N. Deep Learning for solar power forecasting—An approach using AutoEncoder and LSTM Neural Networks. In Proceedings of the 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Budapest, Hungary, 9–12 October 2016; pp. 002858–002865.
42. Bakirci, K. Models of solar radiation with hours of bright sunshine: A review. *Renew. Sustain. Energy Rev.* **2009**, *13*, 2580–2588. [CrossRef]
43. Chen, Y.; Lin, M.; Yu, R.; Wang, T. Research on Simulation and State Prediction of Nuclear Power System Based on LSTM Neural Network. *Sci. Technol. Nucl. Install.* **2021**, *2021*, 8839867. [CrossRef]
44. Zhou, N.; Ren, N.; Zhou, J.; Huang, C. Prediction of remaining mileage of hydrogen energy vehicles based on LSTM. In Proceedings of the 2021 4th International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE), Changsha, China, 26–28 March 2021; pp. 749–753.
45. Turkish State Meteorological Service Official Web Sites. Available online: https://mgm.gov.tr/eng/forecast-cities.aspx (accessed on 1 September 2019).
46. Rosenblatt, F. *The Perceptron: A Perceiving and Recognizing Automaton (Project PARA)*; Report No. 85-460-1; Cornell Aeronautical Laboratory: New York, NY, USA, 1957.

47. Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. In Proceedings of the NIPS 2014 Workshop on Deep Learning, Cornell University, New York, NY, USA, 10 December 2014.
48. Lipton, Z.C.; Berkowitz, J.; Elkan, C. A Critical Review of Recurrent Neural Networks for Sequence Learning. *arXiv* **2015**, arXiv:1506.00019.
49. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.
50. Ghaderi, A.; Sanandaji, B.M.; Ghaderi, F. Deep Forecast: Deep Learning-based Spatio-Temporal Forecasting. *arXiv* **2017**, arXiv:1707.08110.
51. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
52. Lee, H.-Y. Deep Learning Neural Network with Memory (1)—PPT Video Online Download. Available online: https://slideplayer.com/slide/7478657/ (accessed on 2 September 2019).
53. Ramachandran, P.; Zoph, B.; Le, Q.V. Searching for Activation Functions. *arXiv* **2017**, arXiv:1710.05941.
54. Nwankpa, C.; Ijomah, W.; Gachagan, A.; Marshall, S. Activation functions: Comparison of trends in practice and research for deep learning. *arXiv* **2018**, arXiv:1811.03378.
55. Sibi, P.; Jones, S.A.; Siddarth, P. Analysis of different activation functions using back propagation neural networks. *J. Theor. Appl. Inf. Technol.* **2013**, *47*, 5.
56. Sharma, S. Activation functions in neural networks. *Towards Data Sci.* **2017**, *6*, 310–316. [CrossRef]
57. Karlik, B.; Olgac, A.V. Performance analysis of various activation functions in generalized MLP architectures of neural networks. *Int. J. Artif. Intell. Expert Syst.* **2011**, *1*, 111–122.
58. Cheng, J.; Dong, L.; Lapata, M. Long Short-Term Memory-Networks for Machine Reading. *arXiv* **2016**, arXiv:1601.06733.
59. Zurada, J.M. *Introduction to Artificial Neural Systems*; West: Eagan, MN, USA, 1992.
60. Ağbulut, U.; Gürel, A.E.; Biçen, Y. Prediction of daily global solar radiation using different machine learning algorithms: Evaluation and comparison. *Renew. Sustain. Energy Rev.* **2021**, *135*, 110114. [CrossRef]

electronics

Article

Metaheuristic Optimization-Based Path Planning and Tracking of Quadcopter for Payload Hold-Release Mission

Egemen Belge, Aytaç Altan and Rıfat Hacıoğlu *

Department of Electrical and Electronics Engineering, Zonguldak Bülent Ecevit University, 67100 Zonguldak, Turkey; egemenbelge@beun.edu.tr (E.B.); aytacaltan@beun.edu.tr (A.A.)
* Correspondence: hacirif@beun.edu.tr

Abstract: Under harsh geographical conditions where manned flight is not possible, the ability of the unmanned aerial vehicle (UAV) to successfully carry out the payload hold–release mission by avoiding obstacles depends on the optimal path planning and tracking performance of the UAV. The ability of the UAV to plan and track the path with minimum energy and time consumption is possible by using the flight parameters. This study performs the optimum path planning and tracking using Harris hawk optimization (HHO)–grey wolf optimization (GWO), a hybrid metaheuristic optimization algorithm, to enable the UAV to actualize the payload hold–release mission avoiding obstacles. In the study, the hybrid HHO–GWO algorithm, which stands out with its avoidance of local minima and speed convergence, is used to successfully obtain the feasible and effective path. In addition, the effect of the mass change uncertainty of the UAV on optimal path planning and tracking performance is determined. The effectiveness of the proposed approach is tested by comparing it with the metaheuristic swarm optimization algorithms such as particle swarm optimization (PSO) and GWO. The experimental results obtained indicate that the proposed algorithm generates a fast and safe optimal path without becoming stuck with local minima, and the quadcopter tracks the generated path with minimum energy and time consumption.

Keywords: path planning and tracking; metaheuristic optimization; quadcopter; payload hold–release system; obstacle avoidance

Citation: Belge, E.; Altan, A.; Hacıoğlu, R. Metaheuristic Optimization-Based Path Planning and Tracking of Quadcopter for Payload Hold-Release Mission. *Electronics* **2022**, *11*, 1208. https://doi.org/10.3390/electronics11081208

Academic Editors: Luis Hernández-Callejo, Sergio Nesmachnow and Sara Gallardo Saavedra

Received: 12 March 2022
Accepted: 8 April 2022
Published: 11 April 2022

1. Introduction

Path planning and tracking are the main tasks studied for unmanned vehicles, especially unmanned aerial vehicles (UAVs), unmanned ground vehicles, and unmanned underwater vehicles [1–4]. UAVs, which have been used extensively in defense industry and academic studies in recent years, perform tasks such as surveillance, target tracking, search and rescue, and payload transportation [4–7]. The obstacles and their positions in the region where UAVs will operate play an important role in the effective operation of UAVs [8]. Establishing a safe path by determining the risky areas in military operation and natural disaster areas, following the path that has been generated, and releasing the payloads to the predefined regions are critical for the successful performance of the mission [9]. In this study, a new path planning and tracking algorithm based on metaheuristic optimization is developed for the payload hold–release task by avoiding the obstacles at the target points defined around the planned path.

A path planning and tracking is required for the UAV to safely reach the target location from the starting location depending on certain restriction conditions such as minimum flight distance and time [10]. UAVs may be exposed to inconvenient land and weather conditions while performing critical tasks. UAVs try to overcome this problem with their maneuverability and altitude capabilities [1]. This situation causes the UAV to consume more energy [11]. In the presence of obstacles and constraints, optimal path planning is required for the UAV to safely follow the specified path with minimum energy and time consumption [12,13]. The UAV path planning problem is a complex optimization problem that requires efficient algorithms to solve. This optimization problem can be solved with classical algorithms as well as by using quite efficient metaheuristic algorithms. Simple path planning for UAV is performed with the Voroni diagram algorithm [14], the probabilistic roadmaps algorithm [15], the A* algorithm [16], and rapidly discovered tree-based algorithms [17]. However, since the kinematic and dynamic constraints of the UAV are rarely considered, these algorithms are generally not preferred in practical applications. In recent years, various approaches have been proposed for the autonomous path planning of the UAV, including meta-heuristic optimization algorithms. In [18], modeling of the battery performance of the UAV is emphasized. A multi-variable linear regression model has been created for the minimum energy consumption of the UAV on the specified path. The generated energy consumption model is used as a fitness function in the optimization algorithm. The performance of the proposed algorithm has been verified with various scenarios for the minimum energy consumption of the UAV. In [19], a path planning algorithm based on k-degree smoothing is proposed to define the coordinated path planning of the UAV in a safe and efficient manner. In the study, a k-degree smoothing method that aims to obtain a safer path using the ant colony optimization (ACO) algorithm [20] is presented. The proposed algorithm draws attention with its slow convergence speed and high probability of being stuck to local optima. In order to deal with these problems, a hybrid optimization algorithm obtained by combining maximum–minimum ACO (MMACO) and Cauchy mutant (CM) operators is recommended in [21]. As paths with faster convergence speed and better solution optimization are preferred in practical applications, swarm-based bio-inspiring optimization algorithms with low computational complexity and high computational speed are used extensively. In [22], an improved particle swarm optimization (PSO) algorithm has been proposed to achieve faster convergence speed and better solution optimization in the path planning of the UAV. The performance of the algorithm has been tested on various UAVs under many environmental constraints with Monte Carlo simulations. In [23], the 3D path planning problem of the UAV in the presence of obstacles is solved with the grey wolf optimization (GWO) algorithm [24] and the performance of the proposed algorithm is compared with metaheuristic algorithms such as PSO, the whale optimization algorithm (WOA), and the sine cosine algorithm (SCA). In the literature, metaheuristic optimization algorithms play an important role in solving different engineering problems, as well as path planning and tracking [25–27].

UAVs may encounter various obstacles while performing specified missions by sticking to a path. In [28], an obstacle avoidance algorithm based on ellipsoid geometry is proposed for the UAV to remain loyal to its original path and avoid the obstacles in its environment by creating waypoints in the presence of obstacles that obstruct the UAV flight path. The search for an avoidance path in the proposed algorithm is based on the use of ellipsoid geometry as a limited region containing the obstacle. Considering the geometry of the defined obstacle, a limited ellipsoid zone is created, and new crossing points are calculated within this zone. A convolutional neural network (CNN) approach based on depth estimation using molecular camera data to enable the quadrotor UAV to independently avoid collisions with obstacles in unknown and unstructured environments is presented in [29]. In [30], a new algorithm has been proposed that analytically calculates the path efficiently and effectively to create an environment map with a path without collision. In the developed algorithm, an initial path is created by the intersection of two 3D surfaces. Each obstacle position is shaped around the obstacles by adding a radial function to one of the two surfaces. The developed algorithm ensures that the intersection between deformed surfaces does not intersect with obstacles. The algorithm provides that the safe path is created in real time in the UAV's path tracking. In [31], every point in the motion environment is scanned with the 2D lidar on the UAV and the position of the UAV is estimated using the point cloud correction method. Unlike many studies with lidar, the effects of motion on the point cloud have been taken into account. In the proposed algorithm, point cloud features obtained by laser radar are extracted and a clustering is made based on relative distance and density. A robust nonlinear flight controller framework with six dimensional force and torque estimators that includes a model predictive controller (MPC)-based trajectory plan that considers the trajectory planning problem as an optimal control problem with nonlinear obstacle avoiding limitations is proposed in [32].

The ability of the UAV, which has a payload transportation system, to move around the reference trajectory and hold payloads from a certain point and to release payloads to specified targets with minimum error makes the UAV important for critical missions. In releasing the payloads to the specified targets, the UAV should be able to determine the path on its own or stay loyal to the specified path. In [33], neural-network-based real-time UAV control is performed in order to release the payloads to the marked targets by following a certain path with minimum error. The controller structure includes feature extraction and selection stages. In order for the UAV to release the payloads on the predetermined coordinates with the highest accuracy, the full mathematical model of the UAV, as well as the model of the payload transportation system, is needed. In [34], both the dynamic model parameters and the payload transportation system model of the UAV are handled together with the controller approach based on the law of feedback linearization. It is stated that the stabilization of the UAV, especially when releasing payload, is improved with the proposed controller approach. The controller scheme robust to payload changes in various weights is presented in [35]. The proposed controller provides the stabilization of the UAV in the suspended position by compensating the weight changes in the UAV with payload transportation system. In [36], an optimization-based controller algorithm has been developed for the UAV moving around a certain trajectory to make minimum oscillation at maximum payload. It is emphasized that the developed algorithm performs optimal control, especially in maneuvers.

In this study, a new metaheuristic-optimization-based path planning and tracking algorithm with a very high convergence speed is proposed to the UAV with payload transportation system in order to plan a path by avoiding obstacles under constraints such as mass uncertainty, unknown parameters, and unmeasurable external disturbances and to release the payloads to the target points with minimum error while staying loyal to path. The proposed algorithm is robust as it copes with unknown system dynamics and adverse environmental factors. The main contributions of this study are that the new hybrid Harris hawk optimization (HHO)–GWO algorithm for path planning is proposed, the new path planning and tracking control strategy is developed together, and the path-tracking performance of the quadcopter in payload hold–release mission has been analyzed. In addition, the positional error due to the mass uncertainty can be minimized by the proposed control strategy, as well as the energy function. The results of the study are shown that the mass uncertainty and energy of quadcopter during payload hold–release mission have been minimized using the new proposed path planning and tracking algorithm.

The remainder of this paper is organized as follows. The dynamic model of the quadrotor UAV used in the study is given in Section 2. The proposed controller approach for the path planning and tracking of the UAV is introduced in Section 3, including GWO and HHO algorithms. The generated maps are presented in Section 4. The results obtained with the proposed model are discussed in Section 5. Finally, in Section 6, the main results of the study, and future work are highlighted.

2. Dynamic Model of Quadcopter

Quadcopter is an underactuated type of UAV with four motors and six degrees of freedom (three translational and three rotations) and capable of landing and taking off in limited areas [37]. The evaluation of translation and rotation dynamics together in the motion control of a quadcopter is an important control problem. In the solution of this control problem, it is very important to take into account the non-linear parameters in the dynamics of the quadcopter. The main components of the quadcopter, Euler angles (roll, pitch, yaw), body frame, and global frame are illustrated in Figure 1.

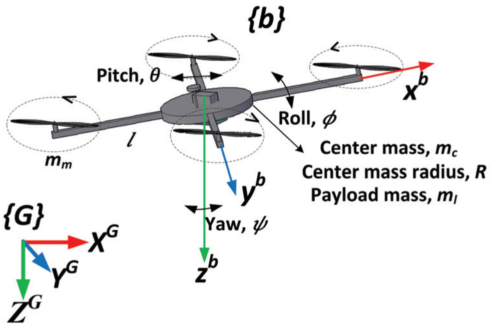


Figure 1. The main components of quadcopter.

The following parameters:

- The frame of the quadcopter is symmetrical and the center of gravity is in the middle of the fuselage;
- The thrust and friction of each motor of a quadcopter is proportional to the square of their motor speed;
- Moment of inertia of the propellers;
- During the flight of the qudcopter, the Earth is flat and non-rotating.

are assumed in the model of quadcopter [38]. Position changes during quadcopter flight are measured in the frame, accelerometer, and gyro values are measured in the body frame. For this reason, it is necessary to define the transformations between body and coordinate systems. In this study, $cos(.)$, $sin(.)$, and $tan(.)$ are represented by $c(.)$, $s(.)$, and $t(.)$, respectively. Considering these transformations, the velocity expression in the frame is obtained by using the velocity in body frame as:

$$\begin{bmatrix} \dot{X}^G \\ \dot{Y}^G \\ \dot{Z}^G \end{bmatrix} = \begin{bmatrix} c(\psi)c(\theta) & c(\psi)s(\phi)s(\theta) - c(\phi)s(\psi) & c(\phi)c(\psi)s(\theta) + s(\phi)s(\psi) \\ s(\psi)c(\theta) & s(\phi)s(\psi)s(\theta) + c(\phi)c(\psi) & c(\phi)s(\psi)s(\theta) - c(\psi)s(\phi) \\ -s(\theta) & c(\theta)s(\phi) & c(\phi)c(\theta) \end{bmatrix} \begin{bmatrix} \dot{x}^b \\ \dot{y}^b \\ \dot{z}^b \end{bmatrix} \tag{1}$$

where $\dot{X}^G$, $\dot{Y}^G$, $\dot{Z}^G$ velocities (m/s) (X, Y, Z) in global frame, ϕ, θ, ψ (roll, pitch, yaw angles), (rad), and $\dot{x}^b$, $\dot{y}^b$, $\dot{z}^b$ velocities (X, Y, Z) in the body frame [38,39]. The equations of motion of the quadcopter consist of two main components, dynamic and kinematic. Dynamic components explain the motion of the quadcopter according to Newton's second laws, while kinematic components explain the quadcopter's transformation equations. The rotational kinematics of the quadcopter describe the relationship between the angular rate and Euler angles. According to this rotation kinematics, since the angular rate is given in the body frame and the Euler angles are given in the frame, the relation between each other is obtained as:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & s(\phi)t(\theta) & c(\phi)t(\theta) \\ 0 & c(\phi) & -s(\phi) \\ 0 & \frac{s(\phi)}{c(\theta)} & \frac{c(\phi)}{c(\theta)} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{2}$$

by using the transformation matrix, where p, q, r roll, pitch, yaw rates (rad/s) and $\dot{\phi}$, $\dot{\theta}$, $\dot{\psi}$, and (rad/s) time derivatives of Euler angles, respectively [38,40]. Translational s describes the linear motion of all forces acting on the quadcopter during flight according to the coordinate frame. Equations of motion resulting from translational s of the quadcopter are obtained as in Equation (3), according to Newton's second law:

$$\begin{bmatrix} \ddot{X}^G \\ \ddot{Y}^G \\ \ddot{Z}^G \end{bmatrix} = \begin{bmatrix} \frac{1}{m}(-[c(\phi)c(\psi)s(\theta)+s(\phi)s(\psi)]u_1 - K_{dx}\dot{X}^G) \\ \frac{1}{m}(-[c(\phi)s(\psi)s(\theta)-c(\psi)s(\phi)]u_1 - K_{dy}\dot{Y}^G) \\ \frac{1}{m}(-[c(\phi)c(\theta)]u_1 - K_{dz}\dot{Z}^G) + g \end{bmatrix} \tag{3}$$

where $\ddot{X}^G$, $\ddot{Y}^G$, $\ddot{Z}^G$ accelerations (m/s^2) X, Y, Z in the coordinates, m mass of quadcopter (kg), K_{dx}, K_{dy}, K_{dz} drag coefficients, $\dot{X}^G$, $\dot{Y}^G$, $\dot{Z}^G$ velocities X, Y, Z in the coordinates, and u_1 is total thrust of all motors, respectively [41]. The rotational s of quadcopter describes the relationship:

$$\begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} \frac{[(J_y - J_z)qr - J_r q(w_1 - w_2 + w_3 - w_4) + lK_T u_2]}{J_x} \\ \frac{[(J_z - J_x)pr + J_r p(w_1 - w_2 + w_3 - w_4) + lK_T u_3]}{J_y} \\ \frac{[(J_x - J_y)pq + K_d u_4]}{J_z} \end{bmatrix} \tag{4}$$

between the second derivatives of Euler angles ($\ddot{\phi}$, $\ddot{\theta}$, $\ddot{\psi}$) (rad/s^2) on each axis depending on the square of its motor speeds (w_1, w_2, w_3, w_4) (rad/s), namely, torques, and J_x, J_y, and J_z (kg m^2) quadcopter moments of ineartia on each axis. u_2 refers roll control input, u_3 describes pitch control input, u_4 indicates yaw control input, K_T is the thrust coefficient, K_d is the drag torque proportionality constant, and l is the arm length of quadcopter (m) as in Equation (4) [38,41]. The quadcopter moments of inertia on each axis and mass of quadcopter are expressed [41]:

$$J_x = J_y = \frac{2(m_c + m_l)R^2}{5} + 2l^2 m_m \qquad J_z = \frac{2(m_c + m_l)R^2}{5} + 4l^2 m_m \qquad m = 4m_m + m_c + m_l \tag{5}$$

where m_c is the center mass of quadcopter (kg), R is the radius of center mass (m), m_m is the motor mass (kg), and m_l is the payload mass (kg). In this study, the total mass in the system model of the quadcopter is changed during the payload hold–release mission depending on the weight of the payload carried, and the moment of inertia in each axis is directly related to this mass change. To summarize, the dynamic and kinematic model of the quadcopter with six degrees of freedom is represented as Equations (1)–(4). The relationship between motor speeds and control variables is defined as:

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} K_T & K_T & K_T & K_T \\ 0 & -lK_T & 0 & lK_T \\ lK_T & 0 & -lK_T & 0 \\ K_d & -K_d & K_d & -K_d \end{bmatrix} \begin{bmatrix} w_1^2 \\ w_2^2 \\ w_3^2 \\ w_4^2 \end{bmatrix} \tag{6}$$

Note that the control variables are directly proportional to the squares of the motor speeds.

3. Proposed Control Approach for Path Planning and Tracking

The control strategy of this study consists of path planning and tracking. The hybrid HHO–GWO algorithm, which has high convergence speed and swarm intelligence that can avoid local minimum points, is proposed in this study in determining the optimum path. The path planning performances of the proposed optimization algorithm are compared with metaheuristic optimization algorithms such as PSO and GWO. The payload hold–release path determined by these optimization algorithms is generated with the shortest distance and avoiding the areas where there are obstacles. By analyzing the multi-objective function with metaheuristic optimization algorithms, waypoints to be followed by the UAV are generated. As seen in Figure 2, after the waypoints that the quadcopter are to follow are generated, the following of these waypoints, namely, the path tracking, is carried out with controller in a nested structure. The main idea of the study is that a new control strategy is proposed to carry out path planning and tracking together for the quadcopter's payload hold–release mission. The section includes not only controller design of quadcopter but also metaheuristic algorithms such as PSO, GWO, and HHO.

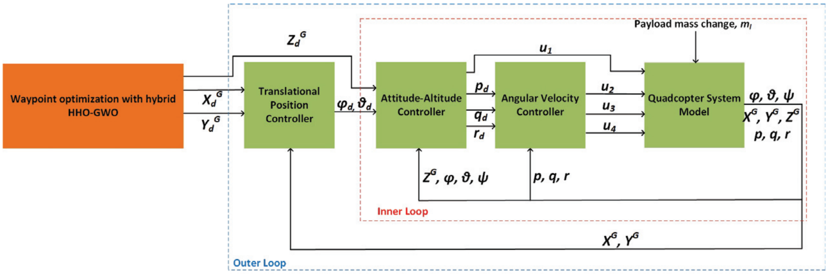

Payload mass change, m_l
Z_d^G
u_1
Waypoint optimization with hybrid HHO-GWO
X_d^G
Y_d^G
Translational Position Controller
φ_d, ϑ_d
Attitude-Altitude Controller
p_d
q_d
r_d
Angular Velocity Controller
u_2
u_3
u_4
Quadcopter System Model
φ, ϑ, ψ
X^G, Y^G, Z^G
p, q, r
$Z^G, \varphi, \vartheta, \psi$
p, q, r
Inner Loop
X^G, Y^G
Outer Loop

Figure 2. The proposed control strategy of the quadcopter.

3.1. Controller Design

The control strategy proposed in this study consists of two main steps: path planning and the tracking of the path. Path planning, which is the first step of the control strategy, is the process of determining the waypoints that the quadcopter is desired to track during payload transportation mission. Metaheuristic optimization algorithms such as PSO, GWO, and hybrid HHO–GWO are used to identify these waypoints. After determining the waypoints that the quadcopter is desired to track, path tracking is performed as the second step of control strategy. The path tracking process consists of four control structures: translational position, attitude–altitude, angular velocity controllers, and the system model of quadcopter. This path tracking controller is nested with each other. The motion control on the X and Y axes, attitude–altitude control and the angular velocity control of quadcopter are performed on the outer part, the inner part, and the innermost part, respectively. In the study, noise that occurs as a disruptive effect in attitude, altitude, and angular velocity control is suppressed by Kalman filter [42]. The position errors of quadcopter in X, Y, and Z axes are indicated as:

$$e_x = X_d^G - X^G \qquad e_y = Y_d^G - Y^G \qquad e_z = Z_d^G - Z^G \tag{7}$$

where e_x, e_y, and e_z refer the position errors; X_d^G, Y_d^G, and Z_d^G denote the desired positions; and X^G, Y^G, and Z^G define the measured positions in the X, Y, and Z axes, respectively. The errors of quadcopter in orientation angles are specified as:

$$e_\phi = \phi_d - \phi \qquad e_\theta = \theta_d - \theta \qquad e_\psi = \psi_d - \psi \tag{8}$$

where e_ϕ, e_θ, and e_ψ describe the orientation errors; ϕ_d, θ_d, and ψ_d define the desired orientation angles; and ϕ, θ, and ψ represent the measured orientation angles in the roll, pitch, and yaw angle, respectively. The angular velocity error is stated as:

$$e_p = p_d - p \qquad e_q = q_d - q \qquad e_r = r_d - r \tag{9}$$

where e_p, e_q, and e_r are the angular velocity errors; p_d, q_d, and r_d define the desired angular velocity; and p, q, and r express the measured angular velocity along x^b, y^b, z^b, respectively.

3.1.1. Translational Position Control

This controller is responsible for minimizing the measurement difference obtained from the desired position and the quadcopter system model output. As specified in Equations (10) and (11):

$$\theta_d(t) = K_{PX} e_x(t) + K_{IX} \int_0^t e_x(\tau) d_\tau + K_{DX} \dot{e}_x(t) \tag{10}$$

$$\phi_d(t) = K_{PY}e_y(t) + K_{IY}\int_0^t e_y(\tau)d_\tau + K_{DY}\dot{e}_y(t) \tag{11}$$

the desired roll and pitch values are identified by the translational position controller as a result of the minimization of this error with the proportional–integral–derivative (PID) controller, and where K_{PX}, K_{IX}, K_{DX} express PID gains that control the movement of the quadcopter in the X position, and K_{PY}, K_{IY}, K_{DY} indicate PID gains that control the movement of quadcopter in the Y position, respectively. The inputs of the translational position controller are desired the position (X_d^G, Y_d^G), the output of the quadcopter system model (X^G, Y^G), and the controller's output are the desired pitch (θ_d) and roll angles (ϕ_d). The proposed controller also performs the tracking of waypoints specified in the X, Y plane with metaheuristic path planning algorithms [38].

3.1.2. Attitude–Altitude Control

The height and orientation angles (roll, pitch, yaw) of the quadcopter are controlled by the attitude–altitude controller. As stated in Equations (12)–(15):

$$u_1(t) = \frac{1}{cos(\phi)cos(\theta)}(K_{PZ}e_z(t) + K_{IZ}\int_0^t e_z(\tau)d_\tau + K_{DZ}\frac{de_z(t)}{dt} + mg) \tag{12}$$

$$p_d(t) = K_{P\phi}e_\phi(t) + K_{I\phi}\int_0^t e_\phi(\tau)d_\tau + K_{D\phi}\frac{de_\phi(t)}{dt} \tag{13}$$

$$q_d(t) = K_{P\theta}e_\theta(t) + K_{I\theta}\int_0^t e_\theta(\tau)d_\tau + K_{D\theta}\frac{de_\theta(t)}{dt} \tag{14}$$

$$r_d(t) = K_{P\psi}e_\psi(t) + K_{I\psi}\int_0^t e_\psi(\tau)d_\tau + K_{D\psi}\frac{de_\psi(t)}{dt} \tag{15}$$

K_{PZ}, K_{IZ}, K_{DZ} express PID gains that control the movement of quadcopter in the Z position; $K_{P\phi}, K_{I\phi}, K_{D\phi}$ specify PID gains that control the roll angle; $K_{P\theta}, K_{I\theta}, K_{D\theta}$ describe PID gains that control the pitch angle; $K_{P\psi}, K_{I\psi}, K_{D\psi}$ denote PID gains that control the yaw angle. The inputs of the controller are desired and measured height, roll, pitch, and yaw angles; the outputs are u_1; and the desired angular velocities are (p_d, q_d, r_d). u_1 obtained at the controller output is input into the quadcopter system, and this control variable enables the quadcopter to increase [38].

3.1.3. Angular Velocity Control

This controller performs angular velocity control, the main task of the controller is the minimization of error between desired and measured angular velocity components. The angular velocity control is denoted as:

$$u_2(t) = K_{PP}e_p(t) + K_{IP}\int_0^t e_p(\tau)d\tau + K_{DP}\frac{de_p(t)}{d_t} \tag{16}$$

$$u_3(t) = K_{PQ}e_q(t) + K_{IQ}\int_0^t e_q(\tau)d\tau + K_{DQ}\frac{de_q(t)}{dt} \tag{17}$$

$$u_4(t) = K_{PR}e_r(t) + K_{IR}\int_0^t e_r(\tau)d\tau + K_{DR}\frac{de_r(t)}{dt} \tag{18}$$

where K_{PP}, K_{IP}, K_{DP} express PID gains that control the angular velocity along x^b; K_{PQ}, K_{IQ}, K_{DQ} indicate PID gains that control the angular velocity along y^b; and K_{PR}, K_{IR}, K_{DR} refer to PID gains that control the angular velocity along z^b. The outputs of controller constitute the orientation control variables (u_2, u_3, u_4) of the quadcopter [38].

3.1.4. Motor Control

Using the height and orientation control variables obtained from attitude–altitude and angular velocity controller outputs, the angular velocities required for the motors are obtained as in Equations (19) and (20) [38]. Thus, the thrust required for the movement of the quadcopter in the specified path is obtained by controlling the speed of the motors w_i, $i = 1, 2, 3, 4$ as:

$$w_1^2 = \frac{u_1}{4K_T} + \frac{u_3}{2lK_T} + \frac{u_4}{4K_d} \qquad w_2^2 = \frac{u_1}{4K_T} - \frac{u_2}{2lK_T} - \frac{u_4}{4K_d} \tag{19}$$

$$w_3^2 = \frac{u_1}{4K_T} - \frac{u_3}{2lK_T} + \frac{u_4}{4K_d} \qquad w_4^2 = \frac{u_1}{4K_T} + \frac{u_2}{2lK_T} - \frac{u_4}{4K_d} \tag{20}$$

The power consumed by each motor of quadcopter is indicated as:

$$P_{m_k} = P_{h_k} = (2\rho A_p)(\frac{K_v K_\tau}{K_t})^3 w_k^3 \qquad k = 1, 2, 3, 4 \tag{21}$$

where P_{m_k} denotes the power consumed by the kth motor, P_{h_k} explains the hovering power consumed by the kth motor, ρ is air density (kg/m^3), A_p refers to the propeller cross-section (m^2), K_v is the back electromotive force (EMF) constant, K_τ is the torque proportionality constant, and K_T is the thrust coefficient.

3.2. *Three-Dimensional Path Planning Model of the Quadcopter*

The 3D path planning algorithm proposed in this study is operated to define the optimum path by avoiding the obstacle region after the starting and ending point are determined. There are two limit values of the objective function, the starting (x_s, y_s, z_s) and the ending points (x_t, y_t, z_t). The number of waypoints to be generated, including starting, ending, payload hold, and payload release points, are entered. Afterwards, the locations of the spherical barriers on the map are defined as central positions (x_{obs}, y_{obs}, z_{obs}) and the radius (r_{obs}), and these locations are given as an input to the algorithm. The 3D path-planning algorithm presented in this study consists of three different objective functions. In the first part, the length of the generated path is indicated as:

$$d = \sum_{i=1}^{N_p - 1} \sqrt{dx_i^2 + dy_i^2 + dz_i^2} \tag{22}$$

where d_x, d_y, d_z are the infinitesimal lengths traversed by the quadcopter along X, Y, Z axes, respectively, and N_p is the number of generated points [43]. In the second part, the total energy consumed by the quadcopter is expressed as:

$$E_t = K_E \sum_{i=1}^{N_p} \Delta t \sum_{k=1}^{4} w_{ki}^3 \tag{23}$$

where $K_E = (2\rho A)(\frac{K_v K_\tau}{K_t})^3$, and Δt is the sampling period. Motor speeds w_k $(k = 1, 2, 3, 4)$ have been calculated in order to obtain d_{xi}, d_{yi}, d_{zi} in each i by using Equation (6) to obtain u_1, u_2, u_3, u_4, as well as the UAV model Equations (3) and (4). The collision of the quadcopter with obstacles is represented as violation function. The violation function is calculated as indicated in Algorithm 1. In the third part, the distance of each point on the pathway to a specific obstacle is expressed as:

$$dobs_i = \sqrt{(X_i - x_{obs})^2 + (Y_i - y_{obs})^2 + (Z_i - z_{obs})^2} + r_{uav} \tag{24}$$

$$r_{uav} = 2(l + \Delta l) \tag{25}$$

where X_i, Y_i, Z_i are the generated points, and Δl is the propeller radius. If this distance is greater than the radius of the obstacle, then it is assumed that the obstacle is outside the quadcopter's field of view. The feasible constraint takes the following form:

$$dobs_i \leq r_{obs} \tag{26}$$

in Equation (26) [44]. The points chosen on the map representing the flight path attain a value calculated as:

$$v_i = \begin{cases} 0, & (1 - \frac{dobs_i}{r_{obs}}) < 0 \\ (1 - \frac{dobs_i}{r_{obs}}), & (1 - \frac{dobs_i}{r_{obs}}) > 0 \end{cases} \tag{27}$$

in Equation (27). The average value of v_i with respect to a particular obstacle is obtained. Subsequently, the average of v_i with respect to the other obstacle is obtained, and the total violation function is expressed as:

$$V = \frac{1}{N_p} \sum_{i=1}^{N_p} \sum_{j=1}^{n_{obs}} v_{ij} \tag{28}$$

in Equation (28). The objective function is calculated by combining Equation (22), Equation (23), and Equation (28):

$$J_E = min\{(d + E_t)(1 + \xi V)\} \tag{29}$$

where ξ and V are the violation coefficient and function, respectively.

Algorithm 1: Pseudo-code of proposed 3D path-planning algorithm.

```
Initialize starting $X, Y, Z$ point  $(x_s, y_s, z_s)$
Initialize ending $X, Y, Z$ point  $(x_t, y_t, z_t)$
Initialize holding $X, Y, Z$ point  $(x_h, y_h, z_h)$
Initialize releasing $X, Y, Z$ point  $(x_r, y_r, z_r)$
Input: The number of measurements  $(N_p)$
Input: Determine the position of obstacle on Map  $(x_{obs}, y_{obs}, z_{obs}, r_{obs})$
Input: The number of obstacles  $(n_{obs})$
Initialize waypoints between starting and ending point randomly.
for Optimization Algorithms (PSO, GWO, hybrid HHO–GWO) do
    for k = 1: number of obstacles  $(n_{obs})$ do
        Calculate the distance of randomly generated path to sperical obstacles
         using Equation (24).  $(dobs)$
        $v = max(1 - \frac{dobs}{r_{obs}(k)}, 0)$
        $V = V + mean(v)$
    Calculate the distance of generated path using Equation (22).  $(d)$
    Calculate the energy of generated path using Equation (23).  $(E_t)$
    Calculate objective function using Equation (29), $J_E = min\{ (d + E_t)(1 + \xi V)\}$
return $X_d^G$, $Y_d^G$, $Z_d^G$
```

3.3. Proposed Path Planning and Tracking Optimization Algorithm

In recent years, many metaheuristic optimization algorithms that imitate living things in nature have been used extensively to solve complex nonlinear engineering problems. These algorithms stand out compared to traditional optimization techniques such as stochastic and deterministic approaches, with their flexibility, simplicity, avoidance of local optima, and ability to search randomly. In this study, in order to overcome the problem of planning the optimum path and tracking this path for the quadcopter, a swarm-based hybrid optimization approach is proposed, which contains GWO and HHO [45] algorithms and has high convergence speed and is capable of avoiding local minima. The proposed optimization algorithm allows the quadcopter to not only avoid obstacles but also to follow the planned path for payload holding-releasing with minimum error. The performance of the proposed algorithm is compared with PSO and GWO algorithms. The PSO, GWO, and hybrid GWO–HHO algorithms used for the quadcopter's path planning and tracking are described in this section.

3.3.1. Particle Swarm Optimization

PSO is a population-based metaheuristic optimization algorithm developed in 1995 by Kennedy and Eberhart, inspired by the behavior of birds living in flocks in nature [46]. Generally, PSO is a population-based probability optimization method, which is preferred to produce solutions for multivariable and multiparameter optimization problems. It is frequently used in different optimization problems due to its high convergence speed and solutions. In adapting to various environmental conditions, such as avoiding predators or finding a rich food source, many animal swarms such as fish and birds communicate with each other, increasing their probability and speed of finding the real target. The essence of the PSO algorithm is a swarm and each particle is a part of it. In this swarm-based optimization algorithm, each particle consists of a position and velocity component, and an update is made in their positions by changing the velocity of the particles. Depending on the optimization problem, the updated positions of the particles are substituted in the objective function [46]. In the minimization process of the objective function, if the position value of the particle is smaller, than the best position value obtained, the new solution is kept in the memory in each iteration as shown in Algorithm 2. The position and velocity vectors of these particles are initially determined randomly, depending on the constraints. The velocities of randomly generated particles are computed as:

$$V_i(t+1) = V_i(t) + c_1 rand(p_{best} - X_i) + c_2 rand(g_{best} - X_i) \tag{30}$$

in the next iteration, where X_i, the position of i. particle, $rand$, is a uniformly random number between [0, 1]; p_{best} is the best position of the swarm; g_{best} is the best position within the group; and c_1, c_2 are two constants which determine the weights of p_{best} and g_{best}, respectively. The position of the particles is obtained by adding the expression of velocity $V_i(t+1)$ to the current position X_i as:

$$X_i(t+1) = X_i(t) + V_i(t+1) \tag{31}$$

In this optimization process, the position of each particle in the population is updated by changing the velocity vector. This update process consists of both the experimental knowledge of the particle and the knowledge it has socially acquired from neighboring particles.

Algorithm 2: Pseudo-code of PSO algorithm.

```
Initialize position vectors  $X_i(i = 1, 2, ..., n)$
Initialize velocity vectors  $V_i(i = 1, 2, ..., n)$
while (t<Max number of iterations) do
    for i = 1: Number of Particles (n) do
        Update the velocity of particles by Equation (30)
        Update the position of particles by Equation (31)
        Evaluate the fitness of $X_i$
        if $f(X_i) < f(p_{best})$ then
            $X_i = p_{best}$
        if $f(X_i) < f(g_{best})$ then
            $X_i = g_{best}$
return $g_{best}$
```

3.3.2. Grey Wolf Optimization

The GWO algorithm, inspired by the hunting hierarchy of grey wolves that live as a swarm in nature, is proposed by Mirjalili et al. [24]. As illustrated in Figure 3, the alpha wolf makes all hunting decisions in the herd, leads the swarm, and is located at the top of hunting pyramid. According to the order of social hierarchy in the herd, the top three wolves are alpha, beta, and omega, respectively. The candidate solutions are randomly generated in optimization process as with other metaheuristic optimization algorithms as shown in Algorithm 3. Among these candidate solutions, the best, the second, and the third candidate solution refers to to alpha (X_α), beta (X_β), and delta (X_δ) positions, respectively. The other low candidate solution refers to the omega (ω) position. The hunting mechanism of grey wolves consists of following the prey and approaching, encircling, and attacking the prey. In the grey wolf optimization algorithm, the process of encircling the prey is carried out:

$$\overrightarrow{D} =| \overrightarrow{C}\,\overrightarrow{X_p}(t) - \overrightarrow{X}(t)| \tag{32}$$

$$\overrightarrow{X}(t+1) = \overrightarrow{X_p}(t) - \overrightarrow{A}\,\overrightarrow{D} \tag{33}$$

where t specifies the current iteration, $\overrightarrow{A}$ and $\overrightarrow{C}$ are constant vectors, $\overrightarrow{X_p}$ is the position vector of the prey, and $\overrightarrow{X}$ defines the position vector of a grey wolf. $\overrightarrow{A}$ and $\overrightarrow{D}$ are calculated by:

$$\overrightarrow{A} = 2\,\overrightarrow{a}\,\overrightarrow{r_1} - \overrightarrow{a} \tag{34}$$

$$\overrightarrow{C} = 2\overrightarrow{r_2} \tag{35}$$

where $\overrightarrow{a}$ is linearly decreased from 2 to 0 over the course of the iterations, and $\overrightarrow{r_1}$ and $\overrightarrow{r_2}$ are random values generated between 0 and 1. The hunting process of grey wolves is expressed as:

$$\overrightarrow{D_\alpha} =|\overrightarrow{C_1}\overrightarrow{X_\alpha} - \overrightarrow{X}| \qquad \overrightarrow{D_\beta} =|\overrightarrow{C_2}\overrightarrow{X_\beta} - \overrightarrow{X}| \qquad \overrightarrow{D_\delta} =|\overrightarrow{C_3}\overrightarrow{X_\delta} - \overrightarrow{X}| \tag{36}$$

$$\overrightarrow{X_1} = \overrightarrow{X_\alpha} - \overrightarrow{A_1}\overrightarrow{D_\alpha} \qquad \overrightarrow{X_2} = \overrightarrow{X_\beta} - \overrightarrow{A_2}\overrightarrow{D_\beta} \qquad \overrightarrow{X_3} = \overrightarrow{X_\delta} - \overrightarrow{A_3}\overrightarrow{D_\delta} \tag{37}$$

$$\overrightarrow{X}(t+1) = \frac{\overrightarrow{X_1} + \overrightarrow{X_2} + \overrightarrow{X_3}}{3} \tag{38}$$

where the positions of the best three agents are indicated by $\overrightarrow{X_\alpha}$, $\overrightarrow{X_\beta}$, $\overrightarrow{X_\delta}$; the distance vectors ($\overrightarrow{D_\alpha}$, $\overrightarrow{D_\beta}$, $\overrightarrow{D_\delta}$) of candidate solutions are calculated according to the best three solutions; ($\overrightarrow{X_1}$, $\overrightarrow{X_2}$, $\overrightarrow{X_3}$) are the updated positions of the search agents; and $\overrightarrow{X}(t+1)$ is the next iteration position.

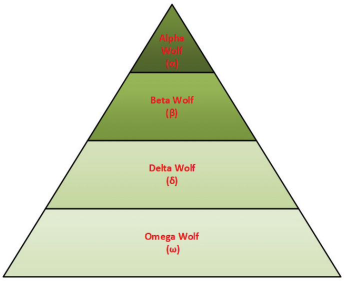


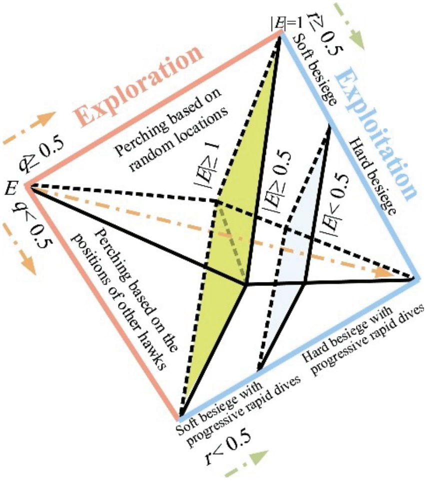
Figure 3. The hunting hierarchy of grey wolves.

Algorithm 3: Pseudo-code of GWO algorithm.

```
Initialize the grey wolf population $X_i (i = 1, 2, ..., n)$
Initialize a, A and C
Calculate the fitness of each search agent
$X_\alpha$ = the best search agent
$X_\beta$ = the second best search agent
$X_\delta$ = the third best search agent
while (t<Max number of iterations) do
    for each search agent do
        Update the position of the current search agent by Equation (37)
        Update a, A and C
        Calculate the fitness of all search agents
        Update $X_\alpha, X_\beta$ and $X_\delta$
    $t = t + 1$
return $X_\alpha$
```

3.3.3. Harris Hawk Optimization

In this section, the exploration, transition from exploration to exploitation, and exploitation phases of the HHO component of the hybrid GWO–HHO algorithm proposed in the study are explained. In this algorithm, the hunting strategy of Harris hawks, one of the smart birds in nature, is imitated. Harris hawks act as a swarm, especially during the rabbit-hunting process. Each swarm has a leader. The leader and other members of the swarm primarily make exploration flights. After the prey is detected, the hunting process begins. HHO is gradient-free optimization method; hence, it can be applied to many nonlinear engineering problems depending on a suitable formulation [45]. Harris hawks' main tactic in hunting is called the "surprise attack". In this clever strategy, several hawks collaboratively try to attack from different directions and simultaneously approach the prey that has been found to have fled outside the shelter. The attack can be completed quickly, with the hawks catching their prey in a matter of seconds. All phases of the HHO's exploration and exploitation processes are shown in Figure 4.



Exploration
Exploitation
|E|=1
E
$q \geq 0.5$
$q < 0.5$
$r \geq 0.5$
$r < 0.5$
Perching based on random locations
Perching based on the positions of other hawks
$|E| \geq 1$
$|E| \geq 0.5$
$|E| < 0.5$
Soft besiege
Hard besiege
Soft besiege with progressive rapid dives
Hard besiege with progressive rapid dives

Figure 4. All phases of Harris hawk optimization algorithm [45].

- **Exploration phase:** Although Harris hawks have strong eyes, sometimes they may not be able to detect their prey easily. In this situation, Harris hawks often wait in the desert area and observe their surroundings. This process continues in a loop. Harris hawks in each loop are identified as candidate solutions. The hawk, who is in the best position in relation to the rabbit in the loop, represents the optimum solution. The HHO algorithm uses two different strategies in the hunt search process. These strategies can be described by [45]:

$$X(t+1) = \begin{cases} X_{rand}(t) - r_1|X_{rand} - 2r_2X(t)| & q \geq 0.5 \\ (X_{rabbit}(t) - X_m(t)) - r_3(LB + r_4(UB - LB)) & q < 0.5 \end{cases} \tag{39}$$

where $X(t+1)$ represents the position of Harris hawks in the next iteration t; $X(t)$ denotes the current position of Harris hawks; X_{rabbit} indicates the position of the rabbit; $X_m(t)$ is the average position of the current population of Harris Hawks; $X_{rand}(t)$ represents a randomly selected Harris hawk from the current population; r_1, r_2, r_3, r_4, and q are random numbers between $[0,1]$; and UB and LB show the upper and lower bounds of the variables, respectively. The average position of hawks is determined by:

$$X_m(t) = \frac{1}{N}\sum_{i=1}^{N} X_i(t) \tag{40}$$

where N represents the total number of Harris hawks, and $X_i(t)$ indicates the location of each Harris hawk in iteration t.

- **Transition from exploration to exploitation phase:** Harris hawks begin the exploitation phase by developing different attack models according to the energy of the prey after the exploration process is completed. This process is modelled in [45] as:

$$E = 2E_0(1 - \frac{t}{T}) \tag{41}$$

where E_0 is the initial energy value of the prey randomly defined in the range of $[0,1]$, E is the energy of the escaping prey, and T is the maximum number of iterations.

- **Exploitation phase:** At this phase, the Harris hawk attacks its prey and makes the surprise attack move. In response to this situation, the prey tries to escape. In this case, the Harris hawk basically develops four different strategies. The energy of the prey and the chance of catching the escaping prey are indicated by E and r, respectively:
  - **Soft besiege** ($r \geq 0.5$ and $|E| \geq 0.5$)
    In this strategy, the Harris hawk makes misleading jumps at its prey and tries to reduce the energy of its prey. This soft besiege strategy is mathematically described by:

$$X(t+1) = \Delta X(t) - E|JX_{rabbit}(t) - X(t)| \tag{42}$$

$$\Delta X(t) = X_{rabbit}(t) - X(t) \tag{43}$$

    where $\Delta X(t)$ is the difference between the current position in the t-th iteration and the current position of the prey, and J is a value that changes with each iteration to simulate the natural motion of the prey.

  - **Hard besiege** ($r \geq 0.5$ and $|E| < 0.5$)
    In this strategy, the energy of the prey is very low. The hawk hardly makes any besiege to throw his surprise claws on its prey. This strategy can be mathematically modeled as:

$$X(t+1) = X_{rabbit}(t) - E|\Delta X(t)| \tag{44}$$

    where $X_{rabbit}(t)$ represents the current position of prey, $\Delta X(t)$ is the difference between the current position in the t-th iteration and the current position of the prey.

  - **Soft besiege with progressive rapid dives** ($r < 0.5$ and $|E| \geq 0.5$)
    In this strategy, the prey has enough energy to escape. The Harris hawk is still performing the soft besiege strategy before the surprise jump. This process is

smarter than the previous strategy. Before the hawks start their soft besiege, they decide their next move based on the following calculation:

$$Y = X_{rabbit}(t) - E|JX_{rabbit}(t) - X(t)| \tag{45}$$

where $X_{rabbit}(t)$ indicates the current position of the prey, and J is a value that changes with each iteration to simulate the natural motion of the prey. This situation is compared with the previous dive to decide whether such a move would be a good dive. If the situation is unfavorable, the hawks dive into their prey suddenly. When deciding on this, a Levy-flight-based movement structure is used. This situation is defined by:

$$Z = Y + S \times LF(D) \tag{46}$$

where Z is the variable that decides whether the hawk will make a move on its prey, Y indicates its position in relation to the decreasing energy of the prey, D is the size of the problem, S is a random vector of size 1 x D, and LF is the Levy flight function and is defined by:

$$LF(x) = 0.01\frac{u \times \sigma}{|v|^{\frac{1}{\beta}}}\sigma = \left(\frac{\Gamma(1+\beta) \times sin(\frac{\pi\beta}{2})}{\Gamma(\frac{1+\beta}{2}) \times \beta \times 2^{(\frac{\beta-1}{2})}}\right)^{\frac{1}{\beta}} \tag{47}$$

where u and v are the random numbers between (0, 1), and β is 1.5. Note that the Levy flight algorithm is added to the exploitation phase to ensure that the local search process can be continued without becoming stuck at local optimum points. The positions of the hawks in the soft besiege phase are updated by:

$$X(t+1) = \begin{cases} Y & \text{if } F(Y){<}F(X(t)) \\ Z & \text{if } F(Z){<}F(X(t)) \end{cases} \tag{48}$$

where Y and Z are obtained using Equations (40) and (41).

– **Hard besiege with progressive rapid dives** ($r < 0.5$ and $|E| < 0.5$) In this strategy, the prey does not have enough energy to escape. The Harris hawk makes a fierce siege before its surprise jump to catch its prey. The hard besiege situation is expressed by:

$$X^{'}(t+1) = \begin{cases} Y^{'} & \text{if } F(Y^{'}){<}F(X(t)) \\ Z^{'} & \text{if } F(Z^{'}){<}F(X(t)) \end{cases} \tag{49}$$

where $Y^{'}$ and $Z^{'}$ are defined as:

$$Y^{'} = X_{rabbit}(t) - E|JX_{rabbit}(t) - X_m(t)| \tag{50}$$

$$Z^{'} = Y^{'} + S \times LF(D) \tag{51}$$

3.3.4. The Proposed Optimization Algorithm

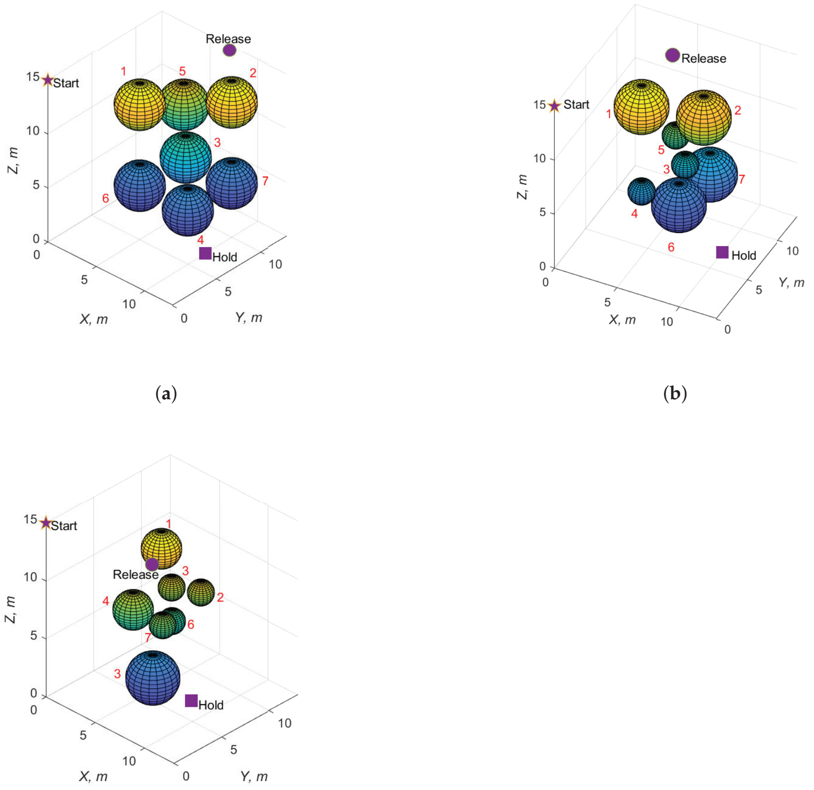
In this study, a hybrid HHO–GWO algorithm is proposed by combining the HHO algorithm with its random search capability and high convergence speed and the GWO algorithm, which has a high performance in avoiding local optima, so that the quadcopter can both avoid obstacles and track the planned path with minimum error. This algorithm ensures the robustness of the controller, even with sudden mass changes in the quadcopter during payload hold and release. The pseudo-code of the proposed algorithm for this study is presented in Algorithm 4.

Algorithm 4: Pseudo-code of hybrid HHO–GWO algorithm

```
Input: The population size $N$ and maximum number of iterations $T$
Output: The location of rabbit and its fitness value
Initialize the random population $X_i (i = 1, 2, ..., N)$
while (stopping condition is not met) do
    Calculate the fitness values of hawks. Set $X_{rabbit}$ as the location of rabbit (best position).
    for $(each\ hawk\ (X_i))$ do
        Update the initial energy $E_0$ and jump strength $J$
        $E_0 = 2rand() - 1, J = 2(1 - rand())$
        Update the $E$ using Equation (41)
        if $|E| \geq 1$ then
            Update the location vector using Equation (39)—(Exploration phase)
        end
        if $|E| < 1$ then
            if $r > 0.5$ and $|E| > 0.5$ then
                Update the position vector using Equation (42)—Soft besiege
            end
            if $r \geq 0.5$ and $|E| < 0.5$ then
                Update the position vector using Equation (44)—Hard besiege
            end
            if $r < 0.5$ and $|E| \geq 0.5$ then
                Update the position vector using Equation (45)—Soft besiege with progressive rapid dives
            end
            if $r < 0.5$ and $|E| < 0.5$ then
                Update the position vector using Equation (50)—Hard besiege with progressive rapid dives
            end
        end
    end
end
return $X_{rabbit}$
Initialize the starting position of search agents as final position vector of Harris Hawk Optimization
Initialize a, A and C
Calculate the fitness of each search agent
$X_\alpha$ = the best search agent
$X_\beta$ = the second best search agent
$X_\delta$ = the third best search agent
while (t<Max number of iterations) do
    for each search agent do
        Update the position of the current search agent by Equation (37)
        Update a, A and C
        Calculate the fitness of all search agents
        Update $X_\alpha$, $X_\beta$ and $X_\delta$
    end
    $t = t + 1$
end
return $X_\alpha$
```

4. Payload Hold-Release Mission Planning

In this study, a path planning and tracking algorithm is proposed on three different maps. In order to guarantee that the algorithms run do not memorize the path, three maps with different starting and ending points, containing obstacles at different locations, are generated. On the first map, there are obstacles of equal size with a radius of 2 m. On the second map, there are obstacles of two different sizes with radii of 1 m and 2 m. On the third map, there are obstacles in three different sizes with radii of 1 m, 1.5 m, and 2 m. The environmental difficulty level of Map 1, Map 2, and Map 3 range from weak to strong, respectively, in performing the payload hold-and-release mission by coping with obstacles. Seven separate spherical obstacles are placed on each of the maps. The locations of these spherical barriers on three different maps are given in Table 1. As stated in the Table 1, the location of each obstacle in 3D space is expressed as the X, Y, Z positions and radius R. These spherical barriers are positioned in 3D space, as shown in Figure 5. Here, the point where the quadcopter starts its mission, holds and releases the payload is shown as star, square and circle, respectively. In addition, the numbers on the figure are used to label the obstacles. The numbers on the figure are used to name the obstacles. Considering the safe and shortest path conditions of the quadrotor on these generated maps, waypoints are determined by metaheuristic optimization algorithms such as PSO, GWO, and hybrid HHO–GWO. By following this determined path, the payload hold–release performance of the quadrotor has been analyzed.



(**a**)

(**b**)

(**c**)

Figure 5. Maps created for testing the performance of quadcopter path planning and tracking (**a**) for Scenario 1, (**b**) for Scenario 2, (**c**) for Scenario 3.

Table 1. Positions of obstacles with 3 different scenarios.

Obstacle Number	Map 1	Map 2	Map 3
		$(x_{obs}, y_{obs}, z_{obs}, r_{obs})$, (m)	
1	(5, 5, 12.5, 2)	(4, 6, 12, 2)	(3, 9, 10, 1.5)
2	(10, 10, 12.5, 2)	(8, 8, 11, 2)	(7, 9, 8, 1)
3	(7.5, 7.5, 7.5, 2)	(7, 7, 7, 1)	(6, 5, 2, 2)
4	(10, 5, 5, 2)	(5, 4, 6, 1)	(4, 5, 7, 1.5)
5	(5, 10, 10, 2)	(6, 7.5, 9, 1)	(5, 8, 8, 1)
6	(5, 5, 5, 2)	(7, 6, 4, 2)	(6, 7, 6, 1)
7	(10, 10, 5, 2)	(8, 9, 5, 2)	(7, 5, 7, 1)

5. Experimental Results and Discussion

The performance comparison of the path planning and tracking control strategy is presented in this section. Firstly, the path planning performance of the quadcopter is examined on three different maps. Generating the shortest and safest path of quadcopter on all three maps is performed with the PSO, GWO, and hybrid HHO–GWO algorithms. The quadcopter at origin point (0, 0, 0) rises by 15 m along the Z-axis in all 3 maps. Afterwards, the payload holds the path from the first to the fifth waypoints, and the payload release path from the fifth to ninth waypoints are generated by metaheuristic optimization algorithms such as PSO, GWO, and hybrid HHO–GWO. The mass of payload is 1 kg. Therefore, the total mass of quadcopter has been changed from 3 kg to 4 kg in all missions. The root mean squared error (*RMSE*) performance criterion in path planning and tracking is denoted as:

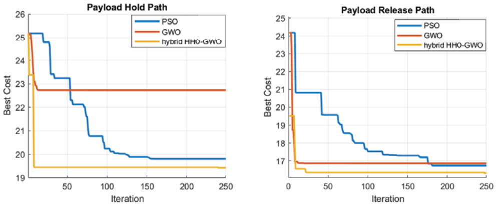
$$RMSE = \frac{1}{N_m}\sum_{i=1}^{N_m}\sqrt{(Xref_i - X_i)^2 + (Yref_i - Y_i)^2 + (Zref_i - Z_i)^2} \tag{52}$$

where $Xref_i$, $Yref_i$, and $Zref_i$ are reference positions of the quadcopter; X_i, Y_i, and Z_i are measured positions of the quadcopter in X, Y, and Z axes, respectively; and the total number of measurements is expressed with N_m. The energy efficiency can be calculated as:

$$E_{eff} = \frac{E_b - E_t}{E_b} \times 100 \tag{53}$$

where E_b is the total energy of the battery, and E_t is the total energy consumed by the quadcopter. The generated waypoints are presented for Scenario 1 in Table 2. The distances of the paths created are 37.53 m, 36.26 m, 35.68 m in Scenario 1 for the PSO, GWO, and hybrid HHO–GWO, respectively. The performance of the payload hold and release path is demonstrated in Figure 6. The path generated by metaheuristic optimization algorithms is illustrated on Scenario 1 with obstacles in Figure 7. When the convergence rate and minimum point are investigated, the maximum convergence rate and minimum point has been obtained for the proposed hybrid HHO–GWO algorithm. The shortest distance path is obtained with the hybrid HHO–GWO on Scenario 1.

The generated waypoints are introduced for Scenario 2 in Table 3. The distances of the paths created are 37.47 m, 40.72 m, and 36.73 m in Scenario 2 for the PSO, GWO, and hybrid HHO–GWO, respectively. The performance of the payload hold-and-release path is displayed in Figure 8. The paths generated by the metaheuristic optimization algorithms are indicated in Scenario 2 with obstacles in Figure 9. When the convergence rate and minimum point are investigated, the maximum convergence rate and minimum point has been obtained for the proposed hybrid HHO–GWO algorithm. The generated minimum path distance is obtained for the hybrid HHO–GWO in Scenario 2.



(**a**) (**b**)

Figure 6. The optimized payload hold (**a**) and release (**b**) path performance of metaheuristic optimization algorithms for Map 1.

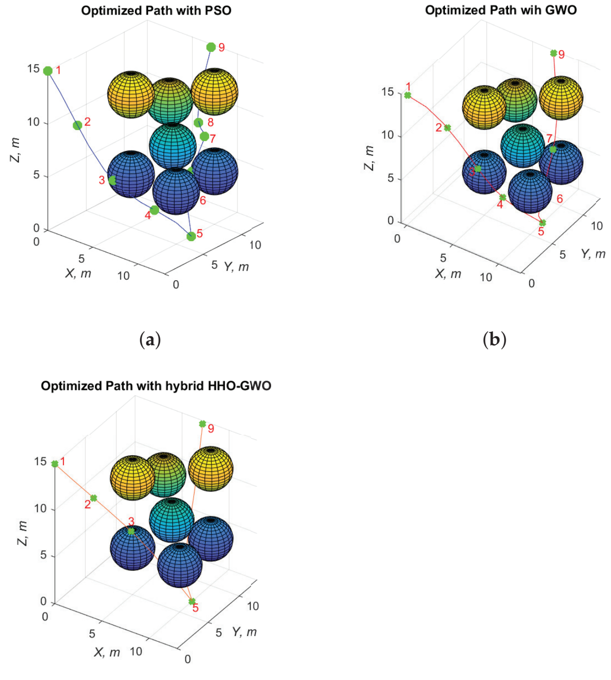
Table 2. Optimized waypoints for Scenario 1.

Waypoint Number	PSO	GWO	Hybrid HHO–GWO
	X_d^G (m), Y_d^G (m), Z_d^G (m)		
1	(0, 0, 15)	(0, 0, 15)	(0, 0, 15)
2	(1.23, 2.44, 9.35)	(2.89, 2.14, 11.10)	(2.94, 1.85, 11.37)
3	(4.78, 2.90, 5.13)	(5.90, 2.62, 7.38)	(5.60, 3.82, 7.69)
4	(8.05, 4.45, 2.82)	(7.29, 4.62, 3.30)	(7.82, 5.26, 4.30)
5	(10, 7, 0)	(10, 7, 0)	(10, 7, 0)
6	(8.9, 8.33, 5.26)	(9.71, 7.21, 2.81)	(8.74, 7.85, 3.39)
7	(9.63, 9.16, 8.48)	(9.93, 8.81, 7.32)	(8.25, 9.41, 7.05)
8	(8.23, 9.97, 9.05)	(9.02, 10.74, 10.6)	(8, 10.68, 10.31)
9	(7.5, 12.5, 15)	(7.5, 12.5, 15)	(7.5, 12.5, 15)
Path Distance	**37.53 m**	**36.26 m**	**35.68 m**

Table 3. Optimized waypoints for Scenario 2.

Waypoint Number	PSO	GWO	Hybrid HHO–GWO
	X_d^G (m), Y_d^G (m), Z_d^G (m)		
1	(0, 0, 15)	(0, 0, 15)	(0, 0, 15)
2	(0.85, 0.84, 12.33)	(5.32, 1.71, 6.99)	(1.96, 1.51, 10.33)
3	(2.71, 2.08, 7.35)	(8.25, 2.73, 5.90)	(4.17, 2.21, 6.81)
4	(6.72, 4.05, 3.26)	(8.77, 4.99, 3.89)	(6.77, 4.14, 3.13)
5	(10, 7, 0)	(10, 7, 0)	(10, 7, 0)
6	(10.75, 9.24, 4.44)	(6.63, 5.61, 1.71)	(10.01, 9.11, 5.95)
7	(8.2, 9.3, 7.85)	(4.23, 4.71, 5.16)	(8.7, 9.32, 7.94)
8	(5.9, 9.49, 10)	(2.52, 7.58, 8.98)	(6.43, 10.25, 12.01)
9	(5, 9, 15)	(5, 9, 15)	(5, 9, 15)
Path Distance	**37.47 m**	**40.72 m**	**36.73 m**

The generated waypoints are demonstrated for Scenario 3 in Table 4. The distances of the paths created are 31.32 m, 32.24 m, and 29.59 m on Map 3 for the PSO, GWO, and hybrid HHO–GWO, respectively. The performance of the payload hold-and-release path is displayed in Figure 10. The paths generated by the metaheuristic optimization algorithms are shown in Scenario 3 with obstacles in Figure 11. The numbers on Figures 7, 9 and 11 are used to label the waypoints obtained by the optimization algorithms. When the convergence rate and minimum point are investigated, the maximum convergence rate and minimum point were obtained for the proposed hybrid HHO–GWO algorithm. The generated minimum path distance is obtained for hybrid HHO–GWO in Scenario 3. To summarize, the path planning on all three maps is obtained for the proposed hybrid HHO–GWO algorithm for minimum distance and the maximum converge rate. The PSO algorithm is run 500 times for Scenarios 1–3, and the average running times for each scenario are determined as 66.14 s, 66.16 s, and 66.01 s, respectively. The GWO algorithm is run 500 times for Scenarios 1–3, and the average running times for each scenario are 65.12 s, 65.25 s, and 65.11 s, respectively. The proposed hybrid HHO–GWO algorithm is run 500 times for Scenarios 1–3, and the average running times for each scenario are measured as 64.09 s, 64.68 s, and 64.71 s, respectively. Note that all algorithms mentioned in the study are run on a PC device, which has an Intel i7-10750H, 6 cores, 2.6 GHz Turbo, and 32 GB RAM. All codes are compiled with MATLAB 2020b.

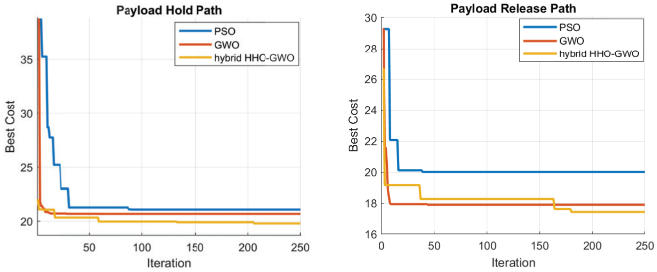


(**a**)

(**b**)

(**c**)

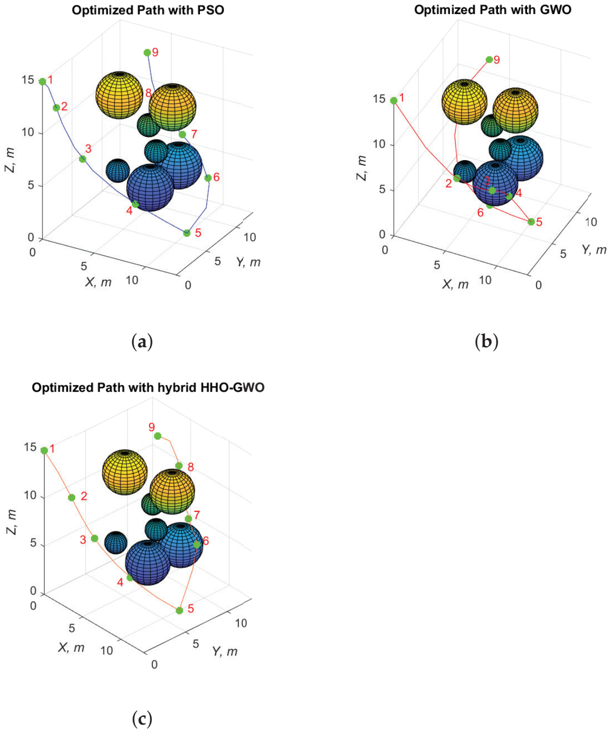
Figure 7. The optimized path for Map 1 (**a**) using PSO, (**b**) using GWO, and (**c**) using hybrid HHO–GWO.



(**a**)

(**b**)

Figure 8. The optimized payload hold (**a**) and release (**b**) path performance of metaheuristic optimization algorithms for Map 2.



(**a**)

(**b**)

(**c**)

Figure 9. The optimized path for Map 2 (**a**) using PSO, (**b**) using GWO, and (**c**) using hybrid HHO–GWO.

Table 4. Optimized waypoints for Scenario 3.

Waypoint Number	PSO	GWO	Hybrid HHO–GWO
		X_d^G (m), Y_d^G (m), Z_d^G (m)	
1	(0, 0, 15)	(0, 0, 15)	(0, 0, 15)
2	(1.98, 2.19, 8.8)	(0.51, 3.15, 9.64)	(1.96, 1.51, 10.33)
3	(6.71, 3.59, 4.31)	(1.71, 4.16, 4.62)	(4.17, 2.21, 6.81)
4	(8.32, 4.95, 1.54)	(4.48, 5.41, 0.74)	(6.77, 4.14, 3.13)
5	(8, 7, 0)	(8, 7, 0)	(8, 7, 0)
6	(7.82, 6.76, 3.22)	(6.44, 7.01, 3.07)	(10.01, 9.11, 5.95)
7	(6.95, 6.45, 7.77)	(7.12, 7.33, 4.89)	(8.7, 9.32, 7.94)
8	(5.82, 6.24, 8.66)	(6.48, 7.1, 8.27)	(6.43, 10.25, 12.01)
9	(4, 7, 10)	(4, 7, 10)	(4, 7, 10)
Path Distance	**31.32 m**	**32.24 m**	**29.59 m**

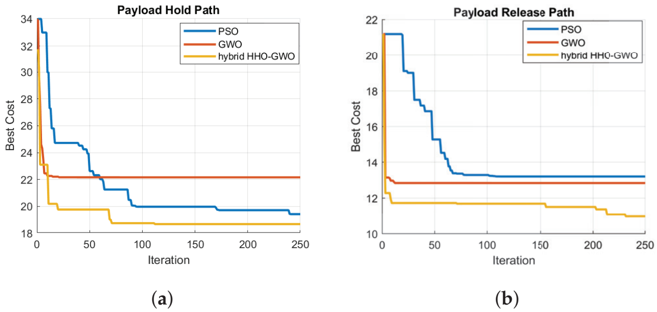


Figure 10. The optimized payload hold (**a**) and release (**b**) path performance of metaheuristic optimization algorithms for Map 3.

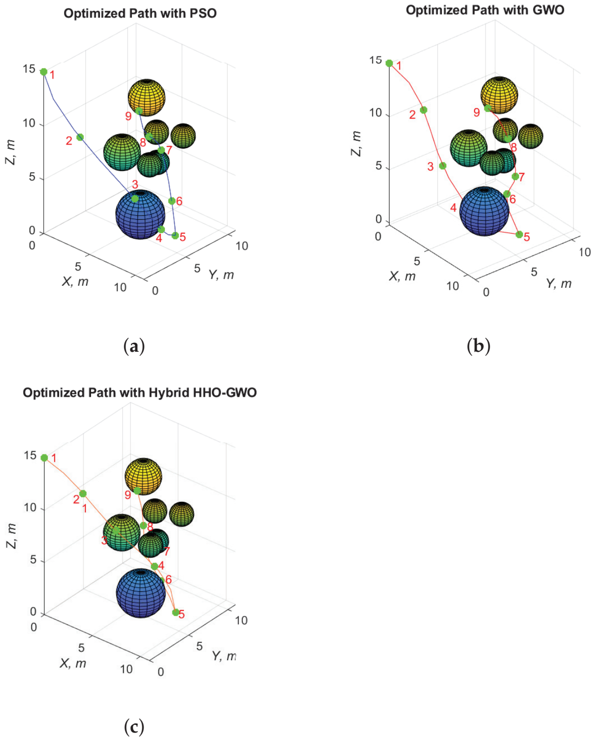


Figure 11. The optimized path for Map 3 (**a**) using PSO, (**b**) using GWO, and (**c**) using hybrid HHO–GWO.

The path tracking performance of the quadcopter is analyzed with these generated waypoints. The payload hold and release are carried out in waypoints 5 and 9, respectively. The performance of the quadcopter under both path tracking and sudden payload change is examined. The path tracking in a payload hold–release mission is illustrated in Figures 12–14 for Scenarios 1, 2, and 3, respectively. The total path, RMSE, target time, and energy efficiency performance criteria of metaheuristic algorithms are presented in Table 5. The total measured paths are 53.025 m, 51.631 m, and 50.7 m, and the mean square errors are 21.76 m, 19.98 m, and 19.57 m for PSO, GWO, and hybrid HHO–GWO, respectively, in Scenario 1. The total times of the payload hold–release mission in Scenario

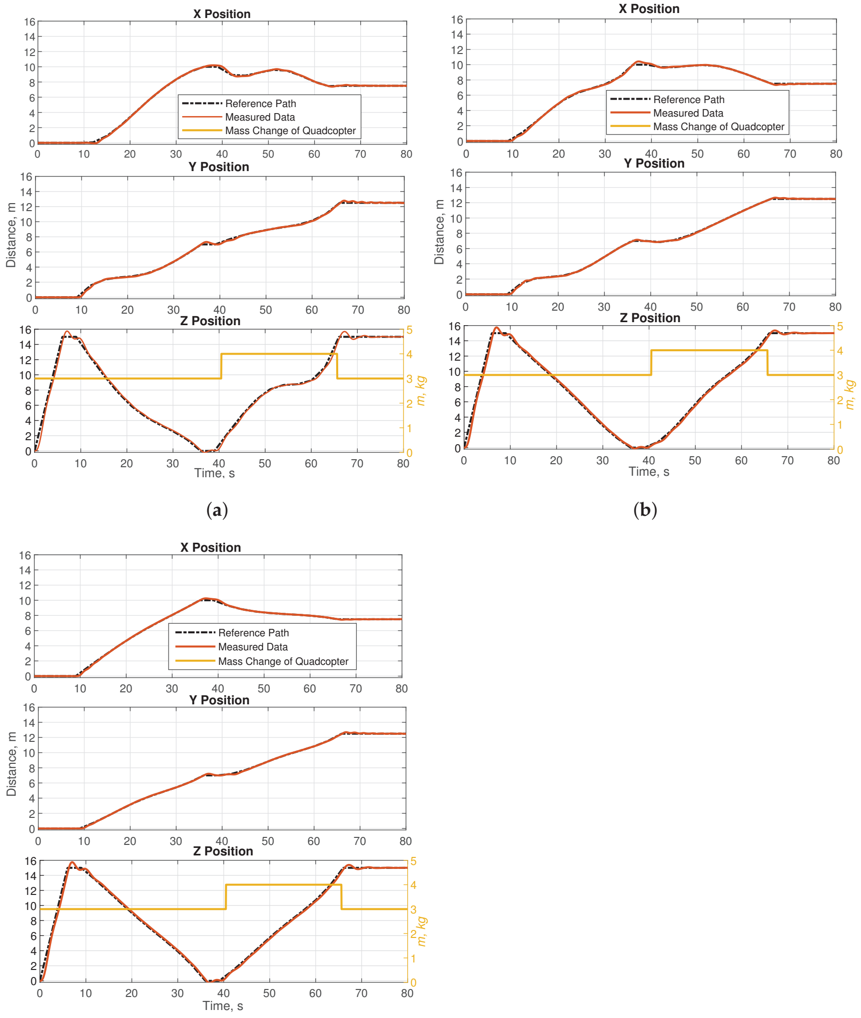
1 are 66.15 s, 65.01 s, and 64.12 s for the PSO, GWO, and hybrid HHO–GWO, respectively. The energy efficiencies of the quadcopter in Scenario 1 are obtained as 64.51%, 67.42%, and 68.08% for the PSO, GWO, and hybrid HHO–GWO respectively. The total measured paths are 52.92 m, 56.52 m, and 52.51 m, and the mean square errors are 19.86 m, 22.7 m, and 19.35 m for the PSO, GWO, and hybrid HHO–GWO, respectively, in Scenario 2. The total mission times are 66.18 s, 65.24 s, and 64.76 s, and the energy efficiencies obtained are 67.92%, 63.33%, and 68.74% for the PSO, GWO, and hybrid HHO–GWO, respectively, in Scenario 2. The total measured paths are 46.87 m, 47.80 m, and 44.72 m, and the mean square errors are 17.65 m, 18.49 m, and 16.92 m for the PSO, GWO, and hybrid HHO–GWO, respectively, in Scenario 3. The total mission times are 65.99 s, 65.01 s, and 64.71 s, and the energy efficiencies are 66.74%, 65.5%, and 68.81% for the PSO, GWO, and hybrid HHO–GWO, respectively, in Scenario 3. The minimum total path, mean square error, target time, and energy efficiency are obtained for the hybrid HHO–GWO in all Scenarios. When the path tracking performance of the quadcopter in Figure 14 for Scenario 3, which has the highest environmental difficulty level, is evaluated, it is seen that the least change in the Z-axis occurs with the proposed algorithm. This shows that the energy is used optimally. The results show that the hybrid HHO–GWO algorithm has the highest energy efficiency.

Table 5. Performance criteria of metaheuristic optimization algorithms for path planning and tracking.

Map Number	Algorithms	Total Path (m)	RMSE (m)	Target Time (s)	Energy Efficiency (%)
1	PSO	53.03	21.76	66.15	64.51
1	GWO	51.63	19.98	65.01	67.41
1	hybrid HHO–GWO	50.70	19.57	64.12	68.08
2	PSO	52.92	19.86	66.18	67.92
2	GWO	56.52	22.70	65.24	63.33
2	hybrid HHO–GWO	52.11	19.35	64.76	68.74
3	PSO	46.87	17.65	65.99	66.74
3	GWO	47.80	18.49	65.01	65.50
3	hybrid HHO–GWO	44.72	16.92	64.71	68.81

To summarize, the path planning and tracking control strategy of the quadcopter have been proposed in this study. The path planning has been achieved via PSO, GWO, and the proposed hybrid HHO–GWO algorithms. The results of path planning show that the shortest and safest paths are obtained for all scenarios. After this, the path-tracking performance of the quadcopter in a payload hold–release mission is investigated for all scenarios. The path-tracking results express that the minimum total path, mean square error, target time and energy efficiency of quadcopter in payload transportation mission have been obtained for all scenarios. The path-tracking error due to the mass uncertainty of the quadcopter has been minimized in all scenarios with obstacles. The contributions of this study are the following:

- A hybrid HHO–GWO optimization algorithm with high convergence speed for path planning has been proposed,
- The position error of the quadcopter caused by the sudden change during payload holding and releasing is examined;
- The errors that occur in path tracking under sudden payload changes are minimized with the newly proposed control strategy.

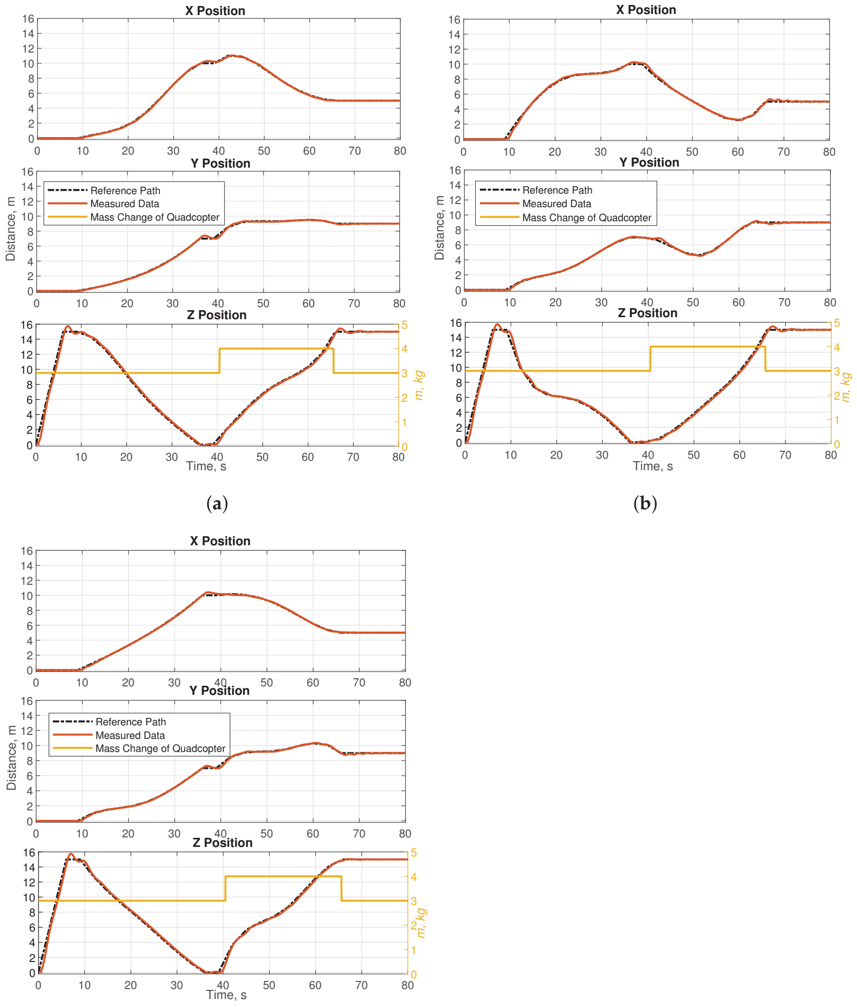


(**a**)

(**b**)

(**c**)

Figure 12. The path-tracking performance of the quadcopter for Map 1 (**a**) using PSO, (**b**) using GWO, and (**c**) using hybrid HHO–GWO.

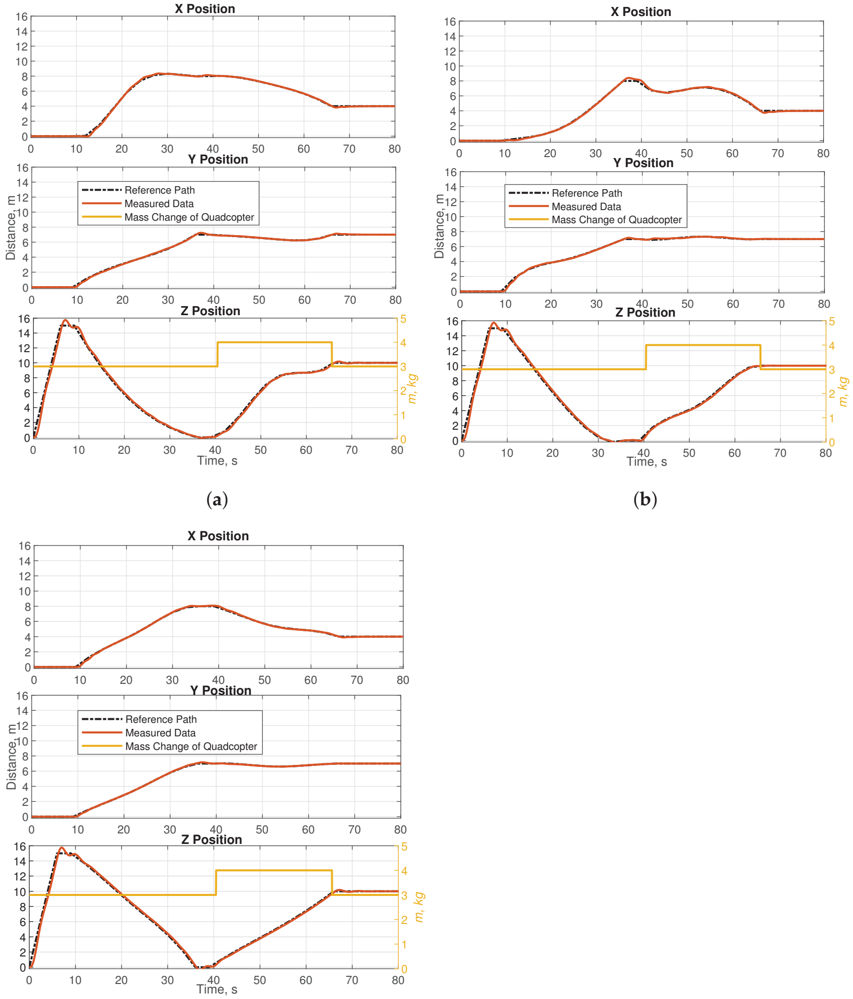


(**a**)

(**b**)

(**c**)

Figure 13. The path-tracking performance of quadcopter for Map 2 (**a**) using PSO, (**b**) using GWO, and (**c**) using hybrid HHO–GWO.



(**a**)

(**b**)

(**c**)

Figure 14. The path-tracking performance of quadcopter for Map 3 (**a**) using PSO, (**b**) using GWO, and (**c**) using hybrid HHO–GWO.

6. Conclusions and Future Work

In this study, a new metaheuristic path planning and tracking algorithm for payload hold–release mission is proposed to avoid obstacles. A hybrid HHO–GWO algorithm is proposed by combining the HHO algorithm, with high convergence speed, and the GWO algorithm, which has a high performance in avoiding local optima, so that the quadcopter can both avoid obstacles and track the planned path with minimum error. The performance of the proposed path-planning algorithm is compared with PSO and GWO. The minimum path distance and maximum convergence rate have been obtained with the newly proposed hybrid HHO–GWO metaheuristic optimization algorithm. The waypoints that the quadcopter desires to track are generated with the optimization algorithm not only minimizing distance but also energy. The path tracking has been carried out by these generated waypoints. The payload hold-and-release mission has been realized with a path-tracking controller. The mass component in the quadcopter model is changed during payload holding and releasing. The position errors occur in path tracking with the sudden mass change of the quadcopter in this payload holding and releasing task. The results of path tracking are indicated by the minimum total path, mean square error, total time and consumed energy have been obtained for the newly proposed hybrid HHO–GWO. The most important contribution of this study is that the proposed control strategy and the position error caused by this mass uncertainty can be minimized. It is planned to perform environmental tests of the proposed metaheuristic-based approach by embedding it on a single UAV in the future. By comparing the results obtained with the results of our current study, studies will be focused on the optimum solution of path generation and tracking problems with the cooperation of multiple UAVs.

Author Contributions: Conceptualization, E.B., A.A. and R.H.; methodology, E.B., A.A. and R.H.; software, E.B., A.A. and R.H.; writing—original draft preparation, E.B. and A.A.; writing—review and editing, E.B., A.A. and R.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Yu, X.; Li, C.; Zhou, J. A constrained differential evolution algorithm to solve UAV path planning in disaster scenarios. *Knowl.-Based Syst.* **2020**, *204*, 106209. [CrossRef]
2. Ropero, F.; Muñoz, P.; R-Moreno, M.D. TERRA: A path planning algorithm for cooperative UGV–UAV exploration. *Eng. Appl. Artif. Intell.* **2019**, *78*, 260–272. [CrossRef]
3. Sivčev, S.; Rossi, M.; Coleman, J.; Omerdić, E.; Dooly, G.; Toal, D. Collision detection for underwater ROV manipulator systems. *Sensors* **2018**, *18*, 1117. [CrossRef] [PubMed]
4. De Vivo, F.; Battipede, M.; Johnson, E. Infra-red line camera data-driven edge detector in UAV forest fire monitoring. *Aerosp. Sci. Technol.* **2021**, *111*, 106574. [CrossRef]
5. Altan, A.; Hacıoğlu, R. Model predictive control of three-axis gimbal system mounted on UAV for real-time target tracking under external disturbances. *Mech. Syst. Signal Process.* **2020**, *138*, 106548. [CrossRef]
6. Silvagni, M.; Tonoli, A.; Zenerino, E.; Chiaberge, M. Multipurpose UAV for search and rescue operations in mountain avalanche events. *Geomat. Nat. Hazards Risk* **2017**, *8*, 18–33. [CrossRef]
7. Cui, L.; Zhang, R.; Yang, H.; Zuo, Z. Adaptive super-twisting trajectory tracking control for an unmanned aerial vehicle under gust winds. *Aerosp. Sci. Technol.* **2021**, *115*, 106833. [CrossRef]
8. Causa, F.; Fasano, G. Multiple UAVs trajectory generation and waypoint assignment in urban environment based on DOP maps. *Aerosp. Sci. Technol.* **2021**, *110*, 106507. [CrossRef]
9. Sun, P.; Boukerche, A. Performance modeling and analysis of a UAV path planning and target detection in a UAV-based wireless sensor network. *Comput. Netw.* **2018**, *146*, 217–231. [CrossRef]
10. Zhang, Z.; Li, J.; Wang, J. Sequential convex programming for nonlinear optimal control problems in UAV path planning. *Aerosp. Sci. Technol.* **2018**, *76*, 280–290. [CrossRef]
11. Huang, Y.; Mei, W.; Xu, J.; Qiu, L.; Zhang, R. Cognitive UAV communication via joint maneuver and power control. *IEEE Trans. Commun.* **2019**, *67*, 7872–7888. [CrossRef]
12. Zhang, Y.; Zhang, R.; Li, H. Graph-based path decision modeling for hypersonic vehicles with no-fly zone constraints. *Aerosp. Sci. Technol.* **2021**, *116*, 106857. [CrossRef]
13. Radmanesh, R.; Kumar, M.; French, D.; Casbeer, D. Towards a PDE-based large-scale decentralized solution for path planning of UAVs in shared airspace. *Aerosp. Sci. Technol.* **2020**, *105*, 105965. [CrossRef]
14. Pehlivanoglu, Y.V. A new vibrational genetic algorithm enhanced with a Voronoi diagram for path planning of autonomous UAV. *Aerosp. Sci. Technol.* **2012**, *16*, 47–55. [CrossRef]
15. Baumann, M.; Leonard, S.; Croft, E.A.; Little, J.J. Path planning for improved visibility using a probabilistic road map. *IEEE Trans. Robot.* **2010**, *26*, 195–200. [CrossRef]
16. Bayili, S.; Polat, F. Limited-Damage A*: A path search algorithm that considers damage as a feasibility criterion. *Knowl.-Based Syst.* **2011**, *24*, 501–512. [CrossRef]
17. Moon, C.B.; Chung, W. Kinodynamic planner dual-tree RRT (DT-RRT) for two-wheeled mobile robots using the rapidly exploring random tree. *IEEE Trans. Ind. Electron.* **2014**, *62*, 1080–1090. [CrossRef]
18. Battulwar, R.; Winkelmaier, G.; Valencia, J.; Naghadehi, M.Z.; Peik, B.; Abbasi, B.; Parvin, B.; Sattarvand, J. A Practical Methodology for Generating High-Resolution 3D Models of Open-Pit Slopes Using UAVs: Flight Path Planning and Optimization. *Remote. Sens.* **2020**, *12*, 2283. [CrossRef]

19. Huang, L.; Qu, H.; Ji, P.; Liu, X.; Fan, Z. A novel coordinated path planning method using k-degree smoothing for multi-UAVs. *Appl. Soft Comput.* **2016**, *48*, 182–192. [CrossRef]
20. Dorigo, M.; Birattari, M.; Stutzle, T. Ant colony optimization. *IEEE Comput. Intell. Mag.* **2006**, *1*, 28–39. [CrossRef]
21. Ali, Z.A.; Zhangang, H.; Hang, W.B. Cooperative path planning of multiple UAVs by using max–min ant colony optimization along with cauchy mutant operator. *Fluct. Noise Lett.* **2021**, *20*, 2150002. [CrossRef]
22. Shao, S.; Peng, Y.; He, C.; Du, Y. Efficient path planning for UAV formation via comprehensively improved particle swarm optimization. *ISA Trans.* **2020**, *97*, 415–430. [CrossRef] [PubMed]
23. Dewangan, R.K.; Shukla, A.; Godfrey, W.W. Three dimensional path planning using grey wolf optimizer for UAVs. *Appl. Intell.* **2019**, *49*, 2201–2217. [CrossRef]
24. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey wolf optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [CrossRef]
25. Ganguly, S. Multi-objective distributed generation penetration planning with load model using particle swarm optimization. *Decis. Mak. Appl. Manag. Eng.* **2020**, *3*, 30–42. [CrossRef]
26. Negi, G.; Kumar, A.; Pant, S.; Ram, M. Optimization of complex system reliability using hybrid grey wolf optimizer. *Decis. Mak. Appl. Manag. Eng.* **2021**, *4*, 241–256. [CrossRef]
27. Das, M.; Roy, A.; Maity, S.; Kar, S.; Sengupta, S. Solving fuzzy dynamic ship routing and scheduling problem through new genetic algorithm. *Decis. Mak. Appl. Manag. Eng.* **2021**. [CrossRef]
28. Sasongko, R.A.; Rawikara, S.; Tampubolon, H.J. UAV obstacle avoidance algorithm based on ellipsoid geometry. *J. Intell. Robot. Syst.* **2017**, *88*, 567–581. [CrossRef]
29. Singla, A.; Padakandla, S.; Bhatnagar, S. Memory-based deep reinforcement learning for obstacle avoidance in UAV with limited environment knowledge. *IEEE Trans. Intell. Transp. Syst.* **2019**, 22, 107–118. [CrossRef]
30. Iacono, M.; Sgorbissa, A. Path following and obstacle avoidance for an autonomous UAV using a depth camera. *Robot. Auton. Syst.* **2018**, *106*, 38–46. [CrossRef]
31. Zheng, L.; Zhang, P.; Tan, J.; Li, F. The obstacle detection method of UAV based on 2D LIDAR. *IEEE Access* **2019**, *7*, 163437–163448. [CrossRef]
32. Liu, Y.; Rajappa, S.; Montenbruck, J.M.; Stegagno, P.; Bülthoff, H.; Allgöwer, F.; Zell, A. Robust nonlinear control approach to nontrivial maneuvers and obstacle avoidance for quadrotor UAV under disturbances. *Robot. Auton. Syst.* **2017**, *98*, 317–332. [CrossRef]
33. Altan, A.; Aslan, Ö.; Hacıoğlu, R. Real-time control based on NARX neural network of hexarotor UAV with load transporting system for path tracking. In Proceedings of the 2018 6th International Conference on Control Engineering & Information Technology (CEIT), Istanbul, Turkey, 25–27 October 2018; pp. 1–6.
34. Shirani, B.; Najafi, M.; Izadi, I. Cooperative load transportation using multiple UAVs. *Aerosp. Sci. Technol.* **2019**, *84*, 158–169. [CrossRef]
35. Lee, H.I.; Yoo, D.W.; Lee, B.Y.; Moon, G.H.; Lee, D.Y.; Tahk, M.J.; Shin, H.S. Parameter-robust linear quadratic Gaussian technique for multi-agent slung load transportation. *Aerosp. Sci. Technol.* **2017**, *71*, 119–127. [CrossRef]
36. Hashemi, D.; Heidari, H. Trajectory planning of quadrotor UAV with maximum payload and minimum oscillation of suspended load using optimal control. *J. Intell. Robot. Syst.* **2020**, *100*, 1369–1381. [CrossRef]
37. Kuantama, E.; Vesselenyi, T.; Dzitac, S.; Tarca, R. PID and Fuzzy-PID control model for quadcopter attitude with disturbance parameter. *Int. J. Comput. Commun. Control.* **2017**, *12*, 519–532. [CrossRef]
38. Selby, W.C. Autonomous Navigation and Tracking of Dynamic Surface Targets On-Board a Computationally Impoverished Aerial Vehicle. Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 2011.
39. Luukkonen, T. Modelling and control of quadcopter. *Indep. Res. Proj. Appl. Math. Espoo* **2011**, *22*, 22.
40. Wang, P.; Man, Z.; Cao, Z.; Zheng, J.; Zhao, Y. Dynamics modelling and linear control of quadcopter. In Proceedings of the 2016 International Conference on Advanced Mechatronic Systems (ICAMechS), Melbourne, VIC, Australia, 30 November–3 December 2016; pp. 498–503.
41. Beard, R. Quadrotor Dynamics and Control Rev 0.1. 2008. Available online: https://scholarsarchive.byu.edu/cgi/viewcontent.cgi?article=2324&context=facpub (accessed on 10 April 2022).
42. Koksal, N.; Jalalmaab, M.; Fidan, B. Adaptive linear quadratic attitude tracking control of a quadrotor UAV based on IMU sensor data fusion. *Sensors* **2019**, *19*, 46. [CrossRef]
43. Abhishek, B.; Ranjit, S.; Shankar, T.; Eappen, G.; Sivasankar, P.; Rajesh, A. Hybrid PSO-HSA and PSO-GA algorithm for 3D path planning in autonomous UAVs. *SN Appl. Sci.* **2020**, *2*, 1805. [CrossRef]
44. Ghambari, S.; Lepagnot, J.; Jourdan, L.; Idoumghar, L. A comparative study of meta-heuristic algorithms for solving UAV path planning. In Proceedings of the 2018 IEEE Symposium Series on Computational Intelligence (SSCI), Bangalore, India, 18–21 November 2018; pp. 174–181.
45. Heidari, A.A.; Mirjalili, S.; Faris, H.; Aljarah, I.; Mafarja, M.; Chen, H. Harris hawks optimization: Algorithm and applications. *Future Gener. Comput. Syst.* **2019**, *97*, 849–872. [CrossRef]
46. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the ICNN'95—International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948.

energies

MDPI

Article

Development of a Predicting Model for Calculating the Geometry and the Characteristic Curves of Pumps Running as Turbines in Both Operating Modes

Silvio Barbarelli *, Vincenzo Pisano and Mario Amelio

Department of Mechanical, Energy and Management Engineering, University of Calabria,
Via Ponte P. Bucci Cubo 44/C, 87036 Rende, Italy; vincent88kr@hotmail.it (V.P.); mario.amelio@unical.it (M.A.)
* Correspondence: silvio.barbarelli@unical.it

Citation: Barbarelli, S.; Pisano, V.; Amelio, M. Development of a Predicting Model for Calculating the Geometry and the Characteristic Curves of Pumps Running as Turbines in Both Operating Modes. *Energies* **2022**, *15*, 2669. https://doi.org/10.3390/en15072669

Academic Editors: Luis Hernández-Callejo, Sergio Nesmachnow and Sara Gallardo Saavedra

Received: 28 February 2022
Accepted: 31 March 2022
Published: 6 April 2022

Abstract: This article is part of a scientific research project dedicated to the study of plants generating electricity from hydraulic sources by exploiting the technology of inverted flow centrifugal pumps, also known as PAT. The main purpose is to provide a contribution to the methodologies already existing in the literature, creating a one-dimensional model capable of predicting the characteristic curves of the machine, in both operating modes, without knowing its geometry. The first part of the work is therefore focused on the description of the fluid dynamic model, capable of determining the losses in the various sections of the machine, using different calculation approaches. The development of this model was carried out using a set of six centrifugal pumps, measured at the DIMEG Department of the University of Calabria and at the University of Trento. For this range of pumps, the characteristic curves were therefore obtained, both in pump and turbine operation. The second part of this work focuses on the description of the geometric model, useful as generally few data are provided in the manufacturer's catalog, which is necessary for the correct installation of the machine. The geometric model can determine, using these parameters and through good design techniques and statistical diagrams, the entire geometry of the machine. This model refers to a pump prototype, having a simplified geometry, for which the characteristic curves of the PAT are obtained in pump operation. These curves are compared with those present in the manufacturer's catalog, and if they show too high deviations, it is possible to act on some geometric parameters, chosen based on a sensitivity analysis. Once satisfactory results have been obtained, it is possible to obtain the characteristic curves also in turbine operation. This procedure has been finally applied to another PAT, taken as an example.

Keywords: pumps as turbines; predicting model; experimental test; optimization procedure

1. Introduction

Over the years, world energy consumption has drastically increased, with different trends in various countries based on the degree of wealth and development and the availability of raw materials and resources. Consequently, as regards the water sector, the idea of using inverted flow pumps, also known as PAT [1,2], has begun to make its way into the market. The first researchers who realized the actual potential in exploiting a pump used as a turbine were Thoma and Kittredge [3]. They began experimenting in laboratories on this technology around the 1930s. There are also traces of the use of this technology in the 1970s; however, it failed to play an important role as energy was cheap and there was still no sensitivity towards recycling and saving. It was therefore more convenient to buy energy directly from the grid, and few tended to invest in new plants to produce a small number of kilowatt-hours. However, in the 1980s, some factors prompted a re-evaluation of the use of this technology, favoring its development and its establishment on the market, including the following:

- The inability of the distribution networks to reach rural settlements. The latter were therefore forced to produce energy independently, also using inverted flow pumps.
- The use of power electronics: the adjustment of the machine using inverters makes it possible to obtain acceptable efficiencies for a range of different flow rates (previously the adjustments were manual and only hydraulic).

PATs are mainly used as pressure-reducing valves (PRVs) [4,5] or in the pico/micro-hydroelectric sector [6]. Regarding the first application, the typical examples are aqueducts: water distribution systems that must always be under pressure. The pressure must remain within certain values: a minimum value, necessary to reach the highest altitudes, but at the same time a maximum value that must not be exceeded, as losses increase and cause problems of operability. It is therefore necessary to adjust the pressure, as the excess energy would be lost, for this reason, it is preferable to use an inverted flow pump. The pico-hydroelectric sector, on the other hand, is mainly linked to self-production. Even small water sources, commonly neglected for economic reasons, are therefore exploited. The aim is to satisfy one's energy needs, without selling the energy or feeding it into the power grid. It is therefore a question of withdrawing as little as possible from the watercourse, which is then released, reducing the impact on the resource. PAT can also be used by harnessing tidal energy [7], i.e., harvesting its height range in natural bays and estuaries or in artificial barrages, or extracting the kinetic energy from the tidal currents across natural and artificial channels [8]. The main feature of PAT pumps is represented by their reversibility, as it is possible, by reversing the direction of the fluid path inside the machine, to produce energy. They are generally marketed as monobloc electric pumps [9,10] in which the motor is of the asynchronous type. It can play two opposite roles: motor in direct operation and electric generator in reverse operation.

The main advantages of the use of PAT technology are represented by the following:

- Lower costs compared to a normal hydraulic turbine, especially for small units, below 50 kW;
- Simplicity of installation and maintenance;
- The wide range of models available on the market.

On the other hand, since pumps are not designed to operate as a turbine, they have the following disadvantages:

- The absence of guide vanes, which excludes the possibility of making hydraulic adjustments;
- The lower efficiency compared to a well-designed hydraulic turbine, especially in the off-design conditions;
- The lack of information about the characteristic curves for the turbine, as the manufacturer of the machine supplies only those for the pump.

PATs can therefore be used as a replacement for traditionally employed turbines (Francis, Kaplan, Pelton) and turbines discussed in other articles [11,12]. However, it is necessary to make sure that the PAT operation is well adapted to the characteristics of the system where the PAT will be installed. It is necessary to establish, knowing the characteristic curves of operation of a pump, the characteristic curves of the machine that operates as a turbine. The aim of the present paper is to develop a combined procedure for assessing both the geometry (which is not given by manufacturers) and the fluid dynamic performances of a generic PAT. For this purpose, many fluid dynamics models calculating losses, head, and efficiencies by changing the flow rate of the machine were calibrated on a sample of six PATs. Then, a geometrical model capable of reconstructing a prototypal geometry, by a rough sizing of the PAT, was developed, involving charts, maps, good design rules, statistical correlations, and so on. Compared to the past approaches, in this work, the possibility of refining the calculus of geometrical parameters of the geometry model is given. The deviation between the curves given by the manufacturer's catalog and the ones foreseen in the pump operation can be reduced or annulled by changing one or more geometrical parameters, calculated by the model. A sensitivity analysis was conducted for this purpose, in such a way as to understand which parameters to act on. The objective of this research is therefore to provide a flexible tool, which allows calculating the performance of any PAT, in both modes of operation, starting from little information available from the catalog provided by the manufacturer. This would make it possible for anyone who decides to approach this technology to facilitate the choice of the PAT that guarantees the best efficiency according to the energy resources available. As already mentioned, only the behavior of the machine when operating as a pump is known, but not the behavior of the machine when operating as a turbine.

State of the Art

In the literature, there are different approaches in the study of PAT that can be grouped as follows:

1. Simple statistical correlations that aim to establish a connection between the point of better efficiency (BEP) in pump operation and that in turbine operation. In more detail, from the examination of the position of the BEP points in the pump and turbine operation, of machines for which these are known, laws are derived which can then be used to predict the location of the BEP of a new machine. For example, Child [13], Sharma [14], Alatorre [15], and Stepanoff [16] combine the best head ratio and the best flow ratio with respect to the total efficiency of the pump; Hancock [17] correlates these reports to the total efficiency of the turbine; Schmield [18] relates these relationships to the hydraulic efficiency of the pump; and Grover [19] and Hergt [20] relate these ratios to the characteristic speed of the turbine.
2. The PAT performance prediction method using specific speed, where flow rate and head are expressed as a function of specific speed [21]. Different expressions of specific speeds are used, which are gradually refined and improved to ensure better accuracy of the results. Some examples are as follows: Derakshan applied the dimensionless specific speed to obtain different relations, valid for centrifugal pumps with specific speed $n_s < 60$ [22]; Nautiyal proposed an additional parameter through which it is possible to obtain the trend of the prevalence and the flow rate [23]; Singh proposed a correlation based on experimentation performed on a sample of 13 pumps and subsequently applied it to the pump under examination, thus obtaining the relationship between the specific speed in turbine operation and that in pump operation [24]; Tan, by testing the hydraulic performance of centrifugal pumps, used both in direct and reverse operation, obtained different linear relationships between the pump and turbine parameters [25]; Stefanizzi established a relationship between specific speed under pump and turbine mode, based on data obtained from the performance of 27 pumps, and subsequently it used to predict the performance of 11 new PATs [26].
3. Empirical correlations: Derakshan's [27] methodology proposes head–flow and power–flow polynomial curves, interpolated on the available PAT sample. These polynomials are dimensionless based on the values of the flow rate, head, and power of the PAT at the BEP and can be used in a universal way for predicting the curves of head, power, and efficiency versus flow rate for any machine.
4. One-dimensional model: Venturini [28] developed a prediction model based on the physics of the machine and consisting in the use of loss coefficients and specific parameters, through an optimization procedure, which is applied to the machine operating as a pump and subsequently as a turbine.
5. Numerical analysis and CFD, for axial flow centrifugal pumps, which allow reconstructing, through a structured step-by-step methodology, the characteristic curve in pump mode, and subsequently in turbine mode, and predicting the behavior of the fluid inside the turbomachinery [29].
6. For commercial centrifugal radial flow pumps, through computer numerical simulations, a methodology has been developed that makes it possible to predict the characteristic curves, in both operating modes, with errors of less than 10% compared to the mathematical model [30]. The operating conditions of the site are then obtained,

providing a methodology that allows the choice of the most suitable turbomachine to obtain electricity in those areas that do not have access to it, exploiting small hydroelectric resources.

All the proposed models can be improved because it is very difficult to foresee the performance curves of any PAT, given the very wide range of machines present on the market. For some PATs, these models provided acceptable results; for others, they did not. The PATs show phenomena of instability in the associated fluid, generating S-curves or cavitation phenomena and making the performance lower than a traditional Kaplan, Francis, or Pelton turbine. These instabilities are linked to the geometric configuration of the machine and to the deviation of the parameters calculated from the design values or measured in the laboratory. Given their use and their high energy consumption, it is therefore essential that their performance be optimized, and different models or design methodologies have been developed in this regard. Recent literature shows that in-depth studies have been carried out concerning the impact that the geometric parameters have on the performance of PAT, studies of a theoretical, numerical, and experimental nature [31]. A model has recently been developed those acts on the shape of the impellers of centrifugal compressors, given its influence on the overall performance of the machine [32]. It references genetic algorithms (GA) and a 3D simulation, which act on certain parameters such as the angle of the blades at the leading and trailing edges and the point where the splitter blades are connected. A new and performing design has therefore been obtained, which contributes to the research and development of compressors, without altering the technical characteristics of the fluid, to be able to replace low-consumption engines with ecological and economical fuel. A further solution, to optimize the PAT performances, refers to a numerical model which can determine the most advantageous geometric structure of the water cut [33]. Its finite thickness interferes with the flow at the entrance to the duct, generating swirling phenomena and deviations of the flow lines. Different stretching and cutting water thickness values at variable inclination are then analyzed using CFD simulations to identify the geometric features that have the greatest impact on machine performance. In recent literature, there is also a discussion that refers to PATs with low specific speed values in pico-hydropower plants [34]. This research was carried out by referring to regenerative pump models, given the characteristics that mark them in terms of stability and constructive simplicity. The approach used is both theoretical, referring to the momentum exchange theory, and through a 3D numerical simulation, to study the behaviour and performance of the machine in turbine operation. It is evident that PATs are the object of study for many researchers, as this technology is still under development and improvement. There are numerous contributions and optimizations that have been made in recent years, with different approaches and innovative ideas, both with a purely theoretical treatment and through a subsequent experimental verification.

2. Materials and Methods

The fluid dynamic models in both pump and turbine operation are shown separately, highlighting the various steps necessary to obtain the characteristic curves of the machine. These models are tuned on a sample of 6 centrifugal pumps, having specific speed changing in the range 9.05 ÷ 43.48. On these pumps, the measurements of the geometric parameters and the experimental tests on the test bench were carried out, as the flow rate varied, both in direct and reverse operation.

2.1. Fluid Dynamic Model

As already mentioned in the introduction, the proposed fluid dynamic models can predict the performance of the PATs, both in pump and turbine operation. The model provides head, efficiency, and loss variations versus flow rates changes. Figure 1 shows a flow chart summarizing the main steps of the development.

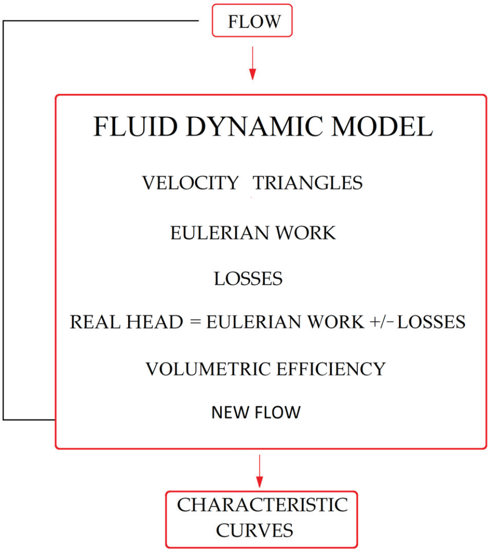


Figure 1. Flow chart summarizing the main steps of the development.

The evaluation of the losses inside a hydraulic machine cannot ignore the knowledge of its geometry, as it is indispensable for doing the calculations. It is convenient to adopt a prototype of a centrifugal pump, already described in another article [35], with characteristics common to most of the pumps available. Figure 2 shows a schematic representation of it, highlighting the passage sections and the symbols used to identify them.

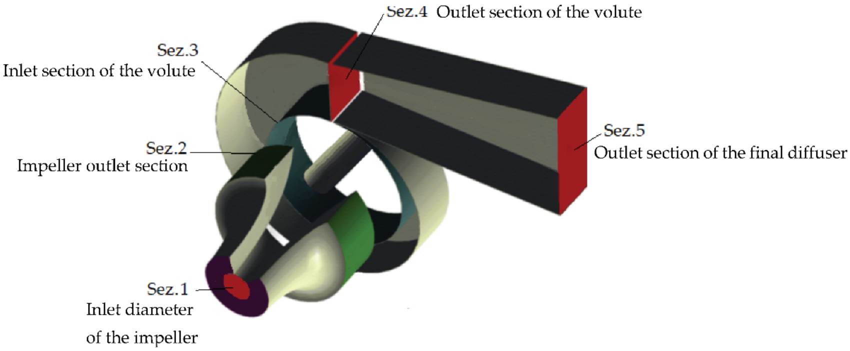

Sez.4 Outlet section of the volute
Sez.3
Inlet section of the volute
Sez.5
Outlet section of the final diffuser
Sez.2
Impeller outlet section
Sez.1
Inlet diameter
of the impeller

Figure 2. Reference geometry.

Regarding the main components that make up the machine, the simplifications adopted are as follows:

Aspiration (Section 0): A conical axial section was adopted, and the hypothesis was also assumed that the inlet diameter d_0 coincides with the blade tip diameter d_{1p} of the inlet section of the impeller.

Impeller (Sections 1–2): As regards the inlet section, a truncated conical surface was considered, having upper and lower base diameters d_{1m} and d_{1p} and laterally delimited by the width of the blades. As a representative diameter [36], the one in correspondence with the average current line has been adopted, which divides the section into two equal parts.

Volute (Sections 3–4): Given the complexity of its geometry, some simplifications have been adopted regarding the section and its evolution. A volute with a square terminal section is assumed, with sides b and h_v. The height of the volute varies linearly along the peripheral direction, until h_v is reached. It is also assumed that the terminal section has a normal along the tangential direction.

Final diffuser (Section 5): It is assumed that the dimensions vary linearly moving from the inlet to the outlet section according to a reflection equal to the tangent of an angle α_d, set equal to 3.5 degrees.

In the next sections, pump and turbine operations are separately analyzed.

2.1.1. Pump Operation

The fluid dynamic model relating to pump operation is shown, referring to the procedure described in Figure 1.

Velocity Triangles

To evaluate the hydraulic losses, the model calculates the flow speed in the various sections of the machine. Initially, this calculus is done at the inlet and outlet of the impeller, given the direct influence on the theoretical head (Eulerian work) estimation. The hypothesis made for the inlet section is that the fluid reaches the impeller in a direction perpendicular to the passage area, therefore with an angle α_1 equal to 90°. The tangential component of the absolute inlet speed is therefore equal to zero (c_{u1} = 0), as can be seen from the velocity triangles represented in Figure 3.

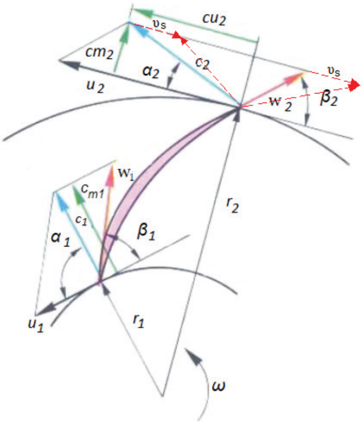


Figure 3. Velocity triangles at the inlet and outlet of the impeller in pump operation under design conditions (BEP) with slip deviation.

A smaller tangential component of the absolute speed (c_{u2}) is obtained, and thus a lower theoretical head is obtained compared to that determined in the one-dimensional design. It represents an inability of the impeller of the machine, having a finite number of blades, to transfer all energy to the fluid. To account for this loss of performance, the slip speed (Figure 4), v_s, has been calculated:

$$v_s = (1 - h_0) \cdot u_2 \tag{1}$$

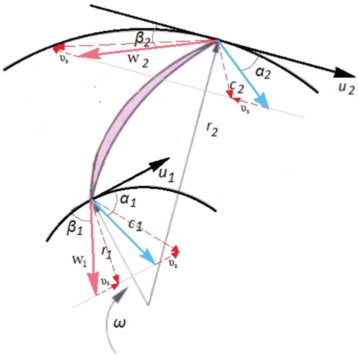


Figure 4. Impeller input–output speed triangles in turbine operation with slip deviation.

The tangential component of the absolute velocity, corrected with the slip, is obtained from the following expression:

$$c_{u2}^{*} = c_{u2} - \upsilon_s \tag{2}$$

In Equation (1), h_0 represents the slip factor.

Many correlations were proposed for assessing the h_0 parameter:

- Stodola [37,38] assumes that the motion of the fluid at the exit of the impeller is the sum of a main flow, which is guided by the blades, and of a vortex, having a rotation speed equal in modulus but in the opposite direction to that of the impeller. The diameter to which this vortex refers corresponds to the minimum passage section at the exit of the impeller.
- Stanitz [39,40], after a series of experimental tests, highlighted how the speed of slip, υ_s, was independent of the angle β_{2p} and not affected by the compressibility effects but was a function exclusively of the number of blades, z.
- Busemann [39,40] considered radial impellers having thick blades in an infinitesimal logarithmic spiral, which transforms into a rectilinear array made of infinite foil profile planes of infinitesimal thickness. Furthermore, he assumed that the upstream machine elements downstream of the rotor are sufficiently distant to have no effect on the behavior of the fluid in the mobile blade.
- Qiu, Mallikarachchi, and Anderson [41,42], who analyzed the various computation models of the slip factor present in the literature, realized their application limits. Based on this analysis procedure, they obtained a unitary formulation of the slip coefficient, which considers both the geometry of the impeller and the flow conditions. This model was derived from the studies of Eckardt [43], who believed that the rotation speed of the vortex was not equal to the rotation speed of the impeller but depended on the blade load, that is, the difference in relative speed between the face under pressure and that in the blade depression.
- Other possible correlations are those obtained from Balje [44] and Yadav and Misra [45].
- Wiesner carried out an in-depth study on the correlations that existed in the literature in that year (1967) to verify which was the most reliable and which provided results as close as possible to those obtained experimentally. From this study, he found that Busemann's correlation [39,40] is the most reliable if applied to pumps with centrifugal impellers.

Stodola's formula, however, proved to be the best option as it provided results more consistent with the characteristic curves of the machines supplied by the manufacturers. By following Stodola's formula,

$$h_0 = 1 - \frac{\pi}{z}\sin\left(\beta_{2p}\right) \tag{3}$$

For some models, the slip phenomenon was considered also at the impeller inlet: the inlet is not axial and a value of the angle α_1 different from 90° has been obtained. Considering the slip phenomena, calculated, also in this case, using Stodola's formula (Equation (3)), the tangential component of the absolute velocity becomes $c_{u1}{}^{*}$ and the theoretical head is calculated as follows:

$$H_{th} = \frac{1}{g}\left(u_2 c_{u2}{}^{*} - u_1 c_{u1}{}^{*}\right) \tag{4}$$

This solution has led to an improvement in the results achieved, allowing the head curve obtained from the model to be brought closer to that provided by the catalog.

Hydraulic Losses

The model evaluates the hydraulic losses by referring to the individual components and are classified into friction losses (Table 1) and dynamic losses (Table 2).

Table 1. Friction losses in pump operation.

Impeller	$h_{fg} = \lambda \frac{w_\infty^2}{2g}\left(\frac{l}{d_{eq}}\right)$	(5)
Vaneless diffuser	$h_{fc} = \frac{\lambda}{2g}\frac{1}{D_{h3}}\frac{c_3^2}{sen(\alpha'_2)}\frac{d_3}{d_2}\left(\frac{d_3 - d_2}{2}\right)$	(6)
Volute	$h_{fv} = \sum_{j=1}^{18} \lambda_j \frac{c_4^2}{2g}\frac{\left(\Delta S_{cl} + \Delta S_{inn} + \Delta S_{cp}\right)_j}{A_{\theta mj}}\frac{Q_j}{Q}$	(7)
Final diffuser	$h_{fd} = \frac{\lambda}{8sen(\alpha_d)}\left[1 - \left(\frac{A_4}{A_5}\right)^2\right]\frac{c_4^2}{2g}$	(8)

In evaluating these losses, the value of the friction coefficient was obtained using the Colebrook–White formula [36], in transition and turbulent conditions. The speed w_∞ which appears in the expression of the losses in the impeller (Equation (5)) represents the average value of the relative speeds calculated between the inlet section and the outlet section of the impeller. The choice to use this expression is the result of an in-depth experimental analysis. Regarding the friction losses inside the volute, the approach recommended by Worster was followed [46]. It is hypothesized that the velocities inside the volute have a purely tangential direction and that a free vortex velocity distribution exists in the volute. The analysis is carried out by dividing the component into 18 sectors and evaluating the friction losses in each of them. The purpose is to analyze the volute considering the variations in both dimensions and speeds. It was considered more correct to evaluate these losses by referring to the average speed inside the sections rather than to that at the outer edge of the volute in the exit section. This last approach would have led to a trend of decreasing friction losses as the flow rate increased. Table 2 shows the dynamic losses obtained for the related machine components.

Table 2. Dynamic losses in pump operation.

Inlet		$h_{inlet} = 0.25\left(\frac{Q}{\frac{\pi d_0^2}{4}}\right)^2 \frac{1}{2g}$	(9)
Impeller	Shock losses	$h_{shock} = \frac{[w_1 sen(i)]^2}{2g}$	(10)
	Wake losses	$h_{dg} = (\xi_2 - 1)^2 \frac{c_{m2}^2}{2g}$	(11)
Vaneless diffuser	Instantaneous expansion losses	$h_{dc} = \frac{c_{m2}^2}{2g}\left(1 - \frac{A_{2r}}{A_3}\right)^2$	(12)
Volute	Mixing losses	$h_{dv} = \frac{c_{m3}^2}{2g}$	(13)
Final diffuser	Diffusion losses [1]	$h_{dd} = \xi_d \frac{c_4^2}{2g}$	(14)

[1] ξ_d represents the localized resistance coefficient, and its value was obtained as a function of the ratio c, reported in Table 3, and taken from [47].

Table 3. Localized resistance coefficient as a function of the parameter c.

$0.025 \leq c \leq 0.075$	$\xi_d = 0.14$
$0.075 < c \leq 0.15$	$\xi_d = 0.20$
$0.15 < c \leq 0.25$	$\xi_d = 0.47$
$0.25 < c \leq 0.35$	$\xi_d = 0.76$
$0.35 < c \leq 0.45$	$\xi_d = 0.95$
$0.45 < c \leq 0.75$	$\xi_d = 1.05$
$0.75 < c \leq 0.90$	$\xi_d = 1.10$

The shape of the final diffuser was assumed as that of a diverging duct with a gradual widening of the section. The value of parameter c is obtained from the following equation, as a function of the input (b) and output (b_5) sections of the component and its length (L_d):

$$c = \frac{b_5 - b}{2\,L_d} \tag{15}$$

In addition to the losses in Table 2, another loss is detected, due to the vortex, which arises above all at low flow rates. The model, being one-dimensional, is unable to consider bidimensional phenomena such as vorticity. This observation was confirmed by the experimental investigations carried out by Van der Braembussche [48]. For overcoming this critical issue, some expressions based on experimental observations were proposed.

For $Q < Q_{bep}$,

$$h_{diff} = \frac{c_{u3}^2 - c_{u3bep}^2}{2g} \tag{16}$$

For $Q > Q_{bep}$,

$$h_{diff} = 0 \tag{17}$$

Once the theoretical head and hydraulic losses have been evaluated, it is possible to obtain the real head by calculating the difference between the theoretical head and the hydraulic losses.

$$H_m = H_{th} - \sum losses \tag{18}$$

2.1.2. Turbine Operation

The path of the fluid is inverted with respect to direct operation: the entry coincides with the final diffuser of the machine and is identified by Section 4, while the discharge is identified by Section 1. The speeds are then calculated starting from the inlet section of the machine.

Velocity Triangles

To evaluate the losses, it was first necessary to calculate the velocities in the various sections of the machine:

- Inlet: For the calculation of the velocities, reference is made to their average value.
- Volute: For this component, a free vortex distribution is hypothesized, and this assumption is confirmed in the experimental analyses carried out by some researchers.
- Impeller inlet/outlet (Figure 4): In both sections, the corrections related to the two-dimensional phenomena of slip were considered. The use of the slip factor in the turbine input has been confirmed in various studies [49,50].

Concerning the phenomena of slip in turbine operation, the following correlations have been used:

For $n_s < 10$, Stodola's formula was used (Equation (3)).

For $n_s > 10$, the Stanitz formula [39,40] was used, expressed as follows:

$$h_0 = 1 - 0.315\left(\frac{2\pi}{z}\sin^{-1}\left(\frac{c_{m2}}{u_2}\right)\right) \tag{19}$$

This criterion is part of the tuning of the model by comparing the theoretical and experimental results.

For the same reason, for $n_s < 10$, the tangential component of the absolute speed, c_{u2}, is calculated by assuming the inlet angle α_2 as Worster suggests [46,51] and is used in other models [52]:

$$\alpha_2 = \tan^{-1}\left[\frac{b}{2\pi b_2}\ \ln\left(1 + \frac{2b}{d_2}\right)\right] \tag{20}$$

Then,

$$c_{u2} = \frac{c_{m2}}{\tan(\alpha_2)} \tag{21}$$

For $n_s > 10$, c_{u2} is calculated by assuming a free vortex distribution of velocities in the area between Section 3 and Section 2, i.e., between the volute and the impeller inlet. The expression used is as follows:

$$c_{u2} = c_{u3}\,\frac{d_3}{d_2} \tag{22}$$

Hydraulic Losses

For the evaluation of hydraulic losses, also in this case, a distinction is made between friction losses and dynamic losses (Table 4). For the former, the formulas adopted are the same as those for direct operation (Table 1). The only difference lies in the evaluation of the speed w_∞ which appears in the friction losses in the impeller, since in this case the slip also occurs in the inlet section, changing the value of the tangential component of the relative speed.

Table 4. Dynamic losses in turbine operation.

Diffuser	Inlet losses	$h_{dd} = \xi_d \frac{c_4^2}{2g}$	(23)
Impeller	Inlet losses	$h_{inlet} = 0.5 \cdot \left(1 - \frac{d_2 b_2}{d_3 b_3}\right) \frac{c_{m3}^2}{2g}$	(24)
	Shock losses [1]	$h_{shock} = \frac{[w_2 sen(i)]^2}{2g}$	(25)
	Instantaneous expansion losses	$h_{dg} = (\xi_1 - 1)^2 \frac{c_{m1}^2}{2g}$	(26)
Volute	Diffusion losses	$h_{diff} = \frac{c_{u3}^2 - c_{u3bep}^2}{2g}$	(27)
Outlet [2]	$h_{inlet} = 0.25 \left(\frac{Q}{\frac{\pi d_0^2}{4}}\right)^2 \frac{1}{2g} + \frac{c_{u1}^2}{2g}$		(28)

[1] The shock losses are computed considering the incidence angle i of the fluid at the runner entry. [2] The additional term $c_{u1}{}^2/2g$ has been added to exhaust losses to consider the dissipative vortex generated by the presence of the tangential component c_{u1}.

In the expression of the diffuser losses, following the flow direction of the fluid in turbine operation, this component is a converging duct. In the losses in the diffuser, ξ_d is obtained as a function of the ratio between the final section volute width and the final section diffuser width, taken from [53] and reported in Table 5:

Table 5. Localized resistance coefficient as a function of the ratio b_5/b.

$1.25 \leq \frac{b_5}{b} \leq 1.75$	$\xi_d = 0.12$
$1.75 < \frac{b_5}{b} \leq 3$	$\xi_d = 0.30$
$3 < \frac{b_5}{b} \leq 5$	$\xi_d = 0.40$

The determination of the losses was carried out with reference to the recommendations of Idel'cick [54]. At this point, the engine head is calculated as the sum of the Eulerian work and the previously exposed hydraulic losses:

$$H_m = H_{th} + \sum losses \tag{29}$$

Even in turbine operation, the actual flow rate differs from that of the plant by an amount equal to the leakage of liquid from the clearances present between the impeller and the casing. The parameter that takes this phenomenon into account, that is, the volumetric efficiency, is determined as follows:

$$\eta_v = \frac{Q - Q_s}{Q} \tag{30}$$

2.2. Geometric Model

The knowledge of the geometry of the machine is fundamental for developing the model [36,37]. However, pump manufacturers provide only the geometrical parameters strictly necessary for the correct installation of the machine. This problem was faced with the preparation of a geometric model, capable of obtaining the missing values. It uses few geometric parameters that can be easily deduced from the catalog and, by exploiting the best design practices provided by the technical–scientific literature and based on statistical graphs derived from the measured machines, calculates the other geometrical parameters. The object of this geometric model is the determination of all those geometric parameters necessary for the evaluation of the losses.

The design criterion is inspired by what is present in [37] and requires knowledge of the following parameters:

- Flow rate and head relative to the best efficiency conditions (Q, H_m);
- Rotational speed at which the machine must work (n);
- Head at the shut-off (H_{mo});
- Absorbed power at the point of best efficiency (Pe);
- Height of the machine (h_2);
- External diameter of the impeller (d_2).

Knowing these parameters, it is possible to size the components listed below.

2.2.1. Calculation of the Shaft Diameter

This parameter was obtained by carrying out a simplification. Since the machines under examination are generally subjected to low stresses, only the torsion to which the shaft is subjected was taken into consideration [37]. Considering τ_a, the maximum allowable tension of the material of which the shaft is composed, and Pe, the maximum power that is reached at the axis of the machine, the shaft diameter was obtained as follows:

$$d_{shf} = \left(\frac{16P_e}{\omega\pi\tau_a}\right)^{\frac{1}{3}} \tag{31}$$

Since it is not always possible to know the value of the admissible voltage for each pump under analysis whose constituent material is known, the value obtained to which reference will be made in the maximum calculations was an average value equal to 7.56 Mpa [37].

2.2.2. Sizing of the Inlet Section

For this section, the parameters determined are as follows:

- The blade tip diameter, d_{1p}, is obtained through an interpolation function, of order two, which correlates the blade tip diameter with the specific speed, that is

$$\frac{d_{1p}}{d_2} = -0.00003n_s^2 + 0.0106n_s + 0.1219 \tag{32}$$

- The internal diameter of impeller inlet d_{1m} is assumed with design criteria, considering that the shaft must be housed in the impeller hub. It is determined as follows:

$$d_{1m} = kd_{shf} \tag{33}$$

The coefficient k, based on the measurements obtained on the range of pumps available (Figure 5), is set equal to 1.65.

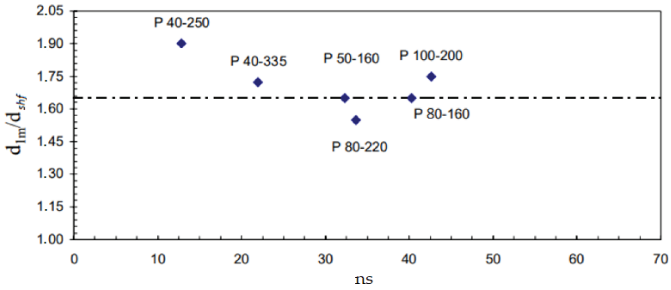


Figure 5. Statistical distribution of the ratio d_{1m}/d_{shf}.

- The inlet angle of the relative velocity vector, β_{1p}, was obtained by imposing that, in correspondence with the design conditions, the geometric angle is equal to the real angle. This evaluation of the angle of entry is aimed at minimizing, under design conditions, the losses due to shocks. For a correct evaluation of the meridian speed, both the volumetric efficiency η_v and the real transit area must be considered. The real transit area considers the overall dimension factor of the blades which in turn depends on the angle β_{1p}. It is clear that there is a need to resort to a recursive procedure for the evaluation of β_{1p}.

$$\begin{aligned} &\beta_{1p\ trial}\\ &\downarrow\\ A_{1r} &= b_1\left(\pi\, d_1 - \frac{z\, t_1}{sen\, \beta_{1p}}\right)\\ &\downarrow\\ c_{m1} &= \eta_v\, \frac{Q}{A_{1r}}\\ &\downarrow\\ \beta_{1p} &= tan^{-1}\left(\frac{c_{m1}}{u_1}\right)_{bep} \end{aligned} \tag{34}$$

- The width of the inlet blade, b_1: Indicating with θ the inclination of the blade edge with respect to the radial direction, the length of the incoming blade was obtained as follows:

$$b_1 = \frac{d_{1p} - d_{1m}}{2}\,\frac{1}{\cos(\theta)} \tag{35}$$

An average value of the angle θ was obtained from a series of measurements carried out on the pump subjects of this analysis, and it is equal to 40°.

2.2.3. Determination of the Geometry of the Seals

For the determination of the spokes of the seals, the hypothesis of linear dependence between the diameters of the seals and the blade tip diameter was assumed. This hypothesis is satisfactorily verified for the front seal while exhibiting less feedback when applied to the rear seal. Given the modest influence of these parameters on the evaluation of volumetric returns, the approximation was considered acceptable. The relations used to evaluate the diameters of the seals as a function of the inlet blade tip diameter are as follows:

$$d_f = 1.3031\ d_{1p} \tag{36}$$

$$d_b = 1.3138\ d_{1p} \tag{37}$$

2.2.4. Determination of the Number of Blades and the Angle β_{2p}

The number of blades and the angle β_{2p} were evaluated by following the procedure described by Lobnanoff [37], using the statistical diagram shown in Figure 6. The use of the diagram presupposes the knowledge of the specific speed of the machine and of the ratio between the vacuum head H_{mo} and the head at BEP, H_m, available in any case from the catalog. Figure 6 shows the points relating to the range of pumps under examination. The values of the parameters chosen are obtained as a function of the curve that comes closest to these points. Being a statistical diagram, obtained from many tests carried out on different models of machines, in some cases it may provide accurate results, in other cases it may not. For this reason, since the number of blades is a simple value to find, it is obviously preferable to use the known value.

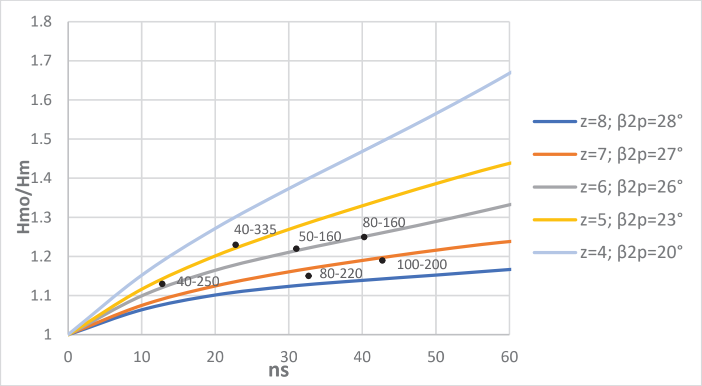


Figure 6. Statistical diagram for the determination of z and β_{2p}.

2.2.5. Calculation of the Blade width at the Outlet

The height b_2 of the unloading blades was obtained after defining the value of flow or flow coefficient, ϕ. The value of this coefficient is chosen from a statistic diagram whose use presupposes the knowledge of the specific speed and of the number of blades z. The height of the blade is obtained through the expression

$$b_2 = \frac{Q}{\varnothing\left(\pi d_2 - \frac{z t_2}{\sin\left(\beta_{2p}\right)}\right)} \tag{38}$$

The thickness of the blades, t_2, was assumed to be 4 mm.

2.2.6. Sizing of the Volute

The diameter of the water cutter (d_3) is obtained considering that it must be slightly higher (about 5%) than the external diameter of the impeller. This is necessary to avoid interference due to machine vibrations. The following dimensioning criterion [38] is adopted:

$$d_3 = 1.05\, d_2 \quad n_s \in [10\ldots 20] \tag{39}$$

$$d_3 = 1.06\, d_2 \quad n_s \in [20\ldots 30] \tag{40}$$

$$d_3 = 1.07\, d_2 \quad n_s \in [30\ldots 50] \tag{41}$$

$$d_3 = 1.09 \quad d_2 n_s \in [50\ldots 75] \tag{42}$$

The geometry relating to the exit section of the volute (A_4) is obtained through the value of the average speed in the volute. It is expressed as a function of a parameter K_v, obtained from a statistical diagram as the specific speed varies.

The procedure used is as follows:

$$K_v = \frac{v_{volute}}{\sqrt{2gH_m}} \tag{43}$$

$$A_4 = \frac{Q}{K_v\sqrt{2gH_m}} \tag{44}$$

$$b = h_v = \sqrt{A_4} \tag{45}$$

A square geometry was assumed for the exit section of the volute (A_4).

2.2.7. Sizing of the Final Diffuser

It has been assumed that the length of the final diffuser, L_d, can be assumed to be equal to the difference between the height h_2 (distance between the axis of the pump and the discharge flange) and the radius r_2. It was therefore possible to dimension the output section of the final diffuser using the following expressions:

$$b_5 = h_5 = 2\ L_d\ \tan(\alpha_d) + b \tag{46}$$

$$A_5 = b_5 h_v \tag{47}$$

The half-opening angle of the diffuser α_d is set equal to 3.5 degrees in such a way as to ensure that fluid vein detachment does not occur.

2.2.8. Impeller–Case Distance s_d

The impeller–case distance was evaluated following an analysis carried out on the geometric characteristics of the pumps under examination. An average value of 9 mm was established. Although there are undeniable differences between this value and the real one, the influence of the parameter on the evaluation of quantities of interest, and in particular the friction efficiency on the disc, is reduced to such an extent that the approximation was deemed acceptable.

2.3. Measurement of Geometric Parameters

To verify the accuracy of the geometric model, it was necessary to measure, for each pump subject of this analysis, the following geometric parameters:

- External diameter (d_1);
- Eye diameter of the impeller (d_2);
- External blade width (b_1);
- Blade width at the eye of the impeller (b_2);
- Outflow angle relative to the external diameter (β_1);
- Outflow angle relative to the eye of the impeller (β_2).

The measurement of some of the parameters listed above such as the diameters and the height of the blades did not present difficulties. To perform such measurements, a vernier caliper was used (0–1000 mm range, 0.15 mm accuracy). The evaluation of the angles β_1 and β_2 was carried out adopting the following methodology:

- To calculate the exit angle β_2 (angle relative to the high-pressure area), it was verified that the profile of the impeller blades in the radial plane was approximated by a logarithmic spiral, using a probe mounted in the spindle of the milling machine to reconstruct its shape. The equation of the logarithmic spiral in polar coordinates r, θ (where r is the generic radius of the profile in a radial plane and θ is the angle that this radius forms with the axis of the machine always in the radial plane) is as follows:

$$r = a\, e^{m\theta} \tag{48}$$

Having a series of pairs of values of r, θ it is possible to evaluate by interpolation the unknown values a and m of Equation (47). To obtain the values r and θ, the various measured impellers were mounted on a divider disc of the type used for milling machines. This disc allowed the rotation of the impeller installed on it exactly by the desired angle, with great precision.

- To determine the angle β_1, it is required that in correspondence with the design conditions (BEP point) there is a correspondence between the geometric angle and the flow angle.

3. Results

The analysis was carried out on five Ksb pumps (P40-335, P80-220, P40-250, P50-160, P100-200) and a Caprari pump (P80-160), which cover a specific speed range from 9.05 to 43.48. These pumps were measured at the DIMEG Department of the University of Calabria, except for the P100-200, which was instead measured at the University of Trento. They are centrifugal pumps with a conical axial inlet and perpendicular discharge, a volute with a square end section, and a height that varies linearly along the peripheral direction. The final diffuser consists of a diverging truncated cone. The following paragraphs show the results of both the geometric and fluid dynamic models.

3.1. Results of the Fluid Dynamics Model

The following Figures 7–11 and 12A,B show the characteristic curves obtained from the fluid dynamic model for the range of pumps under examination, both in turbine and pump operation. In this case, the geometric parameters of the pumps are known, as they have been measured, and therefore the model processes these values according to the machine prototype described above. These curves are compared with those measured experimentally to highlight the deviation and an error band of 5%.

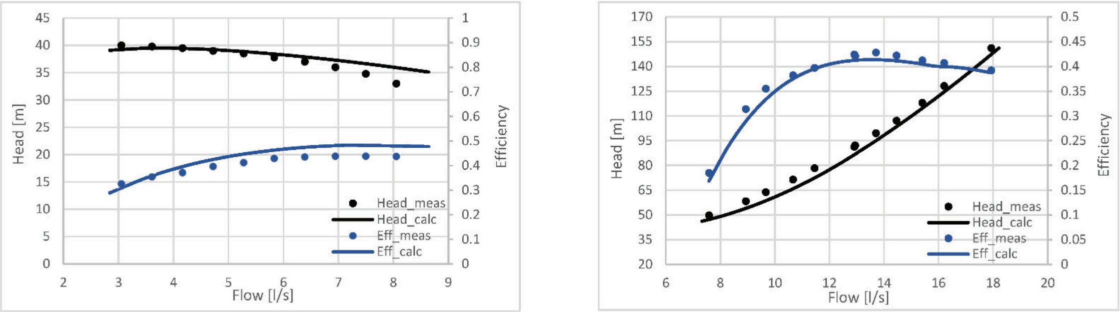


Figure 7. (**left**) Head and efficiency for the PAT 40-335 (ns = 9.05) in pump operation. (**right**) Head and efficiency for the PAT 40-335 (nst = 5.52) in turbine operation.

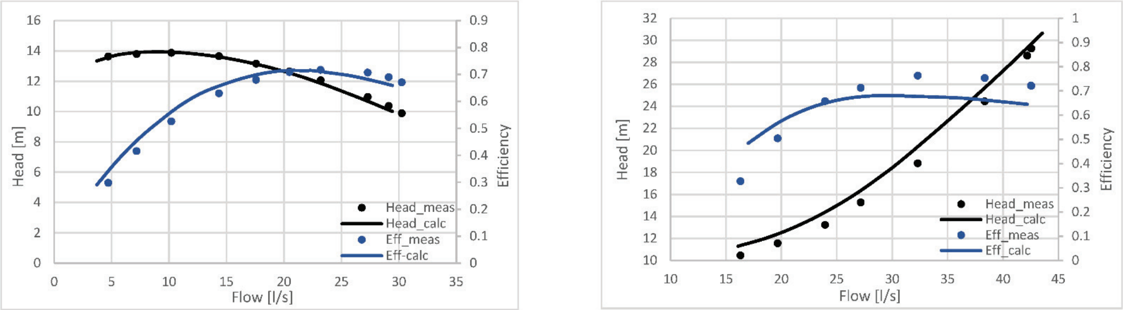


Figure 8. (**left**) Head and efficiency for the PAT 80-220 (ns = 32.69) in pump operation. (**right**) Head and efficiency for the PAT 80-220 (nst = 26.91) in turbine operation.

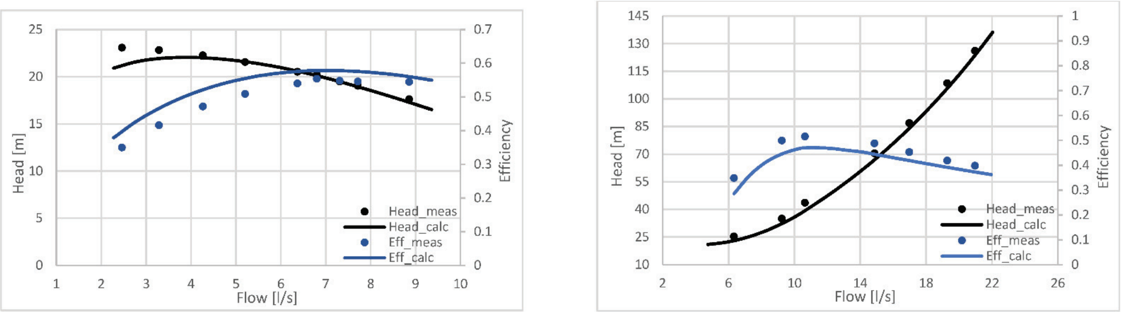


Figure 9. (**left**) Head and efficiency for the PAT 40-250 (ns = 12.78) in pump operation. (**right**) Head and efficiency for the PAT 40-250 (nst = 8.81) in turbine operation.

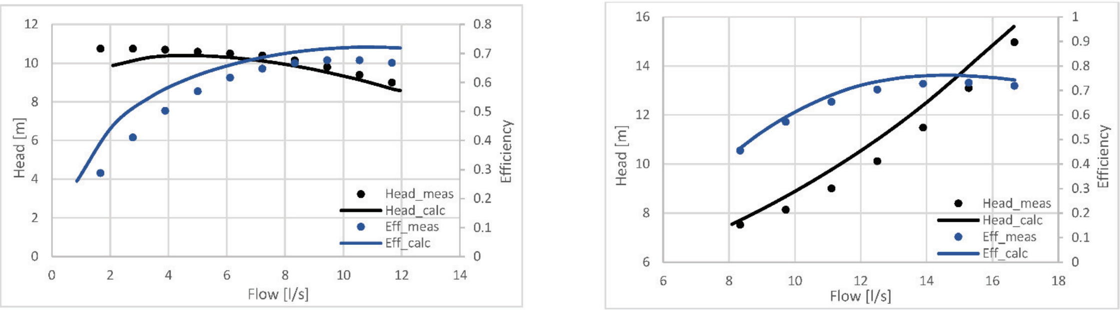


Figure 10. (**left**) Head and efficiency for the PAT 50-160 (ns = 31.01) in pump operation. (**right**) Head and efficiency for the PAT 50-160 (nst = 26.03) in turbine operation.

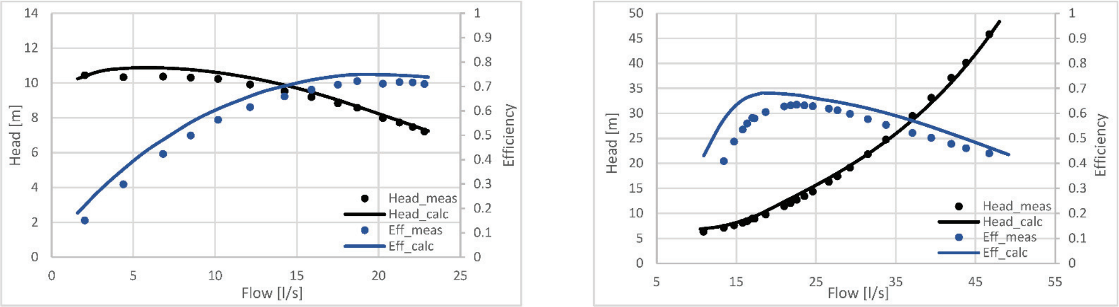


Figure 11. (**left**) Head and efficiency for the PAT 80-160 (ns = 40.24) in pump operation. (**right**) Head and efficiency for the PAT 80-160 (nst = 32.94) in turbine operation.

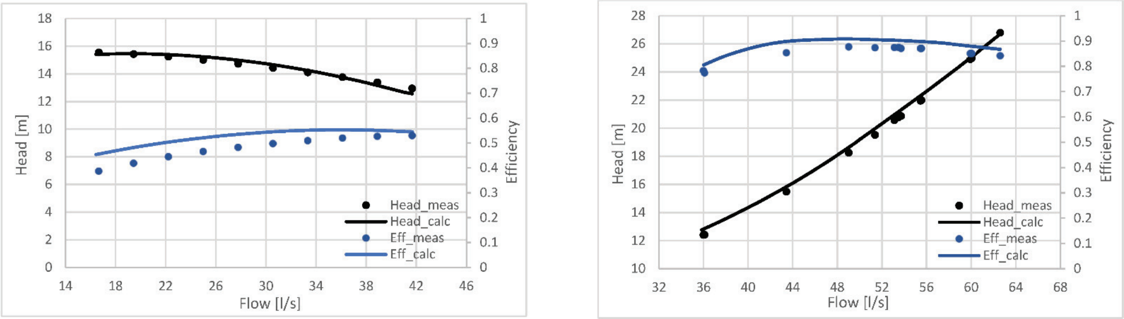


Figure 12. (**left**) Head and efficiency for the PAT 100-200 (nst = 43.48) in pump operation. (**right**) Head and efficiency for the PAT 100-200 (nst = 35.91) in turbine operation.

As can be seen in the Figures 7–11 and 12A,B, in turbine operation, the PATs are able to provide satisfactory performances under nominal operating conditions. To the right of the BEP point, the efficiency drops slowly, and this represents an advantage of the PATs, because it allows working with good efficiency in a wide range of flow rates. Nevertheless, there are no very high efficiencies, if compared to traditional machines, such as small Francis and Pelton turbines. The main cause is linked to the fact that the PATs are not designed to work as a turbine, and for this reason they are not optimized. In addition, there are instability phenomena that occur inside the machine when it operates outside the design conditions. This instability can occur as follows:

- Oscillations of the rotation speed;
- Instability in the torque applied to the motor shaft;
- Instability in the head from the turbine or in the flow rate processed;
- Cavitation due to the presence of low suction pressures;
- Water hammer, which stresses both the piping system and the mechanical parts of the PAT.

There are therefore strong fluctuations in the torque that is transmitted to the machine shaft which, in the transition from pump to turbine operation, requires long synchronization times of the machine with the generator. The PAT is therefore not able to adapt to the variations of the energy required by the network. These fluctuations can also occur during turbine operation, during synchronization with the generator at the frequency of the electrical network in the starting or braking phase, and for low flow rates or when the load applied to the shaft is zero, as the hydraulic energy is entirely dissipated by the friction in the bearings and the impeller does not accelerate. Regarding pump operation, centrifugal pumps are operating machines and, to obtain high efficiencies, they are designed to minimize losses. The best pump efficiencies are obtained for machines with high specific speeds. The shape of the impeller is less critical and allows a more regular fluid passage; there are no abrupt changes in shape or strong bends. However, high specific speed pumps process high flow rates but low heads. For low specific speed values, these machines are designed exclusively to provide high heads at the flow rate, at the expense of efficiency. To obtain high heads, the impeller assumes a 'pan' conformation; it is very flattened, and the fluid threads have strong curves, producing high pressure drops and therefore low performances. For example, the 40-335 pump has low efficiency because it only produces head. It must therefore centrifuge, and it is necessary to increase its diameter, flattening its shape.

3.2. Results of the Geometric Model

This paragraph shows the comparison between the geometric values determined by the model previously described and the actual geometry of the machines measured in the laboratory (Table 6). Furthermore, the difference between the two corresponding values is highlighted to be able to observe the reliability of the procedure used. Since this is a study based on statistical graphs, it is foreseeable that this difference will be minimal for some values, while it will be substantial for others, depending on the actual geometry of the machine, which adapts to the model itself. It is the designer's task to establish, and then evaluate, whether to accept this obtained gap, obviously depending on the sensitivity that the parameter itself has towards the result. This procedure was the basis for the verification of the model and its refinement. The curves that have been obtained from the entire module, which includes the fluid dynamic and geometric model obtained for the machine models under examination, have reported tolerable results for the purpose that had been set.

Table 6. Geometric parameters measured and calculated by the model.

	Pumps											
	40-335		**40-250**		**80-220**		**50-160**		**80-160**		**100-200**	
	Meas	**Calc**	**Meas**	**Calc**	**Meas**	**Calc**	**Meas**	**Calc**	**Meas**	**Calc**	**Meas**	**Calc**
d_0 (mm)	72.5	82.4	65	65.6	115	95.6	77.1	73.4	120	87.5	143	114
$d_{1\mathrm{m}}$ (mm)	49.9	40.2	37.74	35.9	50	45.4	34	29.6	42	35	50	51
d_1 (mm)	60.9	61.3	51.37	50.8	57	70.5	50	51.5	81	61.2	62	82.4
d_f (mm)	84.75	107.3	84.75	85.5	134.75	124.5	89.75	95.6	135	114	159.75	148.4
d_3 (mm)	338	251.2	282	273	230	234.3	185	186.2	185	187.3	224	234.3
d_b (mm)	139.7	108.2	99.76	86.2	150	125.5	90	96.4	135	114.9	150	149.6
b_1 (mm)	7.2	27.5	8.89	11.5	50	32.7	32	28.6	62	34.3	61	41.1
b_2 (mm)	10	16.1	8	7.3	25	21.2	16	16.5	25	21.1	32.5	28.7
b_3 (mm)	16	28.2	8	14.7	42.5	37.2	26	28.8	35	37	61	50.3
b (mm)	26	24	24	27	49	66	67	42	87	47	55	93.6
β_{1p} (°)	20.49	43.83	38.58	51.9	17.53	42.75	26.36	41.38	23.7	43	20	38.97
β_{2p} (°)	24	23	20	23	28.26	27	26	26	23.7	26	26	27
t_1 (mm)	4	4	4	4	2	4	4	4	4	4	4	4
t_2 (mm)	4	4	4	4	2	4	4	4	5	4	4	4
b_5 (mm)	40	64.2	89	52.6	80	89	50	89.9	84	63	89	114.5
cl (mm)	0.125	0.125	0.25	0.125	0.15	0.125	0.25	0.125	0.25	0.125	0.55	0.25
sd (mm)	7	9	17	9	9	9	21	9	9	9	15	9
L_d (mm)	97	131.5	98	95	90	170	102	93	137	136.5	91	170

As previously specified, we notice significant differences for some values. These differences, however, are relative and calculated as a function of the specific measured value. This analysis was carried out on a sufficiently large number of models, in such a way as to allow their improvement. Obviously, this research, being still ongoing, has the potential to provide even better results.

4. Sensitivity Analysis

The proposed model provides more accurate results when using the measured geometry of the machine. The influence of the single geometric parameters on the model results was analyzed to carry out a sensitivity analysis. The geometric parameters to which the model is most sensitive have been identified and are as follows:

- Hub diameter d_{1m};
- Width of the blades entering the impeller b_1;
- Suction diameter d_0;
- Width and height of the volute h_v, b;
- Impeller exit angle β_2;
- Height of the blades exiting the impeller b_2.

A sensitivity analysis was carried out for each of these parameters in correspondence to percentage variations $\pm\Delta/2\%$ and $\pm\Delta\%$. For both pump and turbine operation, the following parameters are determined:

- $Q_{P/T}$ meas: flow rate to BEP measured on the bench;
- $Q_{P/T}$ calc: flow rate to the BEP calculated by the model;
- $H_{P/T}$ meas: head at BEP measured on the bench;
- $H_{P/T}$ calc: head at BEP calculated by the model.

The percentage error is then evaluated as follows:

$$\frac{(Q,H)_{P/T}meas - (Q,H)_{P/T}calc}{(Q,H)_{P/T}meas} \tag{49}$$

Table 7 summarizes the influence of the geometric parameters considered on the values of flow rate and head at the BEP, in direct (Q_P, H_P) and inverse (Q_T, H_T) operation. The stars (*) indicate a greater or lesser degree of sensitivity in proportion to their number.

Table 7. Sensitivity to geometric parameters.

	Q_P	H_P	Q_T	H_T
d_{1m}	*	***	***	***
b_1	*	*	**	****
d_0	**	***	****	****
h_v, b	****	***	***	****
b_2	**	***	*	*
b_2	**	****	**	**

For example, (*) means that the model is not very sensitive to this parameter, for (****) the model is very sensitive to this parameter.

As can be seen from Table 7, in determining the head at BEP in turbine operation (H_T), the most critical parameters for what is necessary to have a value as accurate as possible are the diameter of the hub (d_{1m}), the height of the inlet blades (b_1), the suction diameter (d_0), and the dimensions of the volute (h_v, b). Regarding, instead, the determination of the flow rate at the BEP for the turbine operation (Q_T), the influence of the height of the inlet blades b_1 is always negligible. As an example, Tables 8–10 show the results relating to parameters b and h_v, to which the model is most sensitive, for three representative pump models: P40-335, P40-250, and P80-220. During the analysis, the range of variation of each parameter is between ±20%, except for three parameters which showed a higher sensitivity (diameter of hub and suction band diameter of ±10%, width of the volute band of ±5%).

Table 8. Sensitivity analysis for pump 40-335.

h_v, b (m)	Δ% Variation	h_v, b (m)	Q_p meas	Q_p calc	E%	H_p meas	H_p calc	E%	Q_T meas	Q_T calc	E%	H_T meas	H_T calc	E%
0.024	−5.0%	0.023	26	25.1	3.6%	35	29.90	14.7%	49.3	45.9	7.0%	99.5	109.0	−9.5%
0.024	−2.5%	0.023	26	26.0	0.0%	35	30.30	13.5%	49.3	47.3	4.0%	99.5	105.1	−5.6%
0.024	0.0%	0.024	26	26.0	0.0%	35	31.10	11.1%	49.3	48.8	1.1%	99.5	101.5	−2.0%
0.024	2.5%	0.025	26	27.2	−4.6%	35	31.20	10.9%	49.3	50.2	−1.8%	99.5	98.2	1.4%
0.024	5.0%	0.025	26	28.3	−9.0%	35	31.30	10.6%	49.3	51.7	−4.8%	99.5	95.0	4.5%

Table 9. Sensitivity analysis for pump 40-250.

h_v, b (m)	Δ% Variation	h_v, b (m)	Q_p meas	Q_p calc	E%	H_p meas	H_p calc	E%	Q_T meas	Q_T calc	E%	H_T meas	H_T calc	E%
0.027	−5.0%	0.026	25	25.0	0.0%	20	19.90	3.8%	38.3	37.6	1.8%	43.7	49.0	−12.2%
0.027	−2.5%	0.026	25	25.0	0.0%	20	19.70	1.7%	38.3	38.7	−1.0%	43.7	47.3	−8.4%
0.027	0.0%	0.027	25	25.0	0.0%	20	20.00	0.0%	38.3	39.8	−3.7%	43.7	45.9	−5.0%
0.027	2.5%	0.028	25	25.4	−1.6%	20	20.20	−0.8%	38.3	41.2	−7.4%	43.7	44.9	−2.9%
0.027	5.0%	0.028	25	26.4	−5.7%	20	20.10	−0.4%	38.3	42.3	−10.2%	43.7	43.6	0.1%

Table 10. Sensitivity analysis for pump 80-220.

h_v, b (m)	Δ% Variation	h_v, b (m)	Q_p meas	Q_p calc	E%	H_p meas	H_p calc	E%	Q_T meas	Q_T calc	E%	H_T meas	H_T calc	E%
0.066	−5.0%	0.063	100	100.0	0.0%	14.4	14.28	0.8%	123	117.68	4.3%	20.0	20.7	−3.3%
0.066	−2.5%	0.064	100	100.0	0.0%	14.4	14.35	0.3%	123	120.66	1.9%	20.0	20.2	−1.1%
0.066	0.0%	0.066	100	100.0	0.0%	14.4	14.41	−0.1%	123	122.64	0.3%	20.0	19.6	1.9%
0.066	2.5%	0.068	100	100.0	0.0%	14.4	14.46	−0.4%	123	124.63	−1.3%	20.0	19.1	4.6%
0.066	5.0%	0.069	100	100.0	0.0%	14.4	14.51	−0.8%	123	126.61	−2.9%	20.0	18.6	7.2%

5. Procedure for Predicting the Performance of a Generic Pump

This paragraph describes the global procedure for calculating the performance of a generic PAT, which uses both the pump/turbine fluid dynamic model and the geometric model. In the first phase, the geometry of the machine is built, as described in Section 3, starting from the data Q, H, H_{mo}, h_2, d_2, and n acquired from the catalog. The fluid dynamic model is then applied to calculate the performance of the machine in pump operation, and comparisons are made with the curves available in the catalog. At this point, two situations could occur: the first is that the two curves are very close, and the second is that the two curves do not coincide. In the second case, it is necessary to act on the geometric parameters calculated by the model, starting from those to which the model itself is most sensitive. The manual insertion of these parameters into the model which calculates the geometry (Section 3) is possible. When, after several attempts, the curves concur, the model can be applied in turbine operation. The procedure is shown in Figure 13.

To better illustrate what has just been described, an example carried out on the Caprari pump P65-250 is reported. This pump was supplied by the DIMEG Department of the University of Calabria, together with the five pumps previously described. The geometric model, first of all, calculates its geometry. Then, using these parameters, the fluid dynamic model derives the head and efficiency curve as a function of the flow rate, in pump operation, which will then be compared with the pump curve provided in the catalog (see Figure 14A).

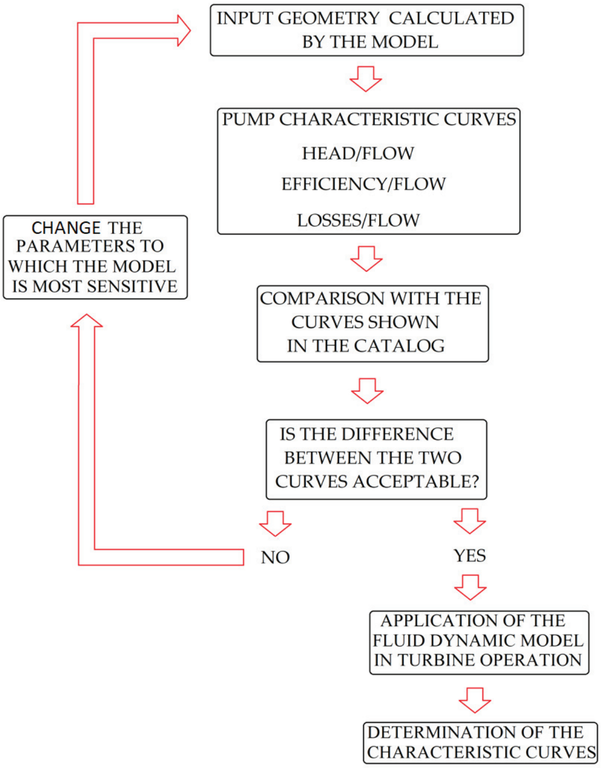

INPUT GEOMETRY CALCULATED BY THE MODEL
PUMP CHARACTERISTIC CURVES
HEAD/FLOW
EFFICIENCY/FLOW
LOSSES/FLOW
COMPARISON WITH THE CURVES SHOWN IN THE CATALOG
IS THE DIFFERENCE BETWEEN THE TWO CURVES ACCEPTABLE?
NO
YES
CHANGE THE PARAMETERS TO WHICH THE MODEL IS MOST SENSITIVE
APPLICATION OF THE FLUID DYNAMIC MODEL IN TURBINE OPERATION
DETERMINATION OF THE CHARACTERISTIC CURVES

Figure 13. Global procedure for determining the performance curves.

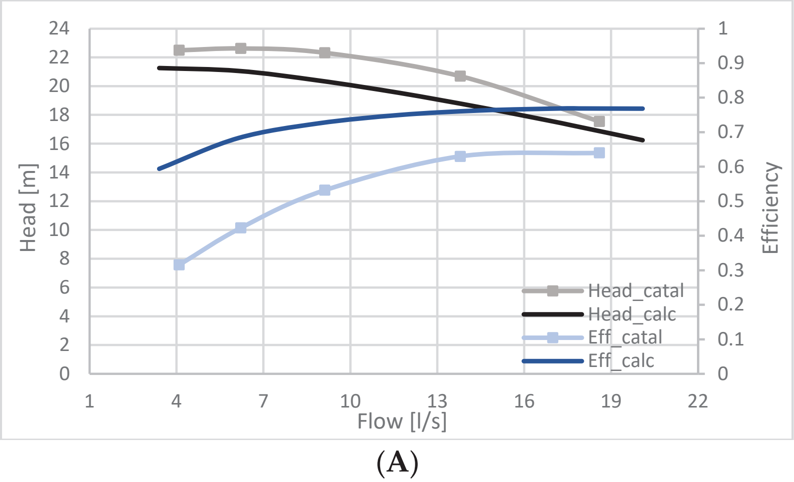
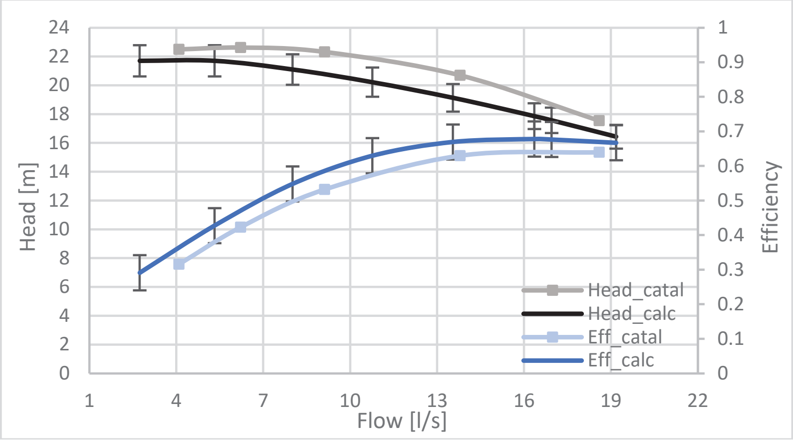


Figure 14. *Cont.*



(**B**)

Figure 14. (**A**) Head and efficiency for the PAT 65-250 (ns = 22.75) in pump operation. (**B**) Head and efficiency for the PAT 65-250 (ns = 22.75) in pump operation with z = 6, b = 0.038 m, b5 = 0.054 m, and cl = 3.5 × 10^{-4} m.

As can be seen from Figure 14A, the real head obtained from the model was too low if compared with the head reported in the catalog, as it is positioned below this curve. The geometric model provided a value of the number of blades equal to 5. However, having the real value of this parameter (z = 6) available, it was possible to replace it. The change of this parameter has modified the losses linked to the slip phenomenon, calculated using the Stodola formula [37,38], which depends on the value of the number of blades. To improve the characteristic curve, it was decided to act on the value of the width of the exit section of the volute (b), reducing it, to then obtain, using the geometric model, the corresponding value of the width of the final diffuser (b_5). By decreasing the value of this parameter, the fluid passage section is reduced (A_4), causing an increase in speed inside the volute (c_4). The b-value initially obtained by the model was 0.048 m, and b_5 was equal to 0.064 m. Finally, as can be seen from the total efficiency curve calculated by the model, the efficiency is too high for low flow rates. This is linked to the assumptions imposed a priori and to having initially considered the volumetric efficiency, for any flow rate value, equal to its value at the BEP, set equal to 0.95. However, it was possible to improve the results by acting on the amplitude value of the seal, *cl*, through which the liquid leaks occur, which is therefore linked to the volumetric losses. By increasing the value of this parameter, the total efficiency curve is considerably lowered. After applying the changes illustrated above, the curves changed as shown in Figure 14B.

At this point, having obtained the conformity of the results for pump operation, it was possible to observe the behavior of the machine in turbine operation. Figure 15A,B shows the curves obtained from the model before and after the modifications to the geometric parameters analyzed. These curves, obtained from the fluid dynamic model relating to turbine operation, are compared, for simplification purposes, with the data obtained experimentally on the PAT. However, this method works even in the absence of experimental data.

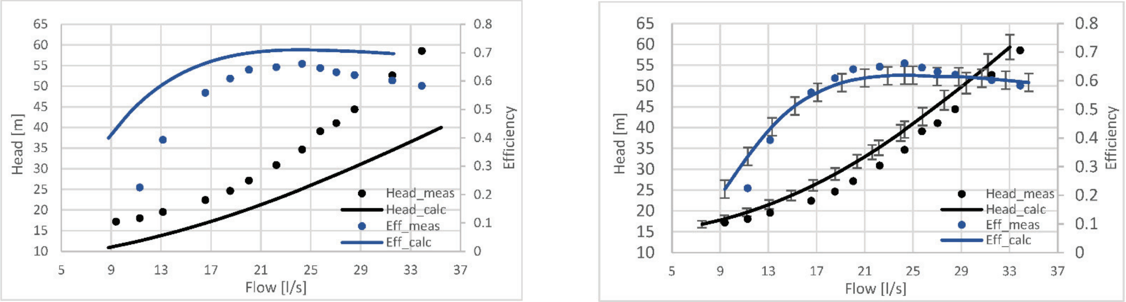


Figure 15. (**left**) Head and efficiency for the PAT 65-250 (nst = 15.83) in turbine operation. (**right**) Head and efficiency for the PAT 65-250 with z = 6 in b = 0.038, b5 = 0.054 m, and cl = 3.5 × 10^{-4} m.

6. Conclusions

The work carried out is part of a theoretical–experimental research context on centrifugal pumps used as turbines (PATs). Their convenience lies mainly in the lower costs incurred compared to a turbine with the same power and in the wide range of models available on the market. However, this advantage cannot be grasped if it is not possible to know the actual behavior of the machine when it is used as a turbine once a specific need has been recognized. The objective of this research is the development of a prediction model capable of obtaining the head–flow rate and efficiency–flow rate curves of the PAT, both in pump and turbine operation. The effort was the development of a series of fluid dynamic models that involve the pressure drops in the various components of the machine, as well as the slip phenomena at the inlet and outlet of the impeller. Furthermore, a geometric model was created for the reconstruction of the geometry of the machine, based on good design techniques, statistical data, and maps of good functioning, available in the literature. This analysis was necessary as the geometry is not provided by the manufacturer's catalog. These models were calibrated based on measurements made on the DIMEG hydraulic test bench on a sample of six machines, tested in both operating modes, whose geometric parameters were measured. The machines measured were six centrifugal pumps, namely five Ksb pumps (P40-335, P80-220, P40-250, P50-160, P100-200) and a Caprari pump (P80-160), which have a specific speed range from 9.05 to 43.48. For these pumps, the head flow rate and flow rate efficiency curves have been obtained in both operating modes. Generally, these curves, if compared with those obtained experimentally in the DIMEG test bench, show a good reliability, because they fall into error bands equal to +/−5%. To the right of the BEP, the efficiency curves are flat, which represents an advantage for those who use this technology, as the machine maintains good performance over a wide range of flow rates. Subsequently, a procedure was set up which envisages the use of the previously mentioned models, useful for calculating the performance curves of the machine both in pump and turbine operation, as well as for the reconstruction of the geometry. In the first phase, based on a few data available from the manufacturer's catalog, the model reconstructs the prototype geometry of the machine and calculates the performance curves of the machine in pump operation. If these curves match those present in the catalog, the geometry calculated by the geometric model is correct; otherwise, it is necessary to change some geometric parameters so that the predicted curves and those in the catalog coincide. For this purpose, a sensitivity analysis comes to the aid of the user, the purpose of which is to identify the parameters to which the model is most sensitive. The sensitivity analysis showed that the geometric parameters to which the model is most sensitive, in turbine operation, are the diameter of the hub (d_{1m}), the height of the inlet blades (b_1), the suction diameter (d_0), and the dimensions of the volute (h_v, b). Finally, once the appropriate parameters have been changed and the compliance of the performance curves with the catalog data has been obtained, it is possible to obtain the curves in turbine operation, which will certainly be reliable, given the adherence of the curves in pump operation.

In conclusion, the article presented a flexible and interactive forecasting tool, with 95% reliability, which allows choosing the most suitable PAT model to exploit the available water resources. A simple model of general application was presented, useful for those who decide to rely on PAT technology.

Author Contributions: Conceptualization, S.B.; Data curation, S.B. and V.P.; Formal analysis, V.P. and M.A.; Investigation, S.B. and V.P.; Methodology, S.B.; Project administration, M.A.; Software, V.P.; Validation, V.P.; Writing–original draft, V.P.; Writing–review & editing, S.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Nomenclature

Symbols

A	generic area
$A_{b,f}$	back/front leakage passage area
$A_{\theta mj}$	inlet area of the jth volute sector
$A_{1,\,2}$	passage area at different points of the impeller
$A_{1r,\,2r}$	real passage area at different points of impeller
A_3	diffusion region passage area
A_4	volute final section area
A_5	final diffuser inlet passage area
$b_{1,\,2}$	width at different points of impeller
b_3	vaneless diffuser width
b_4	final section volute width
b_5	final section diffuser width
$c_{1,\,2,\,3,\,4}$	absolute fluid velocities at different points of PAT
$c_{m1,\,m2,\,m3}$	meridional velocities at different points of PAT
$c_{u1,\,u2,\,u3,\,u4}$	peripheral velocities at different points of PAT
cl	radial clearance of the seal
d	generic diameter
d_o	impeller eye diameter
d_1, d_2, d_3	diameter at different points of PAT
d_{eq}	equivalent hydraulic diameter
d_{shf}	shaft diameter
d_f	diameter of the front seal
d_b	diameter of the rear seal
Eff_meas	measured efficiency
Eff_catal	catalog efficiency
Eff_calc	calculated efficiency
$h_{4,\,5}$	heights at different points of the final diffuser
h_v	volute throat section height
H	head
H_e	head at BEP of the pump
H_m	real head
Head_meas	measured real head
Head_catal	catalog real head
Head_calc	calculated real head
H_{mo}	head at the shut-off
H_{th}	theoretical head (Euler's head)
H_{BEP}	head at BEP of the PAT

K_v	volute velocity coefficient
L_d	diffuser length
n	rotational speed
ns	characteristic speed
sd	clearance between the impeller and the case
P	power
P_e	maximum pump power
Q	flow rate
Q_e	flow rate at BEP of the pump
Qs	leakage flow
Q_{BEP}	flow rate at BEP of the PAT
R_4	final section volute radius
$t_{1,\,2}$	vane thickness
$u_{1,\,2}$	peripheral velocities at different points of impeller
$w_{u1,\,u2}$	peripheral components of relative velocity
$w_{m1,\,m2}$	meridional components of relative velocity
w_∞	average relative velocity
z	number of blades
Greek letters	
α_2	absolute flow angle in the vaneless diffuser
α_d	final diffuser opening angle
β	inclination of relative flow to peripheral direction
$\beta_{1f,\,2f}$	relative flow direction
$\beta_{1p,\,2p}$	blades angles at different points of impeller
ΔS_{cl}	lateral surface area
ΔS_{inn}	increment of inner wall surface
ΔS_{cp}	increment of peripheral volute surface
z	dynamic loss coefficient
η	efficiency
η_{calc}	calculated efficiency
η_H	hydraulic efficiency
η_D	disc efficiency
η_v	volumetric efficiency
η_{tot}	total efficiency
η_{meas}	measured efficiency
θ	inclination of blade to radial direction
λ	friction coefficient
λ_j	friction coefficient of a segment of volute
μ	leakage flow coefficient
υ	kinematic viscosity
$\xi_{1,\,2}$	vanes blockage factor
ξ_d	localized drag coefficient
ρ	density of water
τ_a	torsional stress
ϕ	capacity coefficient
ω	angular velocity

References

1. Williams, A.A. *Pumps as Turbines a User's Guide*; Intermediate Technology Publications: Bradford, UK, 1995. [CrossRef]
2. Barbarelli, S.; Amelio, M.; Florio, G. Experimental activity at test rig validating correlations to select pumps running as turbines in microhydro plants. *Energy Convers. Manag.* **2017**, *149*, 781–797. [CrossRef]
3. Thoma, D.; Kittredge, C. Centrifugal pumps operated under abnormal conditions. *J. Power Sources* **1931**, *73*, 881–884.
4. Patelis, M.; Kanakoudis, V.; Gonelas, K. Pressure management and energy recovery capabilities using Pats. *Procedia Eng.* **2016**, *162*, 503–510. [CrossRef]
5. Lima, G.M.; Luvizotto, E., Jr.; Brentan, B.M. Selection of Pumps as Turbines Substituting Pressure Reducing Valves. *Procedia Eng.* **2017**, *186*, 676–683. [CrossRef]
6. Barbarelli, S.; Amelio, M.; Florio, G.; Scornaienchi, N.M. Procedure Selecting Pumps Running as Turbines in Micro Hydro Plants. *Procedia Eng.* **2017**, *126*, 549–556. [CrossRef]

7. Barbarelli, S.; Nastasi, B. Tides and Tidal Currents—Guidelines for Site and Energy Resource Assessment. *Energies* **2021**, *14*, 6123. [CrossRef]
8. Barbarelli, S.; Florio, G.; Amelio, M.; Scornaienchi, N.M. Preliminary performance assessment of a novel onshore system recovering energy from tidal currents. *Appl. Energy* **2018**, *224*, 717–730. [CrossRef]
9. Agarwal, T. Review of pump as turbine (PAT) for micro-hydropower. *Int. J. Emerg. Technol. Adv. Eng.* **2012**, *2*, 163–169.
10. Williams, A.A. Pumps as turbines for low-cost micro hydro power. *Renew. Energy* **1996**, *9*, 1227–1234. [CrossRef]
11. Barbarelli, S.; Florio, G.; Scornaienchi, N.M. Developing of a Small Power Turbine Recovering Energy from low Enthalpy Steams or Waste Gases: Design, Building and Experimental Measurements. *Therm. Sci. Eng. Prog.* **2018**, *6*, 346–354. [CrossRef]
12. Barbarelli, S.; Florio, G.; Zupone, G.L.; Scornaienchi, N.M. First techno-economic evaluation of array configuration of self-balancing tidal kinetic turbines. *Renew. Energy* **2018**, *129*, 183–200. [CrossRef]
13. Childs, S.M. Convert pumps to turbines and recover HP. *Hydrocarb. Process. Pet. Refin.* **1962**, *41*, 173–174.
14. Sharma, K.R. *Small Hydroelectric Project-Use of Centrifugal Pumps as Turbines*; Kirloskar Electric Co.: Bangalore, India, 1985.
15. Alatorre-Frenk, C.; Thomas, T.H. The pumps as Turbine's approach to small hydropower. In Proceedings of the World congress on Renewable energy, Reading, UK, 23–28 September 1990.
16. Stepanoff, A.J. *Centrifugal and Axial Flow Pumps*; John Wiley: New York, NY, USA, 1957; p. 276.
17. Hancock, J.W. Centrifugal pumps or water turbine. *Pipeline News* **1963**, *6*, 25–27.
18. Schmield, E. *Serien-Kreiselpumpen im Turbinenbetrieb*; Pumpentagung: Karlsruhe, Germany, 1988.
19. Grover, K.M. *Conversion of Pump to Turbines*; GSA Inter Corp.: Katonah, NY, USA, 1980.
20. Lewinsky-Kesslitz, H.P. Pumpen als Turbinen fur Kleinkraftwerke. *Wasserwirtschaft* **1987**, *77*, 531–537.
21. Liu, M.; Tan, L.; Cao, S. Performance Prediction and Geometry Optimization for Application of Pump as Turbine: A Review. *Front. Energy Res.* **2022**, *9*, 818118. [CrossRef]
22. Derakhshan, S.; Nourbakhsh, A. Experimental Study of Characteristic Curves of Centrifugal Pumps Working as Turbines in Different Specific Speeds. *Exp. Therm. Fluid Sci.* **2008**, *32*, 800–807. [CrossRef]
23. Nautiyal, H.; Kumar, A.; Yadav, S. Experimental Investigation of Centrifugal Pump Working as Turbine for Small Hydropower Systems. *Energy Sci. Technol.* **2011**, *1*, 79–86.
24. Singh, P.; Nestmann, F. Internal Hydraulic Analysis of Impeller Rounding in Centrifugal Pumps as Turbines. *Exp. Therm. Fluid Sci.* **2011**, *35*, 121–134. [CrossRef]
25. Tan, X.; Engeda, A. Performance of Centrifugal Pumps Running in Reverse as Turbine: Part II-Systematic Specific Speed and Specific Diameter Based Performance Prediction. *Rev. Energy* **2016**, *99*, 188–197. [CrossRef]
26. Stefanizzi, M.; Torresi, M.; Fortunato, B.; Camporeale, S.M. Experimental Investigation and Performance Prediction Modeling of a Single Stage Centrifugal Pump Operating as Turbine. *Energy Procedia* **2017**, *126*, 589–596. [CrossRef]
27. Derakhshan, S.; Kasaeian, N. Optimization, Numerical, and Experimental Study of a Propeller Pump as Turbine. *ASME J. Energy Resour. Technol.* **2011**, *136*, 012005. [CrossRef]
28. Venturini, M.; Manservigi, L.; Alvisi, S.; Simani, S. Development of a physics-based model to predict the performance of pumps as turbines. *Appl. Energy* **2018**, *231*, 343–354. [CrossRef]
29. Penagos-Vásquez, D.; Graciano-Uribe, J.; Torres, E. Characterization of a Commercial Axial Flow PAT Through a Structured Methodology Step-by-Step. *CFD Lett.* **2022**, *14*, 1–19. [CrossRef]
30. Vásquez, D.P.; Uribe, J.G.; García, S.V.; del Rio, J.S. Characteristic Curve Prediction of a Commercial Centrifugal Pump Operating as a Turbine Through Numerical Simulations. *J. Adv. Res. Fluid Mech. Therm. Sci.* **2021**, *83*, 153–169. [CrossRef]
31. Graciano-Uribe, J.; Sierra, J.; Torres-López, E. Instabilities and Influence of Geometric Parameters on the Efficiency of a Pump Operated as a Turbine for Micro Hydro Power Generation: A Review. *J. Sustain. Dev. Energy Water Environ. Syst.* **2021**, *9*, 1–23. [CrossRef]
32. Omidi, M.; Liu, S.J.; Mohtaram, S.; Lu, H.T.; Zhang, H.C. Improving Centrifugal Compressor Performance by Optimizing the Design of Impellers Using Genetic Algorithm and Computational Fluid Dynamics Methods. *Sustainability* **2019**, *11*, 5409. [CrossRef]
33. Morabito, A.; Vagnoni, E.; Di Matteo, M.; Hendrick, P. Numerical investigation on the volute cutwater for pumps running in turbine mode. *Renew. Energy* **2021**, *175*, 807–824. [CrossRef]
34. Nejadali, J. Analysis and evaluation of the performance and utilization of regenerative flow pump as turbine (PAT) in Pico-hydropower plants. *Energy Sustain. Dev.* **2021**, *64*, 103–117. [CrossRef]
35. Barbarelli, S.; Amelio, M.; Florio, G. Predictive model estimating the performances of centrifugal pumps used as turbines. *Energy* **2016**, *107*, 103–121. [CrossRef]
36. Neumann, B. *The Interaction between Geometry and Performance of a Centrifugal Pump*; Mechanical Engineering Publications: London, UK, 1991.
37. Lobnanoff, V.S. *Centrifugal Pumps–Design and Applications*, 2nd ed.; Gulf Publishing Company: Houston, TX, USA, 1992.
38. Stodola, A. *Steam and Gas Turbines*; McGraw-Hill: New York, NY, USA, 1945.
39. Japikse, D.; Marscher, W.D.; Furst, R.B. *Centrifugal Pump Design and Performance*; Concepts ETI: Wilder, VT, USA, 1997.
40. Micheli, D.; Pinamonti, P. La valutazione del fattore di scorrimento nel dimensionamento di ventilatori centrifughi a pale rovesce. In *42° Congresso Nazionale ATI*; CLEUP Editore: Padova PD, Italy, 1987; Volume 2, p. III-13.

41. De Bellis, V. Simulazione Monodimensionale Stazionaria e non Stazionaria di Turbocompressori per la Sovralimentazione di MCI. Napoli, 30 November 2011. Available online: http://www.fedoa.unina.it/id/eprint/8894 (accessed on 22 September 2021).
42. Qiu, X.; Mallikarachchi, C.; Anderson, M. A new slip factor model for axial and radial impellers. In *Turbo Expo: Power for Land, Sea, and Air*; ASME: Montreal, Canada, 2007; Volume 47950, pp. 957–966.
43. Eckardt, D. Flow Field Analysis of Radial and Backswept Centrifugal Compressor Impellers, Part 1: Flow Measurement using a Laser Velocimeter. In Proceedings of the Twenty-fifth Annual International Gas Turbine Conference and Exhibit and Twenty-second Annual Fluids Engineering Conference, New Orleans, LA, USA, 9–13 March 1980; (A80-36101 14-34). American Society of Mechanical Engineers: New York, NY, USA, 1979; pp. 77–86.
44. Balje, O. *Turbomachines*; John Wiley & Sons: New York, NY, USA, 1981.
45. Cristian, F. *Sviluppo di una Metodologia di Progettazione Integrata per il Dimensionamento di Macchine Operatrici a Flusso Centrifugo*; Università degli studi di Ferrara: Ferrara, Italy, 2012.
46. Worster, R.C. *The Effects of Skin Friction and Roughness on the Losses in Centrifugal Pump Volutes. BHRA Fluid Engineering Centre, Publication No. RR-557*; Cranfield: Bedford, UK, 1957.
47. Perdite di Carico Nelle Condotte. Available online: https://www.edutecnica.it/macchine/carico/carico.htm (accessed on 15 December 2021).
48. Van den Braembussche, R.A. *Flow in Radial Turbomachines*; Lecture Series 1996−01; Von Karman Institute for Fluid Dynamics: Rhode Saint Genèse, Belgium, 1996.
49. Rohlik, H.E. Radial Inflow Turbines. *NASA SP* **1975**, *290*, 279–306.
50. Whitfield, A.; Baines, N.C. *Design of Radial Turbomachines*; Longman Scientific & Technical, Ltd.: Harlow, UK, 1990.
51. Worster, R.C. The flow in volutes and its effect on centrifugal pump performance. *Proc. ImechE* **1957**, 177–843. [CrossRef]
52. Amelio, M.; Barbarelli, S.; Schinello, D. Review of Methods Used for Selecting Pumps as Turbines (PATs) and Predicting their Characteristic Curves. *Energies* **2020**, *13*, 6341. [CrossRef]
53. Fanizzi, L. Scienza & Inquinamento: Le Perdite di Carico nei Circuiti Idraulici. ECOACQUE. 2014. Available online: http://www.ecoacque.it/phocadownload/le%20perdite%20di%20carico%20nei%20circuiti%20idraulici.pdf (accessed on 15 December 2021).
54. Idel'cick, E. *Memento des Pertes de Charge*; Eyrolles Edition: Paris, France, 1986.

energies

Article

Neural Ordinary Differential Equations for Grey-Box Modelling of Lithium-Ion Batteries on the Basis of an Equivalent Circuit Model

Jennifer Brucker *, René Behmann, Wolfgang G. Bessler and Rainer Gasper

Institute of Sustainable Energy Systems, Offenburg University of Applied Sciences, Badstraße 24, 77652 Offenburg, Germany; rene.behmann@hs-offenburg.de (R.B.); wolfgang.bessler@hs-offenburg.de (W.G.B.); rainer.gasper@hs-offenburg.de (R.G.)

* Correspondence: jennifer.brucker@hs-offenburg.de

Abstract: Lithium-ion batteries exhibit a dynamic voltage behaviour depending nonlinearly on current and state of charge. The modelling of lithium-ion batteries is therefore complicated and model parametrisation is often time demanding. Grey-box models combine physical and data-driven modelling to benefit from their respective advantages. Neural ordinary differential equations (NODEs) offer new possibilities for grey-box modelling. Differential equations given by physical laws and NODEs can be combined in a single modelling framework. Here we demonstrate the use of NODEs for grey-box modelling of lithium-ion batteries. A simple equivalent circuit model serves as a basis and represents the physical part of the model. The voltage drop over the resistor–capacitor circuit, including its dependency on current and state of charge, is implemented as a NODE. After training, the grey-box model shows good agreement with experimental full-cycle data and pulse tests on a lithium iron phosphate cell. We test the model against two dynamic load profiles: one consisting of half cycles and one dynamic load profile representing a home-storage system. The dynamic response of the battery is well captured by the model.

Keywords: neural ordinary differential equations; grey-box model; equivalent circuit model; lithium-ion batteries

Citation: Brucker, J.; Behmann, R.; Bessler, W.G.; Gasper R. Neural Ordinary Differential Equations for Grey-Box Modelling of Lithium-Ion Batteries on the Basis of an Equivalent Circuit Model. *Energies* **2022**, *15*, 2661. https://doi.org/10.3390/en15072661

Academic Editors: Luis Hernández-Callejo, Sergio Nesmachnow and Sara Gallardo Saavedra

Received: 24 February 2022
Accepted: 25 March 2022
Published: 5 April 2022

1. Introduction

Lithium-ion batteries are a key technology for electric vehicles, portable devices and stationary applications such as home-storage systems. With the increasing usage of lithium-ion batteries in complex fields of application, the demand for battery models is growing as well. Battery models are necessary to predict the dynamic voltage and current behaviour and to monitor internal states, particularly the state of charge (SOC) and the state of health (SOH). There are many different types of battery models [1,2]. Depending on the required purpose, they can be selected as a compromise between accuracy and simplicity. We introduce here a grey-box (GB) modelling approach that uses a simple equivalent circuit model (ECM) as a basis.

Digitisation has been progressing rapidly in the past decades, and with it the amount of available data increases. This has boosted the development of artificial intelligence and especially neural networks. Neural networks are an important representative of black-box (BB) models. They learn relations between inputs and outputs of systems based on data [3–6]. However, BB models require a huge amount of training data. Therefore, it is reasonable to consider other modelling techniques. White-box (WB) modelling uses prior physical, chemical or engineering knowledge in the form of mathematical equations to describe the behaviour of the corresponding system. WB models are therefore limited to the understanding of the underlying processes. GB models combine WB and BB modelling techniques to benefit from their respective advantages [3–6].

There are many examples in current research where neural networks are used to model lithium-ion batteries. In Ref. [7] a feedforward network with two hidden layers approximates the SOC of a battery based on the actual voltage, current and time. The authors of Ref. [8] predict the SOC of a battery with a recurrent neural network (RNN). The last three values of SOC, battery current, battery voltage and the values of four temperature sensors are taken into account. RNNs enable time series prediction. The authors of Ref. [9] perform online predictions of the remaining capacity of a lithium-ion battery with a long short-term memory network, a special form of RNN. The measured voltages during constant current (CC) charging above a certain battery voltage and the charge throughput till reaching the charge cut-off voltage serve as inputs. The authors of Ref. [10] use neural networks for battery design. They generate their training data with a pseudo-two-dimensional model of a lithium-ion battery by varying different design parameters. The first neural network classifies whether the given parameter combination leads to a possible battery configuration or not. A second neural network estimates the specific energy and the specific power of the battery with the chosen parameters. In Ref. [11] a feedforward network is used for end-of-line prediction. The unmeasured physical battery parameters are estimated by a neural network. The aforementioned approaches represent BB models. The following articles focus on GB modelling of lithium-ion batteries. The authors of Ref. [12] estimate the SOH of a battery with a neural network that takes the fitted parameters of an ECM as input. In Ref. [13] a reduced-order physics-based model is supplemented with two neural networks to predict what the authors call "nonideal voltages" of the positive and negative electrode. An additional Bayesian network approximates the influence of ageing on the battery resistance and the amount of cyclable lithium. The authors of Ref. [14] build GB models of dynamic systems including external variables with neural ordinary differential equations (NODEs). In contrast to the original contribution [15], they call the combination of NODEs and differential equations “universal differential equations”. In Refs. [16,17] NODEs are used for GB modelling of lithium-ion batteries. The authors of Ref. [16] focus on physical battery modelling in combination with NODEs. They consider ageing effects such as solid electrolyte interface formation, lithium plating and active material isolation as well as the increase in the internal resistance. NODEs approximate the remaining deviation between the physical model and the experiment. In our previous work [17] an ECM serves as a basis for a GB model of a lithium-ion battery. NODEs model the voltage drop across the included resistor–capacitor (RC) circuit.

In the present contribution, we continue our previous work [17] by further improving the GB model. For this purpose, we increased the amount of physical knowledge in the model. In contrast to the former contribution, the focus of the current study is on modelling the dynamic properties of the battery. We used additional training data from charging and discharging with pulsed currents to train the time constant of battery dynamics. Furthermore, we tested the trained GB model against two test profiles covering more realistic battery operation. So far we have neither considered temperature dependencies nor ageing effects.

The target battery studied here is a large-format 180 Ah prismatic commercial lithium-ion cell with lithium iron phosphate (LFP)/graphite chemistry. This type of cell is used in stationary storage systems. We have previously investigated the experimental properties of this cell in great detail [18]. LFP cells are attractive for stationary storage applications because they have shown a high cyclic and calendaric lifetime [19,20]. However, their state diagnosis is challenging due to a flat, plateau-like discharge voltage curve and charge–discharge voltage hysteresis [21]. One of the goals of the present study is therefore to investigate the applicability of GB models to this type of cell.

The paper is organised as follows. In Section 2, we describe the fundamentals of the ECM, the NODEs and the combination of both for GB modelling of lithium-ion batteries. In Section 3, we show and discuss the application of the proposed GB model to the simulation of lithium-ion batteries. The training and test results are given as well as their dependencies on hyperparameters, the user-defined parameters of a neural network. Hyperparameters such as the learning rate or the number of hidden layers of a neural network control the learning process. At the end of the paper, we summarise the results and give an outlook.

The measurement data and the code are available in Zenodo. See 'Data Availability Statement' for further information.

2. Methodology

In this section, we introduce NODEs and explain how to use them for modelling dynamic systems. We present an ECM of a lithium-ion battery and derive the GB model from the ECM. Furthermore, we describe the initialisation, normalisation and training procedures as well as the experimental basis used for training and testing.

2.1. Background: Neural Ordinary Differential Equations

Besides the standard feedforward network, a number of other neural network architectures have been developed for different areas of application. The interested reader is referred to Ref. [22] for a detailed overview of neural networks.

RNNs are used for time series prediction. In contrast to feedforward networks, RNNs have recurrent connections. The outputs of a neuron can be used as inputs of a neuron in the same or a previous layer. In Ref. [23] RNNs learn multivariate time series with missing values. The authors of Ref. [24] include external variables in RNNs.

The authors of Ref. [25] introduce residual neural networks (ResNets) to overcome problems with the degradation of the training loss with an increasing number of hidden layers in deep neural networks. ResNets have additional short-cut connections which allow direct addition of the input of a neuron to its output.

In Ref. [26] the connection between ResNets with shared weights (the same weights are used in each layer of the neural network) and special forms of RNNs is established. ResNets can be used for time series prediction as well.

The following recursive formula applies to the state transformation from layer t to layer $t+1$ in a ResNet [25]:

$$\vec{z}_{t+1} = \vec{z}_t + \vec{f}\left(\vec{z}_t, \vec{\theta}_t\right), \quad t = 0, ..., T-1 \tag{1}$$

where, $\vec{z}_t \in \mathbb{R}^d$ is the vector of the hidden states at layer t, $\vec{\theta}_t$ the learned parameters of layer t and $\vec{f} : \mathbb{R}^d \to \mathbb{R}^d$ a learnable function. The vector $\vec{\theta}_t$ of learned parameters summarises the learned weights and biases. Parameter sharing across the layers ($\vec{\theta}_t = \vec{\theta}$ for $t = 0, ..., T-1$) results in the explicit Euler discretisation of the initial value problem [15,27–32],

$$\frac{\mathrm{d}\vec{z}(t)}{\mathrm{d}t} = \vec{f}\left(\vec{z}(t), t, \vec{\theta}\right), \quad \vec{z}(0) = \vec{z}_0. \tag{2}$$

Herein the continuous change in the states $\vec{z}(t)$ is given by the learnable function $\vec{f}$ that represents a neural network. Therefore, the differential equation according to Equation (2) is called NODE. Starting from the initial state $\vec{z}(0)$ a differential equation solver can calculate the output state $\vec{z}(T)$ [15,29,30,32].

Originally, NODEs were developed for initial-value problems. The authors of Ref. [14] expanded the approach to solving differential equations with constraints. In our previous work [17], we showed how to consider external variables $\vec{u}(t)$ (here, the dynamic battery current as input variable) directly based on a simple application example. The differential equation according to Equation (2) is generalised:

$$\frac{\mathrm{d}\vec{z}(t)}{\mathrm{d}t} = f\left(\vec{z}(t), \vec{u}(t), t, \vec{\theta}\right), \quad \vec{z}(0) = \vec{z}_0. \tag{3}$$

The external variables are inputs of the NODE. Therefore, we have to provide a function describing the change in the external variables with time. We could for example interpolate the measured data [17]. Figure 1 illustrates how to use NODEs with external variables schematically.

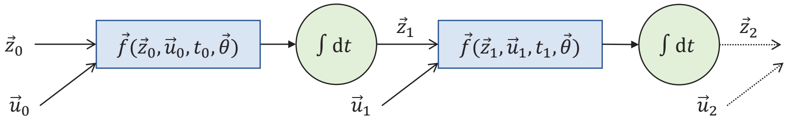


Figure 1. NODEs with external variables; $\vec{z}_t$ represents the state variables at time t and $\vec{u}_t$ represents the respective external variables. Adapted from Figure 1 in [17], which is licensed under CC BY 4.0 (https://creativecommons.org/licenses/by/4.0/, accessed on 23 February 2022).

As stated in Refs. [14,17], NODEs can be used for GB modelling. The differential equations derived from physical insights in the system and NODEs can be combined in one equation system. A WB model is used as a basis for GB modelling. Single dependencies or entire equations in the differential equation system are then replaced with learnable parameters and neural networks. The respective ODEs are transformed into NODEs. Additional assumptions going beyond the physical insights in the system can be added. A differential equation solver delivers the corresponding values of the state variables at the considered time points. Additional algebraic model equations can also be modified using learnable parameters and neural networks.

2.2. Equivalent Circuit Model

Equivalent circuit modelling is a common approach to model lithium-ion batteries. ECMs describe battery dynamics with only a few states and parameters. Due to their simplicity, they are often used to predict the SOC or the SOH of batteries [33,34]. There is no agreement in the literature about the type of equivalent circuit to be used for lithium-ion batteries [2]: Simple empirically oriented versions of ECMs model battery dynamics with a voltage source, a serial resistor and one or more RC elements [33,35–40]. Electrochemically oriented models will typically include a Warburg diffusion element (either in series with the RC element or within the RC element). A more detailed analysis, particularly in the context of the present combination with NODEs, is out of the scope of the present study.

One can take into account that the circuit parameters may depend on SOC, temperature, the battery current, and the cycle number [36,40].

As in Ref. [17], we used a simple ECM as a basis for battery modelling. The chosen ECM is shown in Figure 2. It is composed of an SOC-dependent voltage source, a serial resistor, and one RC circuit. The open-circuit voltage of phase-change active materials such as LFP is known to exhibit a path dependency [21]: The measured voltage is different after discharge with a subsequent rest phase or after charge with a subsequent rest phase at the same SOC. To describe this effect with our model, we included a hysteresis voltage drop representing the particular feature of the studied LFP cell.

The following equation system describes the chosen ECM including parameter dependencies on battery current and SOC:

$$\frac{\mathrm{dSOC}}{\mathrm{d}t} = -\frac{1}{C_\mathrm{bat}} i_\mathrm{bat} \tag{4}$$

$$\frac{\mathrm{d}v_\mathrm{RC1}}{\mathrm{d}t} = \frac{1}{C_1} \cdot \left(i_\mathrm{bat} - \frac{1}{R_1(\mathrm{SOC}, i_\mathrm{bat})} \cdot v_\mathrm{RC1} \right) \tag{5}$$

$$v_\mathrm{bat} = v_\mathrm{OC}(\mathrm{SOC}) - v_\mathrm{hys} \cdot \mathrm{sgn}(i_\mathrm{bat}) - R_\mathrm{S} \cdot i_\mathrm{bat} - v_\mathrm{RC1}, \tag{6}$$

where C_bat is the battery capacity, R_S the serial resistance, $R_1(\mathrm{SOC}, i_\mathrm{bat})$ the charge-transfer resistance in the RC circuit depending on SOC and battery current, and C_1 the double-layer capacitance (which, in our case, may include other physical contributions to voltage dynamics, for example, solid-state diffusion). It should be noted that considering a non-constant C_1 could improve the approximation capability of the model. However, we decided to use a constant double-layer capacitance at the present stage because we wanted to focus on the most important effects which we expect from the charge-transfer resistance and its dependency on the battery current and the SOC. The SOC-dependent open-circuit voltage (OCV) is labelled $v_{\mathrm{OC}}(\mathrm{SOC})$ and the hysteresis voltage drop is given by v_{hys} times the signum function of the battery current $\mathrm{sgn}(i_{\mathrm{bat}})$. The hysteresis voltage drop could have also been modelled by the current- and SOC-dependent resistance R_1. However, we did not include the voltage hysteresis into R_1 to maintain the physical characteristics of both v_{hys} and R_1. The battery voltage v_{bat} is the output of the dynamic system and the battery current i_{bat} is the external variable. We define the current positive for battery discharge and negative for battery charge. Note that Equations (4) and (5) represent 'standard', physics-derived ordinary differential equations (ODEs).

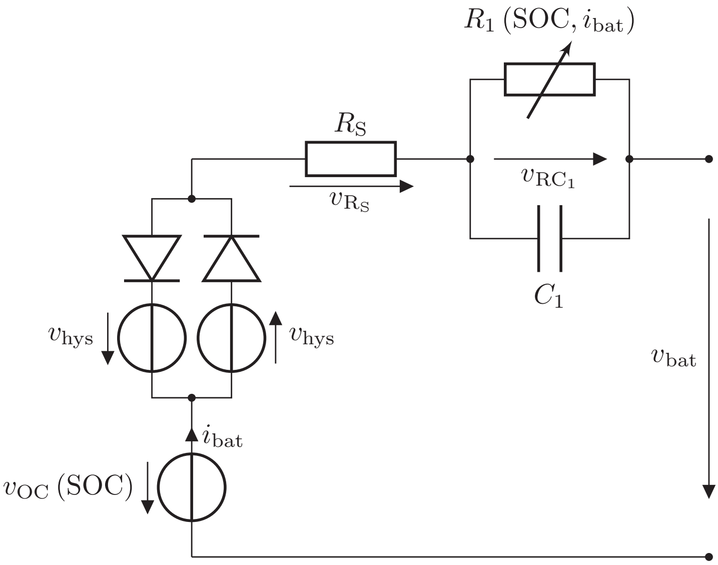


Figure 2. ECM of a battery consisting of an SOC-dependent voltage source, a hysteresis voltage drop, a series resistor, and an RC circuit.

2.3. Grey-Box Model

We took the ECM given by Equations (4) to (6) as a basis for GB modelling. The nominal capacity of a battery is usually given by the manufacturer. It indicates the capacity of a fresh cell. However, the real (experimentally observed) battery capacity C_{bat} can deviate from the manufacturer's claims. For this reason, we considered the capacity C_{bat} in Equation (4) as a learnable parameter. In Equation (5) the double-layer capacitance C_1 and the charge-transfer resistance R_1, as well as its dependency on SOC and battery current, are unknown. Therefore, we introduced a second learnable parameter to represent the capacitance C_1. As we wanted to take into account that the charge-transfer resistance may have different values and characteristics during charging and discharging (as observed experimentally [18]), R_1 is described by two learnable functions. Depending on the sign of the battery current, one of these functions is chosen; at zero current ($i_{\mathrm{bat}} = 0\,\mathrm{A}$) the mean is taken. In the output Equation (6) we had to establish a link between OCV and SOC. The manufacturer usually only provides finite-rate charge/discharge curves. Therefore, we derived $v_{\mathrm{OC}}(\mathrm{SOC})$ from dedicated measurements (so-called quasi-OCV measurements). The hysteresis voltage drop v_{hys} and the serial resistance R_{S} are assumed constant in Equation (6). We introduced two more learnable parameters to approximate these two values. Overall, using these assumptions, the ECM according to Equations (4) to (6) leads to the following GB model:

$$\frac{\mathrm{dSOC}}{\mathrm{d}t} = -\frac{1}{\omega_0} i_{\mathrm{bat}} \tag{7}$$

$$\frac{\mathrm{d}v_{\mathrm{RC1}}}{\mathrm{d}t} = \frac{1}{\omega_1} \cdot \left(i - \frac{1}{R_1(\mathrm{SOC}, i_{\mathrm{bat}})} \cdot v_{\mathrm{RC1}} \right) \tag{8}$$

$$R_1(\mathrm{SOC}, i_{\mathrm{bat}}) = \begin{cases} f\left(\mathrm{SOC}, i_{\mathrm{bat}}, \vec{\theta}_f\right) & \forall\, i_{\mathrm{bat}} < 0 \\ g\left(\mathrm{SOC}, i_{\mathrm{bat}}, \vec{\theta}_g\right) & \forall\, i_{\mathrm{bat}} > 0 \\ \frac{1}{2}\left(f\left(\mathrm{SOC}, i_{\mathrm{bat}}, \vec{\theta}_f\right) + g\left(\mathrm{SOC}, i_{\mathrm{bat}}, \vec{\theta}_g\right)\right) & \text{else} \end{cases} \tag{9}$$

$$v_{\mathrm{bat}} = v_{\mathrm{OC}}(\mathrm{SOC}) - \omega_2 \cdot \mathrm{sgn}(i_{\mathrm{bat}}) - \omega_3 \cdot i_{\mathrm{bat}} - v_{\mathrm{RC1}}. \tag{10}$$

Here, ω_0, ω_1, ω_2 and ω_3 represent learnable parameters. The functions f and g represent feedforward networks with their respective learnable parameters $\vec{\theta}_f$ and $\vec{\theta}_g$. We chose neural networks with one hidden layer and rectified linear unit (ReLU) activation for f and g. We varied the number of neurons in the hidden layer between 10 and 300. Both networks had two inputs, the SOC and the battery current, and one output, the ohmic resistance R_1.

It is worthwhile recognising that, mathematically, this model combines physics-based ODEs and machine-learning-based NODEs in one equation system. The combined equations are solved simultaneously within a single numerical framework.

2.4. Experiments

We applied the proposed GB modelling approach to a single lithium-ion battery cell. All experiments were carried out using a commercial single cell of the Chinese manufacturer CALB, model CA180FI. The large-format prismatic cell has a nominal capacity of 180 Ah and a nominal voltage of 3.2 V. It uses LFP at the positive electrode and graphite at the negative electrode. The cell was investigated experimentally under a controlled laboratory environment (climate chamber CTS 40/200 Li) using a battery cycler with four-wire measurement (Biologic VMP3). Details on the cell and characterisation methods can be found in our previous publication [18]. Here we carried out additional measurements for GB model parameterisation and testing.

We measured experimental data sets representing several different operation scenarios. Constant current constant voltage (CCCV) charge and discharge curves were measured with different C-rates of 0.1 C, 0.28 C and 1 C (corresponding to 18 A, 50 A and 180 A, respectively) during the CC phase. The upper and lower cut-off voltages were 3.65 V and 2.5 V, respectively, and a cut-off current of the CV phase of C/20 was used. Additionally, one charge and one discharge curve were acquired with included current pulses: During 50 A CC operation, every two SOC-percent the current was reduced to 25 A for 30 s. This gives rise to two dynamic voltage answers, one at beginning and one at end of pulse.

Furthermore, two independent measurements for model testing were carried out. Firstly, the cell was cycled with 50 A between 25% and 75% SOC for around 44 h after fully charging, in the following referred to as half cycles. We started from a fully-charged cell and a first discharge to 25% SOC. The SOC cycling range was controlled by Coulomb counting. After 40 half cycles it was fully charged again. Secondly, the cell was fully charged and afterwards subjected to a dynamic load profile over 48 h representing a home storage battery in a single-family house. The synthetic load profile was taken from Ref. [41] (obtained with a load profile generator [42]), where a battery system of 5 kWh was investigated, and downscaled to the energy of the present cell (576 Wh). All measurements were carried out at an ambient temperature of $T = 25\,°\mathrm{C}$.

The number of data points per measurement series was large. Therefore, beginning from the first value, we decided to only keep measurement values if the current varied by $|\Delta i_{\mathrm{bat}}| \geq 0.5$ A or the measured voltage varied by $|\Delta v_{\mathrm{bat}}| \geq 0.5$ mV between two subsequent values. Table 1 summarises the characteristics of the used measurement data. The number of used measurement values and the total duration are given for the different series. It is worth mentioning that these values vary widely. The shortest data set for training only spans $t = 3932\,\mathrm{s}$. The longest training data set takes t = 41,846 s. The test data sets cover much longer durations.

Table 1. Measurement data for training and testing the model.

Data Set	Number of Values	Time Duration/s
discharge 0.1 C	5014	38,148
charge 0.1 C	4492	41,846
discharge 0.28 C	2177	13,787
charge 0.28 C	2181	17,418
discharge 1 C	898	3932
charge 1 C	3120	3936
pulsed discharge	15,575	14,479
pulsed charge	12,660	16,300
half cycles	77,548	162,754
synthetic load profile	69,541	190,231

The measurement data were made available and used as voltage versus time and current versus time series. The measured battery current served as the external input of the model. As proposed in [17], we interpolated the measured current values linearly for providing values at arbitrary times as required by the numerical solver (cf. below).

2.5. Normalisation and Initialisation

The normalisation and initialisation are crucial for the training of the GB model with NODEs. It is recommended to scale the inputs of neural networks [43]: The average of the input variables over the training set should be close to zero (note that this condition is fulfilled for a rechargeable battery, as negative currents for charge and positive currents for discharge integrate to zero). Additionally, their covariances should be about the same.

As the SOC is in the range of 0 to 1, we decided to scale all inputs to values between -1 and 1. Additionally, we normalised the output values of the neural networks to the same value range. We did not use different learning rates for different parameters. Therefore, we also scaled the learnable parameters according to the respective value range and the expected deviation from the chosen initial value.

According to the manufacturer, the cell has a nominal capacity $C_\mathrm{N} = 180\,\mathrm{Ah}$. However, integration of the measured current over time for a whole charging or discharging process leads to an approximate charge throughput of $Q \approx 191.5\,\mathrm{Ah}$. As the manufacturers usually give lower values for the nominal capacity to be on the safe side, we decided to set the initial value to $\omega_0 = 191.5\,\mathrm{Ah}$. In the model, we used SI units. Therefore, we had to include a conversion factor.

To get more information about the ohmic resistances and the capacitance in Equations (5) and (6), or rather their learnable representation in Equations (8) to (10), we examined the measurement data from the pulse tests more closely. Figure 3 shows a detailed view of the current versus time and voltage versus time plot for the charging process with a pulsed current. At $t = 7264\,\mathrm{s}$, there is a current step of $\Delta i_\mathrm{bat} = -25\,\mathrm{A}$ during charging. The battery follows this current step with an ohmic voltage drop $\Delta v_\mathrm{bat,serial} \approx 7\,\mathrm{mV}$. The ohmic voltage drop is modelled through the serial resistance in Equation (6), or rather the learnable parameter ω_3 in Equation (10). For discharging we found similar absolute values. Therefore, $\omega_3 = {}^{|\Delta v_\mathrm{bat,serial}|}/_{|\Delta i_\mathrm{bat}|} = 0.28\,\mathrm{m\Omega}$ should be a good starting point for the learnable parameter. We introduced the normalised parameter $\omega_3^* = 1000 \cdot \omega_3$ instead and initialised it as $\omega_3^* = 0.28\,\Omega$. The value for ω_3, which is the approximation of R_S, is then calculated according to $\omega_3 = {}^{1}/_{1000} \cdot \omega_3^*$. The further course of the battery voltage following the ohmic voltage drop is modelled through the RC circuit in the ECM. We estimated the time constant τ of the RC circuit by applying a tangent to the voltage curve. We found $\tau \approx 15\,\mathrm{s}$. The final battery voltage drop caused by the RC circuit is $\Delta v_{\mathrm{bat,RC}} \approx 8\,\mathrm{mV}$. In the ECM the ohmic resistance R_1 models this voltage drop. It can be approximated as $R_1 = {}^{|\Delta v_{\mathrm{bat,RC}}|}/_{|\Delta i_{\mathrm{bat}}|} = 0.32\,\mathrm{m\Omega}$. The capacitance C_1 was estimated according to $C_1 = {}^{\tau}/_{R_1} = {}^{15\,\mathrm{s}}/_{320\,\mu\Omega} = 47\,\mathrm{kF}$. One has to take into account that the ohmic resistance R_1 in Equation (5) or (8) depends on SOC and battery current. Therefore, this is only a rough reference point. We expected it to be much higher than the estimated value for low and high values of SOC. Again, we introduced normalisation factors to simplify the later training process. The current input to the neural networks f^* and g^* was normalised in relation to the maximum absolute current. The outputs of the neural networks f and g were generated as follows: $f\left(\mathrm{SOC}, i_{\mathrm{bat}}, \vec{\theta}_f\right) = {}^1/_{100} \cdot f^*\left(\mathrm{SOC}, {}^{i_{\mathrm{bat}}}/_{180}, \vec{\theta}_{f^*}\right)$, and $g\left(\mathrm{SOC}, i_{\mathrm{bat}}, \vec{\theta}_g\right) = {}^1/_{100} \cdot g^*\left(\mathrm{SOC}, {}^{i_{\mathrm{bat}}}/_{180}, \vec{\theta}_{g^*}\right)$. We initialised the weights and biases of f^* and g^* from the uniform distribution $\mathcal{U}\left(-\sqrt{k}, \sqrt{k}\right)$, where $k = \frac{1}{l}$ with $l \in \mathbb{N}$ the number of inputs to the respective layer (cf. Ref. [43]). The learnable parameter ω_1 was represented by $\omega_1 = 10^5 \cdot \omega_1^*$, where the normalised parameter ω_1^* was initialised as $\omega_1^* = 0.5\,\mathrm{F}$. We implemented the non-linear $v_{\mathrm{OC}}(\mathrm{SOC})$ curve according to the measurements of Ref. [18] as look-up table. The $v_{\mathrm{OC}}(\mathrm{SOC})$ relationship needed in Equation (10) was obtained from the look-up table via linear interpolation. Due to inaccuracies of the current measurement and the choice of the initial SOC value it could be possible that the calculated SOC was sometimes slightly larger than 1 or slightly lower than 0. In these cases we provided the OCV values for $\mathrm{SOC} = 1$ or $\mathrm{SOC} = 0$, respectively. We approximated the hysteresis voltage drop to find a good initial value as follows. We subtracted the voltage drops over the resistances R_S and R_1 from the difference between the OCV and the measured battery voltage at a medium SOC for $i_{\mathrm{bat}} = -50\,\mathrm{A}$, yielding $v_{\mathrm{hys}} \approx 15\,\mathrm{mV}$. We introduced the respective normalised learnable parameter $\omega_2^* = 10 \cdot \omega_2$. We initialised it to $\omega_2^* = 0.15\,\mathrm{V}$.

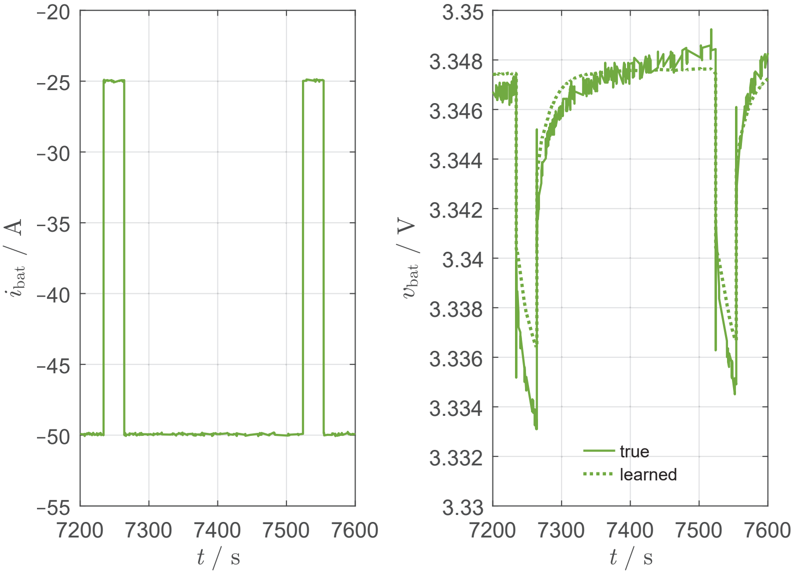


Figure 3. Simulation results using NODEs for grey-box modelling of a lithium-ion battery in comparison to experimental data at $T = 25\,^\circ\mathrm{C}$. The focus is on charging with a pulsed current at a medium SOC; (**left**): battery current versus time; (**right**): battery voltage versus time.

Applying these modifications, the following equations describe the final GB model:

$$\frac{\mathrm{dSOC}}{\mathrm{d}t} = -\frac{1}{3600\,\mathrm{s/h}\,\omega_0} i_{\mathrm{bat}} \tag{11}$$

$$\frac{\mathrm{d}v_{\mathrm{RC1}}}{\mathrm{d}t} = \frac{1}{10^5 \cdot \omega_1^*} \cdot \left(i - \frac{1}{R_1(\mathrm{SOC}, i_{\mathrm{bat}})} \cdot v_{\mathrm{RC1}} \right) \tag{12}$$

$$R_1(\mathrm{SOC}, i_{\mathrm{bat}}) = \begin{cases} \frac{1}{100} \cdot f^* \left(\mathrm{SOC}, i_{\mathrm{bat}}/180, \vec{\theta}_{f^*} \right) & \forall\, i_{\mathrm{bat}} < 0 \\ \frac{1}{100} \cdot g^* \left(\mathrm{SOC}, i_{\mathrm{bat}}/180, \vec{\theta}_{g^*} \right) & \forall\, i_{\mathrm{bat}} > 0 \\ \frac{1}{200} \left(f^* \left(\mathrm{SOC}, i_{\mathrm{bat}}/180, \vec{\theta}_{f^*} \right) + g^* \left(\mathrm{SOC}, i_{\mathrm{bat}}/180, \vec{\theta}_{g*} \right) \right) & \text{else} \end{cases} \tag{13}$$

$$v_{\mathrm{bat}} = v_{\mathrm{OC}}(\mathrm{SOC}) - \frac{1}{10} \cdot \omega_2^* \cdot \mathrm{sgn}(i_{\mathrm{bat}}) - \frac{1}{1000} \cdot \omega_3^* \cdot i_{\mathrm{bat}} - v_{\mathrm{RC1}}, \tag{14}$$

where ω_0, ω_1^*, ω_2^*, and ω_3^* are learnable parameters and the functions f^* and g^* represent neural networks. They were built in analogy to the neural networks f and g in Equation (9). We used feedforward networks with one hidden layer and ReLU activation. The number of hidden neurons was varied.

2.6. Simulation and Optimisation Methodology

We implemented our model in Python (version 3.7.6). We used the open-source machine learning framework PyTorch (version 1.9.0) [44]. PyTorch provides two main features: Tensor computing and automatic differentiation for deep neural networks. Furthermore, we used the torchdiffeq library (version 0.2.1) [45] which builds on PyTorch. It allows solving ODEs and backpropagation through the solutions of the ODEs.

The differential Equations (11) and (12) were solved with the Dopri8 method. Backpropagation was performed with the standard odeint method from torchdiffeq. Finally, an Adam optimiser minimised the loss function.

2.7. Training

The model has a large number of unknown parameters that need to be identified by mathematical optimisation: The four learnable parameters ω_0 to ω_3^*, and $4 \cdot n + 1$ parameters $\vec{\theta}_f^*$ and $\vec{\theta}_g^*$ each in the two learnable functions f^* and g^* with n the number of hidden neurons.

Due to the small amount of available training data, we split the training into two consecutive steps: First, we trained a static network with the CCCV data. Afterward, we used the pulsed data to take the battery dynamics into account. One has to keep in mind that all current flows through the charge-transfer resistance R_1 of the RC circuit at steady-state operation. The double-layer capacitance C_1 is used to capture transient phenomena.

In detail, in the first step we neglected the double-layer capacitance. Therefore, the differential Equation (12) was converted into the algebraic equation

$$v_{\mathrm{RC1}} = R_1(\mathrm{SOC}, i_{\mathrm{bat}}) \cdot i_{\mathrm{bat}}. \tag{15}$$

We trained the resulting simplified GB model using the data covering the six CCCV charging and discharging processes with different C-rates. We initialised the learnable parameters ω_0, ω_2^*, and ω_3^* and the learnable functions f^* and g^* of the simplified model as discussed above. As we have chosen a constant hysteresis voltage for non-zero battery currents, it is important to provide appropriate values for low currents. We decided to set currents with an absolute value $|i_{\mathrm{bat}}| < 0.25\,\mathrm{A}$ to zero. Additionally, we had to provide the initial SOC value. As there was a rest phase before the start of each data set, we assumed that the battery is initially at equilibrium and therefore represented by the OCV curve. We inverted the OCV(SOC)-curve to determine the respective SOC value from the initial voltage. As mentioned above, the Dopri8 method was used to solve Equation (11) with an absolute tolerance of 10^{-5} and relative tolerance of 10^{-3}. We performed backpropagation with the standard odeint method from torchdiffeq. An Adam optimiser with a decaying learning rate between 10^{-2} and 10^{-3} minimised the loss function. The loss function was defined as the sum of the root mean squared error (RMSE) between the simulated battery voltage and the measured battery voltage and an additional penalisation term. Approximated SOC values lower than 0 or higher than 1 were taken into account. Their hundredfold absolute deviation from 0 or 1 was used as the penalisation term. As we had already initialised the other learnable parameters according to the insights from the measurement data, we only optimised $\vec{\theta}_{f^*}$ and $\vec{\theta}_{g^*}$ during the first 50 training epochs. The total number of training epochs was varied. It is a hyperparameter of the training process that controls the number of complete passes through the training data set. During each training epoch, the six data sets were given to the model in random order. All time series were used completely. The optimisation steps were carried out with stochastic gradient descent. The parameters were stored when the total training loss during one epoch decreased.

In the second step, we used the complete GB model according to Equations (11) to (14) for further training. Therefore, we initialised ω_1^* as stated previously. The other parameters were taken from the pre-trained model. The initial SOC was determined as before. Additionally, we had to provide an initial value for the voltage drop v_{RC1} across the RC circuit. Due to the proceeding rest phase we assumed $v_{\mathrm{RC1}}(t = 0) = 0\,\mathrm{V}$. The standard odeint backpropagation was used again. We chose Dopri8 as differential equation solver with an absolute tolerance of 10^{-5} and relative tolerance of 10^{-3}. As before, the loss function was defined as the sum of the RMSE loss of the model output compared to the measured voltage and the penalisation term. The training loss was minimised by an Adam optimiser with a learning rate of 10^{-3}. During the first ten training epochs, we only considered the data from the charging and discharging processes with a pulsed battery current. Afterwards we also considered the data from charging and discharging with the CCCV protocol. Additionally, we froze all learnable parameters except ω_1^* during the first 20 training epochs. Overall, we carried out 30 training epochs with batch gradient descent.

To further test our approach, we investigated GB models with different numbers of neurons in f^* and g^*. Furthermore, we varied the number of training epochs in the first training step between 100 and 1000, leaving training step two unchanged. The results of this study will be discussed in Section 3. We decided to take the trained model with 100 hidden neurons in f^* and g^* and 300 training epochs in training step one as the final version.

2.8. Test

We tested the final GB model against the two remaining experimental data sets (half cycles and synthetic load profile). Again, we used the standard odeint backpropagation method from torchdiffeq. We tried to solve the differential equation system using Dopri8 with an absolute tolerance of 10^{-5} and relative tolerance of 10^{-3}. However, for the half cycles, this resulted in a step size underflow. Therefore, we changed the absolute tolerance to 10^{-3} for the half cycles.

For both test data sets, we had to provide initial values for the SOC and v_{RC1}. We initialised these values as before during training: We set $v_{\mathrm{RC1}}(t = 0) = 0\,\mathrm{V}$ and derived the initial SOC from the battery voltage.

3. Results and Discussion

The training and test results are discussed in the following sections. First, the focus is on the training results, with the goal of selecting an appropriate number of hidden neurons in f^* and g^* and of training epochs. Secondly, we compare the training results to the measurement data. Finally, simulations with the GB model are compared against the further test data sets.

3.1. Training

In total, eight experimental time series of the LFP cell were available and used for training the GB model. In particular, six time series represent charge and discharge with a CCCV protocol at different C-rates, and two time series represent charge and discharge with pulsed current.

The neural networks representing the functions f^* and g^* were used to approximate the dependency of the charge-transfer resistance R_1 on current and SOC. We performed the training with different network sizes for f^* and g^*. Additionally, we varied the number of training epochs in the first training step. Training step two was not changed. Figure 4 shows the results after completing the whole training process. Here the obtained value for R_1 is plotted as a function of SOC for charging with $i_{\mathrm{bat}} = -50\,\mathrm{A}$. The results shown in the left panel of Figure 4 were obtained from the evaluation of function f^* with different numbers of neurons in the hidden layer and 100 epochs during the first training part.

With only 10 hidden neurons, the result takes the form of a combination of two linear branches representing the charge-transfer resistance over the whole range of SOC. With an increasing number of neurons, the dependency of R_1 on SOC gets more complicated. The results vary only slightly when increasing the number of hidden neurons from 100 to up to 300, however at the cost of longer training times. Using a standard notebook and training on the CPU the training time for the first training part with 100 epochs increased from about 15.5 min to about 16.8 min when changing the number of hidden neurons from 100 to 300. Therefore, we decided to choose 100 hidden neurons for f^* and g^*.

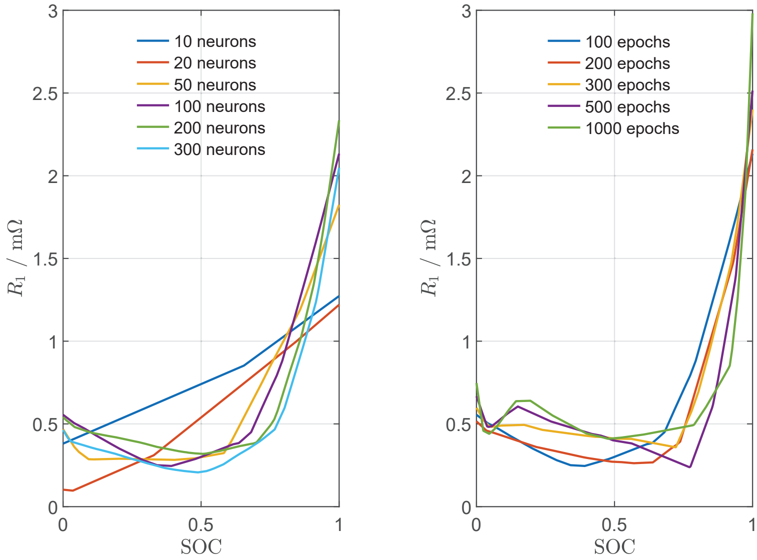


Figure 4. Simulation results: approximation results for R_1 for $i_{\mathrm{bat}} = -50\,\mathrm{A}$ derived from evaluation of function f^*; (**left**): results for a varying number of hidden neurons in f^* and 100 training epochs in the first training part; (**right**): results for 100 hidden neurons in f^* and a varying number of training epochs in the first training part.

We additionally varied the number of training epochs in the first training step. The right panel of Figure 4 illustrates the final results for R_1 at a battery current $i_{\mathrm{bat}} = -50\,\mathrm{A}$ obtained with the neural network f^* with 100 hidden neurons and a varying number of training epochs. With an increasing number of training epochs, the neural network produces more complex behaviour of R_1 as function of SOC.

After training with more than 300 training epochs, the right panel of Figure 4 shows changes in R_1 for low SOC values. We believe that this is due to overfitting. As there were few data available, we did not split off a validation data set. However, we took a closer look at the training and test losses (note that the test results will be discussed in more detail in Section 3.3). We calculated the RMSE between the measured and the approximated battery voltage for all training and test data sets. The overall training and test losses were defined as the average of the RMSE losses of the individual data sets. Figure 5 shows the results as a function of the number of training epochs. The training loss decreases with an increasing number of training epochs in the first training step. However, the test loss reaches a minimum at around 300 training epochs. These results made us choose 300 training epochs in the first training step.

As a final result from this analysis, we represented f^* and g^* with neural networks with one hidden layer with 100 hidden neurons each. We carried out 300 training epochs in the first and another 30 epochs in the second training step.

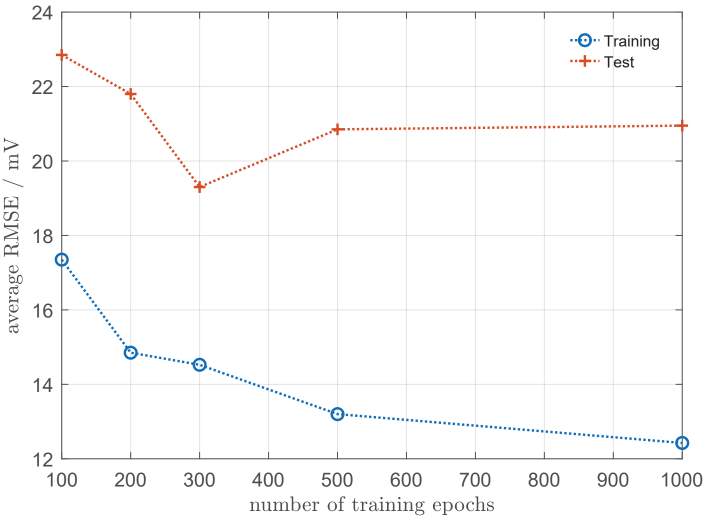


Figure 5. Average training and test losses as a function of the number of training epochs in the first training part.

Figure 6 illustrates the final training results for R_1. The left panel shows the results for charging ($i_{\text{bat}} < 0\,\mathrm{A}$) as evaluated with f^*. The right panel shows the results for discharging ($i_{\text{bat}} > 0\,\mathrm{A}$) as evaluated with g^*. The charge-transfer resistance is in the range of up to several milliohms. It decreases with an increasing absolute battery current for both charging and discharging, and reaches higher values for low and high SOC values compared to a medium SOC. The resistance shows a pronounced asymmetry between charge and discharge: During charge the highest values occur when the cell is (nearly) full. During discharge the highest values occur when the battery is (nearly) empty. This is a typical behaviour observed from lithium-ion batteries with LFP cathode [18]. However, it is difficult to interpret electrochemical details into a simple equivalent circuit. In Ref. [46] the overpotentials of a lithium-ion cell were deconvoluted. The results show that lithium-ion batteries are co-limited by reaction, diffusion, and ohmic losses. In the present paper, the battery is operated at rather low currents (up to 1 C), where diffusion limitations are expected to be not dominant. For a single charge-transfer reaction, the charge-transfer resistance decreases exponentially with increasing direct current in the Tafel region [47]. Therefore, the observed decrease in resistance with increasing current is physically realistic.

After completing the training procedure, the learnable parameters had the following values:

$$\begin{aligned} \omega_0 &= 191.5\,\text{Ah} \\ \omega_1^* &= 0.5069\,\text{F} \\ \omega_2^* &= 0.1125\,\text{V} \\ \omega_3^* &= 0.2814\,\Omega. \end{aligned}$$

This results in the following ECM parameters:

$$\begin{aligned} C_{\text{bat}} &= 191.5\,\text{Ah} \\ C_1 &= 50.69\,\text{kF} \\ v_{\text{hys}} &= 11.25\,\text{mV} \\ R_{\text{S}} &= 281.4\,\mu\Omega. \end{aligned}$$

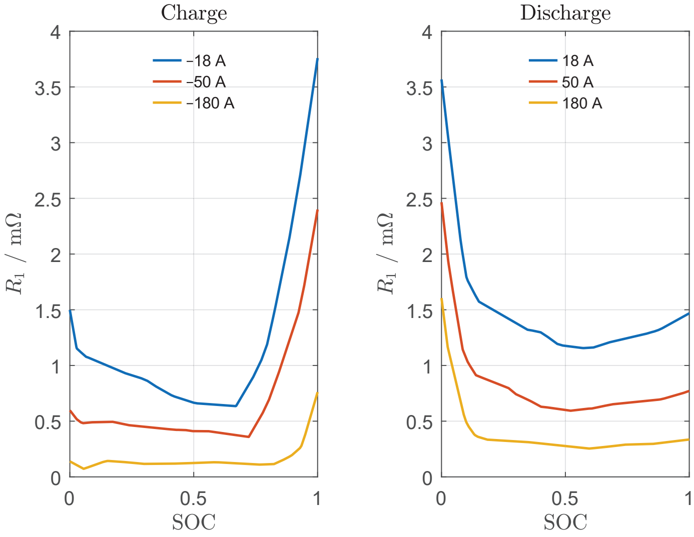
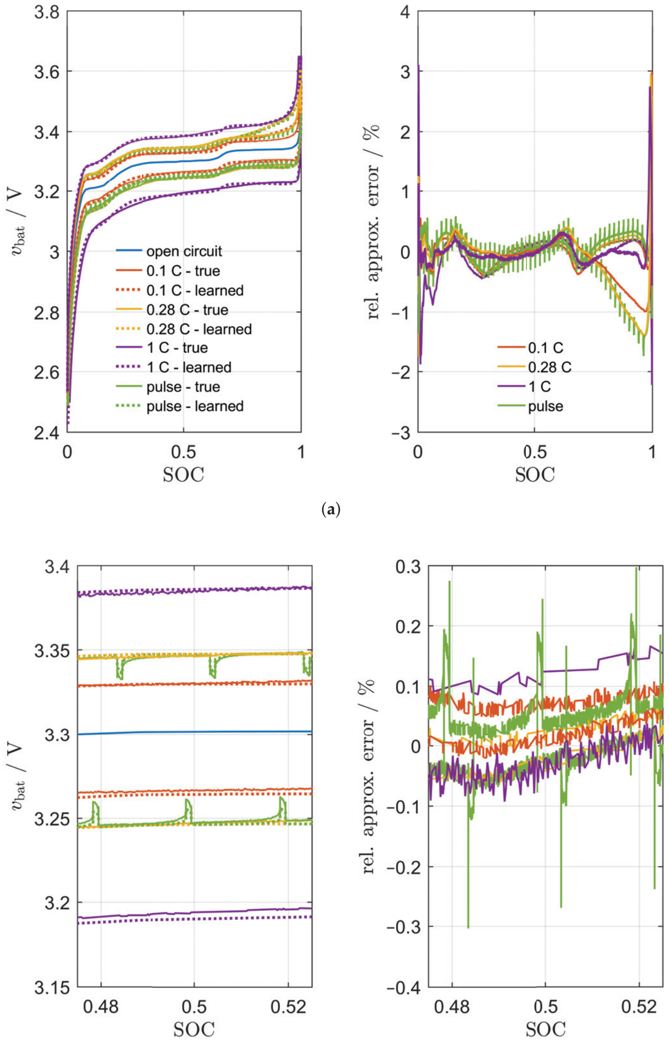


Figure 6. Simulation results: approximation results for R_1 as a function of SOC for different battery currents; (**left**): charging, (**right**): discharging.

3.2. Comparison of Model against Training Data

The measurement data are given as current versus time and voltage versus time series. The current served as the external input of the model which approximated the battery voltage. Figure 7 shows the training results in the form of voltage versus SOC, which allows a better comparison for different C-rates than a voltage versus time plot. The left panel shows the measured and the learned battery voltage as a function of SOC. The right panel shows the approximation error relative to the measured voltage. Figure 7a shows the complete SOC range while Figure 7b focuses on a medium SOC. The simulation results are in good agreement with the experiments over the complete SOC range and for all investigated C-rates. The absolute value of the deviation is smaller than 1% relative to the measured voltage for a wide range of SOC. Only for very low and very high SOC values, the absolute value of the relative approximation error reaches up to around 3%, which is still acceptable. In these ranges the OCV(SOC) curve (shown in blue in Figure 7a,b) is very steep. Therefore, higher approximation errors can be expected.



(**a**)

(**b**)

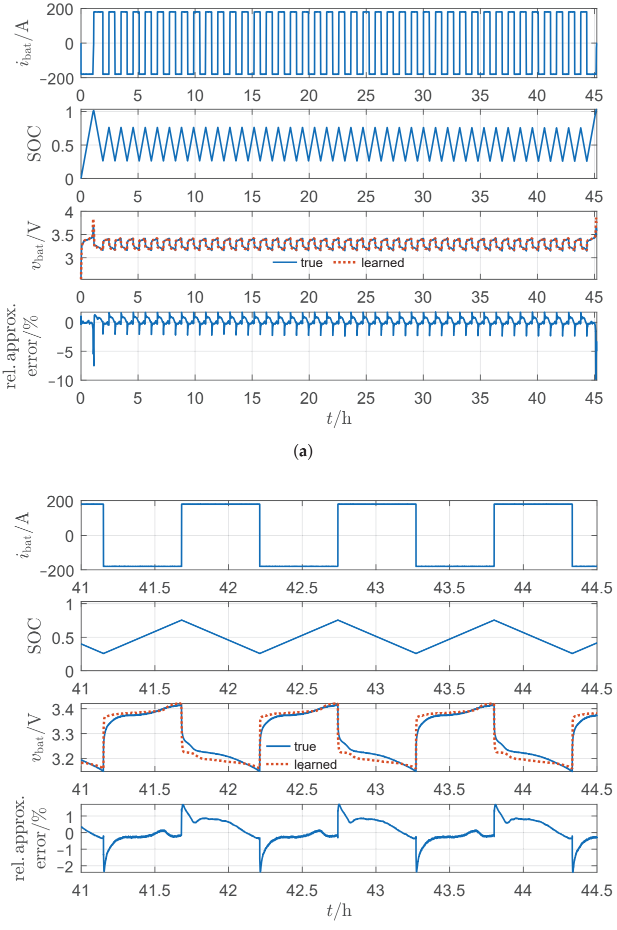
Figure 7. Simulation results using NODEs for grey-box modelling of a lithium-ion battery in comparison to experimental data; left: charge and discharge curves for different C-rates at $T = 25\,°\mathrm{C}$. The lower branches represent discharge (time progresses from right to left), while the upper branches represent charge (time progresses from left to right); right: relative approximation error; (**a**) the whole SOC range (**b**) focus on medium SOC.

Figure 3 compares the training results for a pulsed current charge with the measured voltage. Here, we have chosen a temporal representation. The pulses in Figure 3 are in the area of a medium SOC. The model reproduces the dynamic voltage response of the battery following a current step in a qualitatively correct way. Quantitatively, the absolute voltage drop after the pulse is underestimated by the model. The characteristics of the time behaviour are also different in the simulation compared to the experiment. While the simulation shows an exponential behaviour resulting from the first-order dynamics of the RC element (Equation (12)), the experiment shows a $\sqrt{t}$ behaviour resulting from the solid-state diffusion inside the electrode materials, also referred to as Warburg diffusion [48]. Still, given the relative simplicity of the GB model, the comparison between model and experiment is adequate. Note that we also achieved similar results for other SOC values and for the discharge branch.

In conclusion, the training results show that the GB model can reproduce the training data very well.

3.3. Comparison of Model against Test Data

After finishing the training process we wanted to test the model against data not included in the training. The first test data set consists of consecutive half cycles. The results are shown in Figure 8. Figure 8a shows the test results for the complete time series. In this complete view, the test results are very good. In Figure 8b the focus is on the last three half cycles of the time series. One can see that the dynamics of the battery voltage are modelled well on this scale, although there are deviations between simulation and experiment particularly at the beginning of each half cycle.

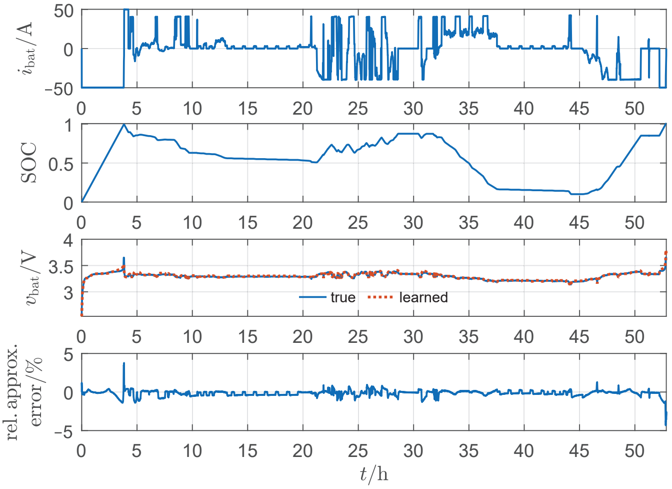


(**a**)

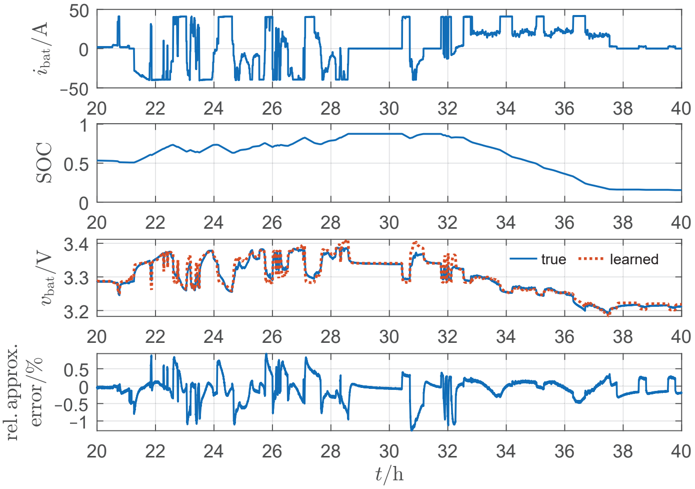
(**b**)

Figure 8. Test results in comparison to experimental data at $T = 25\,°\mathrm{C}$ for half cycles; (**a**) the complete time series; (**b**) focus on the last three half cycles.

We tested the model against a second test data set, a synthetic load profile of a home-storage battery. The results are shown in Figure 9. Figure 9a covers the complete time series, whereas Figure 9b focuses on the segment in the middle covering faster dynamics. The simulations show good agreement with experimental data for the complete load profile. The highest relative approximation errors occur in the area of high SOC values. This was expected because the training error is high at high values of SOC. It is worth mentioning that this synthetic load profile covers the longest measuring time with t = 190,231 s. The longest training time series spanned only t = 41,846 s. Nevertheless, the test results are good for the complete time series.



(**a**)



(**b**)

Figure 9. Test results in comparison to experimental data at T = 25 °C for a synthetic load profile; (**a**) the complete time series (**b**) focus on the segment in the middle.

4. Summary and Conclusions

In this article we have presented the development and application of a GB modelling framework for lithium-ion batteries based on a coupling of NODEs and physics-based ODEs. The model was trained and tested using experimental data of an LFP battery cell used in home-storage applications. The main findings can be summarised as follows.

We showed how to derive a GB model from a physics-based ECM with appropriate choice of learnable functions and parameters. We emphasised the importance of normalisation and initialisation of the parametric parts of the model. The training was split into two training steps: first, a simplified static model was trained where the capacitance of the RC element was neglected. In the second step, the pre-trained parameters were used to train the short-term battery dynamics. When choosing the hyperparameters, especially the number of hidden neurons in f^* and g^* and the number of training epochs, care had to be taken to avoid long training times and overfitting.

The model trained this way was able to reproduce the complete set of training data (CCCV charge and discharge curves as well as pulse tests) with good accuracy (typically < 1% deviation between predicted and measured voltage). In contrast to the GB model proposed in our previous work [17], the present model can approximate the fast (1 s to 30 s) dynamics of the battery. The model was tested against two data sets, half cycles and a synthetic load profile. The simulations showed good agreement with the experimental data. The highest but still acceptable errors occur in the area of low and high SOC values where the OCV curve is very steep. It is worth mentioning that the training database was rather small: only eight time series covering charging and discharging processes were available for training; and the test data sets spanned a much longer time duration than the training data sets.

As an outlook it would be interesting to use more training data, especially from pulse tests with different current steps. Additional data would also improve model validation. For example, a k-fold cross validation could deliver insights into the robustness of the model against the chosen training data. Moreover, the comparison of a WB model and a GB model using NODEs would be of interest.

In conclusion, we have shown that the use of NODEs can be a powerful methodology for modelling lithium-ion batteries.

Author Contributions: Conceptualisation, J.B., W.G.B. and R.G.; methodology, J.B., W.G.B. and R.G.; software, J.B.; validation, J.B., W.G.B. and R.G.; formal analysis, J.B.; investigation, J.B. and R.B.; resources, J.B.; data curation, J.B. and R.B.; writing—original draft preparation, J.B.; writing—review and editing, J.B., R.B., W.G.B. and R.G.; visualisation, J.B.; supervision, W.G.B. and R.G.; project administration, R.G.; funding acquisition, R.G. All authors have read and agreed to the published version of the manuscript.

Funding: J.B. and R.G. acknowledge funding from the Carl Zeiss Foundation and R.B. acknowledges funding from the State of Baden-Württemberg in the framework of the Mittelbauprogramm 2019. Experiments were carried out in the Enerlab 4.0 laboratory which was funded by the Federal Ministry of Education and Research (BMBF) under grant no. 13FH091IN6. The article processing charge was funded by the Baden-Württemberg Ministry of Science, Research and Culture and the Offenburg University of Applied Sciences in the funding programme Open Access Publishing.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The code and measurement data presented in this study are openly available in Zenodo at https://doi.org/10.5281/zenodo.6138075.

Acknowledgments: J.B. is an associated member of the Research Training Group GRK 2218 SiMET—Simulation of mechano-electro-thermal processes in lithium-ion batteries, project number: 281041241. She thanks the German Research Foundation (DFG) for the cooperative support. The authors thank Simone Schede (Offenburg University of Applied Sciences) for proofreading.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

BB	Black-box
CC	Constant current
CCCV	Constant current constant voltage
ECM	Equivalent circuit model
GB	Grey-box
LFP	Lithium iron phosphate
NODE	Neural ordinary differential equation
OCV	Open-circuit voltage
ODE	Ordinary differential equation
RC	Resistor–capacitor
ReLU	Rectified linear unit
ResNet	Residual neural network
RMSE	Root mean squared error
RNN	Recurrent neural network
SOC	State of charge
SOH	State of health
WB	White-box

References

1. Franco, A.A.; Doublet, M.L.; Bessler, W.G. (Eds.) *Physical Multiscale modelling and Numerical Simulation of Electrochemical Devices for Energy Conversion and Storage: From Theory to Engineering to Practice*, 1st ed.; Green Energy and Technology; Springer: London, UK, 2016. [CrossRef]
2. Seaman, A.; Dao, T.S.; McPhee, J. A survey of mathematics-based equivalent-circuit and electrochemical battery models for hybrid and electric vehicle simulation. *J. Power Sources* **2014**, *256*, 410–423. [CrossRef]
3. Estrada-Flores, S.; Merts, I.; de Ketelaere, B.; Lammertyn, J. Development and validation of "grey-box" models for refrigeration applications: A review of key concepts. *Int. J. Refrig.* **2006**, *29*, 931–946. [CrossRef]
4. Oussar, Y.; Dreyfus, G. How to be a gray box: Dynamic semi-physical modelling. *Neural Netw.* **2001**, *14*, 1161–1172. [CrossRef]
5. Duarte, B.; Saraiva, P.M.; Pantelides, C.C. Combined Mechanistic and Empirical Modelling. *Int. J. Chem. React. Eng.* **2004**, *2*. [CrossRef]
6. Hamilton, F.; Lloyd, A.L.; Flores, K.B. Hybrid modelling and prediction of dynamical systems. *PLoS Comput. Biol.* **2017**, *13*, e1005655. [CrossRef]
7. Almeida, G.C.S.; de Souza, A.C.Z.; Ribeiro, P.F. A Neural Network Application for a Lithium-Ion Battery Pack State-of-Charge Estimator with Enhanced Accuracy. *Proceedings* **2020**, *58*, 33. [CrossRef]
8. Jiménez-Bermejo, D.; Fraile-Ardanuy, J.; Castaño-Solis, S.; Merino, J.; Álvaro-Hermana, R. Using Dynamic Neural Networks for Battery State of Charge Estimation in Electric Vehicles. *Procedia Comput. Sci.* **2018**, *130*, 533–540. [CrossRef]
9. Li, W.; Sengupta, N.; Dechent, P.; Howey, D.; Annaswamy, A.; Sauer, D.U. Online capacity estimation of lithium-ion batteries with deep long short-term memory networks. *J. Power Sources* **2021**, *482*, 228863. [CrossRef]
10. Wu, B.; Han, S.; Shin, K.G.; Lu, W. Application of artificial neural networks in design of lithium-ion batteries. *J. Power Sources* **2018**, *395*, 128–136. [CrossRef]
11. Turetskyy, A.; Laue, V.; Lamprecht, R.; Thiede, S.; Krewer, U.; Herrmann, C. Artificial Neural Network enabled P2D Model Deployment for End-of-Line Battery Cell Characterization. In Proceedings of the IEEE 17th International Conference on Industrial Informatics (INDIN), Helsinki-Espoo, Finland, 22–25 July 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 53–58. [CrossRef]
12. Yang, D.; Wang, Y.; Pan, R.; Chen, R.; Chen, Z. A Neural Network Based State-of-Health Estimation of Lithium-ion Battery in Electric Vehicles. *Energy Procedia* **2017**, *105*, 2059–2064. [CrossRef]
13. Nascimento, R.G.; Corbetta, M.; Kulkarni, C.S.; Viana, F.A. Hybrid physics-informed neural networks for lithium-ion battery modelling and prognosis. *J. Power Sources* **2021**, *513*, 230526. [CrossRef]
14. Rackauckas, C.; Ma, Y.; Martensen, J.; Warner, C.; Zubov, K.; Supekar, R.; Skinner, D.; Ramadhan, A.; Edelman, A. Universal Differential Equations for Scientific Machine Learning. *arXiv* **2020**, arXiv:2001.04385.
15. Chen, R.T.Q.; Rubanova, Y.; Bettencourt, J.; Duvenaud, D. Neural Ordinary Differential Equations. *arXiv* **2018**, arXiv:1806.07366.
16. Bills, A.; Sripad, S.; Fredericks, W.L.; Guttenberg, M.; Charles, D.; Frank, E.; Viswanathan, V. Universal Battery Performance and Degradation Model for Electric Aircraft. *ChemRxiv* **2020**. [CrossRef]

17. Brucker, J.; Bessler, W.G.; Gasper, R. Grey-box modelling of lithium-ion batteries using neural ordinary differential equations. *Energy Inform.* **2021**, *4*, 1–13. [CrossRef]
18. Yagci, M.C.; Behmann, R.; Daubert, V.; Braun, J.A.; Velten, D.; Bessler, W.G. Electrical and Structural Characterization of Large–Format Lithium Iron Phosphate Cells used in Home–Storage Systems. *Energy Technol.* **2021**, *9*, 2000911. [CrossRef]
19. Dubarry, M.; Truchot, C.; Liaw, B.Y. Cell degradation in commercial LiFePO4 cells with high-power and high-energy designs. *J. Power Sources* **2014**, *258*, 408–419. [CrossRef]
20. Kupper, C.; Weißhar, B.; Rißmann, S.; Bessler, W.G. End-of-Life Prediction of a Lithium-Ion Battery Cell Based on Mechanistic Aging Models of the Graphite Electrode. *J. Electrochem. Soc.* **2018**, *165*, A3468–A3480. [CrossRef]
21. Dreyer, W.; Jamnik, J.; Guhlke, C.; Huth, R.; Moskon, J.; Gaberscek, M. The thermodynamic origin of hysteresis in insertion batteries. *Nat. Mater.* **2010**, *9*, 448–453. [CrossRef]
22. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; Adaptive Computation and Machine Learning; MIT Press: Cambridge, MA, USA, 2016.
23. Che, Z.; Purushotham, S.; Cho, K.; Sontag, D.; Liu, Y. Recurrent Neural Networks for Multivariate Time Series with Missing Values. *Sci. Rep.* **2018**, *8*, 6085. [CrossRef]
24. Bailer-Jones, C.A.; MacKay, D.J.; Withers, P.J. A recurrent neural network for modelling dynamical systems. *Netw. Comput. Neural Syst.* **1998**, *9*, 531–547. [CrossRef]
25. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 29th IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 770–778.
26. Liao, Q.; Poggio, T. Bridging the Gaps Between Residual Learning, Recurrent Neural Networks and Visual Cortex. *arXiv* **2016**, arXiv:1604.03640.
27. Haber, E.; Ruthotto, L. Stable architectures for deep neural networks. *Inverse Probl.* **2017**, *34*, 014004. [CrossRef]
28. Ruthotto, L.; Haber, E. Deep Neural Networks Motivated by Partial Differential Equations. *J. Math. Imaging Vis.* **2020**, *62*, 352–364. [CrossRef]
29. Dupont, E.; Doucet, A.; Teh, Y.W. Augmented Neural ODEs. In *Advances in Neural Information Processing Systems 32*; Curran Associates, Inc.: Red Hook, NY, USA, 2019; pp. 3140–3150.
30. Zhang, T.; Yao, Z.; Gholami, A.; Keutzer, K.; Gonzalez, J.; Biros, G.; Mahoney, M.W. ANODEV2: A Coupled Neural ODE Evolution Framework. *arXiv* **2019**, arXiv:1906.04596.
31. Haber, E.; Ruthotto, L.; Holtham, E.; Jun, S.H. Learning across scales - Multiscale Methods for Convolution Neural Networks. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; AAAI Press: Palo Alto, CA, USA, 2018; Volume 32.
32. Gholami, A.; Keutzer, K.; Biros, G.; Gholaminejad, A. ANODE: Unconditionally Accurate Memory-Efficient Gradients for Neural ODEs. In Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, Macao, China, 10–16 August 2019; pp. 730–736.
33. He, H.; Xiong, R.; Fan, J. Evaluation of Lithium-Ion Battery Equivalent Circuit Models for State of Charge Estimation by an Experimental Approach. *Energies* **2011**, *4*, 582–598. [CrossRef]
34. Wang, Y.; Fang, H.; Zhou, L.; Wada, T. Revisiting the State-of-Charge Estimation for Lithium-Ion Batteries: A Methodical Investigation of the Extended Kalman Filter Approach. *IEEE Control Syst.* **2017**, *37*, 73–96. [CrossRef]
35. Fleischer, C.; Waag, W.; Heyn, H.M.; Sauer, D.U. On-line adaptive battery impedance parameter and state estimation considering physical principles in reduced order equivalent circuit battery models: Part 1. Requirements, critical review of methods and modelling. *J. Power Sources* **2014**, *260*, 276–291. [CrossRef]
36. Chen, M.; Rincón-Mora, G.A. Accurate Electrical Battery Model Capable of Predicting Runtime and I–V Performance. *IEEE Trans. Energy Convers.* **2006**, *21*, 504–511. [CrossRef]
37. Haifeng, D.; Xuezhe, W.; Zechang, S. A new SOH prediction concept for the power lithium-ion battery used on HEVs. In Proceedings of the 2009 IEEE Vehicle Power and Propulsion Conference, Dearborn, MI, USA, 7–10 September 2009; pp. 1649–1653. [CrossRef]
38. Hu, Y.; Yurkovich, S.; Guezennec, Y.; Yurkovich, B.J. A technique for dynamic battery model identification in automotive applications using linear parameter varying structures. *Control Eng. Pract.* **2009**, *17*, 1190–1201. doi: 10.1016/j.conengprac.2009.05.002. [CrossRef]
39. Tong, S.; Klein, M.P.; Park, J.W. On-line optimisation of battery open circuit voltage for improved state-of-charge and state-of-health estimation. *J. Power Sources* **2015**, *293*, 416–428. [CrossRef]
40. Krewer, U.; Röder, F.; Harinath, E.; Braatz, R.D.; Bedürftig, B.; Findeisen, R. Review—Dynamic Models of Li-Ion Batteries for Diagnosis and Operation: A Review and Perspective. *J. Electrochem. Soc.* **2018**, *165*, A3656–A3673. [CrossRef]
41. Weißhar, B.; Bessler, W.G. Model-based lifetime prediction of an LFP/graphite lithium-ion battery in a stationary photovoltaic battery system. *J. Energy Storage* **2017**, *14*, 179–191. [CrossRef]
42. Pflugradt, N.; Platzer, B. Verhaltensbasierter Lastprofilgenerator für Strom- und Warmwasser-Profile. In *Thermische Solarenergie*; Vajen, K., Ed.; Wissen für Profis, Ostbayerisches Technologie-Transfer-Institut e.V. (OTTI): Regensburg, Germany, 2012; pp. 250–251.
43. LeCun, Y.; Bottou, L.; Orr, G.B.; Müller, K.R. Efficient BackProp. In *Neural Networks: Tricks of the Trade*; Orr, G.B., Müller, K.R., Eds.; Springer: Berlin/Heidelberg, Germany, 1998.

44. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*; Curran Associates, Inc.: Red Hook, NY, USA, 2019; pp. 8024–8035.
45. Chen, R.T.Q. Torchdiffeq. 2021. Available online: https://github.com/rtqichen/torchdiffeq (accessed on 23 February 2022).
46. Quarti, M.; Bessler, W.G. Model-Based Overpotential Deconvolution, Partial Impedance Spectroscopy, and Sensitivity Analysis of a Lithium-Ion Cell with Blend Cathode. *Energy Technol.* **2021**, *9*, 2001122. [CrossRef]
47. Pilla, A.A. A Transient Impedance Technique for the Study of Electrode Kinetics: Application to Potentiostatic Methods. *J. Electrochem. Soc.* **1970**, *117*, 467–477. [CrossRef]
48. Barsoukov, E.; Macdonald, J.R. *Impedance Spectroscopy: Theory, Experiment, and Applications*, 2nd ed.; Wiley-Interscience: Hoboken, NJ, USA, 2005. [CrossRef]

energies

MDPI

Article

Forecasting the Total South African Unplanned Capability Loss Factor Using an Ensemble of Deep Learning Techniques

Sibonelo Motepe [1,*], Ali N. Hasan [2] and Thokozani Shongwe [1]

[1] Department of Electrical and Electronic Engineering Technology, Faculty of Engineering and the Built Environment, University of Johannesburg, Johannesburg 2092, South Africa; tshongwe@uj.ac.za
[2] Department of Electrical Engineering, Faculty of Engineering Science and Technology, Higher Colleges of Technology, Abu Dhabi 25026, United Arab Emirates; alinabeal99@gmail.com
* Correspondence: djscvii@gmail.com

Abstract: Unplanned power plant failures have been seen to be a major cause of power shortages, and thus customer power cuts, in the South African power grid. These failures are measured as the unplanned capability loss factor (UCLF). The study of South Africa's UCLF is almost non-existent. Parameters that affect the future UCLF are, thus, still not well understood, making it challenging to forecast when power shortages may be experienced. This paper presents a novel study of South African UCLF forecasting using state-of-the-art deep learning techniques. The study further introduces a novel deep learning ensemble South African UCLF forecasting system. The performance of three of the best recent forecasting techniques, namely, long short-term memory recurrent neural network (LSTM-RNN), deep belief network (DBN), and optimally pruned extreme learning machines (OP-ELM), as well as their aggregated ensembles, are investigated for South African UCLF forecasting. The impact of three key parameters (installed capacity, demand, and planned capability loss factor) on the future UCLF is investigated. The results showed that the exclusion of installed capacity in the LSTM-RNN, DBN, OP-ELM, and ensemble models doubled the UCLF forecasting error. It was also found that an ensemble model of two LSTM-RNN models achieved the lowest errors with a symmetric mean absolute percentage error (sMAPE) of 6.43%, mean absolute error (MAE) of 7.36%, and root-mean-square error (RMSE) of 9.21%. LSTM-RNN also achieved the lowest errors amongst the individual models.

Keywords: deep learning; forecasting; power outages; coal power plants; recurrent neural networks; ensemble techniques

Citation: Motepe, S.; Hasan, A.N.; Shongwe, T. Forecasting the Total South African Unplanned Capability Loss Factor Using an Ensemble of Deep Learning Techniques. *Energies* **2022**, *15*, 2546. https://doi.org/10.3390/en15072546

Academic Editors: Luis Hernández-Callejo, Sergio Nesmachnow and Sara Gallardo Saavedra

Received: 31 January 2022
Accepted: 7 March 2022
Published: 31 March 2022

1. Introduction

South Africa has been seen to be a late participant in the three key industrial revolutions [1]. The use of artificial intelligence (AI) and data is on the rise in South Africa [2–4]. This rise means that South Africa might not be a late participant in the fourth industrial revolution. In 2007, 2013, 2018, and 2019, South Africa experienced a shortage in power supply due to various challenges, leading to load shedding [1]. South Africa's public power utility, Eskom, has on several occasions stated its inability to accurately predict/forecast the unplanned capability loss factor (UCLF) as one of the major factors leading to an unreliable power supply and unpredictable load shedding [5,6]. UCLF is a term that refers to the measure of unplanned plant breakdown. The behavior of South African UCLF has not been well studied. Pretorius et al. studied the impact of the South African energy crisis on emissions [7]. This study only talks about an increase in UCLF due to maintenance deferral. The study does not talk about how to forecast UCLF, nor the major factors that contribute to UCLF that can help in the forecasting of UCLF. The UCLF, planned capability loss factor (PCLF), and other capability loss factor (OCLF), together with the installed capacity, determine the power available to supply customers. The PCLF is the planned plant outages for the maintenance or refurbishments of the plant. This is typically a planned, set value set by the utility. The utility can decide to change their planned outage/PCLF depending on different factors. The OCLF accounts for other or random losses and is usually significantly smaller than the UCLF [8]. The installed capacity gives the number of megawatts of the installed power plant units. Micali studied the prediction of new coal power plants' availability in the absence of data in South Africa [8]. The author mentions that the work is a precursor to predicting UCLF in new plants. The author proposes using expert opinion with some data from stations where data are available. However, the work in [8] did not focus on the total UCLF, assumed limited availability of data, did not use AI techniques, and depended on expert knowledge. In [9], the authors state that expert knowledge can change from one expert to the next, and thus expert results can be different from the same data. The author, in addition, did not investigate factors that affect power supply and may influence the UCLF [8]. There is, thus, a gap in South Africa in terms of accurately forecasting UCLF. In addition, the study of the total South African UCLF behavior is a gap as only precursor work exists, and the precursor work is focused on new plants. Another gap is the use of intelligent systems that are not reliant on human experts in UCLF forecasting.

To add to the previous paragraph, the knowledge of when the power system might experience a power shortage is still a topic of interest and is not only important for the utility, but also customers. Knowing when there may be a power shortage, and hence a requirement to reduce consumption, helps customers plan their operations. Unplanned failures have been studied before. In [10], real-time prediction of distribution system outage duration using historical outage records to train neural networks was studied. The Netherlands collects information on unplanned outages from its utilities to inform its maintenance and investment policies [11].

South Africa is the highest producer of electricity in Africa and is in the top 25 producers of power in the world [12,13]. Over 80% of South Africa's power is produced by coal-fired power stations and a nuclear power station. The total South African power grid UCLF can, thus, be modeled as that of the coal and nuclear power stations. Despite the recent move towards cleaner energy, the largest power-producing countries, such as India and China, still rely heavily on coal-fired power stations [12]. The study of coal thermal power plants and behavior is, thus, still of interest [14–17]. The study of the South African coal-fired power station UCLF is, therefore, important as coal power plants are still highly used and are still a research topic of interest.

Forecasting and prediction have been topics of interest for many researchers [10,18]. This is mainly due to an interest in understanding and predicting the future behavior of certain variables. Artificial intelligence (AI) techniques have become popular in these forecasting/prediction tasks. One of the reasons for this popularity is their ability to model non-linearity with high accuracy. Khoza and Marwala used an ensemble of the multi-layer perceptron and rough set theory to predict the direction that the South African gross domestic product (GDP) would take [18]. Galius proposed a probabilistic model for modeling power distribution network blackouts [19]. In Egypt, power cable failures were analyzed to help prevent future power outages [20]. In [21], bilateral long short-term memory (LSTM) was used to forecast the short-term cycle of wafer lots for the planning and control of wafer manufacturing. The rise of computational power and access to labeled data has led to an increase in the utilization of deep learning techniques [22]. Deep learning techniques have been seen to have an excellent performance in multiple areas, such as language and speech processing, as well as computer vision [23,24]. Alhussein et al. used a hybrid of convolutional neural networks (CNN) and long short-term memory (LSTM) to forecast individual house loads [25]. Here, the researchers use CNN to select features from the input data and LSTM to learn the sequence. The authors stated a mean absolute percentage error (MAPE) improvement greater than 4% in comparison to LSTM-based models. Kong et al. also combined CNN and LSTM for short-term load forecasting in Singapore [26]. Pandit et al. compared LSTM and Markov chain models in weather forecasting for German offshore wind farms to improve their wind turbine availability and maintenance [27]. Deep learning has also been used to forecast wind speeds at turbine locations [28]. The authors combine CNN and the gated recurrent unit (GRU) to achieve satisfactory results in comparison to existing models. Deep learning techniques have also been used to forecast the Korean postal delivery service demand [29]. This observed performance of deep learning techniques has also led to their adoption in recent load forecasting studies [30,31]. A gap still exists in the application of the state-of-the-art techniques in forecasting UCLF (and South African UCLF), as applied in forecasting in the different engineering areas.

As observed, a number of studies have used a combination of techniques to achieve improved performance [25–29]. This combination of techniques is usually termed ensemble or hybrid techniques. Ensemble techniques have also been used for classification in different engineering applications. Ramotsoela et al. used an ensemble of five artificial intelligence techniques to detect intrusion in water distribution systems [32]. The ensemble model used here combined an artificial neural network (ANN), RNN (recurrent neural network), LSTM, GRU, and CNN in a voting system. The ensemble model classified its output as an anomaly if at least two constituent models classified their outputs as an anomaly. CNN models have been combined to determine driver behavior from multiple data streams [33]. The proposed ensemble model incorporated a voting system to enhance the classification accuracy. A double ensemble model of semi-supervised gated stacked auto-encoders has been used to predict industrial key performance indicators [34]. Drif et al. proposed an ensemble of auto-encoders for recommendations [35]. The authors used an aggression method to combine outputs from the sub-models to form the ensemble model output. Bibi et al. used an ensemble-based technique to forecast electricity spot prices in the Italian electricity market [36]. The authors estimated deterministic components using semi-parametric techniques and then determined stochastic components using time series, and machine learning algorithms. The final forecast is obtained from the estimates of both components [36]. Shah et al. used a similar approach to Bibi et al. in short-term electricity demand forecasting for the Nordic electricity market [37]. The similarity is that the authors separated their approach into a deterministic and a stochastic component and then combined the estimates from them to obtain the final forecast. None of the literature covers the use of ensemble techniques in forecasting UCLF. The use of ensemble techniques in UCLF forecasting is, thus, an existing research gap.

This paper introduces the following contributions: (i) A novel study of the South African UCLF behavior using state-of-the-art AI (deep learning and ensemble) techniques. (ii) An investigation of the impact of the installed capacity, historic demand, and PCLF on the UCLF forecasting accuracy. (iii) An introduction of a novel deep-learning ensemble total South African UCLF forecasting system.

The remainder of this paper is arranged as follows: Section 2 presents the techniques used in this research. Section 3 presents the experimental setup. The proposed UCLF forecasting system is presented in Section 4. Section 5 then presents the experimental results and the discussion of the results. The paper conclusions are presented in Section 6. Section 7 presents the limitations of the study as well as future work. The paper flow chart is shown in Figure 1.

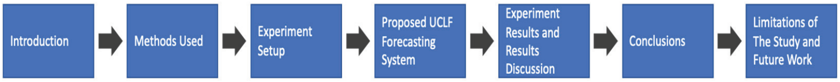

Introduction
Methods Used
Experiment Setup
Proposed UCLF Forecasting System
Experiment Results and Results Discussion
Conclusions
Limitations of The Study and Future Work

Figure 1. The paper arrangement flow chart.

2. Methods Used

This section presents the four techniques used in this research.

2.1. OP-ELM

The optimally pruned extreme learning machine (OP-ELM) is the improved version of the extreme learning machine. This improved technique, introduced by Miche et al., uses the leave-one-out (LOO) method to select the optimal number of neurons [38]. LOO marginalizes the irrelevant neurons built into ELM's network. This marginalization helps overcome the shortfall in the approximation of the training dataset's correlated and irrelevant variables. Given a training set x_i, with a target vector t_i, the OP-ELM's objective is to obtain the minimum possible error function. The OP-ELM equation is given by (1). If there exists an input weight vector connecting the kth hidden neuron and the input (w_k), a kth hidden node's bias (b_k), and an output weight connecting the output and the kth hidden neuron (β_k), such that $\sum_{k=1}^{j} f(w_k, b_k, x_i)\beta_k = y_i$, (1) can be re-written as (2).

$$\sum_{k=1}^{j} f(w_k, b_k, x_i)\beta_k = t_i \tag{1}$$

$$\mathrm{H}\beta = \mathrm{T} \tag{2}$$

$$H = \begin{bmatrix} f(w_1, b_1, x_1) & \cdots & f(w_k, b_k, x_1) \\ \vdots & \ldots & \vdots \\ f(w_1, b_1, x_m) & \ldots & f(w_k, b_k, x_m) \end{bmatrix}_{m \times j} \tag{3}$$

$$\beta = H^* T = \left(HH^T\right)^{-1} HT^T \tag{4}$$

where y_i is the output vector, t_i is the output target vector, H is the hidden layer's output matrix, and k = 1, 2 . . . j. The input weights and biases are assigned at random and do not require tuning. The hidden layer's output matrix parameters are also assigned random values. If H is a square matrix, matrix inversion can be used to determine the output weights. In a case where H is not a square matrix, the Moore–Penrose Equation (4) is used to determine the output weights. The neurons are ranked using multi-response sparse regression, and the LOO is then applied.

2.2. LSTM-RNN

The fading of previously learned patterns is a challenge experienced in standard RNN architectures. The LSTM-RNN has a memory cell to overcome this shortcoming. The memory cell is managed by non-linear gating units. The gated units of an LSTM-RNN unit can be seen in Figure 2. These gated units, the forget gate (f_n), input gate (i_n), and output gate (o_n), are presented by Equations (5)–(7), respectively. Equations (8)–(10), respectively, present the input node (g_n), the state (s_n), and the cell state (h_n). Here, n is the time step, $\varnothing$ is the tanh function, σ is the sigmoid function, and the W matrices are the respective network activation functions' corresponding input weights. The LSTM-RNN cells are stacked after each other to achieve a deep layered LSTM-RNN. The memory cells give the models the ability to sustain memory.

$$f_n = \sigma\left(W_{fz} z_n + W_{fh} h_{n-1} + b_f\right) \tag{5}$$

$$i_n = \sigma(W_{iz} z_n + W_{ih} h_{n-1} + b_i) \tag{6}$$

$$o_n = \sigma(W_{oz} z_n + W_{oh} h_{n-1} + b_o) \tag{7}$$

$$g_n = \varnothing\left(W_{gz} z_n + W_{gh} h_{n-1} + b_g\right) \tag{8}$$

$$s_n = g_n \odot i_n + s_{n-1} \odot f_n \tag{9}$$

$$h_n = \varnothing(s_n) \odot o_n \tag{10}$$

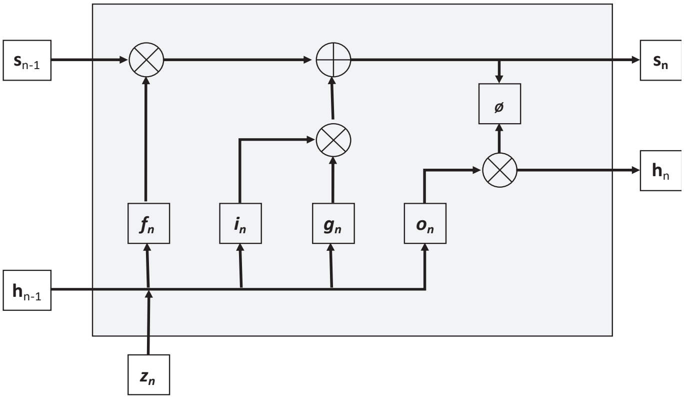


Figure 2. An LSTM−RNN cell with gated units.

2.3. DBN

The deep belief network (DBN) is built by stacking restricted Boltzmann machines (RBM). The technique was introduced in the mid-2000s by Geoffrey Hinton. There are no connections between the neurons on the same layer. There is a symmetrical and bi-directional connection between the layers. The model determines the hidden state, visible state, initial weight, and biases in the first step using unsupervised learning. Supervised learning, using back-propagation, is used to append the unsupervised learning pre-trained model. The joint distribution over the visible and hidden units is given by (11) [39].

$$P(m,h) = \frac{e^{-E(m,h)}}{\sum_n \sum_h e^{-E(m,h)}} \tag{11}$$

where $E(m, h)$ is the energy function. The conditionally independent conditional probabilities are given by (12) and (13). If the values of the hidden and visible units are from 0 to 1, (12) and (13), respectively, become (14) and (15), with $i = 1, 2 \ldots\ k_h$ and $j = 1, 2 \ldots\ k_m$.

$$p(m|h) = \prod_j p(m_j|h) \tag{12}$$

$$p(h|m) = \prod_i p(h_i|m) \tag{13}$$

$$p(m_j = 1|h) = sigmoid\left(\alpha_j + \sum_{i=1}^{k_h} W_{ij}h_i\right) \tag{14}$$

$$p(h_j = 1|m) = sigmoid\left(\beta_i + \sum_{j=1}^{k_m} W_{ij}m_j\right) \tag{15}$$

2.4. Ensemble

Ensembles of models of the three techniques used in this study, LSTM-RNN, OP-ELM, and DBN, are investigated for UCLF forecasting. Ensemble models are a combination of multiple models to try to achieve better performance than that of the individual models. There is a number of different ways that models can be combined to form an ensemble [30]. Figure 3 shows a summary of the aggregate method, which is commonly used in regression problems. Here, models operate in parallel, and their outputs are aggregated to obtain the ensemble model's output. The aggregate ensemble model output, O_φ, can be written as (16). Here, O_{mk} is the ensemble model's *kth* output for models m_1, $m_2 \ldots\ m_n$, and n is the number of models used to develop the assembly model. The equally weighted method was used, where each model's output into the ensemble model is given an equal weight.

$$O_\varphi = \frac{1}{n}\sum_{k=1}^{k=1} O_{mk} \tag{16}$$

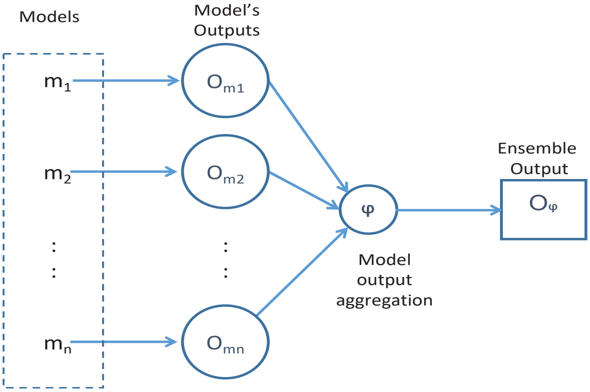


Figure 3. Summary of the aggregate ensemble method.

3. Experimental Setup

This section presents the experiment setup via two sub-sections. The first sub-section presents the South African coal generation plants overview. The second sub-section presents the experimental approach.

3.1. South African Key Coal Power Generation Plants Overview

South Africa has 15 key coal-powered thermal power stations. These stations are owned and operated by Eskom. Two of these stations are the new supercritical power stations, Medupi and Kusile, which are still under construction and at different stages of completion. The power stations are mostly concentrated in Mpumalanga Province, mainly due to the large availability of coal in this province. Twelve coal power stations are located in Mpumalanga, two in Limpopo Province, and one in Free State Province. Figure 4 shows the location of the South African coal-fired power stations [40]. South Africa also has one nuclear power generation station located in the Western Cape Province. This power station has an installed capacity of 1940 MW. This nuclear station and the coal-fired power stations contribute to over 80% of South Africa's installed capacity and supply the country's baseload. The PCLF and UCLF data used in this research are from these coal-fired powered stations and the nuclear power station, collected from a centralized database.

3.2. Data Description

The data used in this study were real utility data collected from January 2010 to December 2019. Figure 5 shows the different periodicities of the UCLF over time. Figure 5c shows the periodicity over weeks in parts of the South African winter (June–July) and summer (November–December) season in the year 2019.

The collected data were for four variables: the installed capacity, demand, PCLF, and UCLF. To investigate how these variables affect the UCLF forecast accuracy of the different techniques, the variables were arranged into five experiments, as shown in Figure 6. A tick indicates that a variable is used in the respective experiment and a cross indicates that the variable was not used in the experiment. The experiment with the best performance will, thus, indicate which variables should be used with which technique to achieve the lowest year-ahead UCLF forecasting error. The installed capacity is the total power that can be generated by the installed power generation plants in megawatts. The demand is the historic total national power demand in megawatts. The PCLF and UCLF are the respective historic variables in megawatts. The UCLF data used for the input in the training and testing of the models were split into the UCLF two years before the target UCLF, *UCLF T-2 Years*, and the UCLF a year before the target, *UCLF T-1 Year*. The UCLF data used was a daily peak value. A variable indicating if it is a weekend or a weekday, the

Weekend Index, was also used as an input. This variable was a 1 for weekends and a 0 for weekdays. This variable was included for the models to be able to differentiate the data for a weekday and the weekend, respectively. This resulted in six input variables. The training period was between 1 January 2012 and 31 December 2018. The testing period was between 1 January 2019 and 31 December 2019. Thus, the forecasts were a daily peak UCLF for the year-ahead forecast period. All the variables, except the weekend index, were normalized to be between 0 and 1. The training input data were, thus, a 2555 × n matrix, where the 2555 is the daily input values over 7 years and n is the number of variables used in the respective experiment, as described next.

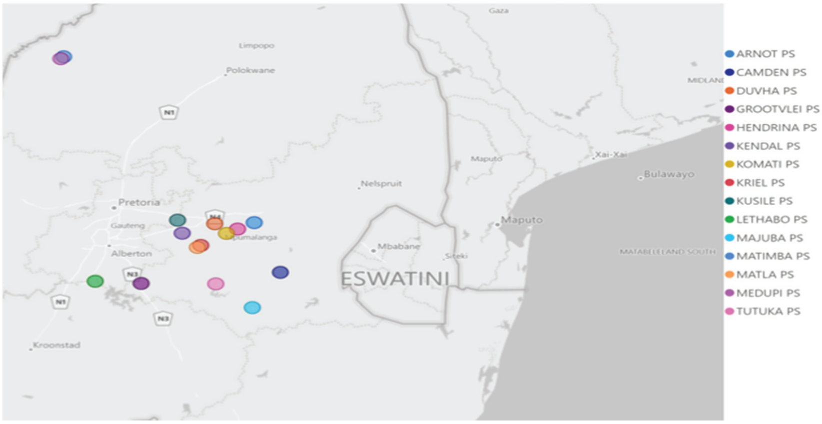


Figure 4. Location of 15 key South African coal-fired power stations.

The training input variable matrix sizes were, thus, 2555 × 6 for Exp 1, 2555 × 5 for Exp 2 to Exp 4, and 2555 × 3 for Exp 5.

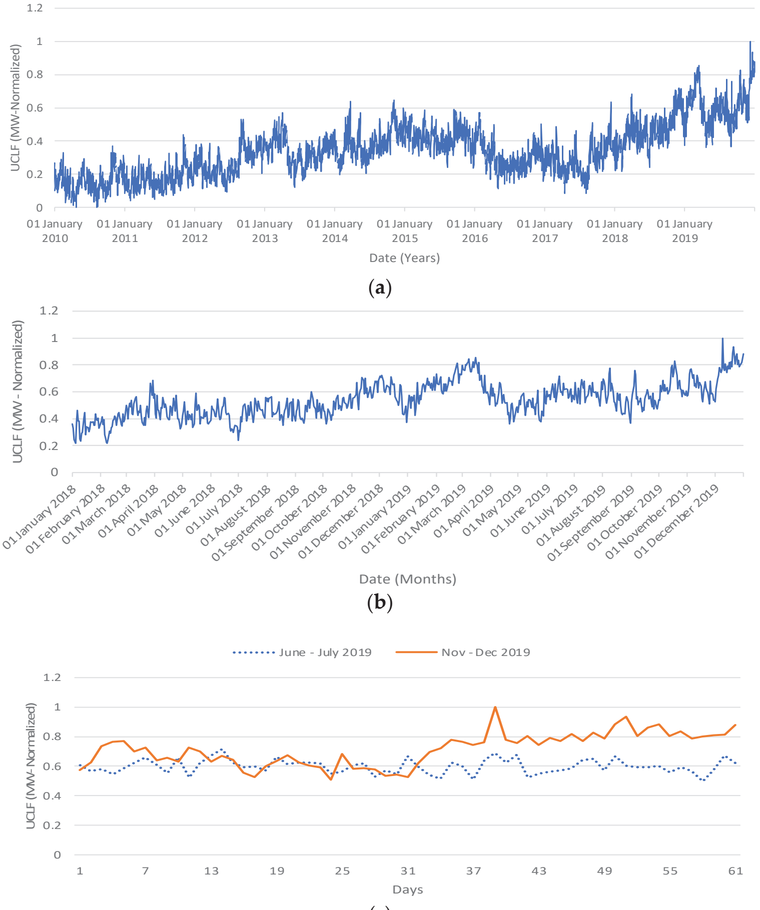
3.3. Experimental Approach

The different techniques' models were, respectively, developed using various approaches.

The OP-ELM models were trained by tuning the model dimensions. A different number of hidden nodes were used to train the model in the respective experiments. Optimal pruning using the LOO method was key in determining the model's dimensions. Various dimensions were investigated and the model with the lowest errors in each experiment was captured and is presented in the results section.

LSTM-RNN models were trained with different numbers of stacked hidden LSTM units. The variation of the hidden units was consistent in all the different experiments. Similar to the OP-ELM, the performance results for the model with the lowest obtained UCLF forecast errors were captured.

Single layered DBN models were developed with the number of hidden units being varied for the respective models, the lowest number of hidden units used was four with the highest number of hidden units being sixteen.

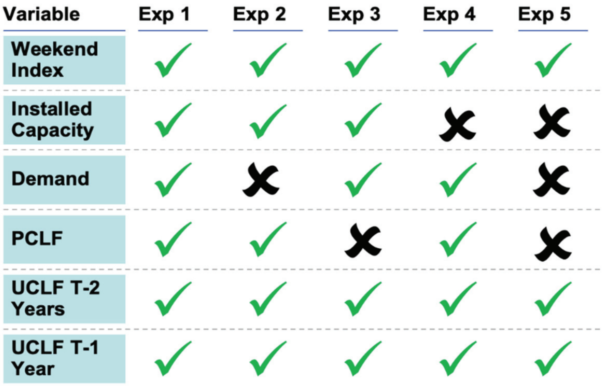


(**a**)

(**b**)

(**c**)

Figure 5. The South African UCLF (MW—normalized): (**a**) UCLF for a period between January 2010 and December 2019; (**b**) monthly periodicity of UCLF between January 2018 and December 2019; (**c**) weekly periodicity for June–July 2019 and November–December 2019.



Variable	Exp 1	Exp 2	Exp 3	Exp 4	Exp 5
Weekend Index	✓	✓	✓	✓	✓
Installed Capacity	✓	✓	✓	✗	✗
Demand	✓	✗	✓	✓	✗
PCLF	✓	✓	✗	✓	✗
UCLF T-2 Years	✓	✓	✓	✓	✓
UCLF T-1 Year	✓	✓	✓	✓	✓

Figure 6. Variables used in the different experiments conducted per technique.

The aggregation ensemble approach was used for the ensemble of the three techniques. These ensembles were of two techniques at a time. Here, the various respective parameters per technique are tuned and combined to form different ensemble models. The performance results of the forecast results with the lowest errors are captured per experiment. For each technique and experiment, the other hyperparameters, such as training rate and the number of layers, were kept the same. In future work, the effect of optimizing the hyperparameters can be investigated.

3.4. Performance Measures Used

Each model's performance was measured using three key performance measures: symmetric mean absolute percentage error (sMAPE), mean absolute error (MAE), and root-mean-square error (RMSE). Motepe et al. state that the MAE, RMSE, MPE, MAPE, and sMAPE are common forecasting error measurements [30]. They further state the challenge that the MAPE faces when target values are too small, which leads to errors being too large. The three used performance measurements in this research are presented in (17)–(19).

$$sMAPE = \frac{2}{N}\sum_{k=1}^{N}\frac{|F_k - T_k|}{|F_k| + |T_k|} \tag{17}$$

$$MAE = \frac{\sum_{k=1}^{N}|F_k - T_k|}{N} \tag{18}$$

$$RMSE = \sqrt{\frac{\sum_{k=1}^{N}(F_k - T_k)^2}{N}} \tag{19}$$

where F_k is the forecasted value, T_k is the target value, and N is the number of forecasted values.

3.5. Statistical Significance Test

After the model performance is measured, the model results can be found to not be statistically different from each other. This means that despite one model achieving results with a lower error in comparison to the next model's results, the model with the lower error does not necessarily outperform the model it is being compared to. A statistical test can be used to determine if model results are statistically significantly different. One such test is the t-test. The t-test uses the mean and the variance to check if two samples are from the same sample. The test calculates a significant value, also termed the *p*-value. A *p*-value less than the acceptable value means that the samples being compared have a significant difference, and vice versa. A *p*-value of 0.05, which is a commonly used value in scientific studies, was used in this study. The statistical significance test is performed, for each technique between the results with the lowest overall errors and results with the lowest errors from Exp 1, Exp 2, Exp 3, Exp 4, and/or Exp 5.

4. Proposed UCLF Forecasting System

Figure 7 presents the proposed UCLF forecasting system. The power stations monitor their plant's performance and report this locally at the station and centrally. These data are then stored in a central database. The UCLF data are part of these stored power station data. A record of the power station units that are on planned outages, PCLF, for maintenance or refurbishment is also stored centrally. These PCLF data are then provided by a central planning department in conjunction with the central operations department. The planning department also provides the installed capacity data to the central database. The system operator or an equivalent department would then provide the demand data. The data are pre-processed, and the variables are then consolidated for input into the deep learning (DL) ensemble UCLF forecasting module. The DL ensemble UCLF forecast module contains a DL ensemble model that forecasts the UCLF. The UCLF forecast is then stored and used by the planning, operations, and system operator. The DL ensemble model is developed and tested offline, and then deployed in the system. The UCLF forecast data together with the actual UCLF data are then used by a model performance evaluation module to periodically check if the model's accuracy is still acceptable based on the utility's requirements.

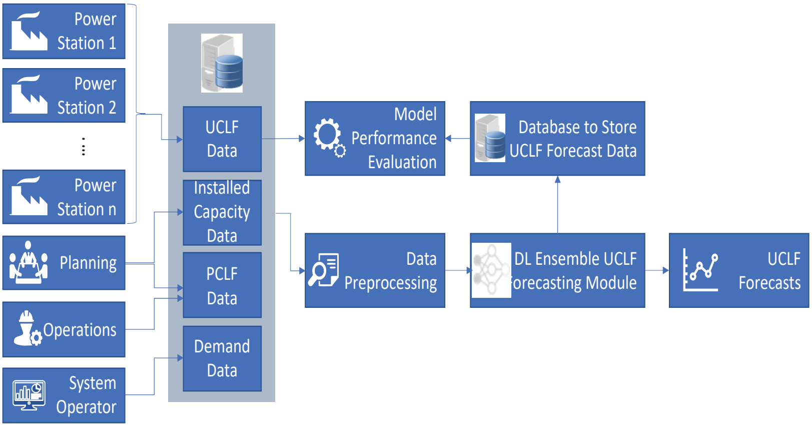


Figure 7. Proposed deep learning UCLF forecasting system.

5. Experiment Results and Results Discussion

This section presents the results of the five different experiments for the four techniques. The results are then discussed.

5.1. OP-ELM Results

The different experiments were conducted with different OP-ELM models, as described in Section 3. The lowest obtained errors per experiment are captured in Table 1. It was found that the OP-ELM model developed using variables for Experiment 2 and 50 hidden nodes achieved the lowest errors. This model achieved an sMAPE of 10.21%, MAE of 11.57%, and RMSE of 14.65%. These performance results are in bold in Table 1. This model was, therefore, developed without the demand as an input. Experiments 4 and 5's lowest obtained errors were higher than the lowest obtained errors in the other three experiments. The exclusion of the installed capacity, in Experiments 4 and 5, was observed to lead to an increase in the errors. In these experiments, the sMAPE increased by over 90% in comparison to the sMAPE in the other experiments. This increase in the errors was also observed to be approximately twice the observed errors in Experiment 2.

Table 1. OP-ELM experiments results.

Experiment	Hidden Nodes	Performance		
		sMAPE	*MAE*	*RMSE*
Exp 1	81	0.208919	0.124393	0.157294
Exp 2	**50**	**0.204172**	**0.115727**	**0.146514**
Exp 3	50	0.231884	0.134929	0.173026
Exp 4	125	0.405396	0.198972	0.228686
Exp 5	18	0.519556	0.246778	0.27476

A statistical significance test was conducted to determine if the results with the lowest errors from each experiment had a significant difference from the results with the overall lowest errors. The statistical significance test results are captured in Table 2. From the significance test results, a p-value of less than 0.05 was observed. Thus, the results are significantly different from each other. The exclusion of the demand, therefore, increases model forecasting error.

Table 2. OP-ELM models' lowest errors statistical significance test.

	Exp 1	Exp 3	Exp 4	Exp 5
***p*-value**	0.020517	0.001810	2.0375×10^{-91}	7.4004×10^{-118}

5.2. LSTM-RNN Results

LSTM-RNN models were developed using the different variables per respective experiment. The performance of the different LSTM-RNN models was observed. The lowest obtained year-ahead UCLF forecast errors, per experiment, are captured in Table 3.

Table 3. LSTM-RNN experiments results.

Experiment	Hidden Units	Performance		
		sMAPE	*MAE*	*RMSE*
Exp 1	**511**	**0.15897**	**0.091421**	**0.114164**
Exp 2	64	0.173154	0.097143	0.117865
Exp 3	511	0.168273	0.09699	0.122862
Exp 4	256	0.343999	0.179548	0.214021
Exp 5	767	0.407081	0.206088	0.237777

A model with 511 hidden units and Experiment 1 variables had the lowest errors. Here, an sMAPE of 7.95%, MAE of 9.14%, and RMSE of 11.42% were achieved. Higher errors were observed in Experiments 4 and 5, where the installed capacity was excluded. These errors were approximately twice the errors in Experiment 1. A statistical significance test was conducted to determine if the results with the lowest errors in each experiment were significantly different from the results with the overall lowest errors. The results were found to be statistically different from each other as a *p*-value of less than 0.05 was observed in all four cases. The obtained *p*-values are captured in Table 4.

Table 4. LSTM-RNN models' lowest errors statistical significance test.

	Exp 2	Exp 3	Exp 4	Exp 5
***p*-value**	0.022794	9.3999×10^{-14}	9.2587×10^{-211}	7.8709×10^{-256}

5.3. DBN Results

The DBN models were developed as discussed in Section 3. The errors for the models' year-ahead UCLF forecast results were observed and the lowest obtained errors per experiment are captured in Table 5. A model with nine hidden nodes developed using all the variables was found to achieve the lowest errors, with an sMAPE of 9.74%, MAE of 11.52%, and RMSE of 13.74%. Experiments 4 and 5 showed an increase that was approximately three times the errors observed in Experiment 1.

The statistical significance test was conducted as described in Section 3.5 and the test result showed that the forecasting results were significantly different. Table 6 shows the statistical significance test results. The *p*-value can be seen to be less than 0.05 in each case, indicating a significant difference in the respective cases.

Table 5. DBN experiments results.

Experiment	Hidden Nodes	Performance		
		sMAPE	*MAE*	*RMSE*
Exp 1	**9**	**0.194736**	**0.115172**	**0.137397**
Exp 2	8	0.328704	0.172461	0.172461
Exp 3	8	0.300888	0.159492	0.189725
Exp 4	4	0.608786	0.279951	0.304046
Exp 5	4	0.588584	0.273245	0.298614

Table 6. DBN models' lowest errors statistical significance test.

	Exp 2	Exp 3	Exp 4	Exp 5
***p*-value**	7.3796×10^{-268}	1.7369×10^{-253}	9.7011×10^{-264}	3.9572×10^{-259}

5.4. Ensemble Results

Ensemble models of the three techniques were developed using the aggregate method with two individual developed models at a time; that is, from Equation (16), $n = 2$. All the individual models developed in this research were ensembled in this manner and their performance was observed. The performance parameters for the ensemble model whose year-ahead UCLF forecast achieved the lowest errors per experiment are presented in Table 7. Thus, not all results are included in Table 7, just the results with the lowest errors per experiment. The ensemble technique name is constructed by combining the name of the original technique used and the number of hidden nodes, for the OP-ELM and DBN, and the number of hidden units, for the LSTM, next to the name. The lowest obtained errors were achieved using an ensemble model of two LSTM models with 192 and 26 hidden units, respectively. This model achieved an sMAPE of 6.43%, MAE of 7.36%, and RMSE of 9.21%, which are bolded in Table 7. The respective errors in Experiments 4 and 5 were approximately twice the errors in Experiment 1. The accuracy of the model in Experiment 2 was higher than that for the models in Experiment 3. The models in Experiments 2 and 3 had lower accuracy than the model in Experiment 1, and higher accuracy than the models in Experiments 4 and 5.

Table 7. Ensemble experiments results.

Experiment	Ensemble Technique	Performance		
		sMAPE	*MAE*	*RMSE*
Exp 1	**LSTM192-LSTM26**	**0.1286794**	**0.073588**	**0.092046**
	LSTM192-DBN9	0.143504	0.080100	0.099055
	LSTM383-OPELM16	0.163263	0.093770	0.120741
	DBN9-DBN8	0.155915	0.089143	0.11129
	DBN9-OPELM16	0.168854	0.096859	0.122968
	OPELM81-OPELM16	0.198971	0.108824	0.140945
Exp 2	LSTM383-LSTM64	0.161214	0.092327	0.112328
	LSTM128-DBN8	0.170443	0.097098	0.124192
	LSTM64-OPELM50	0.167899	0.095670	0.118292
	DBN8-DBN8	0.328704	0.172461	0.205681
	DBN8-OPELM80	0.217165	0.118987	0.150817
	OPELM50-OPELM15	0.206886	0.114157	0.147775

Table 7. *Cont.*

Experiment	Ensemble Technique	Performance		
		sMAPE	*MAE*	*RMSE*
Exp 3	LSTM511-LSTM511	0.168272	0.096990	0.122861
	LSTM511-DBN8	0.210692	0.118462	0.148410
	LSTM511-OPELM50	0.185430	0.106055	0.130075
	DBN8-DBN8	0.300887	0.159491	0.189725
	DBN9-OPELM50	0.225704	0.123672	0.153060
	OPELM50-OPELM15	0.229431	0.127089	0.160191
Exp 4	LSTM256-LSTM256	0.343998	0.179547	0.214021
	LSTM256-DBN4	0.466254	0.228923	0.257488
	LSTM256-OPELM100	0.359248	0.184216	0.223038
	DBN4-DBN4	0.608786	0.279950	0.304045
	DBN4-OPELM100	0.480638	0.230885	0.263031
	OPELM100-OPELM125	0.392240	0.194914	0.232011
Exp 5	LSTM767-LSTM767	0.407080	0.206087	0.237776
	LSTM767-DBN4	0.4929245	0.239221	0.267542
	LSTM767-OPELM18	0.459199	0.225823	0.254908
	DBN4-DBN4	0.588584	0.273245	0.298614
	DBN4-OPELM18	0.551643	0.259683	0.285607
	OPELM18-OPELM18	0.519555	0.246777	0.274759

Table 8 presents the results for a statistical significance test conducted as discussed in Section 3.5. A *p*-value less than 0.05 was observed for each test conducted. This observation indicated that all the results being compared were significantly different from each other.

Table 8. Ensemble models' lowest errors statistical significance test.

	Exp 2	Exp 3	Exp 4	Exp 5
***p*-value**	7.3796×10^{-268}	1.7369×10^{-253}	9.7011×10^{-264}	3.9572×10^{-259}

5.5. Results Discussion

The lowest obtained year-ahead UCLF forecasting errors from each technique are summarized in Table 9. These results show that the lowest UCLF forecasting errors were obtained by the ensemble model. The ensemble model was then followed by the LSTM-RNN, DBN, and then OP-ELM. The two deep learning techniques, thus, achieved higher accuracies than the non-deep learning technique, OP-ELM. It was observed that with all techniques, apart from OP-ELM, the lowest errors were attained in Experiment 1. Experiments 4 and 5 showed a sharp increase in errors, relative to the rest of the experiments with all the techniques. Thus, the exclusion of the installed capacity as an input variable decreased the accuracy of the models of the techniques used. The plots of the target UCLF and the year-ahead forecasted UCLF for the models with the lowest errors per technique are presented in Figures 8–11. These plots are plotted for the period of 1 January 2019 to 31 December 2019. Each plot of the individual models also includes the ensemble model with the lowest forecasting error. The plots of the UCLF forecast by the models that make up the ensemble model are plotted in Figure 9.

Table 9. Summary of lowest obtained errors per used technique.

Technique	**Experiment**	**Performance**		
		sMAPE	***MAE***	***RMSE***
OP-ELM	Exp 2	0.204172	0.115727	0.146514
LSTM-RNN	Exp 1	0.15897	0.091421	0.114164
DBN	Exp 1	0.194736	0.115172	0.137397
Ensemble	**Exp 1**	**0.128679**	**0.073588**	**0.092046**

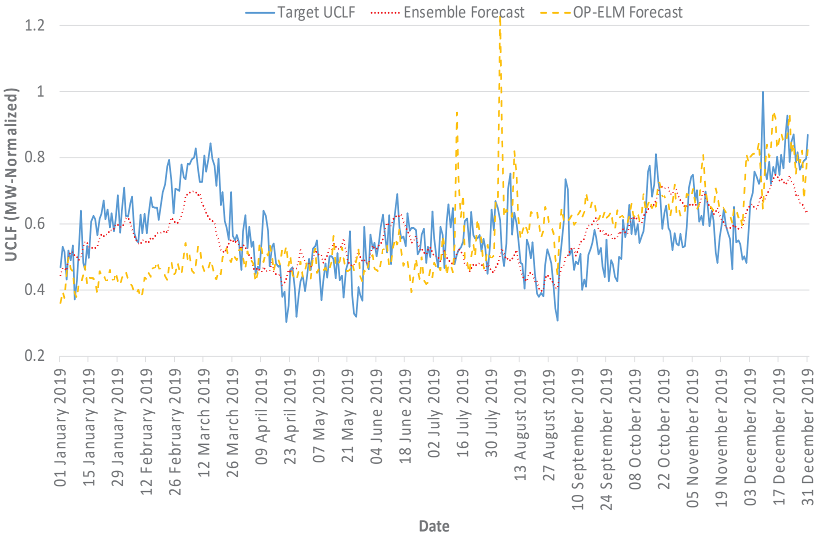


Figure 8. A plot of the OP-ELM and ensemble lowest error model year-ahead UCLF forecast against the target UCLF.

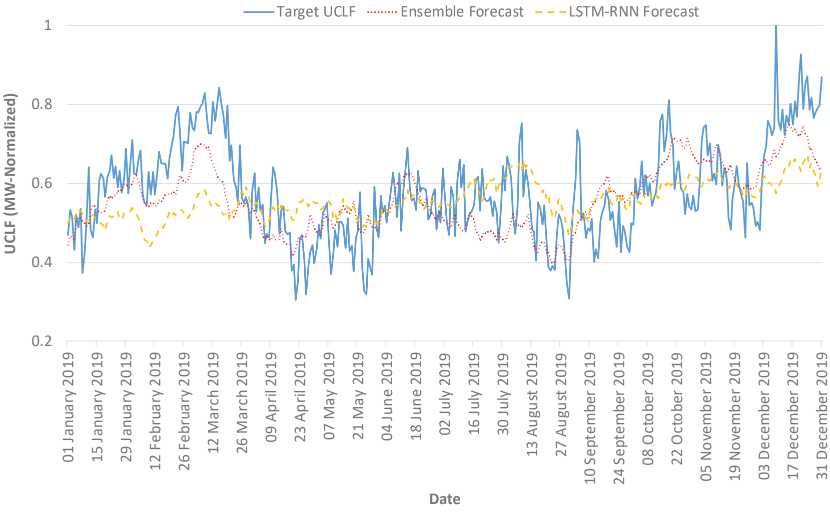


Figure 9. A plot of the LSTM-RNN and ensemble lowest error model year-ahead UCLF forecast against the target UCLF.

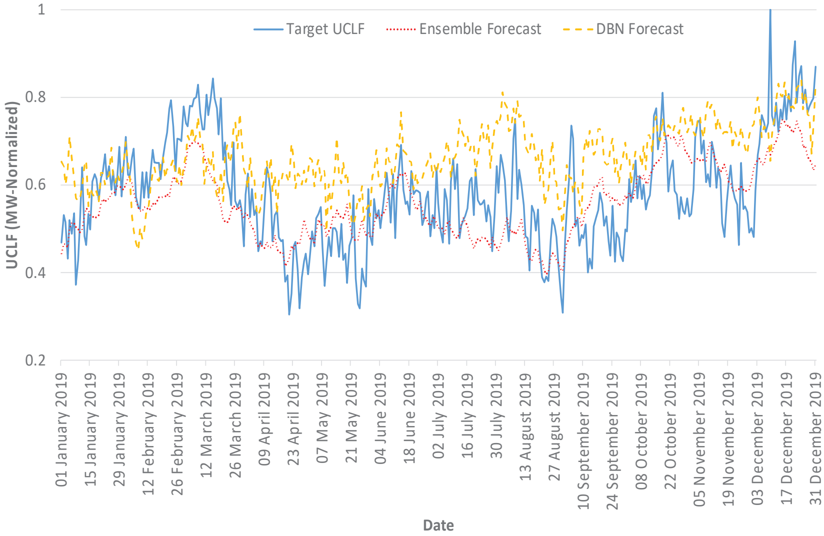


Figure 10. A plot of the DBN and ensemble lowest error model year-ahead UCLF forecast against the target UCLF.

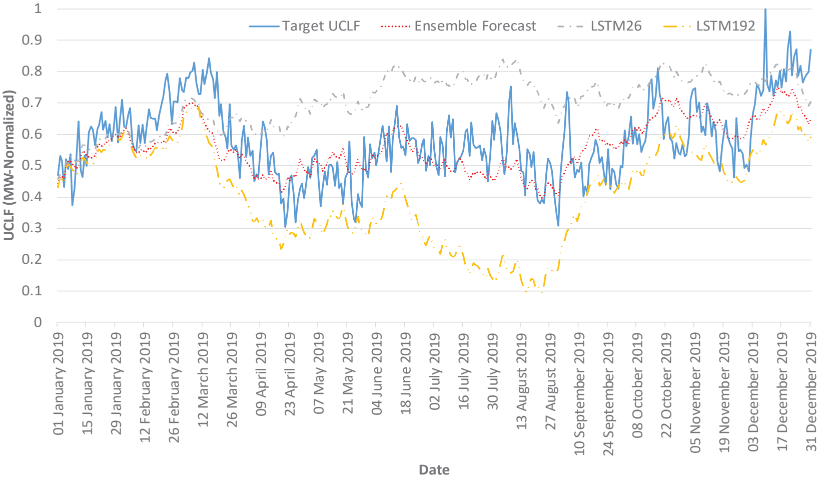


Figure 11. A plot of the ensemble lowest error model and the two aggregated models' year-ahead UCLF forecast against the target UCLF.

6. Conclusions

This paper contributed to the body of knowledge about South African UCLF forecasting. (i) A novel study of the South African UCLF behavior using state-of-the-art AI (deep learning and ensemble) techniques was presented. LSTM-RNN, DBN, OP-ELM, and ensembles of these three techniques' models were investigated in South African UCLF forecasting. (ii) An investigation of the impact of the installed capacity, historic demand, and PCLF on the UCLF forecasting accuracy was presented. It was found that the installed capacity had the biggest impact on the UCLF forecasting error, with the exclusion of this variable doubling the errors with the respective techniques used. (iii) A novel deep-learning ensemble total South African UCLF forecasting system was introduced. It was found that an ensemble of LSTM models achieved the lowest errors with an sMAPE of 6.43%, MAE of 7.36%, and RMSE of 9.21%. The lowest achieved LSTM model UCLF forecast errors were an sMAPE of 7.95%, MAE of 9.14%, and RMSE of 11.42%. The lowest achieved DBN model UCLF forecast errors were an sMAPE of 9.74%, MAE of 11.52%, and RMSE of 13.74%. The lowest achieved OP-ELM model UCLF forecast errors were an sMAPE of 10.21%, MAE of 11.57%, and RMSE of 14.65%. The lowest attained error was, thus, given by the ensemble model, followed by LSTM-RNN. The non-deep learning techniques' lowest achieved error was higher than that of the lowest errors achieved by the other techniques. Thus, ensemble deep learning techniques can be used to effectively forecast the total South African UCLF and, thus, load shedding.

7. Limitations of the Study and Future Work

This section presents the limitation of this study. As with most research, not all research-related aspects can be covered in a single study. As mentioned in Section 1, the study of South African UCLF behavior and UCLF forecasting is a new research area. This study does not focus on the speed of training the models, but rather on how well the models forecast the UCLF. Future work can include looking at the model training performance from the training speed perspective. The study forecast period is a year. This period was selected as it gives a wide enough window for the utility, at a daily resolution, to understand the UCLF behavior for the year. This understanding allows the utility company to plan over the year. The study does not research the performance of the models in shorter-term forecast windows, e.g., hourly, daily, weekly, etc. The performance of the models can, in the future, be studied for different forecast windows. Future research work should also consider looking at recent state-of-the-art techniques, such as temporal convolutional networks (TCN), gated recurrent units (GRU), and quasi-recurrent neural networks (QRNN). Given the performance of the equally weighted ensemble techniques in this paper, weighted ensemble techniques should be considered in future work. This future work can also investigate the ensemble models' performance when combining more than two models. Other benchmark techniques, such as naïve and multilayer perceptron, can be considered in future work.

Author Contributions: Conceptualization, S.M.; methodology, S.M.; software, S.M.; validation, S.M.; formal analysis, S.M.; investigation, S.M.; resources, S.M., A.N.H. and T.S.; data curation, S.M.; writing—original draft preparation, S.M.; writing—review and editing, S.M., A.N.H. and T.S.; visualization, S.M.; supervision, A.N.H. and T.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding. The authors would like to acknowledge the University of Johannesburg's Global Excellence and Stature (GES) 4.0 for funding of this research.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data is not available on any public platform.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Motepe, S.; Hasan, A.N.; Twala, B.; Stopforth, R.; Alajarmeh, N. South African Power Distribution Network Load Forecasting Using Hybrid AI Techniques: ANFIS and OP-ELM. In Proceedings of the Aegean Conference on Electrical Machines and Power Electronics, and Optimization of Electrical & Electronic Equipment Conference (ACEMP-OPTIM), Istanbul, Turkey, 2–4 September 2019.
2. Hasan, A.; Twala, B.; Ouahada, K.; Marwala, T. Energy Usage Optimisation in South African Mines. *Arch. Min. Sci.* **2014**, *59*, 53–69. [CrossRef]
3. Malatji, E.M.; Zhang, J.; Xia, X. A multiple objective optimisation model for building energy efficiency investment decision. *Energy Build.* **2013**, *61*, 81–87. [CrossRef]
4. Motepe, S.; Hassan, A.N.; Stopforth, R. South African Distribution Networks Load Forecasting Using ANFIS. In Proceedings of the IEEE Power Electronics Drivers and Energy Systems (PEDES), Chennai, India, 18–21 December 2018.

5. ESKOM and The Department of Public Enterprise. Update: ESKOM Electricity Supply. Available online: http://www.eskom.co.za/news/Documents/20190403ESKOM_BriefingFINAL.pdf (accessed on 1 April 2020).
6. Pombo-van Zyl, N. Warning: Stage 2 Loadshedding Returns States Eskom. ESI Africa, Africa's Power Journal, February 2020. Available online: https://www.esi-africa.com/industry-sectors/transmission-and-distribution/warning-high-risk-of-loadshedding-returns-states-eskom/ (accessed on 1 May 2020).
7. Pretorius, I.; Piketh, S.J.; Burger, R.P. The impact of the South African energy crisis on emissions. *WIT Trans. Ecol. Environ.* **2015**, *198*, 255–264.
8. Micali, V. Prediction of Availability for new power plant in the absence of data. In Proceedings of the 2012 9th Industrial and Commercial Use of Energy Conference, Stellenbosch, South Africa, 15–16 August 2012.
9. Motepe, S.; Hasan, A.N.; Stopforth, R. Improving Load Forecasting Process for a Power Distribution Network Using Hybrid AI and Deep Learning Algorithms. *IEEE Access* **2019**, *7*, 82584–82598. [CrossRef]
10. Jaech, A.; Zhang, B.; Ostendorf, M.; Kirschen, D.S. Real-Time Prediction of the Duration of Distribution System Outages. *IEEE Trans. Power Syst.* **2019**, *34*, 773–781. [CrossRef]
11. Wolse, H.; Geist, G.; Hoving, B.; Oosterlee, P.; Polman, H. Experience and tendencies after 40 years outage data registration in the Netherlands. *CIRED-Open Access Proc. J.* **2017**, *2017*, 2279–2282. [CrossRef]
12. BP. Statistical Review of World Energy. 69th Edition. 2020. Available online: https://www.bp.com/content/dam/bp/business-sites/en/global/corporate/pdfs/energy-economics/statistical-review/bp-stats-review-2020-full-report.pdf (accessed on 1 May 2020).
13. Index Mundi. Available online: https://www.indexmundi.com/g/r.aspx?t=50&v=79&l=en (accessed on 1 November 2020).
14. Smith, R.K. Analysis of hourly generation patterns at large coal-fired units and implications of transitioning from baseload to load-following electricity supplier. *J. Mod. Power Syst. Clean Energy* **2018**, *7*, 468–474. [CrossRef]
15. Fu, J.; Xiao, H.; Wang, H.; Zhou, J. Control Strategy for Denitrification Efficiency of Coal-Fired Power Plant Based on Deep Reinforcement Learning. *IEEE Access* **2020**, *8*, 65127–65136. [CrossRef]
16. Wang, Y.; Lou, S.; Wu, Y.; Wang, S. Flexible Operation of Retrofitted Coal-Fired Power Plants to Reduce Wind Curtailment Considering Thermal Energy Storage. *IEEE Trans. Power Syst.* **2019**, *35*, 1178–1187. [CrossRef]
17. Che, P.; Liu, Y.; Che, L.; Lang, J. Co-Optimization of Generation Self-Scheduling and Coal Supply for Coal-Fired Power Plants. *IEEE Access* **2020**, *8*, 110633–110642. [CrossRef]
18. Khoza, M.; Marwala, T. Computational intelligence techniques for modelling an economic system. In Proceedings of the 2012 International Joint Conference on Neural Networks (IJCNN), Brisbane, Australia, 10–15 June 2012.
19. Galias, Z. Probabilistic Model for Studying Blackouts in Power Networks. *IEEE J. Emerg. Sel. Top. Circuits Syst.* **2017**, *7*, 218–227. [CrossRef]
20. Attia, A. Analysis of failure in power cables for preventing power outage in Alexandria electricity distribution company in Egypt. *CIRED-Open Access Proc. J.* **2017**, *2017*, 20–24. [CrossRef]
21. Wang, J.; Zhang, J.; Wang, X. Bilateral LSTM: A Two-Dimensional Long Short-Term Memory Model with Multiply Memory Units for Short-Term Cycle Time Forecasting in Re-entrant Manufacturing Systems. *IEEE Trans. Ind. Inform.* **2017**, *14*, 748–758. [CrossRef]
22. Chen, X.-W.; Lin, X. Big Data Deep Learning: Challenges and Perspectives. *IEEE Access* **2014**, *2*, 514–525. [CrossRef]
23. Martín-Doña, J.M.; Gomezs, A.M.; Gonzalez, J.A.; Peinado, A.M. A deep learning loss function based on the perceptual evaluation of the speech quality. *IEEE Signal Process. Lett.* **2018**, *25*, 1680–1684. [CrossRef]
24. Masita, K.L.; Hasan, A.N.; Paul, S. Pedestrian Detection Using R-CNN Object Detector. In Proceedings of the IEEE Latin American Conference on Computational Intelligence (LA-CCI), Gudalajara, Mexico, 7–9 November 2018.
25. Alhussein, M.; Aurangzeb, K.; Haider, S.I. Hybrid CNN-LSTM Model for Short-Term Individual Household Load Forecasting. *IEEE Access* **2020**, *8*, 180544–180557. [CrossRef]
26. Kong, Z.; Zhang, C.; Lv, H.; Xiong, F.; Fu, Z. Multimodal Feature Extraction and Fusion Deep Neural Networks for Short-Term Load Forecasting. *IEEE Access* **2020**, *8*, 185373–185383. [CrossRef]
27. Pandit, R.K.; Kolios, A.; Infield, D. Data-driven weather forecasting models performance comparison for improving offshore wind turbine availability and maintenance. *IET Renew. Power Gener.* **2020**, *14*, 2386–2394. [CrossRef]
28. Kou, P.; Wang, C.; Liang, D.; Cheng, S.; Gao, L. Deep learning approach for wind speed forecasts at turbine locations in a wind farm. *IET Renew. Power Gener.* **2020**, *14*, 2416–2428. [CrossRef]
29. Munkhdalai, L.; Park, K.H.; Batbaatar, E.; Theera-Umpon, N.; Ryu, K.H. Deep Learning-Based Demand Forecasting for Korean Postal Delivery Service. *IEEE Access* **2020**, *8*, 188135–188145. [CrossRef]
30. Motepe, S.; Hasan, A.N.; Twala, B.; Stopforth, R. Effective load forecasting for large power consuming industrial customers using long short-term memory recurrent neural networks. *J. Intell. Fuzzy Syst.* **2019**, *37*, 8219–8235. [CrossRef]
31. Han, L.; Peng, Y.; Li, Y.; Yong, B.; Zhou, Q.; Shu, L. Enhanced Deep Networks for Short-Term and Medium-Term Load Forecasting. *IEEE Access* **2019**, *7*, 4045–4055. [CrossRef]
32. Ramotsoela, T.D.; Hancke, G.P.; Abu-Mahfouz, A.M. Behavioural Intrusion Detection in Water Distribution Systems Using Neural Networks. *IEEE Access* **2020**, *8*, 190403–190416. [CrossRef]
33. Zhang, C.; Li, R.; Kim, W.; Yoon, D.; Patras, P. Driver Behavior Recognition via Interwoven Deep Convolutional Neural Nets with Multi-Stream Inputs. *IEEE Access* **2020**, *8*, 191138–191151. [CrossRef]

34. Sun, Q.; Ge, Z. Deep Learning for Industrial KPI Prediction: When Ensemble Learning Meets Semi-Supervised Data. *IEEE Trans. Ind. Inform.* **2021**, *17*, 260–269. [CrossRef]
35. Drif, A.; Zerrad, H.E.; Cherifi, H. EnsVAE: Ensemble Variational Autoencoders for Recommendations. *IEEE Access* **2020**, *8*, 188335–188351. [CrossRef]
36. Bibi, N.; Shah, I.; Alsubie, A.; Ali, S.; Lone, S.A. Electricity Spot Prices Forecasting Based on Ensemble Learning. *IEEE Access* **2021**, *9*, 150984–150992. [CrossRef]
37. Shah, I.; Iftikhar, H.; Ali, S.; Wang, D. Short-Term Electricity Demand Forecasting Using ComponentsEstimation Technique. *Energies* **2019**, *12*, 2532. [CrossRef]
38. Miche, Y.; Sorjamaa, A.; Lendasse, A. OP-ELM: Theory, Experiments and a Toolbox. In Proceedings of the 18th International Conference on Artificial Neural Networks (ICANN), Prague, Czech Republic, 3–6 September 2008.
39. Motepe, S.; Hasan, A.N.; Twala, B.; Stopforth, R. Power Distribution Networks Load Forecasting Using Deep Belief Networks: The South African Case. In Proceedings of the IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology, Amman, Jordan, 9–11 April 2019.
40. ESKOM. Power station GPS Coordinates. Eskom. Available online: http://www.eskom.co.za/Whatweredoing/ElectricityGeneration/PowerStations/Pages/Power_Station_GPS_Coordinates.aspx (accessed on 6 November 2020).

energies

MDPI

Article

Machine Learning Algorithms for Flow Pattern Classification in Pulsating Heat Pipes

Jose Loyola-Fuentes [1], **Luca Pietrasanta** [2], **Marco Marengo** [2] **and Francesco Coletti** [1,3,*]

[1] Hexxcell Ltd., Foundry Building, 77 Fulham Palace Rd, London W6 8AF, UK; j.loyola@hexxcell.com
[2] Advanced Engineering Centre, School of Architecture, Technology and Engineering, University of Brighton, Lewes Rd, Brighton BN2 4AT, UK; l.pietrasanta2@brighton.ac.uk (L.P.); m.marengo@brighton.ac.uk (M.M.)
[3] Department of Chemical Engineering, Brunel University London, Kingston Lane, Uxbridge UB8 3PH, UK
* Correspondence: f.coletti@hexxcell.com

Abstract: Owing to their simple construction, cost effectiveness, and high thermal efficiency, pulsating heat pipes (PHPs) are growing in popularity as cooling devices for electronic equipment. While PHPs can be very resilient as passive cooling systems, their operation relies on the establishment and persistence of slug/plug flow as the dominant flow regime. It is, therefore, paramount to predict the flow regime accurately as a function of various operating parameters and design geometry. Flow pattern maps that capture flow regimes as a function of nondimensional numbers (e.g., Froude, Weber, and Bond numbers) have been proposed in the literature. However, the prediction of flow patterns based on deterministic models is a challenging task that relies on the ability of explaining the very complex underlying phenomena or the ability to measure parameters, such as the bubble acceleration, which are very difficult to know beforehand. In contrast, machine learning algorithms require limited a priori knowledge of the system and offer an alternative approach for classifying flow regimes. In this work, experimental data collected for two working fluids (ethanol and FC-72) in a PHP at different gravity and power input levels, were used to train three different classification algorithms (namely K-nearest neighbors, random forest, and multilayer perceptron). The data were previously labeled via visual classification using the experimental results. A comparison of the resulting classification accuracy was carried out via confusion matrices and calculation of accuracy scores. The algorithm presenting the highest classification performance was selected for the development of a flow pattern map, which accurately indicated the flow pattern transition boundaries between slug/plug and annular flows. Results indicate that, once experimental data are available, the proposed machine learning approach could help in reducing the uncertainty in the classification of flow patterns and improve the predictions of the flow regimes.

Keywords: two-phase flow; pulsating heat pipes; flow pattern maps; machine learning; classification algorithms

Citation: Loyola-Fuentes, J.; Pietrasanta, L.; Marengo, M.; Coletti, F. Machine Learning Algorithms for Flow Pattern Classification in Pulsating Heat Pipes. *Energies* **2022**, *15*, 1970. https://doi.org/10.3390/en15061970

Academic Editors: Luis Hernández-Callejo, Sergio Nesmachnow, Sara Gallardo Saavedra and Dmitry Eskin

Received: 9 February 2022
Accepted: 4 March 2022
Published: 8 March 2022

1. Introduction

The lifespan and reliability of a wide range of electronic components and electro-mechanical assemblies are often compromised by the poor performance of the thermal control system (TCS). Cooling capacity, weight, and cost requirements are becoming very challenging in high-density PCBs, microprocessors, photovoltaic solar arrays, and actuators, not only limiting the expected performance [1] but also creating safety issues, as in EV battery systems [2]. On the other hand, energy consumption for cooling purposes has critically increased in recent years. Data centers consume 200 TWh each year worldwide [3], where 38% (76 TWh) is estimated to go toward cooling processes. There are a wide variety of available cooling processes for electronics. The most common methods based on two-phase flow are flow boiling [4–10], pool boiling [11–14], and impinging jets [15–18].

Pulsating heat pipes (PHPs) can play a leading role in reducing cooling costs due to their resulting equivalent thermal conductivity that is several times higher than that of pure copper [19]. Furthermore, no pumping power is required for the circulation of the working fluid. This results in a sensible reduction in complexity, volume, and weight of the TCS. The PHP is a thermally driven heat transfer device patented in the 1990s [20,21], which has seen a growing research interest since then. It is simpler in its construction and more cost-effective, compared to other similar heat transfer devices (e.g., heat pipes, loop heat pipes). It is composed either of a tube bent in several turns or of two plates welded with a serpentine-like path milled on one of the surfaces. Once filled and sealed, a working fluid resides in the PHP as an alternation of liquid slugs and vapor plugs due to the dominant effect of capillary forces with respect to buoyancy. When heat is applied to the evaporator zone, the fluid motion inside the tube is activated, and the pressure fluctuations drive a self-excited [22] oscillating motion of liquid plugs and vapor bubbles, also identified as oscillating Taylor flow. This condition significantly enhances the heat transfer [23] by exploiting both sensible and latent heat.

1.1. Flow Patterns in PHPs

Whilst PHPs are drawing the attention of a growing number of research groups, including both experimental and numerical approaches, the industrialization of such technology is still in its preliminary phase, and examples of off-the-shelf PHPs are not yet common and limited to specific applications. The complex interplay of evaporation/condensation phenomena, surface tension, and inertial effects has been the object of several numerical investigations with the aim of developing a robust modeling tool. Nikolayev [24] developed one of the first models able to describe the chaotic self-sustained oscillations in a PHP with an arbitrary number of branches and arbitrary number of bubbles. Further improvements of the same model led to the implementation of the effect of the tube conductivity on the start-up phase [25] and the impact of the PHP orientation on the overall performance [26].

The operation of a PHP is strongly linked to the existence of a dominant slug/plug flow throughout the required range of operating conditions. Due to the variation of flow direction, pressure drop, and liquid film thickness in a PHP, several flow patterns have been observed [27], showing transitions between slug/plug, semi-annular, and annular flow (Figure 1). For a given geometry, the flow pattern is highly influenced by filling ratio and power input [28], due to the effect on the vapor quality, showing a higher ratio of bubble length over tube diameter [29]. As a result, the slug/plug flow pattern can transition into an annular flow, which in the long run can lead to a reduction in thermal performance and a stoppage of the oscillation due to critical drying out of the evaporator. The flow pattern has been extensively investigated in flow boiling in millimeter-scale channels, and it is the result of the interaction of interfacial, inertial, viscous, and gravitational forces. Without an exhaustive knowledge of the flow pattern, the correct thermal and hydraulic design parameters cannot be calculated properly. Despite the crucial role played, the majority of the available flow boiling pressure drop correlations have been formulated without reference to the flow pattern condition they covered [30]. It is also known that the available heat transfer correlations are very sensitive to the flow pattern condition [31]. Frequently, the expected flow pattern is roughly linked to the dimension of the channel. A rough classification proposed by Kandlikar [32] fixed 3 mm as the transition limit between macro-channels and micro-channels, not considering fluid properties, inertial effects, and gravity levels. In varying gravity conditions, a transition from a thermosyphon mode (semi-annular dominant) to PHP mode (slug/plug dominant) impacts the thermal performance, operating range, and start-up power [33].

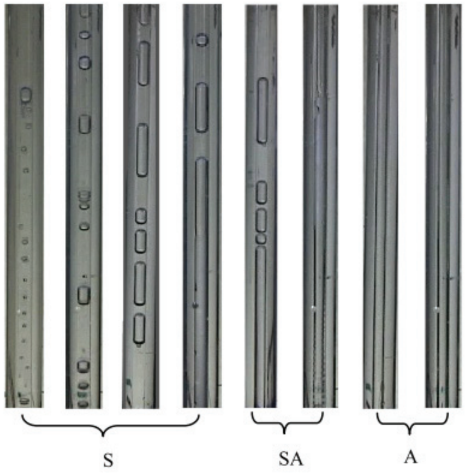


Figure 1. Flow patterns observed in the adiabatic section of a 1.6 mm ID PHP filled with ethanol during a series of experiments [9]. S: slug/plug; SA: semi-annular; A: annular.

One of the main limitations of the available numerical tools is the inability to define the dominant flow pattern given a set of operating conditions. The flow pattern is assumed a priori, and only static criteria are considered. Bond number ($Bo = \rho g d^2/\sigma$), or its relevant form considering the wettability through the contact angle $(\theta < 90^\circ)$ $(Bo = \rho g d^2/(\sigma cos\theta))$, and the confinement number ($Co = 1/Bo^{1/2}$) are implemented to establish whether the initial existence condition for slug/plug flow are met. Once the motion is activated, there is no real control of the transitions of the flow pattern, mainly ignoring inertial effects. Break-ups and coalescence events were reviewed in a numerical investigation from Andredaki et al. [34]. An approach to the development of flow pattern maps for oscillating flows based on dimensionless numbers was proposed by Pietrasanta et al. [29], drawing the attention to break-up and coalescence phenomena in a simplified PHP loop, suggesting the effective use of the actual bubble acceleration rather than the static, nominal g value (i.e., gravitational acceleration) and the actual bubble velocity to describe the transition between slug/plug and semi-annular flow. This last methodology, even if much more accurate than the use of Bond number and other dimensionless numbers such as Weber or Reynolds number, has the disadvantage that it cannot be used for design purposes, but only for a posteriori validation of numerical codes.

Therefore, despite the great effort shown so far, the development of comprehensive design tools, validated over a wide range of operating conditions and able to assist thermal engineers, is not yet complete.

1.2. Machine Learning Algorithms for Two-Phase Flow Heat Transfer

Machine learning is a rapidly growing field that allows data-driven optimization, and it has been recently extended to flow identification and design of cooling devices at different scales, considering the most significant design parameters as inputs and flow regimes or thermal resistance as outputs, depending on the application. The use of these algorithms is predominant in regression problems for heat transfer coefficients and pressure drop, although the classification of flow patterns can still be found in the available literature.

Several challenges related to two-phase flow heat transfer have been addressed via the use of machine learning techniques. The prediction of flow patterns using support vector machines (SVMs) was proposed by Guillén-Rondon et al. [35]. Here, the authors trained an SVM with a large two-phase flow pattern dataset and achieved on average 95% prediction accuracy when testing the algorithm on different groups and combinations of flow patterns. Another interesting contribution is within flow boiling and condensation heat transfer. The prediction of heat transfer coefficients for both phase changes is a challenging task, and the use of machine learning has proven to be beneficial for facilitating these estimations. An example is the work by Zhu et al. [36] which proposed the use artificial neural networks (ANNs) to predict flow boiling and condensation heat transfer coefficients for micro-channel systems with serrated fins. The authors were able to identify the most relevant geometrical and operational parameters to minimize the prediction error and evaluate the influence of specific operational parameters such as mass and heat flux into the prediction accuracy of the ANN. The results were promising, showing that the relative deviation from experimental data was on average 11.4% and 6.10% for flow boiling and condensation, respectively. The use of ANN is also useful in image recognition and analysis. A recent study published by Suh et al. [37] established an automated framework for determining boiling curves from high-quality bubble images using convolutional neural networks (CNNs). The image analysis performed by the neural network was able to capture relevant physical features used for its training and learning of the underlying statistics between bubble dynamics and corresponding boiling curves. The prediction error was reported to be 6% on average.

In terms of the identification of flow patterns in PHP systems, few attempts were found. Most efforts focused on common heat pipes and two-phase systems. Hernandez et al. [38] developed a decision-tree-based classifier to identify flow regimes and select appropriate predictive models for several two-phase flow systems. Zhang et al. [39] proposed two different machine learning classification algorithms for two-phase nuclear systems. The first one was designed for real-time flow regime identification based on SVMs, and the second classifier was designed for transient flow regime classification using CNNs. Both classifiers performed with high accuracy, allowing for a fast response when dealing with complex two-phase systems. Note that the above-mentioned contributions are related to two-phase flow systems, where no pulsating phenomenon occurs, and the transition from one flow regime to another may be less rapid than when the flowrate and its direction are not controlled (as it is the case with PHP systems).

In the context of PHP devices, most research attempts have dedicated their efforts to the prediction of key design parameters, such as thermal resistance and pressure drop. Jokar et al. [40] presented a novel approach for simulation and optimization of PHPs, based on a multilayer perceptron (MLP) neural network. According to the authors, PHPs, as a complex system, can be successfully simulated by means of artificial neural networks. Jalilian et al. [41] extended the study to the optimization of a flat plate PHP for application in a solar collector. The trained network was validated with experimental data and used to evaluate the objective function to maximize the thermal efficiency of the system. A comprehensive discussion of the thermal performance prediction of PHPs based on an artificial neural network (ANN) and regression/correlation analysis (RCA) was proposed by Patel and Metha [42]. The authors investigated the influence of nine major input variables, considering more than 1600 experimental points from the literature. Wang et al. [43] proposed a similar predicting model based on ANN for the optimization of the effects of different working fluids, extending the current state-of-the-art approaches. Table 1 summarizes the main input parameters and machine learning approaches adopted in the abovementioned work. Note that the implementation of these machine learning algorithms is rather recent, indicating that there are still further studies to perform, although promising results have been obtained.

Table 1. Relevant studies on machine learning applied to PHPs.

Author	Year	Model Description	Input Parameters	Output	Prediction Accuracy
Jokar et al. [40]	2016	Combination of ANN and GA 2 hidden layers (50 and 40 neurons) Batch learning method 70% of data used for training	Heat flux Inclination angle Filling ratio	Equivalent thermal resistance	Relative errors of 5–12%
Jalilian et al. [41]	2016	ANN to describe behaviour of PHP in solar collectors GA for optimizing design parameters of solar collector	Solar radiation PHP evaporator length Filling ratio Water tank temperature Inclination angle	Heat gained by collector	Root-mean-square error between 7% and 13%
Patel and Mehta [42]	2018	ANN as a prediction model RCA to find correlation among input and outputs Data collected from 2003 to 2017	Geometrical parameters Working fluids Operational parameters	Thermal resistance	Correlation coefficient of 0.89 for ANN and 0.95 for RCA with dimensionless numbers
Wang et al. [43]	2019	General model for varied working fluids and conditions Use of ANN for prediction Evaporation and condensation temperature estimated from model	Dimensionless numbers related to heat transfer and system geometry Ratio of evaporation length and diameter	Thermal resistance	Mean square error of 0.014 and correlation coefficient of 0.98

On the basis of the findings shown in Table 1, there is still a need for understanding the complex phenomenon of flow regime transition in PHP systems, and for the classification of the flow pattern when the device is in operation. The capability of identifying the flow regime for a set of operating conditions allows for a more accurate prediction of design parameters and for useful insights regarding the behavior of the system during operation. Within this context, the use of machine learning is beneficial, as it leverages the abundance of significant sets of data. The advantages of machine learning techniques, namely, the direct use of data, the variety of methods for specific purposes, and their equation-free nature, provide unique characteristics that can improve the optimization of experiment design, speed in experimental analysis, and scaling to different scenarios.

This work proposes, for the first time, the use of machine learning classifiers to identify flow patterns and flow pattern transition in a single-loop PHP system with two different working fluids and in varying gravity conditions using data from the European Space Agency Parabolic Flight Campaigns [11,25]. Since the single-loop PHP allows the visualization of flow patterns, this makes the present analysis unique in understanding if ML can be successfully trained to recognize PHP flow patterns. The selection of the most suitable classifier is carried out by comparing the accuracies of such classifiers when predicting the flow regime on unseen data (or testing sets). The selected classifier is used for devising flow pattern maps for both working fluids, to identify the location of the flow regime transition zone. It is expected that this capability provides a more systematic approach when identifying flow regimes, reducing observation uncertainty (when used).

2. Methodology

This work was carried out in two stages. First, the experiments were performed, where the data used for the machine learning implementation were generated. Second, these data were preprocessed and prepared for the deployment of machine learning tests and analysis. Velocity measurements were used to estimate acceleration, as described in the work done by Pietrasanta et al. [44]. The length of bubbles was also measured. Pressure measurements also took place in both thermal terminals of the device (i.e., condenser and evaporator). These measurements were used to estimate physical properties for the calculation of dimensionless numbers such as Reynolds (Re), Weber (We), Froude (Fr), and Bond (Bo) numbers, as defined in Pietrasanta et al. [29]. The labeling process was conducted visually while analyzing the high-speed images. These values, along with the label for each observation, were used to train the machine learning algorithms. Note that the entire set of data was split into a training and testing subsets. Cross-validation within the training set was also implemented for hyperparameter selection.

2.1. Experimental Setup

The experimental campaign was conducted on a simplified passive heat transfer loop under varying gravity level and power inputs; ethanol and FC-72 were selected as the working fluid, mainly due to their significant differences in surface tension, density, and latent heat of vaporization. The main fluid properties are listed in Table 2. The varying gravity level was obtained via access to the ESA parabolic flight microgravity platform [45]. The main controlled and observed experimental parameters selected for the setup are detailed in Table 3.

Table 2. Main properties of the two working fluids at 20 °C.

Fluid	σ (N/m)	ρ (kg/m^3)	$h_{l,v}$ (kJ/kg)	μ (Pa·s)	d$_{cr}$ (mm)
Ethanol	0.0224	789.59	927.57	1.22×10^{-3}	3.40
FC-72	0.0118	1701.6	94.024	0.72×10^{-3}	1.69

Table 3. Experimental matrix with controlled parameters and parameters observed.

Controlled Parameters	Value
Working fluid	Ethanol, FC-72
Diameter	2 mm
Gravity level	~10^{-2} g, ~1 g, ~2 g
Total power input (W)	9, 15, 18, 24
Observed parameters	**Range**
Wall temperature (°C)	from 20 to 43
Heat flux (W/cm^2)	from 6.5 to 13.6
Absolute fluid velocity (m/s)	from 0 to 0.6
Absolute fluid acceleration	from 0 to 20 g

The device can be defined as a hybrid pulsating heat pipe/closed loop thermosyphon depending on the working fluid used and on the gravity level. The setup is equipped with wall-side thermocouples, glass tubes for high-speed shadowgraph visualization of the flow pattern, and pressure transducers, and the power input is supplied via three heaters coiled around three sections of the evaporator. The temperature at the condenser is kept constant with an external cooling loop. A detailed description of the experimental apparatus is provided in [29] and a rendering of the experimental setup is depicted in Figure 2. As discussed in [29], the threshold between confined and unconfined flow is conventionally defined through Bo or Co numbers. In both cases, the limit between stratified or unstratified displacement of the fluid (observable if the tube is in a horizontal position), is a function of the diameter, the surface tension, the density of the phases, and the gravity acceleration. When the gravity acceleration is reduced, the confinement conditions (or unstratified displacement of the fluid) are easy to reach. This is due to a change in the hierarchy of the forces acting on the fluid, where capillarity becomes dominant over gravitational forces. The opposite behavior is observed under hyper gravity conditions.

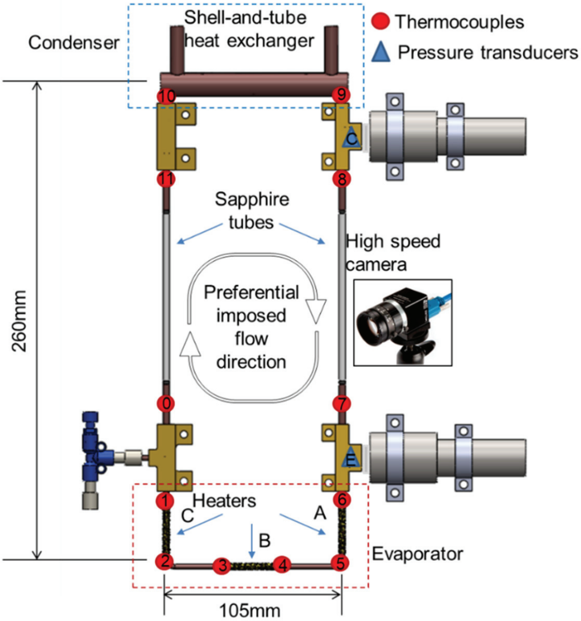


Figure 2. Rendering of the single-loop PHP with position of sensors and camera [45]. Reproduced with permissions.

2.2. Classification Algorithms

Classification is a type of supervised learning. Here, a set of relevant features is associated with a set of categories, which are already labelled (this makes classification a supervised method). This allows for a confident training of the classification model and, later, accurate predictions. When classifying features, different approaches can be implemented, and the specific method for relating features and labels varies from algorithm to algorithm. Hence, evaluating the performance of such algorithms is of great importance, given the context. In this work, three different classification algorithms (namely, K-nearest neighbors, multilayer perceptron, and random forest) are tested and compared. The selection of these methods was based on the fact that each of them presents distinct features that make them unique. This provides a suitable path to cover a wide range of alternatives when classifying an unknown set of features.

The K-nearest neighbors algorithm is a distance-based method, where each data point is put to the test and the distance between such point and its K-nearest neighbors is saved and later compared. Note that no training stage is strictly needed.

The second algorithm, namely, the multilayer perceptron, is an artificial neural network that minimizes a cost function, which allows for accurate predictions once the classification problem is properly trained. The minimization of the cost function can be achieved through a variety of methods, where backpropagation and gradient descent algorithms are popular and accurate choices.

Lastly, the random forest algorithm is an ensemble of decision trees. In this case, predictions are made on the basis of the training of multiple classifiers, and a final prediction takes place via the most repeated forecast of such classifiers. This is applied to improve the robustness of the classification model.

The accuracy of each algorithm is evaluated using the accuracy score function, given in Equation (1).

$$score = \frac{1}{n}\sum_{i=1}^{n-1} 1(\hat{y}_i = y_i), \tag{1}$$

where n is the number of samples, $\hat{y}_i$ is the predicted categorical value, y_i is the true categorical value, and function $1(x)$ is the indicator function, which outputs 1 when $\hat{y}_i = y_i$ and 0 otherwise. A brief description of the algorithms considered in this work is presented below. In addition, a summary of the functionality of each algorithm is depicted in Figure 3, where the logic behind each method is shown via block diagrams. Note that these diagrams are only general, as details regarding the architecture and specific parameters of each algorithm depend on the final configuration of each method, which depends on further studies that decide the suitable values of parameters.

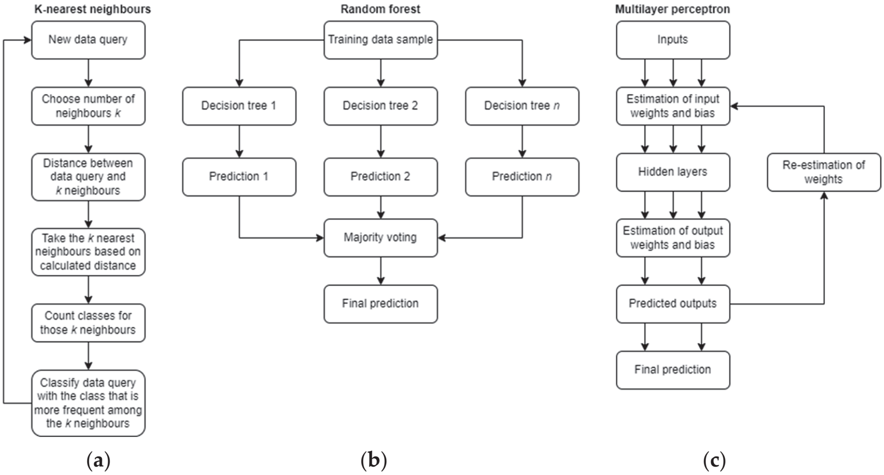

K-nearest neighbours
New data query
Choose number of neighbours *k*
Distance between data query and *k* neighbours
Take the *k* nearest neighbours based on calculated distance
Count classes for those *k* neighbours
Classify data query with the class that is more frequent among the *k* neighbours
(a)
Random forest
Training data sample
Decision tree 1
Decision tree 2
Decision tree *n*
Prediction 1
Prediction 2
Prediction *n*
Majority voting
Final prediction
(b)
Multilayer perceptron
Inputs
Estimation of input weights and bias
Hidden layers
Re-estimation of weights
Estimation of output weights and bias
Predicted outputs
Final prediction
(c)

Figure 3. Schematics of selected classification algorithms. (**a**): K-nearest neighbors, (**b**): random forest, (**c**): multilayer perceptron.

2.2.1. K-Nearest Neighbors

This classification algorithm is fundamentally simple but exhibits relative high performance. The underlying intuition is based on classifying the information from specific features from the categories of its closest neighbors [46]. The number of neighbors (k) can be user-defined or it can vary depending on the local density of the neighborhood. Likewise, to quantify the proximity of such neighbors, different measures of distance can be used, such as Euclidean or Manhattan distance. The distance from a close neighbor can also be weighted so that it provides a higher influence than a farther one.

Major limitations of this algorithm are its lack of performance when dealing with high-dimensional data and its high prediction times for large datasets. The reader is referred to [47] for a deeper description of this algorithm.

2.2.2. Random Forest

The random forest algorithm is based on decision trees, where the data are split into different branches that are created on the basis of specific data subsets. Random forest consists of creating multiple decision trees and randomizing the set of features these trees are fed into [48]. This approach is a trademark for what is known as ensemble learning. The response of each tree is then compared, and, in the case of classification, the mode of the outputs is considered as the categorical prediction. The diverse nature of the random forest algorithm, i.e., the use of multiple classifiers to find a robust prediction, allows for low-variance responses, which is a desired characteristic in any machine learning method [49]. The predictions are also expected to be unbiased.

A particularity of this method is the identification and ranking of the most relevant features in the datasets with respect to the categorical responses. This can be useful as a complement for the study of the effect of single features on the output response.

The main disadvantage of this algorithm is the large computational time required for implementation, which increases with the number of trees to build (defined by the user). More details regarding the algorithm can be found in [50].

2.2.3. Multilayer Perceptron

A multilayer perceptron (MLP) is a type of artificial neural network. It consists of an input layer that receives the data, a set of hidden layers that process the data, and an output layer that contains the response of the classification [49]. The network is trained via backpropagation, which is an optimization technique where a cost function (related to the difference between predictions and true values) is minimized. The function learned by the neural network consists of the linear combination of a set of two parameters, namely, weights and biases. The use of this method allows for flexibility, as linear and nonlinear systems can be fitted to the network and in cases where online predictions are needed.

Major drawbacks of this algorithm are its strong dependence on hyperparameters (i.e., number of neurons, number of hidden layers, etc.) and the presence of local minima when using hidden layers. This means that, when more hidden layers are used to increase accuracy, there is a major risk of deviating from a global optimal solution. A deeper description and the advantages of this algorithm can be found in [51].

3. Results and Discussion

Three different classifiers were built for each of the working fluids (ethanol and FC-72) using the experimental data. These data comprise 9841 observations for ethanol and 8590 observations for FC-72. The input features considered for all classifiers and both working fluids are the modified versions of Weber, Froude, and Bond numbers, represented by We_l^*, Fr_l^*, and Bo_l^*, respectively. These numbers were defined using actual bubble lengths, velocities, and accelerations, estimated via specific image analysis methods. More details regarding the definition of each number and data pre-processing can be found in Pietrasanta et al. [29]. The categorical output data indicate whether a specific observation is classified as slug/plug flow or semi-annular flow, and it was conducted visually. For each working fluid and classifier, the steps described below were carried out.

3.1. Data Splitting

Datasets were randomly split into training and testing sets. This was applied to avoid using entire datasets for training stages, as this could lead to overfitting. The proportion of data used in the training stage was fixed to 70%. This proportion of data splitting is commonly used, along with similar splitting ratios such as 80% or 67%. There is no optimal splitting ratio in machine learning applications (in general), and the decision is based on the original datasets. In this work, the datasets for both working fluids were large enough to perform the selected data split, leading to the values presented in Table 4.

Table 4. Split of data samples for ethanol and FC-722.

Data Sample	Total Data Points	Training Set Data Points	Testing Set Data Points
Ethanol	9841	6888	2953
FC-72	8590	6013	2577

3.2. Data Scaling

The values of the input features in the training and testing set were scaled (i.e., normalized) to avoid issues from different orders of magnitude among feature values. This was achieved by estimating the expected value and standard deviation of the training and testing sets and applying the normalization formula shown in Equation (2), where z_i is the normalized data point, x_i is the original data point, $\overline{\mu}$ is the sample's mean or expected value, and *sd* is the sample's standard deviation. The result from this normalization procedure is a transformed dataset that presents an expected value of 0 and a standard deviation of 1.

$$z_i = \frac{x_i - \overline{\mu}}{sd}. \tag{2}$$

3.3. Classifier Creation

The training set was used to train the classifier and then test it with the testing set. The accuracy score was estimated and stored. At this point, default values for each algorithm's parameters were used. The selection of the most suitable set of parameters for each method was completed later (see Section 3.6).

3.4. Cross-Validation

The training set was further used in cross-validation. This method provides a more general indication of the classification performance. In this work, cross-validation was implemented via the so-called *k*-fold cross-validation method. Here, the training set was further split into *k* subsets, which were smaller than the training set. This was followed by subsequent trainings of the classification algorithm using $k - 1$ subsets as the training set, while the remaining data were set aside for testing. The accuracy score was then estimated for all folds and averaged to get a representation of the overall performance of the classifier. This allowed for a higher training/testing split, as a validation set is not necessary when using cross-validation. A schematic of the *k*-fold method is depicted in Figure 4.

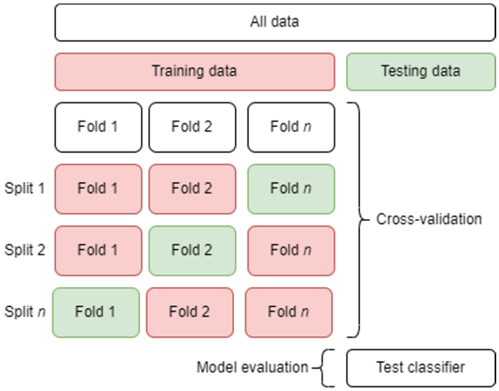


Figure 4. The *k*-fold cross-validation procedure with *n* folds.

3.5. Accuracy Assessment

The accuracy score of the created classifier and the mean score from the cross-validation procedure were compared to assess the general performance of the classifier. This comparison was only for understanding the robustness of the initial classifier. Normally, large differences are expected, leading to the conclusion that the initial set of parameters for each classifier (among other factors such as split ratio, amount of total data, and/or number of input features) should be adjusted.

3.6. Selection of Hyperparameters

To increase classification performance, and to select the most suitable set of parameters for a fair comparison (i.e., comparing only classifiers presenting maximum accuracy), a deep analysis was carried out. This is achieved using a grid search. This procedure consisted of choosing a combination of various values for specific parameters within a classifier and exhaustively performing a cross-validation for each combination of parameters. The accuracy score for each of these combinations was stored for comparison, and the set of parameters with the best (maximum in this case) output value of accuracy score was chosen.

The updated classifier was then tested using the testing set, and a prediction (testing) accuracy score was stored. Note that, in order to apply this procedure to each selected classifier, a set of different hyperparameters was chosen. Table 5 shows the parameters, which were selected on the basis of the authors' criteria and availability within the syntax and structure of the applied algorithms (i.e., Sci-kit learn module in Python).

Table 5. Selected hyperparameters for grid search and cross-validation procedures.

KNN	Random Forest	MLP
Leaf size	Number of trees in the forest	Maximum number of iterations
Number of neighbors	Criterion for split quality	Number of hidden layers
Distance metric	Criterion for maximum features per split	Activation function
-	Minimum samples to split an internal node	Optimization solver
-	Minimum samples to be at a leaf node	Regularization parameter
-	Bootstrap Boolean (resampling)	Learning rate

3.7. Classification Results

Once all classifiers were trained and tested with the default parameters, cross-validation and grid search methods were implemented to select the set of parameters that output the maximum accuracies. On the basis of the list of hyperparameters presented in Table 6, the selected ones for the most accurate classifiers are shown in Tables 6–8 for the K-nearest neighbors, random forest, and multilayer perceptron, respectively. These classifiers were used for comparing accuracy using the testing set in later stages. Note that these optimal parameters varied when choosing a different working fluid, as the results for ethanol could not be extrapolated to those for FC-72. Nevertheless, slight differences could be seen, especially with those classifiers that do not depend on a large number of hyperparameters, such as the K-nearest neighbors classifier. For this classifier, a training set was split even though it was not strictly needed. This was for the sake of consistency when comparing all three classifiers.

Table 6. Optimal hyperparameters for K-nearest neighbors classifier.

Parameter	Value—Ethanol	Value—FC-72
Leaf size	1	1
Number of neighbors	25	26
Distance metric	Manhattan distance	Manhattan distance

Table 7. Optimal hyperparameters for random forest classifier.

Parameter	Value—Ethanol	Value—FC-72
Number of trees in the forest	100	100
Criterion for split quality	Entropy	Entropy
Criterion for maximum features per split	Squared root of features	Squared root of features
Minimum samples to split an internal node	10	2
Minimum samples to be at a leaf node	2	2
Bootstrap Boolean (resampling)	True	True

To assess the performance of individual classifiers after identifying the best set of hyperparameters, normalized confusion matrices were used to visualize the distribution of correct and incorrect classifications on the testing set. A confusion matrix depicts the fraction of correct and incorrect labeled points over the total number of true labels. Thus, a matrix entry of 1, for a specific label, indicates that all points in such a dataset are categorized with the expected label. Normalized values were used for more interpretable visual results.

Table 8. Optimal hyperparameters for multilayer perceptron classifier.

Parameter	Value—Ethanol	Value—FC-72
Maximum number of iterations	1000	100
Number of hidden layers	2	2
Activation function	ReLU	ReLU
Optimization solver	Adam	Adam
Regularization parameter	0.0001	0.0001
Learning rate	adaptive	adaptive

The confusion matrices for the three algorithms in the case of ethanol as the working fluid are shown in Table 9. In general, all algorithms performed similarly, where the classification of slug/plug flow was significantly higher than that of semi-annular flow. This could be due to the increase in observation errors while classifying semi-annular flow or to the innate nature of this flow pattern, which could have led to more erroneous observations.

Table 9. Confusion matrix results for ethanol.

		Actual Slug/Plug	Actual Semi-Annular
KNN	Predicted slug/plug	0.89	0.32
	Predicted semi-annular	0.11	0.68
Random Forest	Predicted slug/plug	0.88	0.32
	Predicted semi-annular	0.12	0.68
MLP	Predicted slug/plug	0.90	0.30
	Predicted semi-annular	0.10	0.70

In the case of ethanol, the multilayer perceptron exhibited a slightly higher number of correct classifications, considering both slug/plug and semi-annular flow. These low differences among the three classifiers suggest that, given the available data and selected input features (limited by design and, thus, subject to potential improvements), a fixed order of accuracy could be reached by all algorithms, with the highest provided by MLP.

In the case of FC-72, the highest fraction of correct classifications was also found using the MLP classifier. The major difference across classifiers was seen in the slug/plug category. The confusion matrices for FC-72 are illustrated in Table 10. As in the case of ethanol, the MLP classifier tended to present the highest accuracy among the algorithms with the testing set, after selecting the most suitable set of hyperparameters.

Table 10. Confusion matrix results for FC-72.

		Actual Slug/Plug	Actual Semi-Annular
KNN	Predicted slug/plug	0.90	0.33
	Predicted semi-annular	0.10	0.67
Random Forest	Predicted slug/plug	0.89	0.34
	Predicted semi-annular	0.11	0.66
MLP	Predicted slug/plug	0.91	0.33
	Predicted semi-annular	0.09	0.67

The overall classification performance is reflected in the value of the accuracy score. These values are reported in Tables 11 and 12, respectively. These values represent the accuracy of those classifiers that presented the highest cross-validated score when selecting the most suitable set of hyperparameters. In accordance with the confusion matrices, the results suggest that the use of MLP provided the highest performance. The lowest performance was shown by the random forest algorithm. In this work, this accuracy score was chosen as a selection criterion for the most suitable classification method; however, it is acknowledged that additional criteria such as computational time or performance when dealing with larger datasets could be included.

Table 11. Accuracy score for each algorithm: ethanol.

Classifier	Accuracy (%)
K-nearest neighbors	82.8
ANN	83.9
Random forest	82.2

Table 12. Accuracy score for each algorithm: FC-72.

Classifier	Accuracy (%)
K-nearest neighbors	75.8
ANN	77.1
Random forest	75.6

An alternative and complementary method for evaluating the performance of each classification algorithm is the analysis of learning curves. A learning curve shows the sensitivity of a particular performance metric (i.e., accuracy score, mean squared error, etc.) with respect to the size of the training set. This allows the user to identify (i) whether it is necessary to include more data samples in the training set, and (ii) whether the classifier under study presents a bias error or a variance one. Generally speaking, a bias error indicates that the classifier could be overly simple/complex with respect to the training set, leading to either overfitting or underfitting cases. Similarly, a variance error indicates that the classifier could vary drastically or remain unaffected when moving from training (seen data) to testing (unseen data). This also leads to overly simplistic/complex models depending on the case [49].

Both these concepts are related, as machine learning models with high/low bias present low/high variance, exhibiting a tradeoff that should always be taken into account [49]. To evaluate this tradeoff in both training and validation stages, the learning curves for each classifier and working fluid were created. Once the hyperparameters of each algorithm were selected via grid search, different sizes of training data were chosen (as a proportion of the initial training set size; see Table 4), and the training accuracy score was estimated for each training set size. For the validation curve, cross-validation was used once again (via *k*-fold cross-validation, as described in Section 3.5), and, for each training set size, accuracy scores were also calculated. The learning curves for all three algorithms and both working fluids are depicted in Figure 5.

Figure 5 reveals that the training error for the KNN classifier in both working fluids (a and d, respectively) was equal to 1, denoting perfect accuracy. Although this might seem like a successful result, there was a relatively large gap between the training score trend and that for the validation set. This gap is indicative of variance error, as the classifier performed well only during training. The accuracy score in both datasets was within the range of 83% to 100% for both working fluids, suggesting a small bias error in both stages. A similar case was seen in the random forest classifier. However, in this case, the variability of the accuracy score in the training set behaved differently as the size of the training set increased for both working fluids. A minimum accuracy score was reached with ethanol as a working fluid, whereas an oscillating trend was seen when working with FC-72. The accuracy score in the validation set was seemingly improved with the increase in data samples. In general, the random forest algorithm presented a lower bias error than KNN. On the other hand, the MLP classifier presented a smaller gap between training and validation stages, indicating a good balance between variance and bias errors, as the accuracy score for both stages was within the range of 84% to 85%. This suggests that the tradeoff for this method was the most balanced.

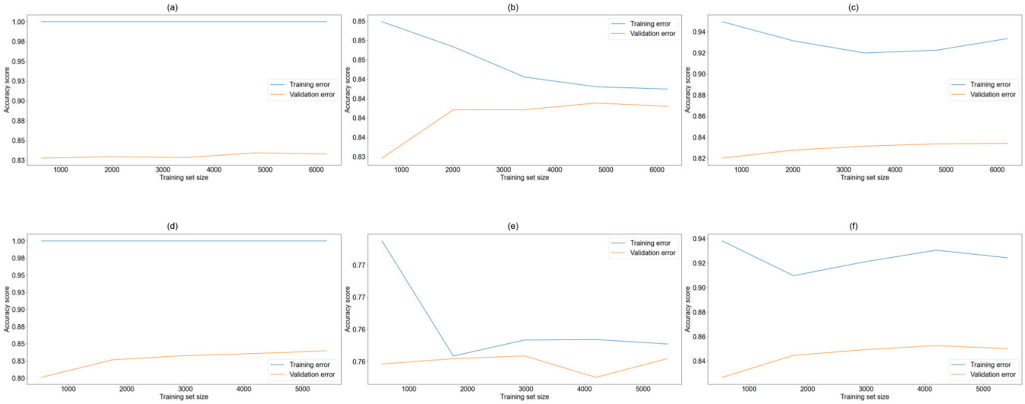


Figure 5. Training and cross-validation curves: (**a**) KNN with ethanol; (**b**) MLP with ethanol; (**c**) random forest with ethanol; (**d**) KNN with FC-72; (**e**) MLP with FC-72; (**f**) random forest with FC-72.

3.8. Flow Pattern Maps

The classification results from the most accurate classifier (MLP in this case) were used to develop a flow pattern map for both ethanol and FC-72. Once the classifier predicted the classes for each data point in the testing set (those in the training set were already stored during training), these points were used to estimate the values for the flow pattern maps. As a result, the maps can be used as a graphic tool for visualizing the outcomes from the classification methods. The x-axis corresponds to $Bo_l^{0.5}$, and the y-axis corresponds to $Fr_l^{0.5}We_l^{0.25}$, in accordance with the correlation between process conditions (velocity and acceleration) and the effect of the different forces acting on the fluid (namely, inertial, external, and related to the surface tension) proposed by Pietrasanta et al. [29]. The resulting flow pattern maps for both fluids developed by the authors are illustrated in Figure 6. These maps can act as a reference for comparison with the flow pattern maps from the MLP classifier.

For both working fluids, a much clearer transition zone was found when comparing the previous flow pattern maps and those based on the MLP classifier. This was due to the inherent improvements brought about by the use of the MLP method, as this algorithm provides a more systematic mean for classification compared to visual categorization or empirical correlations with physical properties.

Figure 7 shows the flow pattern map for ethanol. The map clearly shows a threshold value where the transition from slug/plug to semi-annular flow took place, located approximately where the x-axis was equal to 4. Higher values along this axis indicate semi-annular flow, where surface tension no longer dominated the fluid flow, and the increased acceleration led to higher bubble lengths. Semi-annular flow can be further identified on the y-axis, where for values of $Fr_l^{0.5}We_l^{0.25}$ lower than 2, a relatively high density of points classified as semi-annular flow was encountered. Lower values on both axes indicated the presence of slug-plug flow, either because the PHP device was not active or because the external forces were not strong enough to prevail over the surface tension of the working fluid while operating.

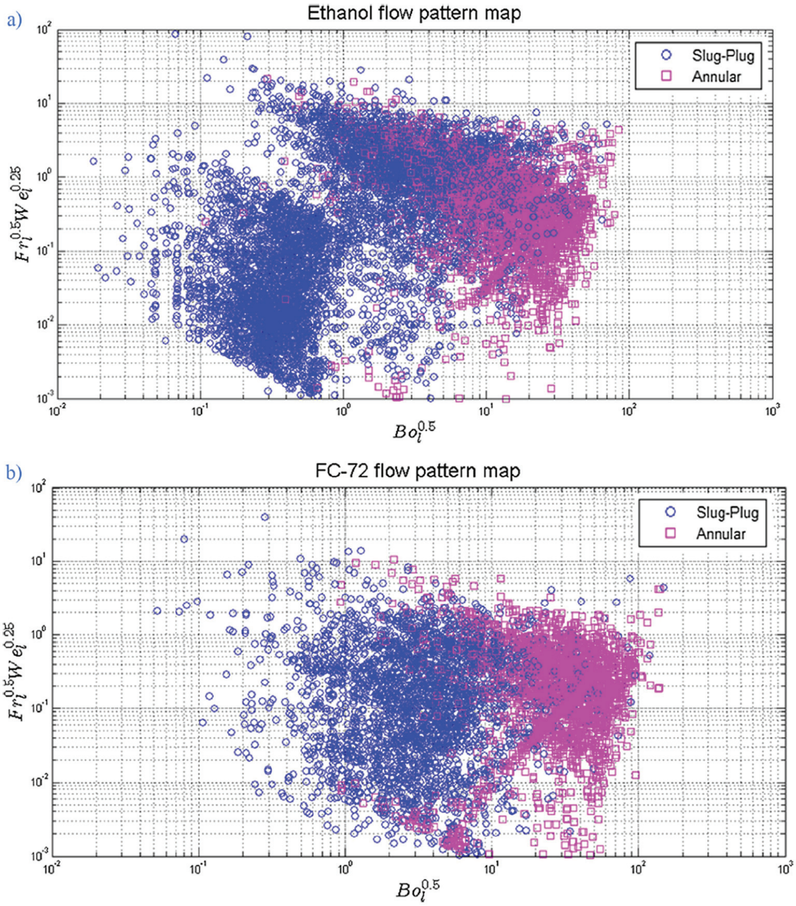


Figure 6. Flow pattern maps for ethanol (**a**) and FC-72 (**b**) proposed by Pietrasanta et al. [29]. Reproduced with permissions.

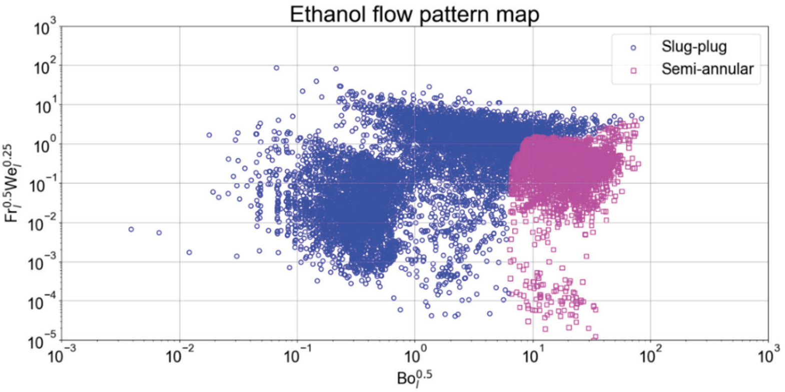


Figure 7. Flow pattern map for ethanol: multilayer perceptron.

In the case of FC-72, its corresponding flow pattern map is shown in Figure 8. Similar to the case of ethanol, a threshold value for the transition zone was found. Here, the threshold was located approximately when $Bo_l^{0.5}$ = 9. This means that slug/plug flow prevailed for higher velocities and bubble lengths when FC-72 was used as working fluid. This can be explained by the differences in the surface tension of both fluids. FC-72 has a lower surface tension than ethanol; hence, for the same dynamic conditions (fluid velocity and acceleration), greater numbers of $Bo_l^{0.5}$ would be reached before the flow regime transition. Abnormal points were found within the slug/plug region for FC-72, which were classified as semi-annular. This phenomenon could have been caused by the propagation of visual errors, as discussed previously. This would also mean that the choice of Bo_l did not properly reflect the surface tension effects, since this number could not capture the regimes for both fluids.

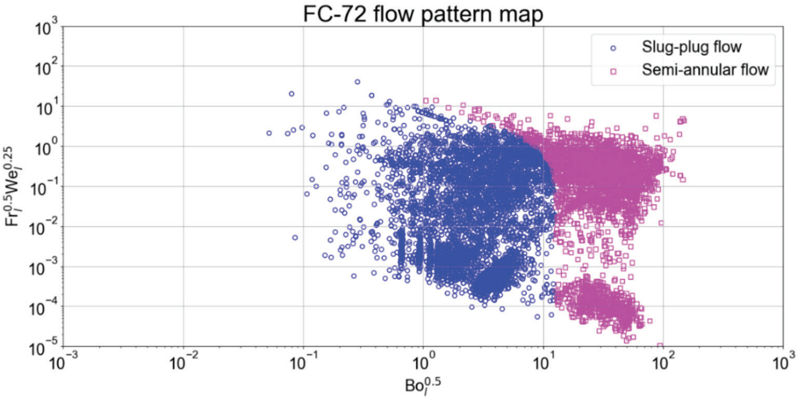


Figure 8. Flow pattern map for FC-72: multilayer perceptron.

To overcome the effect of different surface tension in both working fluids, a modified term was used on the x-axis of both flow pattern maps. The term $Bo_l^{0.5}$ was scaled using the ratio $\frac{\sigma_l}{\sigma_{ref}}$, where σ_l represents the surface tension of the working fluid, and σ_{ref} is a reference surface tension, which in this case was that of ethanol. Surface tension for both fluids was estimated via validated correlations that depend on key operating conditions such as saturation temperature [29]. This correction ratio was calculated for each observation, and updated flow pattern maps for both fluids were developed (using the MLP classifier). Note that, since ethanol was used as reference, no changes were found in its flow pattern map. The updated flow pattern maps for ethanol and FC-72 are depicted in Figures 9 and 10, respectively.

The updated flow pattern maps exhibited more consistent threshold values on both axes, for both working fluids. In the case of ethanol, these values were 6 for the x-axis and 2 for the y-axis, whereas, for FC-72, these limits were located at 5 for the x-axis and 1 for the y-axis. These values allowed for more interpretability, as it was now possible to cluster the observations and determine their corresponding flow regime on the basis of their relative location to the threshold values, with a margin of only ±1 unit on each axis.

Overall, the predictions presented good correspondence with experimental results, and the use of modified numbers plus scaling allowed for a clearer differentiation of flow regimes. However, it is worth noting that, although a relatively large number of data were used, this only represents a single PHP design (e.g., the single-loop PHP). Therefore, any attempt to implement these classifiers in a different system would most likely provide less accurate predictions, and a more extensive dataset would be needed.

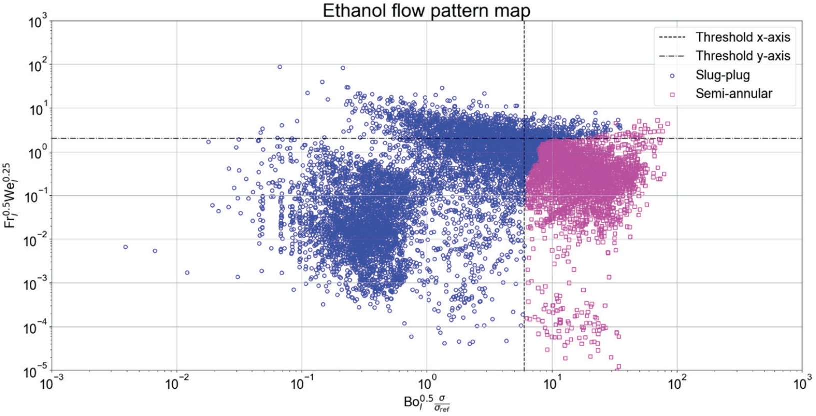


Figure 9. Updated flow pattern map for ethanol: multilayer perceptron.

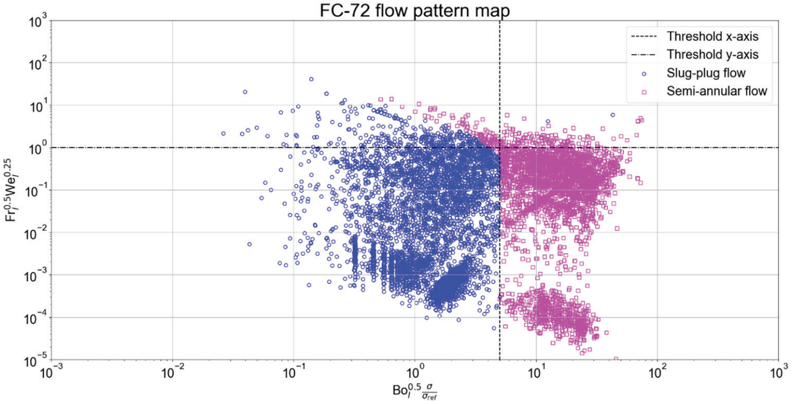


Figure 10. Updated flow pattern map for FC-72: multilayer perceptron.

4. Conclusions

For the purpose of proposing accurate data-driven methods for the flow regime classification in PHP systems, three different machine learning algorithms were tested on experimental data from a PHP device, for two different working fluids (namely, ethanol and FC-72). Both datasets were labeled with their corresponding flow regimes, and the most relevant input features were identified and embedded into specific groups of dimensionless numbers that accurately captured the physical phenomena. All three classifiers showed good performance, whereby the classification of the ethanol data was more accurate than that of FC-72, indicating that the process of labeling the data may have been more challenging in the latter case. The use of the multilayer perceptron (MLP) exhibited the highest performance for both working fluids, whereas the random forest algorithm presented the lowest accuracy, although all algorithms performed similarly. The prediction results from the most accurate classifiers were used to build a flow pattern map for each working fluid. In both cases, clear thresholds were identified, where the transition from slug/plug to semi-annular flow took place. These bounds were obtained after scaling the values of the modified Bond number with those of surface tension for both working fluids. The use of a trained and an automatic classifier in this context could provide a more accurate and less demanding classification of flow regimes. Considering a larger set of data with heat fluxes and geometrical parameters, since effective bubble accelerations and velocities would be dependent variables in this case, this method could effectively offer the chance of overcoming the rough use of Bond numbers to predict confined slug/plug flows in PHPs.

Further extensions of this work include the use of more diverse data, which will improve the robustness of the classification algorithms. In addition, the use of unsupervised learning could be a next step and a significant upgrade. In this way, the labeling process would not be needed, and an appropriate algorithm would identify different clusters of data that may correspond with the flow regimes the clusters belong to. Note that the selection of input features is still of great importance, and the use of the modified Weber, Froude, and Bond numbers can be validated from the results of the clustering.

The use of accurate classifiers in this context allows for a more straightforward identification of flow regimes. This enables the correct selection of models to be used for design, simulation, and optimization of PHP systems. Additionally, regression algorithms can be integrated to the current framework to estimate thermal resistance, which would provide a substantial input for estimating the thermal performance of PHP devices. The results can reveal a clear and robust path to define operational regimes in PHP devices. Moreover, the use of more data from other experiments with different geometries, fluids, and materials can provide a useful resource to improve the applicability of classifiers.

Author Contributions: Conceptualization, J.L.-F., L.P., M.M. and F.C.; methodology, J.L.-F. and F.C.; software, J.L.-F.; data curation, L.P. and J.L.-F.; writing—original draft preparation, L.P. and J.L.-F.; writing—review and editing, L.P., J.L.-F., M.M. and F.C.; supervision, M.M. and F.C.; funding acquisition, M.M. and F.C. All authors read and agreed to the published version of the manuscript.

Funding: This research was funded by EPSRC grant HyHP (EP/P013112/1), the European Space Agency MAP projects TOPDESS and Hexxcell Ltd.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

Nomenclature

a	Fluid acceleration (m/s^2)	$\dot{Q}$	Heat load input (W)
Bo	Bond number	ρ	Density (kg/m^3)
Co	Confinement number	$score$	Accuracy score
d	Diameter (mm)	sd	Standard deviation
f	Frequency (Hz)	σ	Surface tension (N/m)
Fr	Froude number	T	Temperature (°C)
g	Gravitational acceleration (m/s^2)	θ	Angle (°)
h	Enthalpy (J/kg)	u	Velocity (m/ms)
l	length (m)	We	Weber number
μ	Dynamic viscosity (Pa·s)	x_i	Data point
$\overline{\mu}$	Expected or average value	y_i	True categorical value
N	Number of turns	$\hat{y}_i$	Predicted categorical value
n	Total number of data points	z_i	Normalized data point
P	Pressure (Pa)		

Subscripts

b	Bubble	l	Liquid
c	Condenser	l,v	Liquid to vapor
cr	Critical	ref	Reference value
e	Evaporator	v	Vapor
f	Fluid	w	Wall

Abbreviations

ANN	Artificial neural network	PCB	Printed circuit board
ESA	European Space Agency	PHP	Pulsating heat pipe
FR	Filling ratio	RF	Random forest
KNN	K-nearest neighbors	TC	Thermocouples
ML	Machine learning	TCS	Thermal control system
MLP	Multilayer Perceptron	TS	Thermosyphon
p	Parabola		

References

1. Karthik, C.A.; Kalita, P.; Cui, X.; Peng, X. Thermal Management for Prevention of Failures of Lithium Ion Battery Packs in Electric Vehicles: A Review and Critical Future Aspects. *Energy Storage* **2020**, *2*, e137. [CrossRef]
2. Feng, X.; Ren, D.; He, X.; Ouyang, M. Mitigating Thermal Runaway of Lithium-Ion Batteries. *Joule* **2020**, *4*, 743–770. [CrossRef]
3. Jones, N. How to Stop Data Centres from Gobbling up the world's Electricity. *Nature* **2018**, *561*, 163–166. [CrossRef] [PubMed]
4. Kandlikar, S.G. Heat Transfer Mechanisms during Flow Boiling in Microchannels. *J. Heat Transf.* **2004**, *126*, 8–16. [CrossRef]
5. Thome, J.R. Boiling in Microchannels: A Review of Experiment and Theory. *Int. J. Heat Fluid Flow* **2004**, *25*, 128–139. [CrossRef]
6. Liu, D.; Garimella, S.V. Flow Boiling Heat Transfer in Microchannels. *J. Heat Transf.* **2007**, *129*, 1321–1332. [CrossRef]
7. Bar-Cohen, A.; Sheehan, J.R.; Rahim, E. Two-Phase Thermal Transport in Microgap Channels—Theory, Experimental Results, and Predictive Relations. *Microgravity Sci. Technol.* **2011**, *24*, 1–15. [CrossRef]
8. Baldassari, C.; Marengo, M. Flow Boiling in Microchannels and Microgravity. *Prog. Energy Combust. Sci.* **2013**, *39*, 1–36. [CrossRef]
9. Mahmoud, M.M.; Karayiannis, T.G. Flow Pattern Transition Models and Correlations for Flow Boiling in Mini-Tubes. *Exp. Therm. Fluid Sci.* **2016**, *70*, 270–282. [CrossRef]
10. Karayiannis, T.; Mahmoud, M. Flow Boiling in Microchannels: Fundamentals and Applications. *Appl. Therm. Eng.* **2017**, *115*, 1372–1397. [CrossRef]
11. Ahmad, S.W.; Lewis, J.S.; McGlen, R.J.; Karayiannis, T.G. Pool Boiling on Modified Surfaces Using R-123. *Heat Transf. Eng.* **2014**, *35*, 1491–1503. [CrossRef]
12. Berenson, P. Experiments on Pool-Boiling Heat Transfer. *Int. J. Heat Mass Transf.* **1962**, *5*, 985–999. [CrossRef]
13. Marto, P.J.; Lepere, V.J. Pool Boiling Heat Transfer from Enhanced Surfaces to Dielectric Fluids. *J. Heat Transf.* **1982**, *104*, 292–299. [CrossRef]
14. Mudawar, I.; Anderson, T.M. Optimization of Enhanced Surfaces for High Flux Chip Cooling by Pool Boiling. *J. Electron. Packag.* **1993**, *115*, 89–100. [CrossRef]
15. Estes, K.A.; Mudawar, I. Comparison of Two-Phase Electronic Cooling Using Free Jets and Sprays. *J. Electron. Packag.* **1995**, *117*, 323–332. [CrossRef]
16. Bintoro, J.S.; Akbarzadeh, A.; Mochizuki, M. A Closed-Loop Electronics Cooling by Implementing Single Phase Impinging Jet and Mini Channels Heat Exchanger. *Appl. Therm. Eng.* **2005**, *25*, 2740–2753. [CrossRef]
17. Rau, M.J.; Dede, E.M.; Garimella, S.V. Local Single- and Two-Phase Heat Transfer from an Impinging Cross-Shaped Jet. *Int. J. Heat Mass Transf.* **2014**, *79*, 432–436. [CrossRef]
18. De Oliveira, P.A.; Barbosa, J.R. Novel Two-Phase Jet Impingement Heat Sink for Active Cooling of Electronic Devices. *Appl. Therm. Eng.* **2017**, *112*, 952–964. [CrossRef]
19. Thome, J.R. *Encyclopedia of Two-Phase Heat Transfer and Flow IV*; World Scientific Pub Co. Pte. Lt.: Singapore, 2018; Volume 1. [CrossRef]
20. Akachi, H. Structure of a Heat Pipe. U.S. Patent 4921041, 1990. Available online: https://patentimages.storage.googleapis.com/pdfs/US4921041.pdf (accessed on 23 October 2020).
21. Akachi, H. Structure of a Micro-Heat Pipe. U.S. Patent US005219020A, 1993. Available online: https://patentimages.storage.googleapis.com/8d/87/57/18fa8dfa9abc67/US5219020.pdf (accessed on 23 October 2020).
22. Das, S.; Nikolayev, V.; Lefevre, F.; Pottier, B.; Khandekar, S.; Bonjour, J. Thermally Induced Two-Phase Oscillating Flow Inside a Capillary Tube. *Int. J. Heat Mass Transf.* **2010**, *53*, 3905–3913. [CrossRef]
23. Nine, J.; Tanshen, R.; Munkhbayar, B.; Chung, H.; Jeong, H. Analysis of Pressure Fluctuations to Evaluate Thermal Performance of Oscillating Heat Pipe. *Energy* **2014**, *70*, 135–142. [CrossRef]
24. Nikolayev, V.S. A Dynamic Film Model of the Pulsating Heat Pipe. *J. Heat Transf.* **2011**, *133*, 081504. [CrossRef]
25. Nekrashevych, I.; Nikolayev, V.S. Effect of Tube Heat Conduction on the Pulsating Heat Pipe Start-up. *Appl. Therm. Eng.* **2017**, *117*, 24–29. [CrossRef]
26. Nekrashevych, I.; Nikolayev, V.S. Pulsating Heat Pipe Simulations: Impact of PHP Orientation. *Microgravity Sci. Technol.* **2019**, *31*, 241–248. [CrossRef]
27. Liu, S.; Li, J.; Dong, X.; Chen, H. Experimental Study of Flow Patterns and Improved Configurations for Pulsating Heat Pipes. *J. Therm. Sci.* **2007**, *16*, 56–62. [CrossRef]
28. Nazari, M.A.; Ahmadi, M.H.; Ghasempour, R.; Shafii, M.B. How to Improve the Thermal Performance of Pulsating Heat Pipes: A Review on Working Fluid. *Renew. Sustain. Energy Rev.* **2018**, *91*, 630–638. [CrossRef]

29. Pietrasanta, L.; Mameli, M.; Mangini, D.; Georgoulas, A.; Miché, N.; Filippeschi, S.; Marengo, M. Developing Flow Pattern Maps for Accelerated Two-Phase Capillary Flows. *Exp. Therm. Fluid Sci.* **2020**, *112*, 109981. [CrossRef]
30. Cheng, L.; Ribatski, G.; Thome, J.R. Two-Phase Flow Patterns and Flow-Pattern Maps: Fundamentals and Applications. *Appl. Mech. Rev.* **2008**, *61*, 050802. [CrossRef]
31. Özdemir, M.R.; Mahmoud, M.M.; Karayiannis, T.G. Flow Boiling of Water in a Rectangular Metallic Microchannel. *Heat Transf. Eng.* **2021**, *42*, 492–516. [CrossRef]
32. Kandlikar, S.G.; Garimella, S.; Li, D.; Colin, S.; King, M.R. *Heat Transfer and Fluid Flow in Minichannels and Microchannels*; Elsevier: Amsterdam, The Netherlands, 2006; pp. 175–226. [CrossRef]
33. Mameli, M.; Catarsi, A.; Mangini, D.; Pietrasanta, L.; Miché, N.; Marengo, M.; Di Marco, P.; Filippeschi, S. Start-up in Microgravity and Local Thermodynamic States of a Hybrid Loop thermosyphon/Pulsating Heat Pipe. *Appl. Therm. Eng.* **2019**, *158*, 113771. [CrossRef]
34. Andredaki, M.; Georgoulas, A.; Miché, N.; Marengo, M. Accelerating Taylor Bubbles within Circular Capillary Channels: Break-up Mechanisms and Regimes. *Int. J. Multiph. Flow* **2020**, *134*, 103488. [CrossRef]
35. Guillen-Rondon, P.; Robinson, M.D.; Torres, C.; Pereya, E. Support Vector Machine Application for Multiphase Flow Pattern Prediction. *arXiv* **2018**, arXiv:1806.05054.
36. Zhu, G.; Wen, T.; Zhang, D. Machine Learning Based Approach for the Prediction of Flow boiling/Condensation Heat Transfer Performance in Mini Channels with Serrated Fins. *Int. J. Heat Mass Transf.* **2021**, *166*, 120783. [CrossRef]
37. Suh, Y.; Bostanabad, R.; Won, Y. Deep Learning Predicts Boiling Heat Transfer. *Sci. Rep.* **2021**, *11*, 5622. [CrossRef] [PubMed]
38. Hernandez, J.S.; Valencia, C.; Ratkovich, N.; Torres, C.F.; Muñoz, F. Data Driven Methodology for Model Selection in Flow Pattern Prediction. *Heliyon* **2019**, *5*, e02718. [CrossRef] [PubMed]
39. Zhang, Y.; Azman, A.N.; Xu, K.-W.; Kang, C.; Kim, H.-B. Two-Phase Flow Regime Identification Based on the Liquid-Phase Velocity Information and Machine Learning. *Exp. Fluids* **2020**, *61*, 212. [CrossRef]
40. Jokar, A.; Godarzi, A.A.; Saber, M.; Shafii, M.B. Simulation and Optimization of a Pulsating Heat Pipe Using Artificial Neural Network and Genetic Algorithm. *Heat Mass Transf.* **2016**, *52*, 2437–2445. [CrossRef]
41. Jalilian, M.; Kargarsharifabad, H.; Godarzi, A.A.; Ghofrani, A.; Shafii, M.B. Simulation and Optimization of Pulsating Heat Pipe Flat-Plate Solar Collectors Using Neural Networks and Genetic Algorithm: A Semi-Experimental Investigation. *Clean Technol. Environ. Policy* **2016**, *18*, 2251–2264. [CrossRef]
42. Patel, V.M.; Mehta, H.B. Thermal Performance Prediction Models for a Pulsating Heat Pipe Using Artificial Neural Network (ANN) and Regression/Correlation Analysis (RCA). *Sādhanā* **2018**, *43*, 184. [CrossRef]
43. Wang, X.; Yan, Y.; Meng, X.; Chen, G. A General Method to Predict the Performance of Closed Pulsating Heat Pipe by Artificial Neural Network. *Appl. Therm. Eng.* **2019**, *157*, 113761. [CrossRef]
44. Pietrasanta, L.; Mangini, D.; Fioriti, D.; Miché, N.; Andredaki, M.; Georgoulas, A.; Araneo, L.; Marengo, M. A Single Loop Pulsating Heat Pipe in Varying Gravity Conditions: Experimental Results and Numerical Simulations. In Proceedings of the International Heat Transfer Conference 16, Beijing, China, 10–15 August 2018; Volume 16, pp. 4877–4884. [CrossRef]
45. Pletser, V. European Aircraft Parabolic Flights for Microgravity Research, Applications and Exploration: A Review. *Reach* **2016**, *1*, 11–19. [CrossRef]
46. Altman, N.S. An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression. *Am. Stat.* **1992**, *46*, 175–185. [CrossRef]
47. Duda, R.O.; Heart, P.E.; Stork, D.G. *Pattern Classification*, 2nd ed.; Wiley-Interscience: Hoboken, NJ, USA, 2000; ISBN 9780471056690.
48. Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
49. Hastie, T.; Tibshirani, R.; Friedman, J. *The Elements of Statistical Learning*; Springer: Berlin/Heidelberg, Germany, 2009. [CrossRef]
50. Geurts, P.; Ernst, D.; Wehenkel, L. Extremely Randomized Trees. *Mach. Learn.* **2006**, *63*, 3–42. [CrossRef]
51. Pal, S.; Mitra, S. Multilayer Perceptron, Fuzzy Sets, and Classification. *IEEE Trans. Neural Networks* **1992**, *3*, 683–697. [CrossRef]

sustainability

Article

An Unsupervised Mutual Information Feature Selection Method Based on SVM for Main Transformer Condition Diagnosis in Nuclear Power Plants

Wenmin Yu [1], Ren Yu [1,*] and Jun Tao [2]

[1] School of Nuclear Science and Techniques, Naval University of Engineering, Wuhan 430034, China; hust_ywm2012@163.com

[2] China Nuclear Power Operation Management Co., Ltd., Jiaxing 314300, China; taoj@cnnp.com.cn

* Correspondence: 18071068480@163.com

Abstract: Dissolved gas in oil (DGA) is a common means of monitoring the condition of an oil-immersed transformer. The concentration of dissolved gas and the ratio of different gases are important indexes to judge the condition of power transformers. Monitoring devices for dissolved gas in oil are widely installed in main transformers, but there are few recorded fault data of main transformers. The special operation and maintenance modes of main transformers leads to the fault modes particularity of main transformers. In order to solve the problem of insufficient samples and the feature uncertainty, this paper puts forward an unsupervised mutual information method to select the feature verified by the optimized support vector machine (SVM) model of particle swarm optimization (PSO) method and tries to find the feature sequence with better performance. The methos is validated by data from nuclear power transformers.

Keywords: main transformer; condition monitoring; unsupervised mutual information; feature selection; DGA

Citation: Yu, W.; Yu, R.; Tao, J. An Unsupervised Mutual Information Feature Selection Method Based on SVM for Main Transformer Condition Diagnosis in Nuclear Power Plants. *Sustainability* **2022**, *14*, 2700. https://doi.org/10.3390/su14052700

Academic Editors: Luis Hernández-Callejo, Sergio Nesmachnow and Sara Gallardo Saavedra

Received: 25 January 2022
Accepted: 16 February 2022
Published: 25 February 2022

1. Introduction

Power transformers that work under harsh environments would experience thermal decomposition of oil and cellulose insulation materials, such as arcing, corona discharge, low energy sparks, severe overloading, overheating of insulation systems and pump motor failures. These conditions alone or in combination can produce combustible and noncombustible gases [1] Detection of anomalies requires an assessment of the amount of gas produced. Gas in oil-immersed transformers can be used to identify fault types, including thermal and electrical interference. Gases obtained from chromatographic analysis of insulating oils may contain dissolved carbon monoxide (CO), carbon dioxide (CO_2), nitrogen (N_2), hydrogen (H_2), methane (CH_4), acetylene (C_2H_2), ethylene (C_2H_4), and ethane (C_2H_6). The composition, formation rate and specific content ratio of dissolved gas can be used to indicate transformer condition.

The composition and content of dissolved gases in oil of transformer insulation can reflect the operation condition of transformer to a great extent thus dissolved gas analysis (DGA) has become an effective method for fault diagnosis of oil-immersed transformers [2].

Organizations such as the Institute of Electrical and Electronics Engineers (IEEE) and the International Electrotechnical Commission (IEC) recommend a variety of diagnostic techniques [3], depending on the type of transformer and operating conditions. Some of the most commonly used techniques include Doernenburg ratio, Rogers ratio, Duval triangle model, etc. These classical diagnostic methods mostly take the ratio of different gases as the characteristic input and then judge the actual operating condition of the transformer by the threshold value formed by experience or statistical methods. Fuzzy network, support vector machine, artificial neural network, and other commonly used artificial intelligence methods are also generally introduced into the field of power transformer fault diagnosis [4–7]. However, in the studies of different scholars, the features used as the basis of intelligent diagnosis are often different.

In previous studies, in addition to problems in diagnostic methods, there are the following phenomena in monitoring data: less test data, less available data sets and unbalanced data type distribution, which bring great problems to algorithm verification [8].

Main transformers are important equipment for power generation of nuclear facilities. They are in a high-load long-term condition and are more prone to failure caused by aging [9]. Meanwhile, due to the particularity of nuclear power refueling overhaul and the conservative culture of nuclear power [10], the maintenance strategy of nuclear equipment is more rigorous and conservative, and the failure modes of main transformers may be slightly different. The data of the main transformers are classified separately in the IEC database [2], which shows transformer performance difference in nuclear industry.

Due to the particularity of nuclear power transformers, there are less marked data and more constraints on the monitoring data that can be used for research. The features are important inputs of the diagnostic algorithm. High-dimensional features bring high computational cost and the risk of "over-fitting". Dimensionality reduction or selection is an important research direction.

In this paper, an unsupervised mutual information feature algorithm is proposed for feature selection of different features proposed in the current classical algorithm and intelligent algorithm, as a pattern recognition method, SVM can construct the optimal classification hyperplane under the condition of small sample learning and distinguish transformer conditions according to the input features. The main transformer condition diagnosis model based on support vector machine is adopted for diagnosis in this paper, and the case data of main transformer is verified.

2. Framework of the Feature Selection Method Based on the SVM Model for Main Transformers

The research framework for the feature selection method based on the architecture of the SVM model for the main transformers is shown in Figure 1.

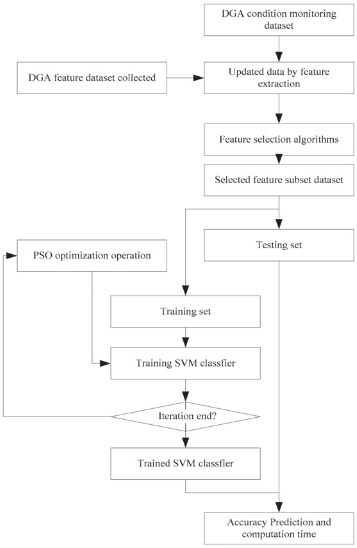

DGA condition monitoring dataset
DGA feature dataset collected
Updated data by feature extraction
Feature selection algorithms
Selected feature subset dataset
Testing set
PSO optimization operation
Training set
Training SVM classfier
Iteration end?
Trained SVM classfier
Accuracy Prediction and computation time

Figure 1. Framework of Feature Selection method based on SVM model for main transformers.

The gas concentration values measured in the continuous operation process of nuclear power transformer are obtained, and there is almost no-fault data.

The features used in various power transformer condition diagnosis methods based on dissolved gas in oil are extensively studied. On this basis, the initial feature set is formed.

The unsupervised mutual information feature extraction algorithm is adopted to extract features, and the set of sequence features is obtained according to the weight coefficient from high to low.

In the feature set, different number of feature sets are selected sequentially and verified by optimized SVM model for transformer fault diagnosis.

In order to reduce the contingency of the experiment, the 5-fold verification method is used to process the training samples and test samples to verify the validity of the selection feature in the diagnosis of the nuclear power transformer condition diagnosis.

Based on the accuracy of diagnosis, the feasibility of different feature extraction algorithms in the condition diagnosis of main transformers is analyzed.

3. Condition Diagnosis Model for Main Transformer

Condition diagnosis model is important to verify the feature selection algorithm and to determine the accuracy and rapidity of transformer condition diagnosis. SVM is a machine learning method based on statistical learning theory, compared with other algorithms, it can well solve practical problems such as small sample, nonlinear [11], the PSO algorithm can converge fast in the parameter optimization. The SVM optimized with PSO in the field of power transformer fault diagnosis has further application [12].

3.1. Support Vector Machine

State diagnosis of main transformers, As the case of a typical nonlinear classification problem, the overall plan of SVM is the first use of a nonlinear transform the input space data is mapped to a high-dimensional feature vector space, and then in the feature space of the optimal separating hyperplane is constructed, linear classification, after the last map back to the original space Became a nonlinear classification of input space [13].

SVM settings

At present, the commonly used kernel functions are mainly polynomial kernel function, radial basis (RBF) kernel function, hyperbolic tangent (sigmoid) kernel function, and so on. This paper mainly uses the RBF kernel function to the apply to SVM model.

3.2. PSO for Optimal Parameters

PSO is a kind of evolutionary computation, the basic idea of which is to find the optimal solution through the cooperation and information sharing between individuals in the group. It mimics a bird in a flock by designing a massless particle with just two properties: speed, which represents how fast it is moving, and position, which represents the direction it is moving. Each particle separately searches for the optimal solution in the search space, and records it as the current individual extreme value, and shares the individual extreme value with other particles in the whole particle swarm and finds the optimal individual extreme value as the current global optimal solution of the whole particle swarm. All particles in a swarm adjust their speed and position based on the current individual extremum they find, and the current global optimal solution shared by the whole swarm [14].

PSO-SVM Parameter Settings

C_1: the initial value is 1.5, local search capability of PSO parameters

C_2: 1.7 initially, PSO parameter global search capability

Maxgen: The initial value is 200, the maximum number of evolutions

Sizepop: the initial value is 20 and the maximum size of the population

K: initial 0.6 (k belongs to [0.1, 1.0]), the relationship between the speed and x ($V = KX$)

WV: The initial value is 1 (wV best belongs to [0.8, 1.2]), and the rate updates the elasti coefficient before the speed in the formula

WP: The initial value is 1, the elastic coefficient in front of the velocity in the population renewal formula

V: Initial 5, SVM Cross Validation parameter

Popcmax: the maximum value of the change in the SVM parameter C, initially 100.

Popcmin: the initial value is 0.1, the minimum change of SVM parameter C.

Popgmax: the initial value is 1000, the maximum value of the change of THE SVM parameter G.

Popgmin: the initial value is 0.01, the minimum change value of the SVM parameter C.

4. Feature Selection Algorithms for Main Transformer Condition

This chapter introduces the unsupervised mutual information filtering feature sorting method used in feature selection. In feature selection, the relevance of each feature is calculated first, the importance of the feature is evaluated by the forward sequential search, and finally an ordered feature sequence is output.

4.1. Stepwise Feature Selection Process

The process of stepwise feature selection is to select a feature from the unselected feature set each time and add the feature set S. In accordance with the selection order, the feature set outputs an ordered feature sequence.

When initializing, the feature set is empty. The unselected feature set is the complete set of all known features.

After each step selection, the feature set increases the feature set selected in this step, while the feature set not selected reduces the feature set selected in this step. Until the unselected feature set is empty.

4.2. Selection Principle

The principle of ʻʻminimum redundancy—maximum correlation" which is similar to the famous supervised feature selection method is adopted [15], and the selection of the mth feature is based on:

$$l_m = arg\ max_{f_i \in U_m}\{Rel(f_i) - \frac{1}{m-1}\sum_{f_t \in S_{m-1}} Red(f_i, f_t)\} \tag{1}$$

where U_m represents the set of unselected features in the current step

f_i represents a feature in the unselected feature set in the current step;

$Rel(f_i)$ represents Relevance of feature f_i, which is the average mutual information between feature f_i and any other one in the whole feature set is defined as $Rel(f_i)$. $Rel(f_i)$ can be calculated with Formula (2).

$$Rel(f_i) = \frac{1}{n}\sum_{t=1}^{n} I(f_i; f_t) = \frac{1}{n}(H(f_i) + \sum_{1 \le t \le n, t \ne i} I(f_i; f_t)) \tag{2}$$

S_{m-1} is the selected feature set in the current step;

$Red(f_i, f_t)$is the redundancy of feature f_i relative to selected feature f_t. $Red(f_i, f)_t$ can be calculated with Formula (3).

$$Red(f_i, f_t) = Rel(f_t) - \mathrm{Rel}(f_t|f_i) \tag{3}$$

$\mathrm{Rel}(f_t|f_i)$ is conditional relevance of f_t with f_i, $\mathrm{Rel}(f_t|f_i)$ can be calculated with Formula (4).

$$\mathrm{Rel}(f_t|f_i) = \frac{H(f_t|f_i)}{H(f_t)} \times Rel(f_t) \tag{4}$$

4.3. Relationship with Supervised Algorithms

When the data type is supervised, the labels of the class can represent the information of the whole feature set.

Then relevance of feature f_i can be defined as

$$Rel(f_i) = I(f_i, c) \tag{5}$$

where c in Formula (5) is the class label [16].

Redundancy between feature f_i and the selected feature f_t is defined as

$$Red(f_i, f_t) = I(f_i, f_t) \tag{6}$$

According to the principle of mathematics [17], relevance in an unsupervised algorithm is the lower bound of relevance in a supervised algorithm, and redundancy in an unsupervised algorithm is proportional to the redundancy in a supervised algorithm. When the initial feature set is approximately equal to the labels of the class, the sequence features obtained by the unsupervised algorithm are highly correlated with the sequence features obtained by the supervised algorithm.

5. Experiment and Validation

5.1. Experiment Description

The internal fault mode of power transformer is mainly mechanical fault, thermal fault and electrical fault, the latter two types of faults is the major issues, and mechanical fault is often shown in the form of thermal fault or electrical fault [18]. General power transformers are often subdivided into fault modes according to the degree of heating or arcing.

Due to safety culture of nuclear power plants, maintenance strategy for the main transformers tends to be conservative and strict; The failure of nuclear power transformers is rare to happen, and the failure data that can be accessed to publicly is very few. Available fault data cannot cover all the modes. Therefore, the condition of the main transformers are divided into the following three types in this paper, and only two summative failure modes are reserved and the corresponding as illustrated in Table 1.

Table 1. Code of power transformer operation condition.

NO.	FAILURE MODE	CONDITION TYPE	CODE
1	Partial discharge	Electrical fault	1
2	Low-energy discharge		
3	High-energy discharge		
4	Thermal fault < 300 °C	Thermal fault	2
5	Thermal fault 300 to 700 °C		
6	Thermal fault > 700 °C		
7		Normal	3

The fault data used in the experiment in this paper are reactor-related transformer data obtained from IEC TC database, and the feature selection data and normal data are the monitoring values of a nuclear power main transformer under normal operation. The data can be obtained in the Supplementary Material. A nuclear power plant generator, 24 kV voltage, is stepped up to 500 kV and connected to the 500 KV power grid through the main transformer. The main transformer is a three single-phase transformer, each phase capacity 410 MVA. The neutral points on the high voltage side are connected and directly grounded. Oil is regularly sampled and analyzed once every 3 months manually. Sampling intervals can sometimes be uneven, depending on special focus judged by sampling staff or adjustment by work schedules.

5.1.1. Sample Data

Gas concentration of DGA method is analyzed in one part per million (PPM). To facilitate the presentation and analysis of the characteristics in the figure, logarithmic processing is performed for each monitoring value in Figures 2 and 3. The box diagram of the monitored data value in the sample data is shown in Figure 2. Sample type distribution and monitoring value distribution of each basic feature are presented in Figure 3.

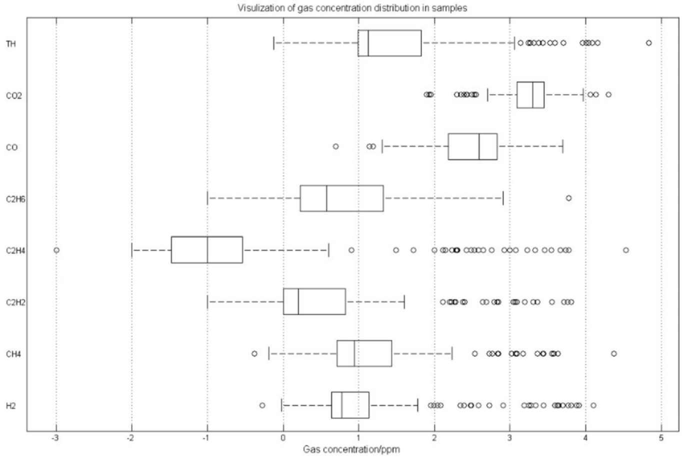


Figure 2. Box diagram of the monitored data value in the sample data.

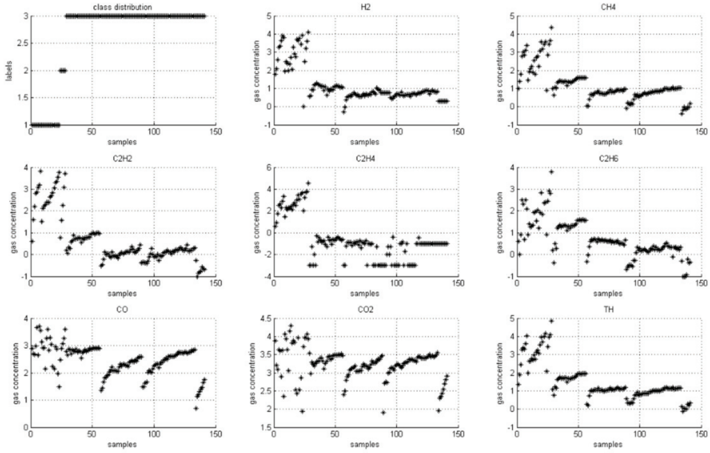


Figure 3. Sample type distribution and monitoring value distribution of each basic feature.

5.1.2. Basic Features by DGA Condition Monitoring

Typical gases measured in the DGA method of the main transformers include H_2, CH_4, C_2H_2, C_2H_4, C_2H_6, CO, CO_2 as illustrated in Figures 2 and 3.

$$TH = CH_4 + C_2H_2 + C_2H_4 + C_2H_6$$

Other features commonly used in transformer condition diagnosis methods are shown in Table 2.

Table 2. Common features of transformer DGA condition diagnosis methods.

NO.	FEATURE	DIAGNOSIS METHOD
1	$\%C_2H_2$	The Duval Triangle method [2]
2	$\%C_2H_4$	
3	$\%CH_4$	
4	CH_4/H_2	The three basic gas ratios of IEC 599/IEC 60599 [19]
5	C_2H_4/C_2H_6	
6	C_2H_2/C_2H_4	
7	C_2H_2/CH_4	Doernenberg Ratios [19]
8	C_2H_6/C_2H_2	
9	C_2H_2/H_2	Two new Gas ratios in IEC 60599
10	CO_2/CO	
11	$\varphi(H_2)$	Other approach [20]
12	$\varphi(CH_4)$	
13	$\varphi(C_2H_6)$	
14	$\varphi(C_2H_4)$	
15	$\varphi(C_2H_2)$	
16	$\%H_2$	fourth % ratio [21]

The symbols in the Duval triangle method shown in Table 1 are denoted as $\%C_2H_2 = 100x/(x+y+z)$; $\%C_2H_4 = 100y/(x+y+z)$; $\%CH_4 = 100z/(x+y+z)$; with $x = (C_2H_2)$; $y = (C_2H_4)$; $z = (CH_4)$ in PPM.

$\varphi(H_2)$, $\varphi(CH_4)$, $\varphi(C_2H_6)$, $\varphi(C_2H_4)$ and $\varphi(C_2H_2)$ in Table 1 represent the contents of five characteristic gases, respectively, and Total Combustion Gases (TCG) as in: TCG = $H_2 + CH_4 + +C_2H_4 + C_2H_6 + C_2H_2$; $\varphi(H_2) = H_2/TCG$; $\varphi(CH_4) = CH_4/TCG$; $\varphi(C_2H_6) = C_2H_6/TCG$; $\varphi(C_2H_4) = C_2H_4/TCG$; $\varphi(C_2H_2) = C_2H_2/TCG$;

And $\%H_2 = 100 * H_2/(H_2 + C_2H_6 + CO + CO_2)$.

As shown in Figures 2 and 3, the class of the sample is not balanced and several basic features are similar in distribution. Unsupervised feature extraction is adopted to obtain the feature set with maximum correlation and minimum redundancy.

5.2. Extracted Feature Sequence by Unsupervised MI V.S. Supervised MI

Table 3 shows the comparison of feature sequences extracted by the unsupervised mutual information method and supervised mutual information method and their corresponding weights. As obtained from Table 3.

Table 3 Extracted feature sequence by unsupervised MI V.S. supervised MI the mutual information of unsupervised feature extraction method selects the same first feature from cases of nuclear power transformer DGA with that of supervised method, while the weight value, generated by supervised feature extraction algorithm is greater than the corresponding values in the unsupervised algorithm, which means supervised method has more dynamic to choose the first feature; In other steps, selected feature is not the same, but the unsupervised algorithm has a strong dynamic at each step.

5.3. Results of the Diagnosis by Optimized SVM

5.3.1. Diagnostic Precision with Supervised Mutual Information Feature Selection Method versus Unsupervised Approach

According to the framework described in Section 2, a PSO-optimized SVM model is applied to classify cases to reflect the fitness of the selected features.

The feature sequences selected by the supervised mutual information feature selection algorithm and the unsupervised mutual information feature selection algorithm are applied, respectively. Different numbers of features are selected from feature sequences obtained by both methods, and the diagnostic accuracy of the both method is shown in Figure 4, in which the red point is where the best fitness is obtained.

Table 3. Extracted feature sequence by unsupervised MI V.S. supervised MI.

No.	Supervised MI		Unsupervised MI	
	Feature	Weight	Feature	Weight
1	$\%C_2H_4$	0.380665	$\%C_2H_4$	0.31706
2	$\%C_2H_2$	0.021415	$\%H_2$	0.276004
3	$\varphi(C_2H_4)$	0.095868	$\varphi(C_2H_2)$	0.302481
4	C_2H_4/C_2H_6	0.052677	$\varphi(C_2H_4)$	0.306476
5	C_2H_2/CH_4	0.058468	$\varphi(C_2H_6)$	0.299351
6	C_2H_6	0.066181	$\varphi(CH_4)$	0.29588
7	$\%CH_4$	0.082497	$\varphi(H_2)$	0.292157
8	$\varphi(C_2H_2)$	0.067574	CO_2/CO	0.298659
9	CO_2	0.066806	C_2H_2/H_2	0.304623
10	C_2H_6/C_2H_2	0.068373	C_2H_6/C_2H_2	0.305679
11	C_2H_2	0.059614	C_2H_2/CH_4	0.30906
12	CH_4/H_2	0.055447	C_2H_2/C_2H_4	0.301607
13	C_2H_4	0.057832	C_2H_4/C_2H_6	0.306031
14	$\varphi(CH_4)$	0.064196	CH_4/H_2	0.301547
15	$\%H_2$	0.032973	$\%CH_4$	0.303705
16	$\varphi(C_2H_6)$	−0.00899	$\%C_2H_2$	0.279383
17	CH_4	0.002819	TH	0.26508
18	CO	−0.00451	CO_2	0.243314
19	C_2H_2/C_2H_4	−0.01682	CO	0.245554
20	TH	−0.00641	C_2H_6	0.246708
21	C_2H_2/H_2	−0.02399	C_2H_4	0.229841
22	$\%H_2$	−0.02539	C_2H_2	0.213115
23	$\varphi(H_2)$	−0.0789	CH_4	0.202028
24	CO_2/CO	−0.08773	H_2	0.187682

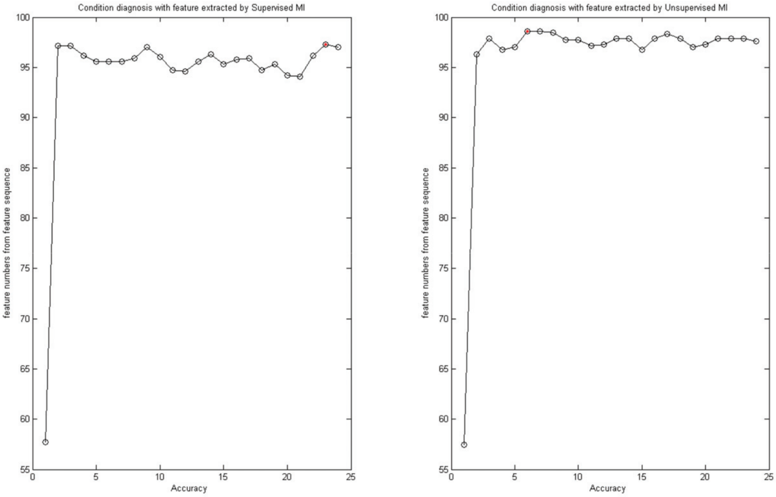


Figure 4. Diagnostic precision with supervised mutual information feature selection vs. unsupervised approach.

As can be seen from Figure 4, features selected from both unsupervised and supervised feature selection methods have good performance as input to diagnostic accuracy in cases, and both of the models achieve greatly increased diagnostic accuracy in the second feature.

5.3.2. Diagnostic Precision by Features of the Unsupervised Approach with Best Fitness vs. Other Classical Feature Set

Feature set with the highest diagnostic accuracy obtained by unsupervised mutual information feature selection method is used as input to optimized SVM diagnosis model for main transformer cases, other typical feature sets are used in contrast as shown in Table 4.

Table 4. Diagnostic precision by features of the unsupervised approach with best fitness vs. other classic feature set (percent).

TYPE	1	2	3	4
5-FOLD-1	100.00	89.29	89.29	96.43
5-FOLD-2	100.00	100.00	100.00	100.00
5-FOLD-3	100.00	96.43	96.43	100.00
5-FOLD-4	100.00	92.86	100.00	100.00
5-FOLD-5	92.86	96.43	92.86	92.86
AVERAGE	98.57	95.00	95.71	97.86

Note: 1 refers to the features of the unsupervised approach with the best fitness obtained as shown in Figure 4 as input to the optimized svm model; 2 refers to the features used in the three ratios method [1]; 3 refers to the features used in some intelligent methods [16]; 4 refers to the features used in the Duval Triangle method [2].

As shown in Table 4, the feature set obtained by the unsupervised mutual information feature selection algorithm, is used as the input of the optimized SVM diagnosis model and performs better than other inputs of the feature set in the case of diagnosis of the main transformer condition diagnosis. Therefore, the algorithm has high applicability.

6. Conclusions and Analysis

Fault data of main transformer lacks. In addition, the fault mode of the main transformer is different from that of other power transformers. This paper proposes an unsupervised mutual information feature selection method to calculate DGA monitoring data of main transformer and output feature selection sequence. Compared with the supervised mutual information feature selection algorithm, the unsupervised mutual information feature selection algorithm is highly correlated with the sequence features output by the supervised feature selection algorithm in feature selection. In the samples, the training samples and test samples were designed by five-fold method based on the appropriate feature set obtained by the unsupervised mutual information feature selection algorithm. The PSO optimized support vector machine model was used to verify the main transformer fault diagnostic, and the diagnosis accuracy was high. This method is suitable for feature extraction in main transformer fault diagnosis. However, the feature extraction method based on unsupervised mutual information is essentially an embedded feature extraction method with some significant advantages and disadvantages at the same time. The redundancy between features in the selected feature set is minimized, and its limitations depend on the evaluation of candidate solutions by the classification algorithm, which is computationally more expensive. Therefore, the offline data set can be used for training and verification in practical application, and the obtained feature set can be used to judge the condition of nuclear power transformers online.

Supplementary Materials: The following are available online at https://www.mdpi.com/article/10.3390/su14052700/s1, the experimental data of DGA samples used in Section 5.1 of this paper.

Author Contributions: Conceptualization, R.Y.; data collection, J.T.; methodology and the other, W.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable for studies not involving humans or animals.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data can be obtained in the Supplementary Materials of the paper.

Conflicts of Interest: Jun Tao is an employer from CNNC Nuclear Power Operation Management Co., Ltd. The other authors declare that they have no conflict of interest.

References

1. Bustamante, S.; Manana, M.; Arroyo, A.; Castro, P.; Laso, A.; Martinez, R. Dissolved Gas Analysis Equipment for Online Monitoring of Transformer Oil: A Review. *Sensors* **2019**, *19*, 4057–4078. [CrossRef] [PubMed]
2. Duval, M. A review of faults detectable by gas-in-oil analysis in transformers. *IEEE Electr. Insul. Mag.* **2002**, *18*, 8–17. [CrossRef]
3. Duval, M.; Depabla, A. Interpretation of gas-in-oil analysis using new IEC publication 60599 and IEC TC 10 data-bases. *Electr. Insul. Mag. IEEE* **2002**, *17*, 31–41. [CrossRef]
4. Liao, R.; Zheng, H.; Grzybowski, S.; Yang, L.; Zhang, Y.; Liao, Y. An integrated decision-making model for condition assessment of power transformers using fuzzy approach and evidential reasoning. *IEEE Trans. Power Deliv.* **2011**, *26*, 1111–1118. [CrossRef]
5. Kari, T.; Gao, W.; Zhao, D.; Abiderexiti, K.; Mo, W.; Wang, Y.; Luan, L. Hybrid feature selection approach for power transformer fault diagnosis based on support vector machine and genetic algorithm. *IET Gener. Transm. Distrib.* **2018**, *12*, 5672–5680. [CrossRef]
6. Miranda, V.; Castro, A.R.G.; Lima, S. Diagnosing Faults in Power Transformers with Autoassociative Neural Networks and Mean Shift. *IEEE Trans. Power Deliv.* **2012**, *27*, 1350–1357. [CrossRef]
7. Amora, M.; Braga, A.; Lisboa, L.; Pontes, R.; Almeida, O.; Barbosa, F. Improved DGA method based on rules extracted from high-dimension input space. *Electron. Lett.* **2012**, *48*, 1048–1049. [CrossRef]
8. Mirowski, P.; LeCun, Y. Statistical Machine Learning and Dissolved Gas Analysis: A Review. *IEEE Trans. Power Deliv.* **2012**, *27*, 1791–1799. [CrossRef]
9. Alexander, G.W.; Corbin, S.L.; Mcnutt, W.J. Influence of design and operating practices on excitation of generator step-up transformers. *IEEE Trans. Power Appar. Syst.* **1966**, *8*, 901–909. [CrossRef]
10. Agarwal, V.; Lybeck, N.J.; Pham, B.T. *Diagnostic and Prognostic Models for Generator Step-Up Transformers*; Idaho National Lab. (INL): Idaho Falls, ID, USA, 2014.
11. Guo, H.; Wang, W. An active learning-based SVM multi-class classification model. *Pattern Recognit.* **2015**, *48*, 1577–1597. [CrossRef]
12. Illias, H.A.; Zhao Liang, W. Identification of transformer fault based on dissolved gas analysis using hybrid support vector machine-modified evolutionary particle swarm optimisation. *PLoS ONE* **2018**, *13*, e0191366. [CrossRef] [PubMed]
13. Baesens, B.; Viaene, S.; Van Gestel, T.; Suykens, J.; Dedene, G.; De Moor, B.; Vanthienen, J. *Least Squares Support Vector Machine Classifiers: An Empirical Evaluation*; Access & Download Statistics; Departement Toegepaste Economische Wetenschappen, K.U. Leuven: Leuven, Belgium, 2000.
14. Akhtar, M.S.; Gupta, D.; Ekbal, A.; Bhattacharyya, P. Feature selection and ensemble construction: A two-step method for aspect based sentiment analysis. *Knowl.-Based Syst.* **2017**, *125*, 116–135. [CrossRef]
15. Peng, H.; Long, F.; Ding, C. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Trans. Pattern Anal. Mach. Intell.* **2005**, *27*, 1226–1238. [CrossRef] [PubMed]
16. Pohjalainen, J.; Räsänen, O.; Kadioglu, S. Feature selection methods and their combinations in high-dimensional classification of speaker likability, intelligibility and personality traits. *Comput. Speech Lang.* **2015**, *29*, 145–171. [CrossRef]
17. Xu, J.; Zhou, Y.; Chen, L.; Xu, B. An unsupervised feature selection approach based on mutual information. *J. Comput. Res. Dev.* **2012**, *49*, 372–382.
18. Christina, A.; Salam, M.A.; Rahman, Q.M.; Wen, F.; Ang, S.P.; Voon, W. Causes of transformer failures and diagnostic methods–A review. *Renew. Sustain. Energy Rev.* **2018**, *82 Pt 1*, 1442–1456.
19. Singh, J.; Sood, Y.R.; Verma, P. Dissolved Gas Analysis for Transformers. *Electr. India* **2007**, *47*, 72–74, 76, 78, 82.
20. Ghoneim, S.S.; Taha, I.B. A new approach of DGA interpretation technique for transformer fault diagnosis. *Int. J. Electr. Power Energy Syst.* **2016**, *81*, 265–274. [CrossRef]
21. Li, X.; Wu, H.; Wu, D. DGA Interpretation Scheme Derived from Case Study. *IEEE Trans. Power Deliv.* **2010**, *26*, 1292–1293. [CrossRef]

sustainability

MDPI

Article

Performance Analysis of Machine Learning Algorithms for Energy Demand–Supply Prediction in Smart Grids

Eric Cebekhulu †, Adeiza James Onumanyi *,† and Sherrin John Isaac †

Advanced Internet of Things, Next Generation Enterprises and Institutions,
Council for Scientific and Industrial Research, Pretoria 0001, South Africa; ecebekhulu@csir.co.za (E.C.);
sisaac@csir.co.za (S.J.I.)
* Correspondence: aonumanyi@csir.co.za
† These authors contributed equally to this work.

Abstract: The use of machine learning (ML) algorithms for power demand and supply prediction is becoming increasingly popular in smart grid systems. Due to the fact that there exist many simple ML algorithms/models in the literature, the question arises as to whether there is any significant advantage(s) among these different ML algorithms, particularly as it pertains to power demand/supply prediction use cases. Toward answering this question, we examined six well-known ML algorithms for power prediction in smart grid systems, including the artificial neural network, Gaussian regression (GR), k-nearest neighbor, linear regression, random forest, and support vector machine (SVM). First, fairness was ensured by undertaking a thorough hyperparameter tuning exercise of the models under consideration. As a second step, power demand and supply statistics from the Eskom database were selected for day-ahead forecasting purposes. These datasets were based on system hourly demand as well as renewable generation sources. Hence, when their hyperparameters were properly tuned, the results obtained within the boundaries of the datasets utilized showed that there was little/no significant difference in the quantitative and qualitative performance of the different ML algorithms. As compared to photovoltaic (PV) power generation, we observed that these algorithms performed poorly in predicting wind power output. This could be related to the unpredictable wind-generated power obtained within the time range of the datasets employed. Furthermore, while the SVM algorithm achieved the slightly quickest empirical processing time, statistical tests revealed that there was no significant difference in the timing performance of the various algorithms, except for the GR algorithm. As a result, our preliminary findings suggest that using a variety of existing ML algorithms for power demand/supply prediction may not always yield statistically significant comparative prediction results, particularly for sources with regular patterns, such as solar PV or daily consumption rates, provided that the hyperparameters of such algorithms are properly fine tuned.

Keywords: Eskom; forecasting; hyperparameter; machine learning; tuning; wind

Citation: Cebekhulu, E.; Onumanyi, A.J.; Isaac, S.J. Performance Analysis of Machine Learning Algorithms for Energy Demand–Supply Prediction in Smart Grids. *Sustainability* **2022**, *14*, 2546. https://doi.org/10.3390/su14052546

Academic Editors: Luis Hernández-Callejo, Sergio Nesmachnow and Sara Gallardo Saavedra

Received: 22 December 2021
Accepted: 27 January 2022
Published: 22 February 2022

1. Introduction

Accurate forecasting of the power being generated and consumed in smart grid systems is crucial to ensuring grid sustainability [1]. Consequently, power demand/supply forecasting continues to be an area of contemporary research, and for this reason, machine learning (ML) algorithms have become key instruments for such forecasting obligations [2].

However, it remains unclear as to which ML algorithm performs best for power demand/supply forecasting in smart grid (SG) systems. Some specific reasons for such uncertainties are well documented in many review articles [3,4], with a few noted as follows:

- It is noted that the number of simple and complex ML algorithms/models in the literature has grown exponentially, thus making it almost impossible to compare all available models [3].

- There are many contradictory conclusions regarding the best performing algorithm(s) mainly due to the lack of proper statistical significance analyses of many output results. For example, the authors in [5,6] claim that statistical techniques (i.e., regression-based approaches) perform better than simple ML methods, whereas the findings in [7–9] suggest that ML methods typically outperform statistical techniques. Thus, such contradictory reports exist in the literature.
- Some authors can be prejudiced toward publishing only those metrics that demonstrate how well their approach may have outperformed other methods, while failing to report other relevant metrics of concern [3]. Such practices can distort the findings of such studies in favor of the suggested method(s) over existing ones, which should not be the case.
- Furthermore, many research works neglect to perform proper hyperparameter tuning exercises of the various algorithms under consideration before conducting comparison assessments. In other cases, crucial information about the source of the training and testing data is omitted, as is the proportion of the training and test split, making it difficult to replicate previously published results [3]. Note that the difference between a hyperparameter and hyperparameter tuning is that a hyperparameter is a parameter whose value is used to control the learning process and to determine the values of the model parameters of a learning algorithm, whereas hyperparameter tuning is the problem of selecting the optimal set of hyperparameters for a learning algorithm [10].

Consequently, following the above concerns, the current article describes an independent investigation of the performance of some well-known ML algorithms in terms of their use in power supply/demand prediction. This article does not propose a new ML method; rather, it provides evidence as to whether there is a true difference in using these different ML algorithms for power prediction use cases. Thus, our findings are intended to help smart grid designers make better decisions about which ML algorithm to use in their designs. Furthermore, the goal of this paper is to inform the smart grid research community that, as long as these algorithms are properly fine tuned, it may be possible to deploy any of these algorithms for prediction purposes in smart grid systems since within the limits of the dataset used in our study, there existed little or no statistically significant difference in their performance. Additionally, our paper emphasizes the importance of adhering to the best practices proposed in comparing different ML algorithms (see [3]), such as ensuring that a thorough statistical significance analysis of the output results is conducted, using multiple metrics of comparison, and providing in-depth details about the training and testing data used in the study. Thus, summarily, the contributions of the present article can be stated in the following:

1. We conducted a comparative performance analyses of six well-known ML algorithms, namely the artificial neural network (ANN), Gaussian regression (GR), k-nearest neighbor (k-NN), linear regression (LR), random forest (RF), and the support vector machine (SVM).
2. We examined three different data sources spanning across the system hourly demand, photovoltaic, and wind generation datasets from the Eskom database. We observed that the different ML algorithms considered herein performed poorly, particularly on the wind power generation dataset, which we attributed to the highly stochastic nature of the wind source.
3. A thorough statistical significance analysis of the different methods revealed that within the confines of the datasets used in this study, there was little/no significant difference in the performance of the different ML algorithms. Thus, this early observation suggests that any of the simple ML algorithms considered here can be used for demand/supply forecasting, albeit after a thorough hyperparameter tuning exercise is conducted.

The remainder of the paper is structured as follows: Section 2 presents a summary of the related work. Section 3 details the methodology to include a summary of the ML algorithms, datasets, and the metrics of performance considered in our study. Section 4 presents the results and discussion, with the conclusion drawn in Section 5. A list of mathematical symbols used in this article is provided in the Abbreviations part.

2. Related Work

This section discusses the related work, particularly those concerned with prediction analysis in smart grids using various ML algorithms. Different prediction models for microgrids are also discussed, many of which are focused on power generation and consumption. Many of these models typically implement ML techniques to forecast short-term and day-ahead electricity demands.

First, it is noted that prediction errors can lead to an imbalance between power supply and demand; thus, load forecasting is essential for transmission system operators because of the impact of prediction accuracy on power system operations. Hence, improving energy demand prediction methods could enable a power grid to become more stable. A comparison of different ML techniques for short-term demand prediction on microgrids was conducted in [11] to improve prediction accuracy. The comparison was between ensemble prediction network (EPN) and long-short term memory (LSTM), neural network, and multi-layer perceptron. The EPN technique outperformed other forecasting methods when error was evaluated on a wide range of data. It was shown that prediction accuracy influences the operational cost of energy too. In [12], the kernel-based extreme learning machine (KELM) algorithm was compared to the extreme learning machine (ELM) and the Gaussian kernel for predicting short-term electricity prices on a yearly dataset from the Australian market. The KELM technique was shown to outperform other kernel methods, but the Gaussian kernel-based ELM was more efficient for dealing with complexities in electricity pricing data and accurately predicting the price profile pattern. An automated reinforcement learning-based multi-period method for forecasting renewable power generation and load was proposed in another interesting article [13]. It was demonstrated that, when compared to traditional scheduling methods, the proposed method, along with its forecasting capability, significantly reduced operating costs while calculating at a faster rate. In a separate work, a least squares SVM (LS-SVM) coupled with the bacterial colony chemotaxis (BCC) optimization algorithm was proposed to improve the accuracy and speed of short-term load forecasting. The method was determined to achieve better accuracy and faster processing speed, compared with the ANN and LS-SVM based on grid search [14].

Various predicting techniques have been proposed to sustain the amount of energy generated to meet the demands of consumers, and some methods have been developed to enhance existing ones. For example, in [15], the ANN was compared with the multi-variable regression (MVR) and support vector machine (SVM) for improving energy dispatch for a residential microgrid based on solar power forecasting. The ANN model was most efficient in this case, with an accurate model for forecasting hourly irradiance and generated power. Unlike in [16], which perceived K-means as a new algorithm to predict irradiance intervals for stochastic generation in microgrids, improvements are always made as technology advances, as seen in [17], which indicates that the use of the regression technique is the way to go. They demonstrated that it yields improved performance since it has longer reliability and less processing time for the prediction of power generated in microgrids. Power forecasting will also be vital to the success of future energy management schemes, such as in transactive energy models [18]. In addition, IoT devices in smart grids will require efficient communication protocols for transmitting forecast data to a remote or cloud server. An efficient interface for such a purpose between CoAP and OSGP was proposed in [19], which can ensure that data are exchanged effectively between IoT devices used in home and industrial applications and an SG infrastructure. Other device development and prediction concepts can be gleaned from [20] in order to develop systems that can be used for smart grid prediction use-cases.

Furthermore, many methods have been used to forecast energy consumption, from an hour ahead to a day ahead, depending on various weather conditions. For comparison purposes, Ref. [21] stated that the SVM outperformed other algorithms for hourly prediction of load power consumption in a building. Power consumption prediction algorithms for the day ahead are either ML or AI. In [22], a hybrid AI model (a combination of feed-forward artificial neural network (FFANN), wavelet transform (WT), and simulated annealing (SA)) was used to predict power demand for a day ahead. The hybrid model was shown to be more efficient as compared to using just one method, as in [23], which implemented the neural network technique for similar day-ahead prediction conditions. Ref. [24] focused on the use of ensemble learning techniques to predict the power consumption of a building with given weather forecast information. They noted that the gradient boosted trees yielded the best performance among the different ensemble methods used. Ref. [25] evaluated different AI algorithms (ARIMA, SARIMA, SVM, XGBoost, RNN, LSTM, and LSTM-RNN) at a university campus microgrid to predict power consumption. They suggested that RNN, LSTM, and RNN-LSTM provided the best MSR, MAE, MAPE, and R-squared when compared to the other techniques used.

Table 1 is essentially a summary of these various related comparative studies. Many of these studies, like previous observations in the literature, compare only a few ML algorithms, and frequently only within the same class.

Table 1. Summary of the related studies, with key characteristics from each study compared to the others.

Ref.	Year	Methods Compared	Metrics of Comparison	Was Statistical Significance Analysis Performed?	Was Processing Time Measured?	Findings
[11]	2021	EPN, LSTM, ANN	RMSE, MAPE	No	Yes: EPN was fastest	EPN outperformed LSTM, MLP, SVR and ETR in terms of RMSE over a wide variety of data
[12]	2021	ELM Kernel-based technique	MAE, MAPE, RMSE	No	No	Kernel based methods performed better than ELM; Gaussian kernel performed better than other kernel methods.
[15]	2020	ANN, MVR, SVM	MAPE, MSE	No	No	The developed neural network model outperformed the MVR and SVM
[17]	2020	Regression, ANN	MSE, RMSE, R-squared Chi-squared	No	No	Regression approach has a better performance than some state-of-the-art method such as feed forward neural network.
[21]	2019	LR, ANN, SMO regression, SVM	MAE, RMSE, CC	No	Yes, but only for SVM	SVM performed better than other algorithms compared with.
[22]	2019	FFANN, WT, SA	MAPE, RMSE, NMAE	No	No	FFANN performed better than BP-, GA-, and PSO-FFANN schemes
[23]	2020	LSTM	RMSE, MAE	No	No	No comparison
[24]	2018	MLR, decision tree, RF, Gradient boosted trees	ARE	No	No	Gradient boosted trees performed better than others.
[25]	2021	ARIMA, SARIMA, SVM, XGBoost, RNN, LSTM, LSTM-RNN	MAE, MAPE, MSE, R-squared	No	No	Deep learning approaches such as RNN, LSTM achieved better results than time series and machine learning. While hybrid of RNN-LSTM achieved the best accuracy
Present Article	2022	ANN, GR, k-NN, LR, RF, SVM	CC, RAE, RRSE, MAE, RMSE	Yes	Yes	There was no statistical significant difference in the performance of the different methods

Furthermore, it is clear that only a few metrics are used to compare these methods, which tends to bias the conclusions that can be drawn from the comparison exercise. Most importantly, none of the recently published studies performed a statistical significance test on their output results. As a result, their conclusions may be biased, making it difficult to determine which algorithm truly performs best. Additionally notable is the absence of timing performance, which limits an ML designer's ability to make appropriate choices. Finally, the findings of these studies demonstrate that no single ML algorithm performs best across all studies. As a result, in the absence of thorough statistical significance tests, many of these conclusions may not be truly reliable. Thus, in this article, we attempt to conduct an independent study of these well-known ML algorithms in order to determine whether there is any significant difference in their performance based on thorough significance tests. Our findings will help to inform the research community in this area, as well as assist designers in making sound decisions when developing smart grid systems.

3. Methodology

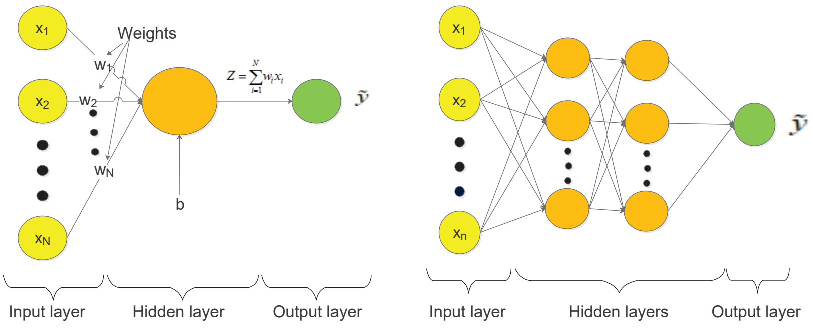
We discuss in this section the different simple ML algorithms considered in our study, the datasets used, and the metrics used to analyze the performance of the different algorithms.

3.1. Machine Learning Algorithms

There are many platforms and learning libraries that can be used to implement different ML algorithms, many of which circumvent the need to write codes. However, in this section, we present only a summary of each ML algorithm as a basis for understanding how they work. According to the meta-analysis of the recent literature provided in Table 1, the underlying ML algorithms include the regression and artificial neural network-based approaches. As a result, we considered these foundational techniques in our research because, in addition to being simple, they use fewer computational and memory resources than the more recent deep learning approaches. Furthermore, in the aftermath of new smart grid applications, which are based on the Internet of Things (IoT), it is critical to consider these simpler methods due to the limited processing and memory capacities of many IoT-based devices, which justifies the inclusion of the methods discussed below in our study.

3.1.1. Artificial Neural Network

There are many works that describe the ANN [26–28], and we aim only to present its basic structure and how it works. The ANN, also called the multilayer perceptron (MLP), typically comprises an input layer, single or multiple hidden layers, and a single or multiple output layer(s) (depending on the specific application), with each layer comprising a different number of nodes as typically represented in Figure 1.



(**a**) (**b**)

Figure 1. Representations of an ANN: (**a**) Single-layer (**b**) Multi-layer.

The input data, which contain the independent variables, also called features or attributes, are denoted as $x_1, x_2, \cdots, x_n$, whereas the output (i.e., dependent) variable is denoted as $\tilde{y}$. The weights connecting the input and hidden nodes are denoted as $w_1, w_2, \cdots, w_n$. The ANN aims to minimize the error, which is the difference between the correct y and the predicted values $\tilde{y}$ via a cost function [26]. The cost function computes this error, wherein the term "cost" refers simply to the error. The steps taken by the ANN can be summarized as follows, but in-depth details can be found in [27,29]:

1. The dot product between the inputs and weights is computed. This involves multiplying each input by its corresponding weight and then summing them up along with a bias term b. This is obtained as

$$Z = \sum_{i=1}^{N} w_i x_i + b \quad (1)$$

2. The summation of the dot products is passed through an activation function. The activation function bounds the input values between 0 and 1, and a popular function, which we used in our study, is the sigmoid activation function, stated mathematically as

$$\phi(Z) = \frac{1}{1 + e^{-Z}} \quad (2)$$

The sigmoid function returns values close to 1 when the input is a large positive value, returns 0 for large negative values, and returns 0.5 when the input is zero. It is best suited for predicting the output as a probability, which ranges between 0 and 1, which makes it the right choice for our forecasting problem. The result of the activation function is essentially the predicted output for the input features.

3. Backpropagation is conducted by first calculating the cost via the cost function, which can simply be the mean square error (MSE) given as

$$MSE = \frac{1}{N} \sum_{i}^{N} (\tilde{y}_i - y_i)^2 \quad (3)$$

where y_i is the target output value, $\tilde{y}_i$ is the predicted output value, and N is the number of observations (also called instances). Then, the cost function is minimized, where the weights and the bias are fine tuned to ensure that the function returns the smallest value possible. The smaller the cost, the more accurate the predictions. Minimization is conducted via the gradient descent algorithm, which can be mathematically represented as

$$W_x^* = W_x - a\left(\frac{\partial Error}{\partial W_x}\right) \quad (4)$$

where W_x^* is the new weight, W_x is the old weight, a is the learning rate, and $\frac{\partial Error}{\partial W_x}$ is the derivative of the error with respect to the weight, where $\partial Error$ is the cost function. The learning rate determines how fast the algorithm learns. The gradient descent algorithm iterates repeatedly (called the number of epochs) until the cost is minimized. Consequently, the steps followed can be summarized as follows:

(a) Define the inputs (i.e., features) and output variables.
(b) Define the hyperparameters.
(c) Define the activation function and its derivatives.
(d) Train the model and make predictions.

Following the preceding steps, the ANN's hyperparameters can be fine-tuned using the GridsearchCV method, with details of using the GridsearchCV well documented in [30]. The number of neurons, activation function, learning rate, momentum, batch size, and epochs are among the hyperparameters fine tuned in our study.

3.1.2. Linear and Gaussian Regression

When making day-ahead energy demand and supply predictions, we are often faced with a single input variable system, and thus, a simple linear regression model can be used for prediction purposes. Here, the model comprises an input or predictor variable that helps to predict the output variable, and this is represented by a simple linear equation. However, for generalization sake, the idea behind the regression is to estimate from a sample the parameters of the model generally written as [31]

$$\hat{y} = \beta_0 + \beta_1 x_1 + \cdots + \beta_N x_N + \epsilon \tag{5}$$

where $\hat{y}$ is the predicted output, $\beta_1, \beta_2, \cdots, \beta_N$ are the parameters (i.e., the model coefficients), $x_1, x_2, \cdots, x_N$ are the input variables (or features), and ϵ is a random error with $\epsilon \sim \mathcal{N}(0, \sigma^2)$, where σ^2 is the variance value. By determining these parameter values (i.e., β), a line of best fit can be obtained and used for prediction purposes. The method of ordinary least squares can be used for parameter estimation, and this involves minimizing the squared differences between the target and predicted outcomes (i.e., the residuals) [31]. The sum of squares of the error, termed the residual sum of squares (RSS), is computed as $RSS = \sum_i^N (y_i - \hat{y}_i)^2$, which can then be minimized using, for example, the gradient descent algorithm instead of the ordinary least squares approach. The gradient descent algorithm commences with sets of initial parameter values and advances iteratively toward a set of parameter values that minimize the function. The iterative minimization is accomplished via derivatives, which involves taking steps in the negative direction of the function gradient.

However, in using the linear regression approach, it is essential that we make assumptions regarding the structure of the function to be used, for example, by making a choice as to which is a better fit: a linear or a quadratic function. Such a choice can be independently decided upon by certain methods, such as the Gaussian regression (GR) (also called Gaussian process regression) approach [32]. Essentially, the GR generates a number of candidate functions that can model the observed data, and it attempts to find the function that best fits the data. Such a best fit function is then used for predicting future occurrences. The main difference between the GR and LR is that the GR uses a kernel, which typically represents the covariance matrix of the data [33]. Thus, the choice of the kernel function often influences strongly the performance of the GR algorithm. Further theoretical details regarding the GR algorithm can be found in [34]. The hyperparameters of the LR and GR fine tuned in this study include the attribute selection method, kernel and the filter type.

3.1.3. k-Nearest Neighbour

The k-NN algorithm is a learning algorithm that predicts the outcome of a test value (input data) based on the k nearest neighbors (i.e., other close data points) and the distance computed between them [35]. By calculating the distance between the k points in the training data closest to the test value, the test value is considered to belong to the category with the least distance. The distance measure can be based on the Euclidean, Manhattan, or Minkowski methods [36].

In using the k-NN algorithm, first the data may need to be standardized/normalized since the outcome may be fairly different due to some features having large variances. Then, it is essential to determine an optimal k value, which is often obtained via a hyperparameter tuning exercise. In this case, a range of k values are tested, and a good value is obtained that minimizes the error rate of the model.

The k-NN has remained viable in many application areas because of its simplicity and ease of application, its dependence on only two main metrics, namely the k and the distance metric, and its ability to easily add new data to the algorithm. The hyperparameters fine tuned for the k-NN are the k value and the type of neighbor search algorithm.

3.1.4. Random Forest

The RF is an ensemble of decision trees used for performing both regression and classification tasks [37]. It is built on the basis of the decision tree algorithm, which is capable of fitting complex datasets. The concept of the tree is to search for a variable–value pair within the training set and then to split this to obtain the best two child subsets. Essentially, when making predictions, each data point begins at the top of the tree (as shown in Figure 2) and then down through the branches until it reaches a leaf node, where no further branching can be achieved.

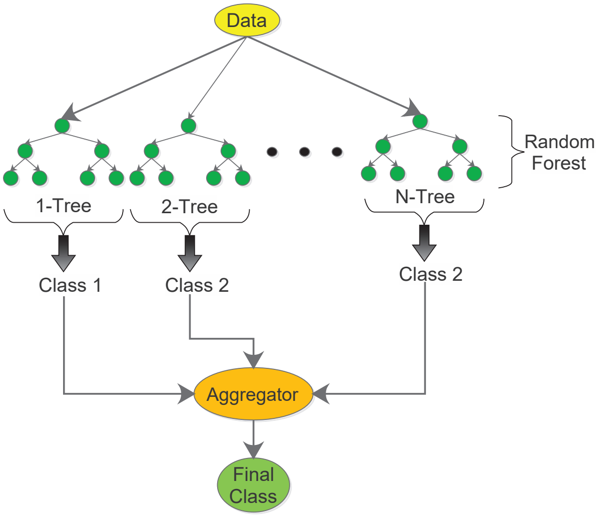


Figure 2. A representation of the concept of the random forest algorithm.

Being an ensemble approach, the RF aggregates multiple outputs generated via different sets of decision trees toward obtaining better results. The idea is to take an average over the outcome of each predictor, thus reducing the variance toward arriving at a better prediction model that presents fewer cases of overfitting the training data [38]. Thus, the RF becomes a strong learner, whereas the individual decision trees are considered weak learners. The RF algorithm is trained via the bagging method or bootstrap aggregating approach, which comprises randomly sampling subsets of the training data [39]. It then fits the model to these smaller datasets and then aggregates the predictions. This approach allows for many instances to be used repeatedly during the training phase. Essentially, the RF can be a slow algorithm since it has to grow many trees during the training stage. Further technical details regarding the RF algorithm can be accessed in [37]. The fine-tuned hyperparameters of the RF algorithm include the maximum depth and the number of iterations.

3.1.5. Support Vector Machine

The SVM aims to determine a hyperplane in an N-dimensional space, where N is the number of features, which distinctly classifies the data points [40]. The SVM is a generalized version of the maximal margin classifier, with provision for more data types to be better classified. Essentially, the SVM uses the hyperplane to separate optimally two convex hulls of points (data instances), by ensuring that the hyperplane is equidistant from the two convex hulls.

In this situation, the hyperplane is a classification border. It is usually a $N-1$ dimensional flat affine subspace, which is a line in two dimensions and a plane in three dimensions [41]. In terms of classification, the goal is to find the plane that optimizes the distance between two classes of data points. The hyperplane's size depends on the number of features. In the SVM domain, support vectors are data points on or near the hyperplane. Using these support vectors allows us to optimize the hyperplane's margin. They are called support vectors because any change in their position affects the hyperplane.

For non-linearly separated data points, the SVM adopts the concept of kernels to classify such points. A kernel refers to a function that maps lower dimensional data into higher dimensional ones. The function takes two vectors of any dimension as input and outputs a score (a dot product) that quantifies how similar the input vectors are. If the dot product is small, then the vectors are different, whereas if they are large, the vectors are similar. The SVM can use a variety of kernel functions, but one popular kernel is the Gaussian radial basis function (RBF). The RBF Gaussian kernel $K(x,y)$ is calculated as follows:

$$K(x,y) = \exp - \left(\frac{||x^2 - y^2||^2}{2\sigma^2} \right) \tag{6}$$

where x and y are N-dimensional independent variables, and σ is assumed to be the standard deviation of x and y, with $|| \bullet ||$ being the Euclidean norm. Further formal and detailed explanation and use of the SVM can be obtained in [42,43], with the following hyperparameters fine tuned in the present study, namely, the kernel and filter type.

3.2. Dataset

We considered the system hourly demand and renewable power generation data obtained from the Eskom database in our study (https://www.eskom.co.za/dataportal/, accessed on 1 October 2021). Eskom is a South African electricity public utility company established to be the Electricity Supply Commission [44]. It owns and operates a number of noteworthy power plants that provide roughly 95% of South Africa's electricity [45]. Eskom provides data on power generated, consumed, and from open cycle gas turbines, renewables and power outages. Our research focused on demand-side data and renewable energy sources, which are reflective of a typical smart grid. Thus, the use cases are as follows:

1. System hourly demand: This dataset presents the hourly power demand measured from 5 to 18 October 2021 (https://www.eskom.co.za/dataportal/demand-side/system-hourly-actual-and-forecasted-demand/, accessed on 1 October 2021). It is classed into the residual and the Republic of South Africa (RSA) contracted demand. However, we considered only the residual demand data in our study, which suffices to compare the different algorithms. The entire dataset comprised 528 data points, with the residual demand data comprising 336 data points collated from 5 to 10 October 2021, and 192 data points from the residual forecast dataset provided from 11 to 18 October 2021. In this case, a training to testing split ratio of 65% to 35% was used, respectively.
2. Hourly renewable generation: This dataset presents the hourly renewable generation per resource type, namely, from photovoltaic (PV) and wind sources (https://www.eskom.co.za/dataportal/renewables-performance/hourly-renewable-generation/, accessed on 1 October 2021). These datasets reflect only renewable sources owned by Eskom or that Eskom has contracts with. The PV and wind datasets comprised 960 data points in total, each measured per hour from 1 September 2021 to 10 October 2021. For both the PV and wind use cases, we used 80% of the dataset for training and 20% for testing. This implies that 770 data points from 1 September 2021 to 2 October 2021 were used to train each model, whereas 190 data points from 3 to 10 October 2021 were used for testing purposes. It should be noted that the term "target" used henceforth in this article refers to the actual data against which the different models are compared with during the testing phase.

The above use cases were selected because a typical grid-tied microgrid in a smart grid system can be expected to supply power to consumers both from the main grid as well as from local renewable sources. Consequently, when in the grid-tied mode, the system hourly actual demand dataset suffices as a relevant use case for prediction purposes. On the other hand, when in an island mode, the hourly renewable generation dataset becomes relevant for forecasting sake. Thus, both use cases are useful for designing energy management systems capable of day-ahead or week-ahead demand and supply forecasting in smart grids. Such knowledge helps to anticipate the amount of electricity that will be used hourly so that sufficient generation can be made available to meet such electricity demand.

3.3. Performance Metrics

As noted in Section 1, the possibility for bias arises when only a few metrics are reported while other notable metrics remain unreported [3]. Since each metric reports different information regarding an algorithm's performance, some articles tend to only report those metrics that reflect their method's improved performance while shunning others. However, to avoid such unacceptable practices, we used a variety of notable metrics to compare the different models with aim to provide a broader perspective regarding the performance of the different models. To this effect, five well-known evaluation metrics were considered, and they are discussed as follows.

3.3.1. Correlation Coefficient

The Pearson correlation coefficient (CC) r_{xy} can be computed for any model as follows

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{7}$$

where n is the sample size, x_i, y_i are the individual sample points (i.e., paired instances) indexed with i for a pair of random variables (X, Y), and

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{8}$$

is the sample mean for X and similarly obtained for Y as follows

$$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i \tag{9}$$

Essentially, the value of r_{xy} ranges from -1 to 1, where a value of 1 means that the relationship between X and Y can be described by a linear equation. In this case, all data points fall on a line. The correlation sign ($-$ or $+$) follows from the regression slope, where a $+$ sign means that Y increases as X increases and vice versa for a $-$ sign. The case of $r_{xy} = 0$ means that no correlation exists between X and Y. Other intermediate values (i.e., $0 < r_{xy} < 1$ and $-1 < r_{xy} < 0$) describe partial correlations with values closer to 1 or -1 representing a better model based on the context and purpose of the experiment.

3.3.2. Relative Absolute Error

The relative absolute error (RAE) is the ratio of the total absolute error produced by a model to the total absolute error of a simple predictor. In this case, the simple predictor is just the average of the target values. The RAE is thus computed as

$$RAE = \frac{\sum_{i=1}^{n}|P_i - A_i|}{\sum_{i=1}^{n}|\bar{A} - A_i|} \times 100\% \tag{10}$$

where P_i is the predicted value by a model for an instance i out of a total number of n instances, A_i is the target value for the instance i, and $\bar{A}$ is the mean of the target values given by

$$\bar{A} = \frac{1}{n}\sum_{i=1}^{n} A_i \tag{11}$$

One advantage of the RAE metric compared to the root mean square error (RMSE) described later is that it treats each error equally by ensuring that only the absolute value is considered and not the square of the error. Consequently, systems that are invariant to the effects of outliers can be best evaluated by the RAE instead of the RMSE.

3.3.3. Root Relative Square Error

The root relative square error (RRSE) is the ratio of the square root of the sum of the squared errors to the sum of the squared errors of a simple predictor. Again, the simple predictor is the average of the target values. The RRSE is given as

$$RRSE = \sqrt{\frac{\sum_{i=1}^{n}(P_i - A_i)^2}{\sum_{i=1}^{n}(\bar{A} - A_i)^2}} \times 100\% \tag{12}$$

where all terms remain as previously defined. By computing the square root of the relative squared error, the RRSE reduces the error to a similar magnitude range as the RAE. However, unlike the RAE, the RRSE penalizes outliers with large error values, thus allowing models with plausible outliers to be easily identified.

3.3.4. Mean Absolute Error

The mean absolute error (MAE) is a measure of the error between a pair of random variables expressing the same event. It is computed as

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|P_i - A_i| \tag{13}$$

Following (13), it can be observed that an errorless model will generate a zero MAE value, since $P_i = A_j$, thus indicating that the MAE ranges from 0 to infinity, with 0 being an ideal model. For this reason, the MAE is a boundless metric and thus, is data specific. Nevertheless, it remains a valuable metric for comparing models that are based on the same input data.

3.3.5. Root Mean Square Error

The root mean square error (RMSE) is a measure of accuracy for comparing the forecast errors of different models based on the same dataset. It is the square root of the average of the squared errors, mathematically computed as follows

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(P_i - A_i)^2}{n}}. \tag{14}$$

Since computing the RMSE involves squaring the difference between the predicted and the target values, thus, a few large differences will definitely increase the RMSE compared to the MAE. Consequently, the RMSE is sensitive to outliers, and hence useful for analyzing models with outlier tendencies.

4. Results and Discussion

In this section, we present and discuss both quantitative and qualitative results obtained from the evaluation and analysis of the ML models considered in our study. By quantitative analysis, we present and discuss the evaluation metrics as they relate to the performance of the different models. By qualitative analysis, we refer to the visual assessment of the different displays of the predicted against the target values of the different models. To this effect, firstly, we conducted a parameter tuning exercise toward ensuring that all models are evaluated based on their best parameter values. For this purpose, the GridSearchCV tool was used with discrete sets of parameter values designated per model. The system hourly demand dataset was used for the fine-tuning process. Thereafter, the fine-tuned models were tested and compared based on the hourly renewable generation dataset, and the results are discussed. The models were simulated using the Waikato Environment for Knowledge Analysis platform upon on a computer having an i7-10750H central processing unit and an NVIDIA GeForce GTX 1650 Ti GPU.

4.1. Hyper-Parameter Optimization

4.1.1. Artificial Neural Network

ANN algorithms are typically characterized by a number of hyperparameters that should be properly fine tuned to obtain models that perform optimally. These hyperparameters are the number of neurons, activation function, learning rate, momentum, batch size, and epochs. However, since the hyperparameter tuning procedure can be a cumbersome and time-consuming process, consequently, we used the sigmoid as the activation function and we kept the batch size fixed for all methods at 100, whereas the epoch was fixed at 500. All other hyperparameters were then fine tuned accordingly.

Table 2 presents the different parameter settings and their respective performances based on the CC, RAE, and RRSE. The number of hidden layers and nodes per layer is denoted as $(x_1, x_2, \cdots, x_n)$, where the number of elements (i.e., index) n represents the number of hidden layers, while the value of each element denotes the number of nodes per layer. Essentially, we examined a maximum of two hidden layers with the number of nodes per hidden layer increased from 6, 9, to 12. We then considered three states for the learning rate classed as low (0.1), medium (0.3), and high (0.5). For the momentum parameter, we examined three values at 0.1, 0.2, and 0.4. These values were selected to understand how the model performs under increasing or decreasing values. The following are our findings.

Table 2. Performance of different ANN parameter settings.

Hidden Layer	Learning Rate	Momentum	CC	RAE (%)	RRSE (%)
6	0.1	0.4	0.8909	44.6982	45.9884
6	0.3	0.2	0.8897	48.2524	49.068
6,6	0.1	0.1	0.8895	44.1209	45.73
6,6	0.5	0.4	0.8795	47.2956	49.1213
9	0.1	0.4	0.8909	44.7817	46.0154
9	0.3	0.2	0.8884	48.9404	49.2918
9,9	0.1	0.1	0.8894	44.0081	45.7508
9,9	0.5	0.4	0.8844	47.4725	51.764
12	0.1	0.4	0.8909	44.8448	46.0465
12	0.3	0.2	0.888	48.7637	49.0518
12,12	0.1	0.1	0.8894	43.8782	45.764
12,12	0.5	0.4	0.8851	46.897	51.3911

1. The model’s performance typically decreases under an increased learning rate and momentum values, irrespective of the number of hidden layers used. This implies that a low learning rate and momentum values are best suitable for an ANN model, with the values of 0.1 and 0.1 yielding the lowest error rates, respectively. This can be easily explained noting that low learning rate values imply smaller step sizes and thus higher resolutions, which leads to improved convergence to better approximations.
2. A model with two hidden layers with 12 nodes per layer yielded the lowest error rates under a low learning rate and momentum values. Although this configuration cannot be generalized for all ANN models, it yielded the lowest error rate for the present use case. Furthermore, we note that increasing the number of nodes above 12 produced no improvement in model performance.
3. Generally, under the same low learning rate and momentum values, we observed that the double-layered model performed marginally better than the single layer configuration. For example, considering in Table 2 the best model of (12,12) hidden layer configuration, and learning rate and momentum of 0.1 each, we obtained a 2.155% decrease in the error rate when using the double-layered model instead of the single-layered model of same number of nodes.

4. Since there is no single fixed global configuration or model for all possible use cases, it becomes vital to ensure that a model's hyperparameters are accurately fine tuned. For example, by fine tuning our model, we achieved a 10.344% error reduction rate in using a double-layer model with 12 nodes per layer (learning rate = 0.1, momentum = 0.1) over a single-layer model with 9 nodes (learning rate = 0.3, momentum = 0.2).

4.1.2. Gaussian Regression

The following hyperparameters of the GR algorithm were fine tuned, namely, the kernel and the filter type. The kernels considered were the polynomial (poly) kernel, radial basis function (RBF), and the normalized polynomial kernel. The filter types included either the normalization or standardization of the training data. Our findings from the results in Table 3 are noted as follows:

1. A combination of the poly kernel and standardization of the training data led to the best model, which yielded the lowest RAE and RRSE values of 44.7277% and 45.5645%, respectively.
2. Hyperparameter tuning of the GR algorithm can achieve as much as 51.387% and 50.301% error reduction rate in the RAE and RRSE, respectively, thus emphasizing the importance of hyperparameter tuning.
3. With RAE and RRSE differences of 3.121% and 3.645%, respectively, there exists little/no significant advantage in using either the normalization or standardization of the training data as it pertains to the poly kernel. Consequently, the most important parameter is simply the choice of the kernel to be used.
4. We presume that the RBF kernel may have performed poorly owing to the large size of the training dataset, which is a well-known limitation of the RBF. Nevertheless, it is noted that performance improvement can yet be achieved by standardizing the training data.

Table 3. Performance of different GR parameter settings.

Kernel	Filter Type	CC	RAE (%)	RRSE (%)
Poly kernel	Normalized training data	0.8905	46.1688	47.2879
Poly kernel	Standardize training data	0.8905	44.7277	45.5645
RBF	Normalized training data	0.8905	92.0074	91.6804
RBF	Standardize training data	0.8905	52.0316	52.7159
Normalized Poly Kernel	Normalized training data	0.8905	46.1688	47.2879
Normalized Poly Kernel	Standardize training data	0.8905	46.045	47.1676

4.1.3. k-Nearest Neighbor

The k-NN algorithm is characterized by one major parameter, which is the k parameter. The neighbor search method is another parameter; however, the linear search approach based on the Euclidean distance metric was used in our study. The following k values were selected as $k = 1, 3, 5$, and 10. The results obtained are presented in Table 4.

Table 4. Performance of different k-NN parameter settings.

K	Neighbour Search Algorithm	CC	RAE (%)	RRSE (%)
1	Euclidean (LNNSearch)	0.8905	44.7124	45.5615
3	Euclidean (LNNSearch)	0.8905	44.7124	45.5615
5	Euclidean (LNNSearch)	0.8905	44.7124	45.5615
10	Euclidean (LNNSearch)	0	100	100

We observed that the same error rate (i.e, RAE and RRSE values) was obtained for parameters $k = 1, 3$, and 5. At $k = 10$, a large error rate of 100% was realized, thus implying that large K values are inappropriate for use under the present use case. With lower k values yielding the same results, it is suitable to use $k = 1$ since it presents the least computational demand for the model.

4.1.4. Linear Regression

The linear regression (LR) method can be improved based on the choice of the attribute selection method. We tested the LR algorithm without any attribute selection method, as well as with the M5 and greedy attribute selection method. The results obtained are presented in Table 5 with little or no difference between the different selection methods. Using an attribute selection method achieved only about 0.764% reduction in the error rate over the use of the no-selection method. Thus, for power demand prediction purposes, it is sufficient to apply the LR method without any attribute selection method. This is expected since there exist only the day and time as the main input attributes for forecasting purposes, thus attribute selection introduces no significant performance advantage.

Table 5. Performance of different LR parameter settings.

Attribute Selection Method	CC	RAE (%)	RRSE (%)
No attribute selection	0.8905	44.7124	45.5615
M5 method	0.89	44.3709	45.657
Greedy methods	0.89	44.3709	45.657

4.1.5. Random Forest

The random forest (RF) algorithm has a few parameters to be fine tuned, namely, the maximum depth and the number of iterations. In our study, a combination of three parameters were examined with progressively increasing values and the results obtained are presented in Table 6. We found that increasing the maximum depth and number of iterations barely resulted in 0.409% and 0.378% decreases in the RAE and RRSE error rates, respectively. This insignificant difference in the error rate implies that using low maximum depth values and number of iterations will be suitable in using the RF algorithm for power demand prediction purposes. It also may present faster computational time since fewer iterative steps are considered during the algorithmic process.

Table 6. Performance of different RF parameter settings.

Max. Depth	Iterations	CC	RAE (%)	RRSE (%)
0	100	0.8901	44.909	45.7124
10	200	0.8904	44.8519	45.6262
20	500	0.8907	44.7255	45.5394

4.1.6. Support Vector Machine

The support vector machine (SVM) algorithm was optimized by tuning the kernel and filter type parameters to improve its performance. The results obtained are presented in Table 7. We found that a combination of the poly kernel and the normalization of the training data resulted in the least error rates across both RAE and RRSE. In this case, both the polynomial and normalized polynomial kernel combined with normalization of the training data achieved the same performance. However, we note that it will be of greater value computation wise to avoid the normalization expenses of the poly kernel, thus implying that using the simple poly kernel should suffice for the present case. Similar to the GR algorithm, the RBF kernel yielded the largest error rates with the same plausible reasons as stated for the GR algorithm applying as well to the SVM algorithm. Summarily, an average of 38.15% reduction in the error rate was achieved by using the poly kernel over the RBF, thus reemphasizing the importance of hyperparameter tuning in the use of ML algorithms.

Table 7. Performance of different SVM parameter settings.

Kernel	Filter Type	CC	RAE (%)	RRSE (%)
Poly kernel	Normalized training data	0.8836	44.2875	47.1576
Poly kernel	Standardize training data	0.8835	44.2983	47.1763
RBF	Normalized training data	0.8654	71.6087	73.5864
RBF	Standardize training data	0.8703	46.9236	50.3859
Normalize Poly Kernel	Normalized training data	0.8836	44.2875	47.1576
Normalize Poly Kernel	Standardize training data	0.8835	44.3023	47.1848

4.1.7. Comparison of the Different Methods

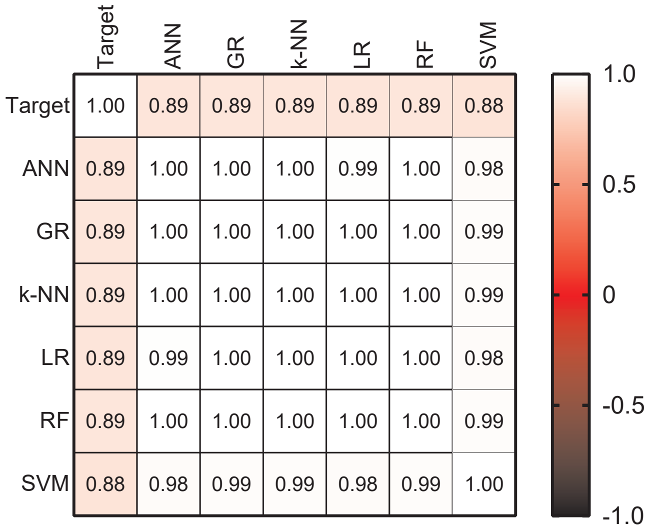
Following the hyperparameter tuning process, the best performing models of the different algorithms were compared, and the results obtained are presented in Table 8. To this effect, the following metrics were compared across the different models, namely CC, RAE, RRSE, MAE, and RMSE. Our findings indicate that although it may seem that some algorithms performed better than others, nevertheless, the performance gap suffices only marginally. For example, there existed only a 1.899% reduction in the error rate in using the ANN over the GR model in terms of their RAE. A difference of about 3.553% in the RRSE existed between the SVM and the RF algorithm. Thus, suggesting an insignificant difference between the different models, sequel to a proper hyperparameter tuning process.

Table 8. Comparison of the different methods based on their best parameter settings.

Methods	CC	RAE (%)	RRSE (%)	MAE	RMSE
ANN	0.8894	43.8782	45.764	833.8811	1046.1255
GR	0.8905	44.7277	45.5645	850.8696	1040.5409
k-NN	0.8905	44.7124	45.5615	850.5788	1040.4722
LR	0.89	44.3709	45.657	844.0817	1042.6523
RF	0.8907	44.7255	45.5394	850.5656	1039.2409
SVM	0.8836	44.2875	47.1576	842.4953	1076.9213

No model performed best across all the different metrics, thus emphasizing the need to avoid comparing different ML models using only a single metric. For example, although the ANN performed best considering the RAE, it generated the smallest RRSE values compared to the other models. Since these different metrics tell different stories, it is essential to consider our analysis across each metric as against a single metric. To this effect, by rendering a higher RRSE value, we note that the ANN model may have been plagued by more outliers than the other methods. This observation is again supported by examining the MAE against the RMSE in Table 8, which shows a higher RMSE than other methods, except the SVM.

In addition, we examined the CC values of the different models, with results of the correlation matrix presented in Figure 3. By comparing the CC achieved by the different models against the target demand, we observed that a $CC < 0.9$ was obtained across all models. This implies a good positive correlation between the predicted and the target demand values. In addition, we can observe that a $CC \approx 1$ was obtained between the different models, further emphasizing that the different models all predicted the same values. In particular, the ANN, GR, KNN and RF models all performed equally with little to distinguish them.



	Target	ANN	GR	k-NN	LR	RF	SVM
Target	1.00	0.89	0.89	0.89	0.89	0.89	0.88
ANN	0.89	1.00	1.00	1.00	0.99	1.00	0.98
GR	0.89	1.00	1.00	1.00	1.00	1.00	0.99
k-NN	0.89	1.00	1.00	1.00	1.00	1.00	0.99
LR	0.89	0.99	1.00	1.00	1.00	1.00	0.98
RF	0.89	1.00	1.00	1.00	1.00	1.00	0.99
SVM	0.88	0.98	0.99	0.99	0.98	0.99	1.00

Figure 3. The correlation matrix of the different methods for the system hourly demand dataset.

Finally, in quantitative terms, a Tukey comparative test of the different models was performed, and the output results are documented in Table 9. The Tukey test is a multiple comparison procedure that can be used to find means that are significantly different from each other. The aim in using this test is to quantitatively determine whether there exists a significant difference between the mean results obtained across the different models or not. Further details regarding the Tukey test can be accessed in [46]. The symbols used to interpret the range of the *p*-values, *p*, obtained for all the Tukey tests reported in this article, are provided in Table 10. An examination of Table 9 indicates that there was no significant (ns) difference between the target and predicted data of the different models (see column 5 of Table 9). It also confirms that there was no significant difference between all other methods as well, with *p*-values all averagely being greater than 0.997. These results support the correlation findings of Figure 3, further emphasizing that following a proper hyperparameter tuning exercise of the different algorithms, they all perform, on average, the same, with little or no significant difference between them.

4.1.8. Visual Assessment of Predicted Values of the Different Methods

Figure 4 presents the target and predicted values generated by the different models. It is immediately obvious that the graphs are practically overlapped, which further confirms that the models achieved similar performance levels. Essentially, there was only very little difference between the predictive values and the target data prior to the 120th h (i.e., day 5), beyond which a significant error difference was observed. This can be explained noting that a stable pattern existed within the first 5 days, followed by some drop in the target demand level between the 6th and 7th day (i.e., from 120 to 168 h), a period which was not properly tracked by the different methods. This implies that the different ML methods may perform best under conditions with well-defined patterns, and otherwise under heavy stochastic conditions.

4.2. Hourly Renewable Generation

In this section, we discuss our findings pertaining to the photovoltaic (PV) and wind hourly generation datasets. We note that the best-performing models obtained in the hyperparameter tuning section were used here.

Table 9. System hourly demand: Tukey test comparison of the performance of the different models.

Comparison	Mean Diff.	95.00% CI of Diff.	Below Threshold?	Summary	Adjusted *p* Value
Target vs. ANN	−48.69	−669.1 to 571.7	No	ns	>0.9999
Target vs. GR	58.07	−562.3 to 678.5	No	ns	>0.9999
Target vs. k-NN	58.07	−562.3 to 678.5	No	ns	>0.9999
Target vs. LR	58.07	−562.3 to 678.5	No	ns	>0.9999
Target vs. RF	63.45	−556.9 to 683.8	No	ns	>0.9999
Target vs. SVM	−125.1	−745.5 to 495.3	No	ns	0.997
ANN vs. GR	106.8	−513.6 to 727.2	No	ns	0.9987
ANN vs. k-NN	106.8	−513.6 to 727.2	No	ns	0.9987
ANN vs. LR	106.8	−513.6 to 727.2	No	ns	0.9987
ANN vs. RF	112.1	−508.2 to 732.5	No	ns	0.9983
ANN vs. SVM	−76.37	−696.8 to 544.0	No	ns	0.9998
GR vs. k-NN	−0.00167	−620.4 to 620.4	No	ns	>0.9999
GR vs. LR	0.00125	−620.4 to 620.4	No	ns	>0.9999
GR vs. RF	5.383	−615.0 to 625.8	No	ns	>0.9999
GR vs. SVM	−183.1	−803.5 to 437.3	No	ns	0.9767
k-NN vs. LR	0.002917	−620.4 to 620.4	No	ns	>0.9999
k-NN vs. RF	5.385	−615.0 to 625.8	No	ns	>0.9999
k-NN vs. SVM	−183.1	−803.5 to 437.3	No	ns	0.9767
LR vs. RF	5.382	−615.0 to 625.8	No	ns	>0.9999
LR vs. SVM	−183.1	−803.5 to 437.3	No	ns	0.9767
RF vs. SVM	−188.5	−808.9 to 431.9	No	ns	0.973

Table 10. The *p*-value range and their corresponding symbol and interpretation used in the Tukey tables.

Symbol	Range	Interpretation
ns	$p > 0.05$	not significant
*	$p \leq 0.05$	weakly significant
**	$p \leq 0.01$	significant
***	$p \leq 0.001$	very significant
****	$p \leq 0.0001$	extremely significant

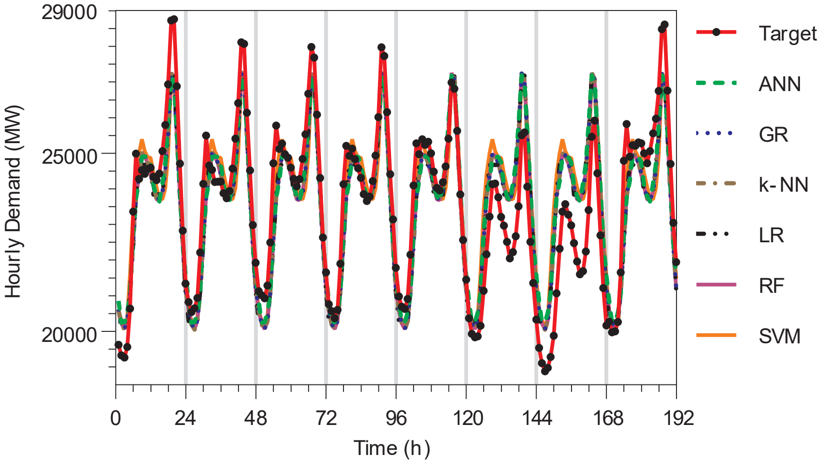


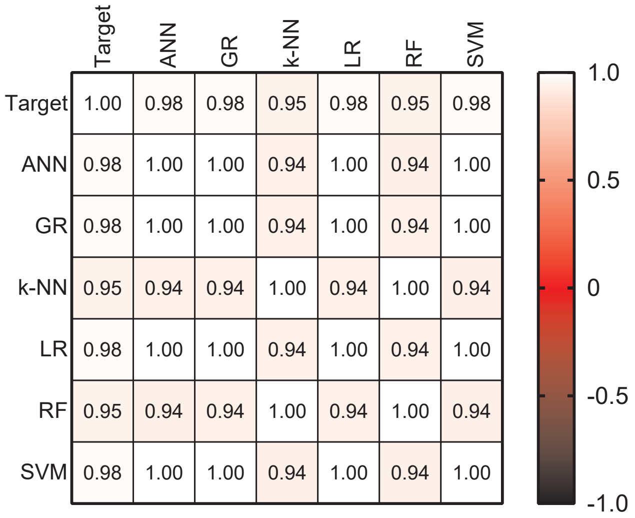
Figure 4. System hourly demand: target demand compared against the predicted demand generated by the different models using their best hyper-parameter values.

4.2.1. Photovoltaic Generation

The performance results of the different models based on the PV dataset are presented in Table 11. In terms of the CC, although the GR model achieved only a slightly higher margin (<0.001) than the ANN and LR models, we can easily conclude that there was no significant difference between the different models. This implies that the predicted results are highly positively correlated with the target demand. Similar high CC values were also obtained between the different models as shown in the correlation matrix of Figure 5. Therein, it can be seen that only the k-NN and RF models had slightly lower CC values to the other models. This may be because both models achieved the lowest CC value as against the target demand. Nevertheless, for use cases where only the data pattern suffices as the main interest to the designer, then any ML method can be used.

Table 11. Performance of the different methods for photovoltaic (PV) power generation.

Methods	CC	RAE (%)	RRSE (%)	MAE	RMSE
ANN	0.9833	21.7619	27.9011	163.8525	231.0859
GR	0.9835	16.292	23.1943	122.6677	192.1026
k-NN	0.9460	16.212	23.1527	122.0658	191.7581
LR	0.9834	16.333	23.1825	122.9761	192.0049
RF	0.9460	16.2378	23.1726	122.2594	191.9225
SVM	0.9824	15.3522	22.3847	115.5917	185.3969



	Target	ANN	GR	k-NN	LR	RF	SVM
Target	1.00	0.98	0.98	0.95	0.98	0.95	0.98
ANN	0.98	1.00	1.00	0.94	1.00	0.94	1.00
GR	0.98	1.00	1.00	0.94	1.00	0.94	1.00
k-NN	0.95	0.94	0.94	1.00	0.94	1.00	0.94
LR	0.98	1.00	1.00	0.94	1.00	0.94	1.00
RF	0.95	0.94	0.94	1.00	0.94	1.00	0.94
SVM	0.98	1.00	1.00	0.94	1.00	0.94	1.00

Figure 5. The correlation matrix of the different methods for the photovoltaic power generation dataset.

By examining the error performance of the different models via the RAE and RRSE in Table 11, it can be observed that the ANN performed the poorest. Thus, it can be said that a 25.503% decrease in the RAE can be achieved by using the k-NN instead of the ANN for the PV power prediction use case. Although this seems large, nevertheless, further analysis following the Tukey comparative test suggests that there was no significant difference between the predicted means of the different models. This can be seen in Table 12, where it is concluded that there was no significant difference in the predicted means of the different models. Thus, this suggests that any model may suffice for PV power forecasting purposes sequel to a proper hyperparameter tuning exercise.

Table 12. Photovoltaic power generation: Tukey test comparison of the performance of the different models.

Comparison	Mean Diff.	95.00% CI of Diff.	Below Threshold?	Summary	Adjusted *p* Value
Target vs. ANN	144.9	−81.66 to 371.4	No	ns	0.4884
Target vs. GR	97.92	−128.6 to 324.4	No	ns	0.8628
Target vs. k-NN	97.91	−128.6 to 324.4	No	ns	0.8628
Target vs. LR	97.92	−128.6 to 324.4	No	ns	0.8628
Target vs. RF	98.13	−128.4 to 324.7	No	ns	0.8616
Target vs. SVM	87.07	−139.4 to 313.6	No	ns	0.9173
ANN vs. GR	−46.94	−273.5 to 179.6	No	ns	0.9965
ANN vs. k-NN	−46.95	−273.5 to 179.6	No	ns	0.9965
ANN vs. LR	−46.94	−273.5 to 179.6	No	ns	0.9965
ANN vs. RF	−46.73	−273.3 to 179.8	No	ns	0.9965
ANN vs. SVM	−57.79	−284.3 to 168.7	No	ns	0.9891
GR vs. k-NN	−0.00628	−226.5 to 226.5	No	ns	>0.9999
GR vs. LR	0.00178	−226.5 to 226.5	No	ns	>0.9999
GR vs. RF	0.2102	−226.3 to 226.7	No	ns	>0.9999
GR vs. SVM	−10.85	−237.4 to 215.7	No	ns	>0.9999
k-NN vs. LR	0.008063	−226.5 to 226.5	No	ns	>0.9999
k-NN vs. RF	0.2165	−226.3 to 226.7	No	ns	>0.9999
k-NN vs. SVM	−10.84	−237.4 to 215.7	No	ns	>0.9999
LR vs. RF	0.2084	−226.3 to 226.7	No	ns	>0.9999
LR vs. SVM	−10.85	−237.4 to 215.7	No	ns	>0.9999
RF vs. SVM	−11.06	−237.6 to 215.5	No	ns	>0.9999

Finally, a visual assessment of the predicted against the target PV generation results is presented in Figure 6. Here, it is observed that a close performance was achieved between the predicted values of the different models and the target data. The overlapping graphs in Figure 6 also confirm that the models all performed similarly with little to distinguish them visually. Since the pattern obtained for PV generation demonstrates strong regularity with peak generation often obtained during midday (at peak sunshine), consequently any model can be used for predictive purposes, typically after properly tuning the model's hyperparameters.

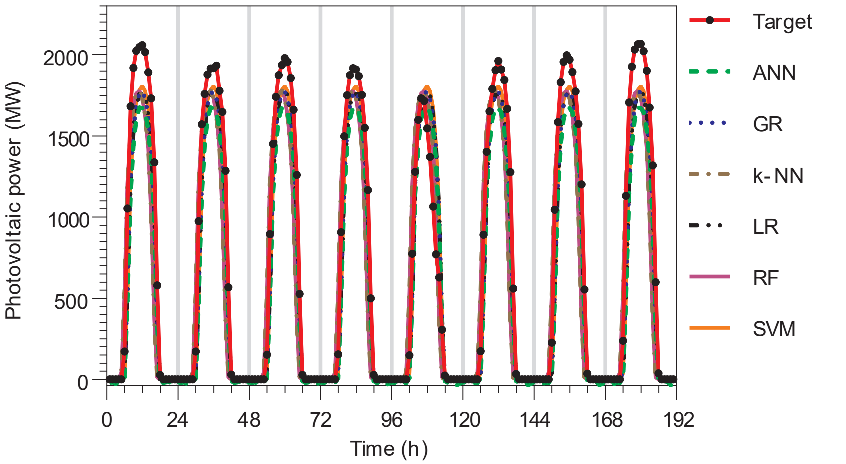


Figure 6. Photovoltaic power generation: Target and predicted demand generated by the different models.

4.2.2. Wind Generation

The performance of the different models based on the wind power generation dataset is presented in Table 13. It is immediately clear that the models performed poorly under this use case, with each model particularly suffering from very high error rates and low CC values. Albeit low, it is also seen that the CC values are, on average, the same for the models when compared to the target data. However, high and positive CC values are obtained when compared between the different models as shown in the correlation matrix of Figure 7. This confirms that the different models all performed similarly with almost perfect correlation between their predicted values.

Table 13. Performance of the different methods for wind power generation.

Methods	CC	RAE (%)	RRSE (%)	MAE	RMSE
ANN	0.558	98.7794	99.8968	279.4048	344.7094
GR	0.5559	80.9548	85.1445	228.9867	293.8042
k-NN	0.5559	80.9542	85.1432	228.9848	293.7998
LR	0.5486	81.229	85.609	229.7621	295.4071
RF	0.5565	80.7043	84.9551	228.2781	293.1508
SVM	0.5884	77.8575	81.5118	220.2257	281.269

The error rates as measured via the RAE and RRSE indicate that the ANN generated the highest error rate, and thus is reported as the poorest performer. Essentially, the RRSE values of each model return higher than their corresponding RAE values, which indicates the presence of outliers across the different models under the wind prediction use case. On the other hand, the SVM model suffices as the best performer, as it achieved a 21.18% reduction in the RAE as compared with the ANN model.

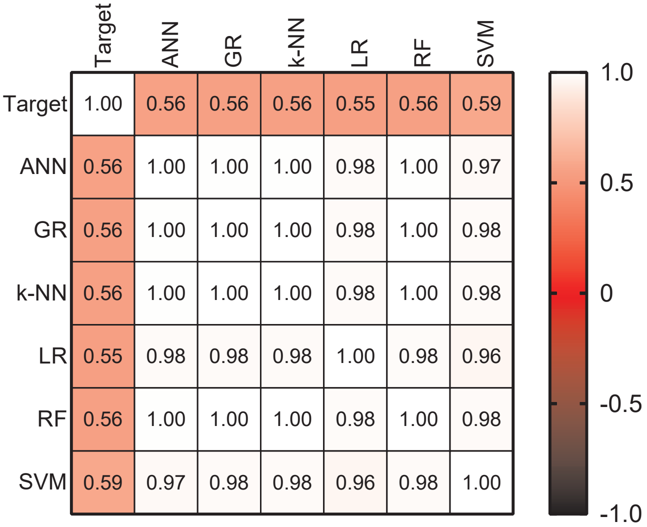


Figure 7. The correlation matrix of the different methods for the wind power generation dataset.

A Tukey test was conducted to examine the differences in the mean values of the models, and the results obtained are presented in Table 14. We observed that, unlike in the PV power prediction and the system hourly demand datasets, there was a significant difference in the mean performance of the different models and the target data. This can be seen in column 4 of Table 14 with very low associated *p*-values, where the performance of the ANN model is also indicated to be significantly different from all other models.

Table 14. Wind power generation: Tukey test comparison of the performance of the different models.

Comparison	Mean Diff.	95.00% CI of Diff.	Below Threshold?	Summary	Adjusted p Value
Target vs. ANN	−213	−274.4 to −151.6	Yes	****	<0.0001
Target vs. GR	−112.4	−173.8 to −50.96	Yes	****	<0.0001
Target vs. k-NN	−112.4	−173.8 to −50.96	Yes	****	<0.0001
Target vs. LR	−112.7	−174.1 to −51.27	Yes	****	<0.0001
Target vs. RF	−111	−172.4 to −49.56	Yes	****	<0.0001
Target vs. SVM	−94.63	−156.1 to −33.19	Yes	***	0.0001
ANN vs. GR	100.6	39.16 to 162.0	Yes	****	<0.0001
ANN vs. k-NN	100.6	39.16 to 162.0	Yes	****	<0.0001
ANN vs. LR	100.3	38.86 to 161.7	Yes	****	<0.0001
ANN vs. RF	102	40.56 to 163.4	Yes	****	<0.0001
ANN vs. SVM	118.4	56.93 to 179.8	Yes	****	<0.0001
GR vs. k-NN	−0.00093	−61.44 to 61.43	No	ns	>0.9999
GR vs. LR	−0.3068	−61.74 to 61.13	No	ns	>0.9999
GR vs. RF	1.396	−60.04 to 62.83	No	ns	>0.9999
GR vs. SVM	17.77	−43.67 to 79.21	No	ns	0.979
k-NN vs. LR	−0.3059	−61.74 to 61.13	No	ns	>0.9999
k-NN vs. RF	1.397	−60.04 to 62.83	No	ns	>0.9999
k-NN vs. SVM	17.77	−43.67 to 79.21	No	ns	0.979
LR vs. RF	1.703	−59.73 to 63.14	No	ns	>0.9999
LR vs. SVM	18.08	−43.36 to 79.51	No	ns	0.9771
RF vs. SVM	16.37	−45.06 to 77.81	No	ns	0.9862

Finally, a visual assessment of the predicted values of the different models can be made, based on the results of Figure 8. We observed that the different models only matched the rising patterns of the target data while failing to track periods of low wind power generation. This implies that the inherent irregularities in the wind power generation pattern typically limited the output performance of the different models. We also observed that the predicted values of the ANN model deviated largely from the target as well as from the other models, thus justifying its poor performance as noted in Tables 13 and 14. Consequently, because of the highly stochastic nature of wind, it may be difficult to apply ML models for predicting wind power generation, thus warranting the need for improved methods in this regard.

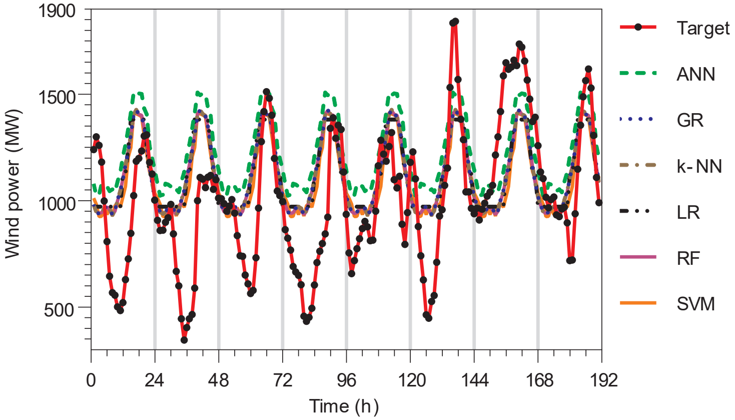


Figure 8. Wind generation: Target and predicted demand generated by the different models.

4.2.3. Runtime Performance of the Different Algorithms

We performed a runtime evaluation of the various algorithms on both the PV and wind datasets, and the results are shown in Table 15. To begin, it is important to note that the following conditions were met prior to conducting these experiments:

1. The same datasets (i.e., PV and wind data) were used to evaluate each algorithm.
2. Both the training and testing runtime performance was measured and reported.
3. To ensure that no extra processing time was incurred by the PC, only the simulation software was kept running as the foreground process during each simulation period. This was accomplished by closing all other foreground processes in the PC's task manager.
4. Finally, the timing results shown in Table 15 were obtained by averaging the results of 50 independent runs of each algorithm.

Table 15. Timing performance of the different algorithms under both the PV and wind datasets.

	PV		Wind	
Methods	**Training Time (s)**	**Test Time (s)**	**Training Time (s)**	**Test Time (s)**
ANN	2.14	0.08	2.24	0.07
GR	0.63	0.26	0.55	0.28
k-NN	-	0.1	-	0.09
LR	0.04	0.09	0.02	0.07
RF	0.25	0.15	0.12	0.08
SVM	0.52	0.07	0.14	0.07

Table 15 shows the empirical run-time results of each algorithm. However, it should be noted that because the k-NN is an unsupervised method, there was no need for a training process, and thus, no results are provided for it. According to these results, the LR achieved the shortest training time in both datasets, while the SVM algorithm achieved the quickest testing time in the PV dataset while having the same testing time as the ANN and LR in the wind dataset. Often, because testing time is most important to the user during real-time operation, we note that the SVM performed best; however, statistical significant analysis of these timing results shows otherwise in Table 16.

Table 16. Statistical significance test (Tukey's comparison test) of the test time of the different algorithms.

Tukey's Multiple Comparisons Test	Mean Diff.	95.00% CI of Diff.	Below Threshold?	Summary	Adjusted p Value
ANN vs. GR	−0.195	−0.2832 to −0.1068	Yes	***	0.001
ANN vs. k-NN	−0.02	−0.1082 to 0.06825	No	ns	0.9326
ANN vs. LR	−0.005	−0.09325 to 0.08325	No	ns	0.9999
ANN vs. RF	−0.04	−0.1282 to 0.04825	No	ns	0.5242
ANN vs. SVM	0.005	−0.08325 to 0.09325	No	ns	0.9999
GR vs. k-NN	0.175	0.08675 to 0.2632	Yes	**	0.0017
GR vs. LR	0.19	0.1018 to 0.2782	Yes	**	0.0011
GR vs. RF	0.155	0.06675 to 0.2432	Yes	**	0.0033
GR vs. SVM	0.2	0.1118 to 0.2882	Yes	***	0.0008
k-NN vs. LR	0.015	−0.07325 to 0.1032	No	ns	0.9784
k-NN vs. RF	−0.02	−0.1082 to 0.06825	No	ns	0.9326
k-NN vs. SVM	0.025	−0.06325 to 0.1132	No	ns	0.8545
LR vs. RF	−0.035	−0.1232 to 0.05325	No	ns	0.6371
LR vs. SVM	0.01	−0.07825 to 0.09825	No	ns	0.9964
RF vs. SVM	0.045	−0.04325 to 0.1332	No	ns	0.4217

It should be noted that only the test time results of Table 15 for both the PV and wind datasets were subjected to the Tukey statistical test. Thus, the Tukey test results in Table 16 reveal that there were no statistically significant (ns) differences in the test time of the different algorithms, albeit for the GR algorithm, which yielded the longest test time compared to the other methods. The GR algorithm's relatively slower performance may be attributed to the effect of the Gaussian kernel, which is known to add additional processing requirements to the method. However, because the difference in the testing time performance was less than 0.195 s across all methods (see column 2 of Table 16), it is possible to conclude that any of these algorithms can be used for real-time power demand/supply prediction use cases in smart grid systems.

5. Conclusions

The goal of this study was to determine whether there is a statistically significant difference in the performance of various well-known simple machine learning (ML) models when they are applied to the prediction of power demand and supply. In order to accomplish this, six well-known machine learning methods were tested using data from the Eskom database, which included hourly system demand and renewable generation datasets. The ML algorithms considered include the artificial neural network, Gaussian regression, K-nearest neighbor, linear regression, random forest, and the support vector machine, among other methods of data analysis. Fairness was achieved by ensuring that the hyperparameters of each algorithm were fine tuned to the greatest degree possible. Our findings suggest that, within the confines of the datasets used in this study, there was little/no statistically significant difference between the different models in terms of both quantitative and qualitative measures, which is particularly noteworthy, given that they were all meticulously fine tuned. Additionally mentioned is the importance of reporting as many metrics as possible, particularly the correlation coefficient and absolute and squared errors, in order to ensure that fair conclusions are formed when comparing different machine learning algorithms. Based on the fact that each metric often reports a separate performance measure and that selective reporting may result in erroneous conclusions, this requirement is recommended. Furthermore, when it came to estimating the wind power generation dataset, all of the models performed poorly, which we attributed to the extremely stochastic nature of wind energy as a source of energy, as previously stated in the literature. This may imply that improved models for smart grid systems may be required, particularly in areas where wind power constitutes a significant portion of the generated electricity. In spite of this, it is possible that any ML model can still be used for power prediction in smart grid systems, particularly in situations where demand and generation follow regular patterns, and provided that the model's hyperparameters are properly tuned based on the type of input data being used. Finally, we stress that further robust investigations, particularly those based on the use of larger datasets from a wider range of sources, should be strongly encouraged in order to either substantiate or refute the conclusions of the present paper.

Author Contributions: These authors E.C., A.J.O. and S.J.I. contributed equally to this work. Conceptualization, A.J.O., E.C. and S.J.I.; methodology, E.C. and A.J.O.; writing—original draft preparation, A.J.O. and E.C.; writing—review and editing, A.J.O. and S.J.I.; supervision, A.J.O. and S.J.I.; funding acquisition, S.J.I. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the COUNCIL FOR SCIENTIFIC AND INDUSTRIAL RESEARCH (CSIR) with project number 05400 054AT KR2EEMG. and The APC was funded by project number 05400 054AT KR2EEMG.

Institutional Review Board Statement: Not applicable

Informed Consent Statement: Not applicable

Data Availability Statement: Not applicable

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this article:

ANN	Artificial neural network
CC	Correlation coefficient
ELM	Extreme learning machine
EPN	Ensemble prediction network
FFANN	Feed-forward artificial neural network
GR	Gaussian regression
KELM	Kernel-based extreme learning machine
k-NN	k-nearest neighbor
LR	Linear regression
LSTM	Long short-term memory
MAE	Mean absolute error
ML	Machine learning
MLP	Multilayer perceptron
MSE	Mean square error
MVR	Multi-variable regression
PV	Photovoltaic
RAE	Relative absolute error
RBF	Radial basis function
RF	Random forest
RMSE	Root mean square error
RRSE	Root relative square error
RSA	Republic of South Africa
RSS	Residual sum of squares
SA	Simulated annealing
SVM	Support vector machine
WT	Wavelet transform
$[\bullet]$	Brackets
$(\bullet)$	Parentheses
$\sqrt{\bullet}$	Square root
dy/dx	Derivative
$\|\| \bullet \|\|$	Euclidean norm
$\sum$	Summation
$\| \bullet \|$	Absolute value

References

1. Danish, M.S.S.; Senjyu, T.; Funabashia, T.; Ahmadi, M.; Ibrahimi, A.M.; Ohta, R.; Howlader, H.O.R.; Zaheb, H.; Sabory, N.R.; Sediqi, M.M. A sustainable microgrid: A sustainability and management-oriented approach. *Energy Procedia* **2019**, *159*, 160–167. [CrossRef]
2. Nespoli, A.; Ogliari, E.; Pretto, S.; Gavazzeni, M.; Vigani, S.; Paccanelli, F. Electrical Load Forecast by Means of LSTM: The Impact of Data Quality. *Forecasting* **2021**, *3*, 91–101. [CrossRef]
3. Lago, J.; Marcjasz, G.; Schutter, B.D.; Weron, R. Forecasting day-ahead electricity prices: A review of state-of-the-art algorithms, best practices and an open-access benchmark. *Appl. Energy* **2021**, *293*, 1–21. [CrossRef]
4. Hong, T.; Pinson, P.; Wang, Y.; Weron, R.; Yang, D.; Zareipour, H. Energy Forecasting: A Review and Outlook. *IEEE Open Access J. Power Energy* **2020**, *7*, 376–388. [CrossRef]
5. Uniejewski, B.; Weron, R.; Ziel, F. Variance Stabilizing Transformations for Electricity Spot Price Forecasting. *IEEE Trans. Power Syst.* **2018**, *33*, 2219–2229. [CrossRef]
6. Marcjasz, G.; Uniejewski, B.; Weron, R. On the importance of the long-term seasonal component in day-ahead electricity price forecasting with NARX neural networks. *Int. J. Forecast.* **2019**, *35*, 1520–1532. [CrossRef]
7. Wang, L.; Zhang, Z.; Chen, J. Short-Term Electricity Price Forecasting with Stacked Denoising Autoencoders. *IEEE Trans. Power Syst.* **2017**, *32*, 2673–2681. [CrossRef]
8. Ugurlu, U.; Oksuz, I.; Tas, O. Electricity Price Forecasting Using Recurrent Neural Networks. *Energies* **2018**, *11*, 1255. [CrossRef]

9. Chen, Y.; Wang, Y.; Ma, J.; Jin, Q. BRIM: An Accurate Electricity Spot Price Prediction Scheme-Based Bidirectional Recurrent Neural Network and Integrated Market. *Energies* **2019**, *12*, 2241. [CrossRef]
10. Rijn, J.N.; Hutter, F. Hyperparameter Importance Across Datasets. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, UK, 19–23 August 2018; pp. 2367–2376. [CrossRef]
11. Bhotto, M.Z.A.; Jones, R.; Makonin, S.; Bajic, I.V. Short-Term Demand Prediction Using an Ensemble of Linearly-Constrained Estimators. *IEEE Trans. Power Syst.* **2021**, *36*, 3163–3175. [CrossRef]
12. Muni, S.P.; Sharma, R. Short-term electricity price prediction using kernel-based machine learning techniques. In Proceedings of the 2021 1st Odisha International Conference on Electrical Power Engineering, Communication and Computing Technology (ODICON), Bhubaneswar, India, 8–9 January 2021; pp. 1–5. [CrossRef]
13. Li, Y.; Wang, R.; Yang, Z. Optimal Scheduling of Isolated Microgrids Using Automated Reinforcement Learning-Based Multi-Period Forecasting. *IEEE Trans. Sustain. Energy* **2022**, *13*, 159–169. [CrossRef]
14. Shi, Z.B.; Li, Y.; Yu, T. Short-Term Load Forecasting Based on LS-SVM Optimized by Bacterial Colony Chemotaxis Algorithm. In Proceedings of the 2009 International Conference on Information and Multimedia Technology, Beijing, China, 16–18 December 2009; pp. 306–309. [CrossRef]
15. Sabzehgar, R.; Amirhosseini, D.Z.; Rasouli, M. Solar power forecast for a residential smart microgrid based on numerical weather predictions using artificial intelligence methods. *J. Build. Eng.* **2020**, *32*, 101629. [CrossRef]
16. Scolari, E.; Sossan, F.; Paolone, M. Irradiance prediction intervals for PV stochastic generation in microgrid applications. *Sol. Energy* **2016**, *139*, 116–129. [CrossRef]
17. Mohamed, M.; Chandra, A.; Abd, M.A.; Singh, B. Application of machine learning for prediction of solar microgrid system. In Proceedings of the 2020 IEEE International Conference on Power Electronics, Drives and Energy Systems (PEDES), Jaipur, India, 16–19 December 2020; pp. 1–5. [CrossRef]
18. Onumanyi, A.J.; Isaac, S.J.; Kruger, C.P.; Abu-Mahfouz, A.M. Transactive Energy: State-of-the-Art in Control Strategies, Architectures, and Simulators. *IEEE Access* **2021**, *9*, 131552–131573. [CrossRef]
19. Viel, F.; Silva, L.A.; Leithardt, V.R.Q.; Santana, J.F.D.P.; Teive, R.C.G.; Zeferino, C.A. An Efficient Interface for the Integration of IoT Devices with Smart Grids. *Sensors* **2020**, *20*, 2849. [CrossRef]
20. Helfer, G.A.; Barbosa, J.L.V.; Alves, D.; da Costa, A.B.; Beko, M.; Leithardt, V.R.Q. Multispectral Cameras and Machine Learning Integrated into Portable Devices as Clay Prediction Technology. *J. Sens. Actuator Netw.* **2021**, *10*, 40. [CrossRef]
21. Dalai, I.; Mudali, P.; Pattanayak, A.S.; Pattnaik, B.S. Hourly prediction of load using edge intelligence over IoT. In Proceedings of the 2019 11th International Conference on Advanced Computing (ICoAC), Chennai, India, 18–20 December 2019; pp. 117–121. [CrossRef]
22. Ma, Y.J.; Zhai, M.Y. Day-Ahead Prediction of Microgrid Electricity Demand Using a Hybrid Artificial Intelligence Model. *Processes* **2019**, *7*, 320. [CrossRef]
23. Dridi, A.; Moungla, H.; Afifi, H.; Badosa, J.; Ossart, F.; Kamal, A.E. Machine Learning Application to Priority Scheduling in Smart Microgrids. In Proceedings of the 2020 International Wireless Communications and Mobile Computing (IWCMC), Limassol, Cyprus, 15–19 June 2020; pp. 1695–1700. [CrossRef]
24. Tian, W.; Lei, C.; Tian, M. Dynamic prediction of building HVAC energy consumption by ensemble learning approach. In Proceedings of the 2018 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 12–14 December 2018; Volume 8, pp. 254–257. [CrossRef]
25. Hajjaji, I.; Alami, H.E.; El-Fenni, M.R.; Dahmouni, H. Evaluation of Artificial Intelligence Algorithms for Predicting Power Consumption in University Campus Microgrid. In Proceedings of the 2021 International Wireless Communications and Mobile Computing (IWCMC), Harbin, China, 28 June–2 July 2021; pp. 2121–2126. [CrossRef]
26. Kubat, M. Artificial Neural Networks. In *An Introduction to Machine Learning*; Springer International Publishing: New York, NY, USA, 2021; pp. 117–143._6. [CrossRef]
27. Graupe, D. *Principles of Artificial Neural Networks*, 3rd ed.; Advanced Series in Circuits and Systems; World Scientific Publishers: Singapore, 2013; Volume 7, pp. 1–382. [CrossRef]
28. Hajian, A.; Styles, P. Artificial Neural Networks. In *Application of Soft Computing and Intelligent Methods in Geophysics*; Springer International Publishing: New York, NY, USA, 2018; pp. 3–69._1. [CrossRef]
29. Principe, J. Artificial Neural Networks. In *Electrical Engineering Handbook*; CRC Press: New York, NY, USA, 1997. [CrossRef]
30. Pirjatullah; Kartini, D.; Nugrahadi, D.T.; Muliadi; Farmadi, A. Hyperparameter Tuning using GridsearchCV on The Comparison of The Activation Function of The ELM Method to The Classification of Pneumonia in Toddlers. In Proceedings of the 2021 4th International Conference of Computer and Informatics Engineering (IC2IE), Jakarta, Indonesia, 22–24 October 2021; pp. 390–395. [CrossRef]
31. Fontenla-Romero, O.; Erdogmus, D.; Principe, J.C.; Alonso-Betanzos, A.; Castillo, E. Linear Least-Squares Based Methods for Neural Networks Learning. In *Artificial Neural Networks and Neural Information Processing—ICANN/ICONIP 2003*; Springer: Istanbul, Turkey, 2003; pp. 84–91._11. [CrossRef]
32. Schulz, E.; Speekenbrink, M.; Krause, A. A tutorial on Gaussian process regression: Modelling, exploring, and exploiting functions. *J. Math. Psychol.* **2018**, *85*, 1–16. [CrossRef]
33. Banerjee, A.; Dunson, D.B.; Tokdar, S.T. Efficient Gaussian process regression for large datasets. *Biometrika* **2013**, *100*, 75–89. [CrossRef]
34. Gramacy, R.B. Gaussian Process Regression. In *Surrogates*; Chapman and Hall/CRC: London, UK, 2020; pp. 143–221. [CrossRef]

35. Cunningham, P.; Delany, S.J. k-Nearest Neighbour Classifiers - A Tutorial. *ACM Comput. Surv.* **2021**, *54*, 1–25. [CrossRef]
36. Ali, N.; Neagu, D.; Trundle, P. Evaluation of k-nearest neighbour classifier performance for heterogeneous data sets. *SN Appl. Sci.* **2019**, *1*, 1–5. [CrossRef]
37. Biau, G.; Scornet, E. A random forest guided tour. *TEST* **2016**, *25*, 197–227. [CrossRef]
38. Probst, P.; Boulesteix, A.L. To tune or not to tune the number of trees in random forest? *J. Mach. Learn. Res.* **2017**, *18*, 1–18.
39. Probst, P.; Wright, M.N.; Boulesteix, A. Hyperparameters and tuning strategies for random forest. *WIREs Data Min. Knowl. Discov.* **2019**, *9*, e1301. [CrossRef]
40. Yang, X.S. Support vector machine and regression.Chapter Support vector machine and regression. In *Introduction to Algorithms for Data Mining and Machine Learning*; Yang, X.S., Ed.; Elsevier: Amsterdam, The Netherlands, 2019; pp. 129–138. [CrossRef]
41. Pisner, D.A.; Schnyer, D.M. Support vector machine. In *Machine Learning*; Elsevier: Amsterdam, The Netherlands, 2020; pp. 101–121. [CrossRef]
42. Suthaharan, S. Support Vector Machine. In *Machine Learning Models and Algorithms for Big Data Classification*; Springer: New York, NY, USA, 2016; pp. 207–235._9. [CrossRef]
43. Cervantes, J.; Garcia-Lamont, F.; Rodríguez-Mazahua, L.; Lopez, A. A comprehensive survey on support vector machine classification: Applications, challenges and trends. *Neurocomputing* **2020**, *408*, 189–215. [CrossRef]
44. Mondi, L. Eskom: Electricity and technopolitics in South Africa by Syvly Jaglin, Alain Dubresson. *Transform. Crit. Perspect. South. Afr.* **2017**, *93*, 176–185. [CrossRef]
45. Roy-Aikins, J. Challenges in Meeting the Electricity Needs of South Africa. In Proceedings of the ASME 2016 Power Conference, Charlotte, NC, USA, 26–30 June 2016. [CrossRef]
46. Lee, S.; Lee, D.K. What is the proper way to apply the multiple comparison test? *Korean J. Anesthesiol.* **2018**, *71*, 353–360. [CrossRef] [PubMed]

energies

Article

Generalized Sliding Mode Observers for Simultaneous Fault Reconstruction in the Presence of Uncertainty and Disturbance

Ashkan Taherkhani [1], Farhad Bayat [1,*], Kaveh Hooshmandi [2] and Andrzej Bartoszewicz [3]

[1] Department of Electrical Engineering, University of Zanjan, Zanjan 45371-38791, Iran; ashkantaherkhani7@gmail.com
[2] Department of Electrical Engineering, Arak University of Technology, Arak 38181-46763, Iran; k.hooshmandi@arakut.ac.ir
[3] Institute of Automatic Control, Lodz University of Technology, 90924 Lodz, Poland; andrzej.bartoszewicz@p.lodz.pl
* Correspondence: bayat.farhad@znu.ac.ir

Abstract: In this paper, a generalized sliding mode observer design method is proposed for the robust reconstruction of sensors and actuators faults in the presence of both unknown disturbances and uncertainties. For this purpose, the effect of uncertainty and disturbance on the system has been considered in generalized state-space form, and the LMI tool is combined with the concept of an equivalent output error injection method to reduce the effects of them on the reconstruction process. The upper bound of the disturbance and uncertainty are minimized in the design of the sliding motion so that the reconstruction of the faults will be minimized. The design method is applied for actuator faults in the generalized state-space form, and then with some suitable filtering, the method extends as sensors and actuators coincidentally faults. Since in the proposed approach, the state trajectories do not leave the sliding manifold even in simultaneous sensors and actuators faults, then the faults are reconstructed based upon information retrieved from the equivalent output error injection signal. Due to the importance of the robust fault reconstruction in the wind energy conversion system (WECS), the proposed approach is successfully applied to a 5 MW wind turbine system. The simulation results verify the robust performances of the proposed approach in the presence of unknown perturbations and uncertainties.

Keywords: sliding mode observer; fault detection; robust fault reconstruction; linear matrix inequalities (LMIs)

Citation: Taherkhani, A.; Bayat, F.; Hooshmandi, K.; Bartoszewicz, A. Generalized Sliding Mode Observers for Simultaneous Fault Reconstruction in the Presence of Uncertainty and Disturbance. *Energies* **2022**, *15*, 1411. https://doi.org/10.3390/en15041411

Academic Editor: Sergio Nesmachnow

Received: 8 January 2022
Accepted: 4 February 2022
Published: 15 February 2022

1. Introduction

In recent decades, industrial processes are becoming more and more complex; thus, ensuring the operational reliability of these processes is an important task. Among them, fault detection and isolation (FDI) methods play a pivotal role in making the process reliable. The sensor and actuator faults are known as the most frequent faults that occur in many control systems such as satellite/aircraft [1,2], wind turbines [3,4], vehicles suspension system [5,6], offshore platforms [7], motor drives [8], power systems, and renewable energies [9,10]. In the event of a fault occurrence, the reliability and efficiency of the system are severely affected, and thus, the fault reconstruction is an important issue in the context of FDI approaches, and various types of research have been done in this field. However, when the system is subject to the uncertainty and disturbance, at the same time, identifying and reconstructing simultaneous sensor and actuator faults are still challenging issues that need to be addressed carefully.

In [11], a PI observer is proposed for fault estimation purposes based on convex structures and by employing nonquadratic Lyapunov functions. As a result, less conservative conditions in the form of LMIs are obtained. In [12], the sensor and actuator faults reconstruction problem is addressed by only considering the uncertainty in the model.

In [13], the actuator fault reconstruction (AFR) problem is investigated by introducing two observers: one to estimate unknown inputs and the other to facilitate fault reconstruction. A particular kind of actuator faults in manipulator systems, i.e., joint luck failure, is considered in [14], and two kinds of reconfiguration schemes are proposed to cope with this issue. In [15], a fault-tolerant control technique is studied for electro-hydraulic actuators. In this reference, an unknown input observer is used to reconstruct sensor faults in the presence of disturbances. In [16], a fault-tolerant sliding mode controller was designed for a class of fuzzy T-S systems subject to actuator saturation, external disturbances, and time-varying delay. Sliding mode control is a variable structure control method that is well known in nonlinear system control. In [17], integral sliding mode control is proposed to a new five-dimensional four-wing hyper chaotic system with hidden attractor. An adaptive finite-time sliding mode control is proposed in [18] to construct a family of nine new chameleon chaotic systems subjected to uncertainties and disturbances. In [19], a new synchronous quasi-sliding mode control (QSMC) is studied for Rikitake chaotic system. A selection on switching surface and the existing of QSMC is also considered in this reference. A composite sliding mode observer is proposed in [20] to study multi-sensor fault diagnosis and active fault-tolerant control in a PMSM drive system. For the FDI of a class of uncertain Lipschitz nonlinear systems, an adaptive robust sliding mode observer (SMO) is proposed in [21], where both external disturbance and faults are considered. A second-order sliding mode observer is considered in [22] to reconstruct sensor faults in an air-path system of a heavy-duty diesel engine in the presence of disturbance. In [23], an adaptive estimation approach is proposed to recover the bias fault of sensors in a class of nonlinear systems subject to unstructured uncertainty. In [9], the fault detection and fault-tolerant control problem for multi-area power systems with sensor failures were considered using a descriptor form SMO. In [24], an adaptive SMO and a descriptive form observer are combined to reconstruct the sensor and actuator faults where the stability analysis was performed by the Lyapunov method. For a linear system with disturbance and time-varying delay, an adaptive estimation approach is presented in [25] for AFR. The problem of fault-tolerant controller design for a synchronization problem of complex dynamical networks subject to actuator faults and saturation was investigated in [26,27]. In [28], a time shift approach for AFR with a time-delay of output is introduced by using an SMO. For wind turbine faults detecting, a new technique is proposed in [29] as a signal reconstruction modeling technique. In the mentioned paper, to detect faults at an early stage, multiple indicators are also calculated. A new data-driven sensor FDI technique is presented in [30] using interval-valued data and an enhanced reconstruction approach to develop fault isolation. Various methods for a simultaneous actuator and sensor faults reconstruction have been proposed in the literature. Inspired by a singular system theory, a descriptor observer design is presented in [31] to reconstruct the actuator fault based on the transformed coordinate system. In [32], both faults are simultaneously reconstructed in a special class of nonlinear system described by the Takagi–Sugeno model. In [33], a new robust and simultaneous actuator and sensor faults estimation is proposed for a class of LPV systems described with polytopic representation where the parameters evolve in the hypercube domain.

Discrepancies between the actual process and its model such as model uncertainties and disturbances cause misleading of fault detection and reconstruction. The problem of simultaneous fault detection and reconstruction of sensors and actuators in the presence of both uncertainties and unknown disturbances has been addressed in this paper. A noticeable feature of the proposed approach is that the inherent differences between the effect of uncertainty and disturbance on the system have been considered in the design of sliding mode observers in a generalized state-space form when faults occur at both sensors and actuators coincidentally. This problem is efficiently addressed in this paper, where two different distribution matrices are incorporated to represent perturbations and uncertainties in the system. Then, LMI and the equivalent output error injection (EOEI) methods have been used to design a robust SMO. Since the state trajectories of SMO do not leave the sliding manifold in the presence of the uncertainties and disturbances, then the sensor and actuator faults are reconstructed based upon information retrieved from the equivalent output error injection signal. In order to verify the robust performances of the proposed approach, we applied it to a 5 MW wind turbine system. The wind energy conversion system (WECS) is a typical large and complex nonlinear system with random and intermittent wind force. In the electrical power system, the safety of electrical equipment is the basis for ensuring the stability and reliability. Fault reconstruction's aim is to guarantee the security of electrical power system operation and industry production. For this reason, we proposed fault reconstruction to ensure the safe and efficient operation of wind turbines. The wind turbine systems actuators and sensors have the highest probability of failure, which has the greatest impact on the WECS safe and efficient operation. A robust fault-tolerant control for a Takagi–Sugeno fuzzy model is studied for the wind energy conversion system in [34].

The rest of this paper is organized as follows. Description of the system in the presence of an actuator and sensor fault, disturbances and uncertainties, and design of the proposed SMO are presented in Section 2. A robust AFR employing the EOEI approach is presented in Section 3. Sensor fault reconstruction is studied similar to the actuator fault method by introducing a new state in Section 4. Simulation results and concluding remarks are provided in Section 5.

2. Description of the Problem

We consider a class of uncertain systems in the presence of fault and disturbance given as:

$$\begin{aligned} \dot{z}(t) &= Az(t) + Bu(t) + M\partial(t,y,u) + Dd(t) + Ff_a(t) \\ y(t) &= Cz(t) + F_s f_s(t) \end{aligned} \tag{1}$$

where $B \in R^{n\times m}$, $A \in R^{n\times n}$, $C \in R^{p\times n}$, $M \in R^{n\times k}$, $D \in R^{n\times q}$, $F \in R^{n\times r}$, and $F_s \in R^{p\times l}$ denote the matrices of inputs, states, outputs, unknown bounded uncertainties, disturbances, actuator faults, and sensor faults, respectively. We assume $p \geq q$, $p \geq l$, $n > p \geq r$, and F and C are full column rank matrices. We also assume that $f_a(t)$ is a bounded unknown function indicating the fault of actuators, where $\|f_a(t)\| \leq \alpha(t)$, and α is a known function. Furthermore, the unknown bounded function $\partial(t,y,u)$ denotes the system's uncertainty and $\|\partial(t,y,u)\| \leq \beta$, where $\beta > 0$ is a known parameter. Moreover, $d(t)$ denotes the disturbance signal, which is bounded $\|d(t)\| \leq d_0$, where d_0 is a positive constant.

Assumption 1. *It is assumed that $rank(CF) = rank(F) = r$ and the system with (A, F, C) matrices has all its invariant zeros in the LHP.*

It is important to note that $p < n$ implies that some states may not be observable. To cope with this issue, the following theorem is utilized to extract the observable and unobservable parts of the system in (1) with $f_s(t) = 0$ where the matrix F only appears in the observable subsystem.

Theorem 1. *Assuming the conditions of Assumption 1 are satisfied and $f_s(t) = 0$, then, there exist linear nonsingular transformations $\tilde{z} = \mathcal{T}_b\bar{z}$ and $\bar{z} = \mathcal{T}_c z$ such that:*

$$\tilde{A} = \begin{bmatrix} A_1 & A_2 \\ A_3 & A_4 \end{bmatrix}, \quad \tilde{C} = [0_{p\times(n-p)}, \mathcal{T}], \quad \tilde{F} = \begin{bmatrix} 0_{r\times r} \\ F_2 \end{bmatrix} \tag{2}$$

where $F_2 = \begin{bmatrix} 0_{(p-r)\times r} \\ F_{22} \end{bmatrix} \in R^{r\times r}$ and $\mathcal{T} \in R^{p\times p}$ is orthogonal and nonsingular, $A_1 \in R^{(n-p)\times(n-p)}$, $\tilde{B}^T = [B_1^T, B_2^T]$, $\tilde{D}^T = [D^T_{1_{(n-p)\times q}}, D^T_{2_{p\times q}}]$, $\tilde{M}^T = [M^T_{1_{(n-p)\times k}}, M^T_{2_{p\times k}}]$.

Proof. First, consider $\mathcal{T}_c = [N_c, C^T]^T$, where columns of N_c span the null space of C and are orthonormal. Then, one obtains:

$$\begin{aligned}\dot{\bar{z}}(t) &= \underbrace{\mathcal{T}_c A \mathcal{T}_c^{-1}}_{A_c}\bar{z}(t) + \underbrace{\mathcal{T}_c B}_{B_c} u(t) + \underbrace{\mathcal{T}_c M}_{M_c}\partial(t,y,u) \\ &+ \underbrace{\mathcal{T}_c D}_{D_c} d(t) + \underbrace{\mathcal{T}_c F}_{F_c} f_a(t) \\ \bar{y} &= \underbrace{\mathcal{T}_c^{-1} C}_{C_c}\bar{z} = [\ 0_{(n-p)} \quad I_p\ ]\bar{z}.\end{aligned} \tag{3}$$

It can be seen that only the last p states are present at the output. Now, considering $F_c = \begin{bmatrix} f_{1_{(r\times r)}} \\ f_{2_{(p\times r)}} \end{bmatrix}$, we define a nonsingular linear transformation matrix $\mathcal{T}_b$ as:

$$\mathcal{T}_b = \begin{bmatrix} I_{(n-p)} & -f_1\left(f_2^T f_2\right)^{-1} f_2^T \\ 0_{p\times(n-p)} & \mathcal{T}^T \end{bmatrix} \tag{4}$$

where the QR decomposition of f_2 is used to obtain $\mathcal{T}$. Then, by using $\tilde{z} = \mathcal{T}_b\bar{z}$, one obtains:

$$\begin{aligned}\dot{\tilde{z}}(t) &= \tilde{A}\tilde{z}(t) + \tilde{B}u(t) + \tilde{M}\partial(t,y,u) + \tilde{D}d(t) + \tilde{F}f_a(t) \\ \tilde{y}(t) &= \tilde{C}\tilde{z}(t)\end{aligned} \tag{5}$$

where

$$\begin{aligned}&\tilde{A} = \begin{bmatrix} A_1 & A_2 \\ A_3 & A_4 \end{bmatrix}, \tilde{F} = \begin{bmatrix} 0_{r\times r} \\ F_2 \end{bmatrix}, \tilde{D} = \begin{bmatrix} D_1 \\ D_2 \end{bmatrix} \\ &\tilde{C} = \left[0_{p\times(n-p)}, \mathcal{T}\right], \tilde{B} = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}, \tilde{M} = \begin{bmatrix} M_1 \\ M_2 \end{bmatrix}.\end{aligned} \tag{6}$$

□

The following SMO is considered:

$$\begin{aligned}\dot{\hat{\tilde{z}}}(t) &= \tilde{A}\hat{\tilde{z}}(t) + \tilde{B}u(t) - \tilde{G}_l e_{\tilde{y}}(t) + \tilde{G}_n v \\ \hat{\tilde{y}}(t) &= \tilde{C}\hat{\tilde{z}}(t).\end{aligned} \tag{7}$$

where $\hat{\tilde{y}}(t)$ and $\hat{\tilde{z}}(t)$ denote the estimation of outputs and states, respectively. The output estimation error is represented by $e_{\tilde{y}}(t) = \hat{\tilde{y}}(t) - \tilde{y}(t)$. Furthermore, the observer gains $\tilde{G}_n, \tilde{G}_l \in R^{n\times p}$ will be defined in the following.

The sliding variable v has a nonlinear discontinuous term to maintain the sliding motion, which is given as:

$$v = \begin{cases} 0, & \forall e_{\tilde{y}} = 0 \\ -\rho(t,y,u)\left\|e_{\tilde{y}}\right\|^{-1} e_{\tilde{y}}, & \forall e_{\tilde{y}} \neq 0 \end{cases} \tag{8}$$

where the upper bound for the fault plus uncertainty and disturbance is represented by the gain factor $\rho(t,y,u) \in R$.

The gain $\tilde{G}_n$ is chosen as:

$$\tilde{G}_n = \begin{bmatrix} -L\mathcal{T}^T \\ \mathcal{T}^T \end{bmatrix} P_0^{-1} \tag{9}$$

where $P_0 = P_0^T \in R^{p \times p}$ is a PDF design matrix that will be calculated in the following and L is defined as:

$$L = \begin{bmatrix} L_0 & 0 \end{bmatrix} \in R^{(n-p) \times p} \tag{10}$$

where $L_0 \in R^{(n-p)\times(p-r)}$ is adjusted such that $(L_0 A_{31} + A_1)$ is Hurwitz, where A_{31} represents the first $p - q$ rows of A_3.

Now, the following theorem is recalled from [35].

Theorem 2. *Assume an observer dynamic as given in (7), a Lyapunov matrix $\tilde{P}$, and a matrix $\tilde{G}_l$ satisfying:*

$$\begin{array}{c} \tilde{P} = \begin{bmatrix} P_1 & P_1 L \\ L^T P_1 & \mathcal{T}^T P_0 \mathcal{T} + L^T P_1 L \end{bmatrix} \\ (\tilde{A} - \tilde{G}_l \tilde{C})^T \tilde{P} + \tilde{P}(\tilde{A} - \tilde{G}_l \tilde{C}) < 0 \end{array} \tag{11}$$

where L defined in (10) and $P_1 \in R^{(n-p)\times(n-p)}$. Then, the observation error $e(t) \overset{\Delta}{=} \hat{\tilde{z}}(t) - \tilde{z}(t)$ is asymptotically stable.

Considering Assumption 1, it can be shown that there exists a stable sliding motion on the sliding surface given as [36]:

$$S = \left\{ e(t) | \tilde{C}e(t) = 0 \right\}. \tag{12}$$

Then, one obtains:

$$\begin{array}{l} \dot{e}(t) = (\tilde{A} - \tilde{G}_l \tilde{C})e(t) - \tilde{M}\partial(t, y, u) \\ \qquad -\tilde{D}d(t) - \tilde{F}f_a(t) + \tilde{G}_n v \end{array} . \tag{13}$$

Lemma 1. *The error dynamics in* (13) *is bounded in the region Ω defined as:*

$$\Omega = \left\{ e | \|e\| < 2(\mu_2 \beta + \mu_1 d_0)/\mu_0 \right\} \tag{14}$$

where $\mu_0 = -\lambda_{\max}(\tilde{A}_c)$, $\mu_1 = \|\tilde{P}\tilde{D}\|$, $\mu_2 = \|\tilde{P}\tilde{M}\|$, $\tilde{A}_c = -(\tilde{G}_l\tilde{C} - \tilde{A})^T\tilde{P} - \tilde{P}(\tilde{G}_l\tilde{C} - \tilde{A})$.

Proof. Define $V = e^T \tilde{P} e$. Then

$$\begin{array}{l} \dot{V} = e^T \tilde{A}_c e - 2e^T \tilde{P}\tilde{M}\partial(t, y, u) - 2e^T \tilde{P}\tilde{D}d(t) \\ \qquad -2e^T \tilde{P}\tilde{F}f_a(t) + 2e^T \tilde{P}\tilde{G}_n v \end{array} . \tag{15}$$

From (16) and considering $\|\partial(t, y, u)\| \le \beta$ and $\|d(t)\| \le d_0$ and utilizing the Cauchy–Schwartz inequality, yield:

$$\dot{V} \le -\mu_0 \|e\|^2 + 2\|e\|\mu_1 d_0 + 2\|e\|\mu_2 \beta - 2e^T \tilde{P}\tilde{F}f_a(t) + 2e^T \tilde{P}\tilde{G}_n v. \tag{16}$$

Using (6), (9), and (11), it is simply verified that $\tilde{P}\tilde{F} = \tilde{C}^T P_0 \tilde{C}\tilde{F}$ and $\tilde{P}\tilde{G}_n = \tilde{C}^T$. Then, considering $e_{\tilde{y}}(t) = \tilde{C}e(t) = \tilde{C}(\tilde{z}(t) - z(t))$, $\|f_a(t)\| \le \alpha$ and (8), one obtains:

$$\begin{array}{l} \dot{V} \le 2\|e\|\mu_1 d_0 - \mu_0 \|e\|^2 + 2\|e\|\mu_2 \beta \\ \qquad -2\left(\rho(t, y, u) - \alpha(t, u)\|P_0 \tilde{C}\tilde{F}\|\right)\|e_{\tilde{y}}\| \\ \quad \le -\|e\|(\mu_0\|e\| - 2\mu_1 d_0 - 2\mu_2 \beta). \end{array} \tag{17}$$

Therefore, if $\|e\| > 2(\mu_2\beta + \mu_1 d_0)/\mu_0$, then $\dot{V} < 0$, and this implies that $e(t)$ will converge to the following bounded region:

$$\Omega = \left\{ e | \|e\| < (2\mu_1 d_0 + 2\mu_2 \beta)/\mu_0 \right\}. \tag{18}$$

□

Now, we show that with the proper selection of $\rho(y, u, t)$, the sliding surface in (13) is reached in finite time. Define:

$$\mathcal{T}_L = \begin{bmatrix} I_{n-p} & L \\ 0 & \mathcal{T} \end{bmatrix}. \tag{19}$$

Using this transformation, the matrices in (6), (9), and (11) are converted to the following form:

$$\begin{array}{l} \mathcal{A} = \mathcal{T}_L \tilde{A} \mathcal{T}_L^{-1} = \begin{bmatrix} \mathcal{A}_1 & \mathcal{A}_2 \\ \mathcal{A}_3 & \mathcal{A}_4 \end{bmatrix}, \quad \mathcal{M} = \mathcal{T}_L \tilde{M} = \begin{bmatrix} \mathcal{M}_1 \\ \mathcal{M}_2 \end{bmatrix} \\ \mathcal{G}_n = \mathcal{T}_L \tilde{G}_n = \begin{bmatrix} 0 \\ P_0^{-1} \end{bmatrix}, \quad \mathcal{F} = \mathcal{T}_L \tilde{F} = \begin{bmatrix} 0_{(n-p)\times r} \\ \mathcal{F}_2 \end{bmatrix} \\ \mathcal{D} = \mathcal{T}_L \tilde{D} = \begin{bmatrix} \mathcal{D}_1 \\ \mathcal{D}_2 \end{bmatrix}, \quad \mathcal{C} = \tilde{C} \mathcal{T}_L^{-1} = \begin{bmatrix} 0_{p\times(n-p)} & I_p \end{bmatrix} \\ \mathcal{P} = (\mathcal{T}_L^{-1})^T \tilde{P} \mathcal{T}_L^{-1} = \begin{bmatrix} P_1 & 0 \\ 0 & P_0 \end{bmatrix} \end{array} \tag{20}$$

where:

$$\begin{array}{l} \mathcal{A}_1 = A_1 + LA_3, \quad \mathcal{M}_1 = M_1 + LM_2, \quad \mathcal{D}_1 = D_1 + LD_2 \\ \mathcal{A}_3 = \mathcal{T} A_3, \quad \mathcal{M}_2 = \mathcal{T} M_2, \quad \mathcal{D}_2 = \mathcal{T} D_2, \quad \mathcal{F}_2 = \mathcal{T} F_2. \end{array} \tag{21}$$

Therefore, the error in (14) becomes:

$$\begin{array}{l} \dot{e}_l(t) = (\mathcal{A} - \mathcal{G}_L \mathcal{C}) e_l(t) - \mathcal{M}\partial(t, y, u) - \mathcal{D} d(t) \\ \qquad - \mathcal{F} f_a(t) + \mathcal{G}_n v \end{array} \tag{22}$$

where

$$e_l = \mathcal{T}_L e = \begin{bmatrix} e_1 \\ e_{\tilde{y}} \end{bmatrix}, \quad \mathcal{G}_L = \mathcal{T}_L G_l = \begin{bmatrix} \mathcal{G}_{L1} \\ \mathcal{G}_{L2} \end{bmatrix}. \tag{23}$$

Using this, (23) can be decomposed as:

$$\begin{array}{l} \dot{e}_1(t) = \mathcal{A}_1 e_1(t) + (\mathcal{A}_2 - \mathcal{G}_{L1}) e_{\tilde{y}}(t) \\ \qquad - \mathcal{M}_1 \partial(t, y, u) - \mathcal{D}_1 d(t) s \\ \dot{e}_{\tilde{y}}(t) = \mathcal{A}_3 e_1(t) + (\mathcal{A}_4 - \mathcal{G}_{L2}) e_{\tilde{y}}(t) + P_0^{-1} v \\ \qquad - \mathcal{M}_2 \partial(t, y, u) - \mathcal{D}_2 d(t) - \mathcal{F}_2 f_a(t). \end{array} \tag{24}$$

The following theorem proposes a proper choice of ρ to guarantee finite time convergence to the sliding surface S.

Theorem 3. *The error dynamic (23) reaches the sliding surface S in finite-time* $T_s \leq \frac{\sqrt{V(0)}}{\eta_0 \sqrt{\lambda_{\min}\left(P_0^{-1}\right)}}$ *and stays there forever, if:*

$$\begin{array}{l} \rho(t, y, u) \geq \|P_0 \mathcal{D}_2\| d_0 + \|P_0 \mathcal{M}_2\| \beta + \|P_0 \mathcal{F}_2\| \alpha + \\ \qquad 2\|P_0 \mathcal{A}_3\| (\mu_1 d_0 + \mu_2 \beta)/\mu_0 + \eta_0. \end{array} \tag{25}$$

Proof. Define the candidate Lyapunov function $V = e_{\tilde{y}}^T P_0 e_{\tilde{y}}$. Then:

$$\begin{aligned}\dot{V} = e_{\tilde{y}}^T \Big(P_0(\mathcal{A}_4 - \mathcal{G}_{L2}) + (\mathcal{A}_4 - \mathcal{G}_{L2})^T P_0\Big) e_{\tilde{y}} + \\ 2e_{\tilde{y}}^T P_0 \mathcal{A}_3 e_1 - 2e_{\tilde{y}}^T P_0 \mathcal{F}_2 f - 2e_{\tilde{y}}^T P_0 \mathcal{D}_2 d - \\ 2e_{\tilde{y}}^T P_0 \mathcal{M}_2 \partial + 2e_{\tilde{y}}^T v\end{aligned} \tag{26}$$

where $2e_{\tilde{y}}^T v = -2\rho \| e_{\tilde{y}} \|$. Then, by using the Cauchy–Schwartz inequality, one gets:

$$\dot{V} \le -2\|e_{\tilde{y}}\| \begin{pmatrix} \rho - \|P_0\mathcal{A}_3\|\|e_1\| - \|P_0\mathcal{F}_2\|\alpha - \\ \|P_0\mathcal{D}_2\|d_0 - \|P_0\mathcal{M}_2\|\beta \end{pmatrix}. \tag{27}$$

From (15), (24), and (25) we conclude that $\rho - \|P_0\mathcal{A}_3\|\|e_1\| - \|P_0\mathcal{M}_2\|\beta - \|P_0\mathcal{D}_2\|d_0 - \|P_0\mathcal{F}_2\|\alpha = \eta_0 > 0$ and $\|e_1\| < 2(\mu_2\beta + \mu_1 d_0)/\mu_0$. This results:

$$\dot{V} \le -2\eta_0 \|e_{\tilde{y}}\| \le -2\eta_0 \sqrt{\lambda_{\min}\left(P_0^{-1}\right)} \sqrt{V}. \tag{28}$$

Therefore, using (29) and the finite-time stability theorem (see Theorem 4.2 of [37]), we conclude that the estimation error converges to zero, and the sliding motion reaches S in finite-time $T_s \le \frac{1}{\eta_0}\sqrt{\frac{V(0)}{\lambda_{\min}\left(P_0^{-1}\right)}}$. □

Now, an LMI-based approach is proposed to obtain an appropriate gain matrix $\tilde{G}_l$. In this regard, Theorem 2.2 requires finding a matrix $\tilde{P}$ that satisfies:

$$(\tilde{A} - \tilde{G}_l\tilde{C})^T \tilde{P} + \tilde{P}(\tilde{A} - \tilde{G}_l\tilde{C}) < 0. \tag{29}$$

As discussed in [36], an inequality of the form (30) can be alternatively solved by the following set of inequalities:

$$\tilde{P} > 0, \quad \tilde{A}^T\tilde{P} + \tilde{P}\tilde{A} - \tilde{C}^T U^{-1}\tilde{C} + \tilde{P}Q\tilde{P} < 0 \tag{30}$$

where $U \in R^{p\times p}$ and $Q \in R^{n\times n}$ are PSD matrices. Applying the Schur lemma, (31) is converted to the following LMI:

$$\begin{bmatrix} \tilde{P}\tilde{A} + \tilde{A}^T\tilde{P} - \tilde{C}^T U^{-1}\tilde{C} & \tilde{P} \\ \tilde{P} & -Q^{-1} \end{bmatrix} < 0. \tag{31}$$

The matrix $\tilde{P}$ is obtained by solving the LMI (32), and then:

$$\tilde{G}_l = \tilde{P}^{-1}\tilde{C}^T U^{-1}. \tag{32}$$

3. Robust Actuator Faults Reconstruction

In this part, assuming that the proposed SMO gains in (7) are well-designed, an efficient approach is proposed for a robust AFR procedure. Relying on the results of Theorem 2.3, one obtains that $e_{\tilde{y}} = \dot{e}_{\tilde{y}} = 0$ as $t \to \infty$. Then:

$$\begin{aligned}\dot{e}_1(t) &= \mathcal{A}_1 e_1(t) - \mathcal{M}_1\partial(t,y,u) - \mathcal{D}_1 d(t) \\ 0 &= \mathcal{A}_3 e_1(t) - \mathcal{M}_2\partial(t,y,u) - \mathcal{D}_2 d(t) - \mathcal{F}_2 f_a(t) + P_0^{-1} v_{eq}\end{aligned} \tag{33}$$

where v_{eq} is obtained by approximating v in (8):

$$v_{eq} = -\rho(t,y,u) e_{\tilde{y}} \left(\varepsilon + \|e_{\tilde{y}}\|\right)^{-1} \tag{34}$$

where $\varepsilon > 0$. From (34), one obtains:

$$\begin{aligned} \dot{e}_1(t) = {} & (LA_3 + A_1)e_1(t) - (M_1 + LM_2)\partial(t,y,u) \\ & - (D_1 + LD_2)d(t) \\ 0 = {} & \mathcal{T}\begin{pmatrix} A_3 e_1(t) - M_2\partial(t,y,u) - \\ D_2 d(t) - F_2 f_a(t) \end{pmatrix} + P_0^{-1} v_{eq}. \end{aligned} \tag{35}$$

This implies:

$$P_0^{-1} v_{eq} = \mathcal{T}\begin{pmatrix} -A_3 e_1(t) + M_2\partial(t,y,u) + \\ D_2 d(t) + F_2 f_a(t) \end{pmatrix}. \tag{36}$$

Now, the goal is to minimize or eliminate the effects of disturbance and uncertainty signals on the AFR. To this end, the reconstruction signal is defined as:

$$\hat{f}_i = W\mathcal{T}^T P_0^{-1} v_{eq} \tag{37}$$

where $W = [W_1, F_{22}^{-1}]$. Multiplication of both sides in (37) by $W\mathcal{T}^T$ implies:

$$\hat{f}_i(t) = \ + f_a(t) - WA_3 e_1(t) + [WD_2, WM_2]\begin{bmatrix} d(t) \\ \partial(t,y,u) \end{bmatrix}. \tag{38}$$

From (36), we have:

$$\begin{aligned} e_1(s) = {} & -(sI - (LA_3 + A_1))^{-1} \times \\ & [LD_2 + D_1, LM_2 + M_1]\begin{bmatrix} d(t) \\ \partial(t,y,u) \end{bmatrix}. \end{aligned} \tag{39}$$

Substitution of (40) in (39) results:

$$\begin{aligned} & \hat{f}_i(t) = f_a(t) + G(s)\begin{bmatrix} d(t) \\ \partial(t,y,u) \end{bmatrix} \\ & G(s) = \begin{bmatrix} WD_2 & WM_2 \end{bmatrix} + \\ & \quad WA_3(sI - (LA_3 + A_1))^{-1} \times \\ & \quad \begin{bmatrix} LD_2 + D_1 & LM_2 + M_1 \end{bmatrix}. \end{aligned} \tag{40}$$

Therefore, the effect of $\begin{bmatrix} d(t) \\ \partial(t,y,u) \end{bmatrix}$ on the fault reconstruction signal will be minimized or bounded if:

$$\|G(s)\|_\infty < \xi \tag{41}$$

where ξ is a small constant. Let define $\tilde{P}$ in (31) as:

$$\tilde{P} = \begin{bmatrix} \tilde{P}_{11} & \tilde{P}_{12} \\ \tilde{P}_{12}^T & \tilde{P}_{22} \end{bmatrix} > 0 \tag{42}$$

where $\tilde{P}_{22} \in R^{p\times p}$ and $\tilde{P}_{11} \in R^{(n-p)\times(n-p)}$. By applying the Bounded Real Lemma (BRL) [38], the inequality (42) is converted to:

$$\begin{aligned} & \begin{bmatrix} \Phi_{11} & \Phi_{12} & -(WA_3)^T \\ \Phi_{12}^T & -\xi I & (W\begin{bmatrix} D_2 & M_2 \end{bmatrix})^T \\ -WA_3 & W\begin{bmatrix} D_2 & M_2 \end{bmatrix} & -\xi I \end{bmatrix} < 0 \\ & \Phi_{11} = \tilde{P}_{11}A_1 + A_1^T\tilde{P}_{11} + \tilde{P}_{12}A_3 + A_3^T\tilde{P}_{12}^T \\ & \Phi_{12} = -(\tilde{P}_{11}\begin{bmatrix} D_1 & M_1 \end{bmatrix} + \tilde{P}_{12}\begin{bmatrix} D_2 & M_2 \end{bmatrix}). \end{aligned} \tag{43}$$

By solving (44), one obtains W and $\tilde{P}$. Then, by substituting W in (38) results:

$$\hat{f}_i(t) \simeq f_a(t). \tag{44}$$

4. Robust Sensor Fault Reconstruction

In this case and without loss of generality, we define a new state $y_n(t) \in R^p$ that converts (1) with $f_a(t) = 0$ to a similar form presented in the previous section, i.e., (1) with $f_s(t) = 0$, so that a similar algorithm can be used. To this aim, let us define:

$$\dot{y}_n(t) + A_n y_n(t) = A_n y(t) \tag{45}$$

where A_n is a stable PD matrix. Then, one obtains:

$$\dot{y}_n(t) = -A_n y_n(t) + A_n C z(t) + A_n F_s f_s(t). \tag{46}$$

Now, an augmented system with $n + p$ states is defined as:

$$\begin{aligned} \begin{bmatrix} \dot{z}(t) \\ \dot{y}_n(t) \end{bmatrix} &= \underbrace{\begin{bmatrix} A & 0 \\ A_n C & -A_n \end{bmatrix}}_{A_N} \begin{bmatrix} z(t) \\ y_n(t) \end{bmatrix} + \underbrace{\begin{bmatrix} B \\ 0 \end{bmatrix}}_{B_N} u(t) \\ &\quad + \underbrace{\begin{bmatrix} 0 \\ A_n F_s \end{bmatrix}}_{F_N} f_s(t) + \underbrace{\begin{bmatrix} D \\ 0 \end{bmatrix}}_{D_N} d(t) + \underbrace{\begin{bmatrix} M \\ 0 \end{bmatrix}}_{M_N} \partial(t, y, u) \\ y_n(t) &= \underbrace{\begin{bmatrix} 0 & I_p \end{bmatrix}}_{C_N} \begin{bmatrix} z(t) \\ y_n(t) \end{bmatrix}. \end{aligned} \tag{47}$$

Using this augmented model, it is evident that the sensor fault reconstruction (SFR) can be handled similar to the AFR procedure discussed in the previous section.

5. Simultaneous Sensor and Actuator Faults Reconstruction

In this section, a more general case is investigated where the sensor and actuator faults occur simultaneously, i.e., $f_s(t) \neq 0$ and $f_a(t) \neq 0$. Before proceeding to the main results, some precalculations need to be done. As discussed earlier, the nonsingular transformation matrices $H = [H_1, H_2]^T$ and $\mathcal{T}$ exist such that:

$$\begin{aligned} &\mathcal{T} A \mathcal{T}^{-1} = \begin{bmatrix} A_1 & A_2 \\ A_3 & A_4 \end{bmatrix}, \mathcal{T} B = \begin{bmatrix} B_1 \\ 0 \end{bmatrix}, \\ &\mathcal{T} F_a = \begin{bmatrix} F_{a_1} \\ 0 \end{bmatrix}, \mathcal{T} D = \begin{bmatrix} D_1 \\ 0 \end{bmatrix}, \mathcal{T} M = \begin{bmatrix} M_1 \\ 0 \end{bmatrix}, \\ &H C \mathcal{T}^{-1} = \begin{bmatrix} C_1 & 0 \\ 0 & C_4 \end{bmatrix}, H F_s = \begin{bmatrix} 0 \\ F_{s_2} \end{bmatrix}. \end{aligned} \tag{48}$$

Then, the system (1) in the new coordinates $\bar{z} = \mathcal{T} z = \begin{bmatrix} \bar{z}_1 \\ \bar{z}_2 \end{bmatrix}$ and $\bar{y} = H y = \begin{bmatrix} \bar{y}_1 \\ \bar{y}_2 \end{bmatrix}$ is decomposed as

$$\begin{cases} \dot{\bar{z}}_1(t) = A_1 \bar{z}_1(t) + A_2 \bar{z}_2(t) + B_1 u(t) \\ \qquad\quad + F_{a_1} f_a(t) + D_1 d(t) + M_1 \partial(t, y, u) \\ \bar{y}_1(t) = C_1 \bar{z}_1(t) \end{cases} \tag{49}$$

$$\begin{cases} \dot{\bar{z}}_2(t) = A_3 \bar{z}_1(t) + A_4 \bar{z}_2(t) \\ \bar{y}_2(t) = C_4 \bar{z}_2(t) + F_{s_2} f_s(t) \end{cases}. \tag{50}$$

By defining $\bar{z}_3 = \begin{bmatrix} \bar{z}_2 \\ f_s \end{bmatrix}$ and $C_5 = \begin{bmatrix} C_4 & F_{s_2} \end{bmatrix}$, Equation (51) can be rewritten as:

$$\begin{aligned} N\dot{\bar{z}}_3 &= \bar{A}_3\bar{z}_1 + \bar{A}_4\bar{z}_3 + \bar{F}_{s_2}f_s \\ \bar{y}_2 &= C_5\bar{z}_3 \end{aligned} \tag{51}$$

where

$$\begin{cases} \dot{\bar{z}}_1(t) = A_1\bar{z}_1(t) + A_2\bar{z}_2(t) + B_1u(t) \\ \qquad\quad +F_{a_1}f_a(t) + D_1d(t) + M_1\partial(t,y,u) \\ \bar{y}_1(t) = C_1\bar{z}_1(t) \end{cases} \tag{52}$$

$$\begin{cases} \dot{\bar{z}}_2(t) = A_3\bar{z}_1(t) + A_4\bar{z}_2(t) \\ \bar{y}_2(t) = C_4\bar{z}_2(t) + F_{s_2}f_s(t) \end{cases}. \tag{53}$$

Considering $\bar{A}_2 = [A_2, 0]$, from (50) one obtains:

$$\begin{aligned} \dot{\bar{z}}_1(t) = {} & A_1\bar{z}_1(t) + \bar{A}_2\bar{z}_3(t) + B_1u(t) + F_{a_1}f_a(t) \\ & +D_1d(t) + M_1\partial(t,y,u) \end{aligned}. \tag{54}$$

Combining (50)–(54), we get:

$$\begin{cases} \dot{\bar{z}}_1(t) = A_1\bar{z}_1(t) + \bar{A}_2\bar{z}_3(t) + B_1u(t) \\ \qquad\quad +F_{a_1}f_a(t) + K_1\psi(t,y,u) \\ \bar{y}_1(t) = C_1\bar{z}_1(t) \end{cases} \tag{55}$$

$$\begin{cases} N\dot{\bar{z}}_3 = \bar{A}_3\bar{z}_1 + \bar{A}_4\bar{z}_3 + \bar{F}_{s_2}f_s \\ \bar{y}_2 = C_5\bar{z}_3 \end{cases} \tag{56}$$

where $K_1 = [D_1, M_1], \bar{K}_2 = [\bar{D}_2, \bar{M}_2], \psi(t,y,u) = \begin{bmatrix} d(t) \\ \partial(t,y,u) \end{bmatrix}$. Based on the above results, the following theorem characterizes the proposed method for simultaneous reconstruction of the sensor and actuator faults in the presence of disturbances and uncertainties.

Theorem 4. *Consider the faulty system (55) and (56), and assume the observer structure as:*

$$\begin{aligned} (N + VC_5)\dot{x} = {} & (\bar{A}_4 - L_1C_5)x + L_2(y_1 - C_1\hat{\bar{z}}_1) \\ & +\bar{A}_3\hat{\bar{z}}_1 + \bar{A}_4(N + VC_5)^{-1}Vy_2 \end{aligned} \tag{57}$$

$$\hat{\bar{z}}_3 = x + (N + VC_5)^{-1}Vy_2 \tag{58}$$

$$\dot{\hat{\bar{z}}}_1 = A_1\hat{\bar{z}}_1 + \bar{A}_2\hat{\bar{z}}_3 + B_1u(t) + \bar{G}_nv(t) - \bar{G}_le_{y_1}. \tag{59}$$

Then, the observer error is bounded if there exist $P_1 = P_1^T$, $P_3 = P_3^T$, and K satisfying the following LMIs

$$\underbrace{\begin{bmatrix} Q_{11} & P_1\bar{A}_2 & P_1K \\ * & Q_{22} & Q_{23} \\ * & * & -I \end{bmatrix}}_{Q} < 0 \tag{60}$$

$$P_1 > 0, \;\; P_3 > 0 \tag{61}$$

where $*$ denotes the transpose of each symmetric element, and

$$\begin{aligned} Q_{11} &= A_1^TP_1 + P_1A_1 + \lambda^2\|\mathcal{T}^{-1}\|^2I_m, \\ Q_{23} &= P_3(N + VC_5)^{-1} \\ Q_{22} &= Q_{23}\bar{A}_4 - KC_5 - \bar{A}_4^TQ_{23}^T - C_5^TK^T \\ & \quad +\lambda^2\|\mathcal{T}^{-1}\|^2I_{n+p-2m} \end{aligned}. \tag{62}$$

Furthermore, the observer error bound is given as follows

$$\|e\| < \sigma = 2\lambda_{\min}^{-1}(Q)\left\| P_3 \left[\begin{array}{c} 0 \\ (V_2\bar{F}_{s_2})^{-1} \end{array} \right] \right\| + \sigma_0 \tag{63}$$

where σ_0 is a small positive scalar.

Proof. For the observer (57)–(59), with $e_{y_1} = \hat{y}_1 - y_1 = C_1 e_1$ we have:

$$v(t) = -\rho_0(\hat{\bar{z}}_1 - \bar{z}_1)(\|(\hat{\bar{z}}_1 - \bar{z}_1)\|)^{-1} \tag{64}$$

$$\begin{array}{l} L_2 = \bar{A}_3 C_1^{-1} \\ L_1 = (N + VC_5)P_3 K \end{array} \tag{65}$$

where ρ_0 is a positive scalar. Defining $e_1 = \hat{\bar{z}}_1 - \bar{z}_1$, $e_3 = \hat{\bar{z}}_3 - \bar{z}_3$ and using (57)–(59) one obtains:

$$\begin{array}{r} \dot{e}_1 = (A_1 - \bar{G}_l C_5)e_1 + \bar{A}_2 e_3 - F_{a_1} f_a(t) \\ +\bar{G}_n v(t) - K_1 \psi(t, y, u) \\ \dot{e}_3 = -\underbrace{Q_{23}\bar{A}_4(N + VC_5)^{-1}}_{Q'_{23}} e_3 - Q_{23}\bar{F}_{s_2} f_s \end{array} . \tag{66}$$

Defining $s(t) = e_1(t)$ and $V = \frac{1}{2}s^T P_1 s$, we get:

$$\dot{V} \leq \|P_1 e_1\| \left(\begin{array}{c} \|(A_1 - \bar{G}_l C_5)e_1 + \bar{A}_2 e_3\| - \rho_0 \bar{G}_n \\ -\|F_{a_1} f_a(t)\| - \|K_1 \psi(t, y, u)\| \end{array} \right) \tag{67}$$

where $\bar{G}_n v(t) = -\rho_0 \bar{G}_n e_1 \|e_1\|^{-1}$. Choosing $\rho_0 \geq \|\bar{G}_n^{-1}\| (l\|(A_1 - \bar{G}_l C_5)\| + \|\bar{A}_2\| - \|F_{a_1}\| - \|K_1\|)\varepsilon$, implies that after a finite time, we have $e_1(t) = \dot{e}_1(t) = 0$. Then, one obtains

$$0 = \bar{A}_2 e_3 - F_{a_1} f_a(t) + \bar{G}_n v(t) - K_1 \psi(t, y, u). \tag{68}$$

Now, the following actuator reconstruction signal is defined:

$$\hat{f}_a = W \bar{G}_n v_{eq}(t) \tag{69}$$

where $W = F_{a_1}^{-1}$. To preserve sliding motion, $v(t)$ must take in the average $v_{eq}(t) = -\rho_0 e_1(\|e_1\| + \varepsilon)^{-1}$. Then, multiplying (68) by W results:

$$\hat{f}_a = -W\bar{A}_2 e_3 + f_a(t) + WK_1 \psi(t, y, u). \tag{70}$$

Then, one obtains

$$\begin{array}{l} \hat{f}_a - f_a = -W\bar{A}_2 e_3 + \underbrace{WK_1}_{Z} \psi(t, y, u) \\ \rightarrow \left\| \hat{f}_a - f_a \right\| < v' + \|Z\| \psi(t, y, u) \end{array} \tag{71}$$

where $v' > \|W\bar{A}_2 e_3\|$ is a small positive constant. Assuming $\|Z\|_\infty < v''$ with $v'' > 0$, results:

$$\left\| \hat{f}_a - f_a \right\| < v. \tag{72}$$

For small v, it implies $\hat{f}_a \approx f_a$. Then, from (66) one obtains

$$\dot{e}_3 + Q'_{23} e_3 = \sigma = -Q_{23}\bar{F}_{s_2} f_s. \tag{73}$$

Now, the following sensor reconstruction signal is defined:

$$\hat{f}_s = W_s \sigma \tag{74}$$

where $W_s = (-Q_{23}\bar{F}_{s_2})^{-1}$. Then, one obtains

$$\|\hat{\bar{z}}_3 - \bar{z}_3\| \leq \sigma. \tag{75}$$

This implies $\hat{f}_s \approx f_s$ and completes the proof. □

Remark 1. *Regarding the design parameters tuning, it is worthwhile to mention some points. The parameter ε in (34) should be initially selected as a very small scalar and then gradually increased such that it approximates the output error injection v and fulfills the design requirements. The parameter ξ in (41) is a small constant that satisfies the LMI conditions (43).*

Remark 2. *In practice, both sensor and actuator faults may occur simultaneously. Therefore, unlike most existing approaches dealing with sensor and actuator faults separately, our proposed approach takes care of simultaneous sensor and actuator faults, which can be a critical issue in some systems such as aircraft, wind turbines, etc., which need some more technical cares to have much better performance and efficiency.*

The step-by-step procedure of applying the proposed design algorithm is summarized as follows:

(I): Use (2) and (4) to obtain the new coordinate system as defined in Theorem 2.1 ($\mathcal{T}_c$, $\mathcal{T}_b$, and $\mathcal{T}$ are obtained).

(II): In the case of robust AFR, solve the LMI (44) to obtain $\tilde{P}$, which minimizes the effects of disturbances and uncertainties. Then, design the observer gains using (9) and (33). Finally, the actuator fault is reconstructed using (38).

(III): In the case of robust SFR, first construct the augmented system equations proposed in (48). Then, apply (I) and (II) to the augmented system to solve SFR.

(IV): In the case of simultaneous sensor and AFR, first use (49) to obtain the new coordinate system.

(V): Considering the proposed observer structure in Theorem 5.1, solve the LMIs (60) and (61) to obtain P_1, P_3, and K. Then, design the observer gains using (64) and (65). Finally, the actuator and sensor faults are reconstructed using (69) and (74).

6. Simulation Results

To verify the effectiveness of the proposed approaches, we consider a 5 MW wind turbine subject to the actuator and sensor faults in the presence of disturbances and uncertainties. The model and the parameters of the wind turbine used in the simulations are taken from [4] as following:

$$\begin{aligned}
\dot{x}(t) &= Ax(t) + Bu(t) + Dd(t) + M\partial(t,y) + Ff_a(t) \\
y(t) &= Cx(t) + F_s f_s(t)
\end{aligned}$$

$$A = \begin{bmatrix} 0 & 1.0000 & -0.0406 & 0 & 0 \\ -88.8900 & -0.8889 & 0.0361 & 6.685e-45 & 0 \\ 32552 & 325.2 & -13.22 & 0 & -0.1 \\ 0 & 0 & 0 & -6.6670 & 0 \\ 0 & 0 & 0 & 0 & -10 \end{bmatrix}, B = F = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 10 & 0 \\ 0 & 6.6667 \end{bmatrix}$$

$$C = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}, D = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}, M = \begin{bmatrix} 1 \\ -0.5 \\ 1 \\ 0 \\ 0 \end{bmatrix}, F_s = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

The state and the control input vectors are denoted as

$$x(t) = \begin{bmatrix} \Theta(t) & \Omega_r(t) & \Omega_g(t) & \beta(t) & T_g(t) \end{bmatrix}^T$$
$$u(t) = \begin{bmatrix} \beta_r(t) & T_{g,d}(t) \end{bmatrix}^T$$

where $\Theta(t)$ the torsion angle, $\Omega_r(t)$ the rotor speed, $\Omega_g(t)$ the generator speed, $\beta(t)$ the pitch angle, and $T_g(t)$ the generator torque are the state variables and $T_{g,d}(t)$ the desired generator torque and $\beta_r(t)$ the pitch angle command are the control input of the wind turbine model.

6.1. Actuator Fault Reconstruction

First, the following stabilizing controller is designed:

$$u(t) = \begin{bmatrix} -14.34 & -1.26 & -0.01 & 0.33 & -3.06 \\ 21.89 & 0.28 & 0.07 & -0.82 & 9.37 \end{bmatrix} x(t).$$

During the simulation, we assume $x(0) = [0.5, 1, 1, 1.5, 0.5]^T$; the disturbance $d(t) = u(t-25)$ and the uncertainty $\partial(t, y, u) = [0, 0.5, 2]y$ are also considered. It is easy to check that Assumption 1 is satisfied for this system, so the proposed method is applicable. Using the results in Theorem 2, the transformation matrix $\mathcal{T}_b$ is calculated as:

$$T_b = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 \end{bmatrix}.$$

Then, the LMI (44) is solved to minimize the effects of disturbances and uncertainties. Consequently, the observer gains and the AFR are obtained using (9), (33), and (38) as:

$$G_l = \begin{bmatrix} -0.1 & -3.3 & 0 \\ 0.01 & 0.1 & 0 \\ -1.55 & 71.6 & 0 \\ -0.8 & 1.55 & 0 \\ 0 & 0 & -0.1 \end{bmatrix}, G_n = \begin{bmatrix} -0.001 & 0.07 & 0 \\ 0.001 & -0.07 & 0 \\ 2.75 & 2.68 & 0 \\ 1.37 & 1.3 & 0 \\ 0 & 0 & -0.13 \end{bmatrix}.$$

The associated matrices L and P_0 are calculated as:

$$L = \begin{bmatrix} 1 & 1 & 0 \\ -1 & -1 & 0 \\ 0 & -1 & 1 \end{bmatrix}, P_0 = \begin{bmatrix} -13.35 & 14.06 & 0 \\ 14.06 & -14.05 & 0 \\ 0 & 0 & 7.53 \end{bmatrix}.$$

The Lyapunov matrix P is also obtained from (30):

$$P = \begin{bmatrix} 8.41 & 0 & 0.1 & -0.8 & 0 \\ 0 & 5.65 & -0.01 & 0 & 0 \\ 0.1 & 0 & 0 & -0.1 & 0 \\ -0.81 & 0 & -0.1 & 0.72 & 0 \\ 0 & 0 & 0 & 0 & 7.53 \end{bmatrix}.$$

The parameters ε and ρ are selected as 0.5 and 10, respectively. Then, choosing $\xi = 1 \times 10^{-3}$, the matrix W is calculated as $W = [-0.676, -0.581, -0.28]$.

Figures 1 and 2 show the effectiveness of the proposed AFR algorithm reconstructing faults simultaneously occurring in both actuators in the presence of the mentioned unknown disturbances/uncertainties.

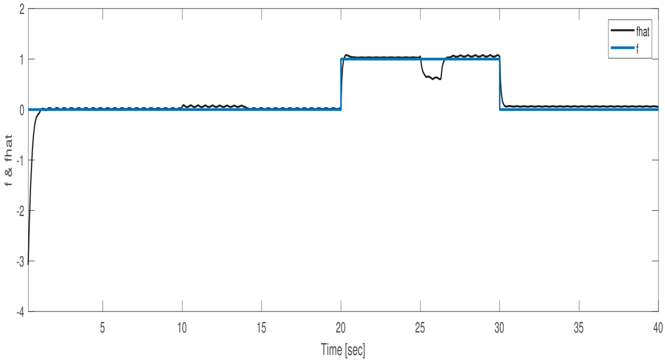


Figure 1. Illustration of robust actuator fault reconstruction (fault on the first actuator).

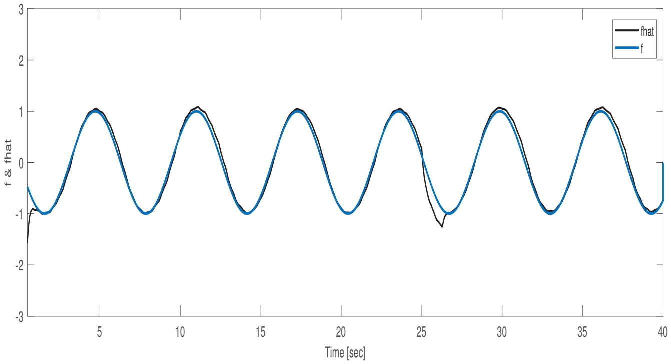


Figure 2. Illustration of robust actuator fault reconstruction (fault on the second actuator).

6.2. Sensor Fault Reconstruction

First, by choosing $A_n = 20I_{3\times3}$, the matrices of the associated augmented model in (48) are obtained.

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 8.9 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 25.5 & 32.25 & -13 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -7 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -10 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -20 & 0 & 0 & 0 \\ 0 & 20 & 0 & 0 & 0 & -20 & 0 & 0 \\ 0 & 0 & 20 & 0 & 0 & 0 & -20 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 10 & 0 \\ 0 & 6.68 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, D = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, M = \begin{bmatrix} 1 \\ 0.5 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, F_s = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 20 \end{bmatrix}.$$

Then, using a similar procedure the matrix T_b is obtained as:

$$T_b = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}.$$

Using this coordinate transformation, the equivalent model is obtained. Then, a stabilizing controller is designed as $u(t) = Kz(t)$, where:

$$K = \begin{bmatrix} -201.7 & -10.1 & -0.1 & 2.7 & -0.1 & 1.3 & -8.4 & 56 \\ -4214.3 & -6 & -18.7 & 31.5 & 0.3 & 1.2 & -188.2 & 84.4 \end{bmatrix}.$$

In this case, we assume $x(0) = [1, 0.5, 1, 0.5, 1.5, 1, 2, 0.5]^T, \partial(t, y, u) = [0.3, -0.5, 0], y = 0.3z_5 - 0.5z_6$, and $d(t) = u(t - 20)$. Using a similar procedure, the observer gains are obtained for the augmented system as:

$$G_n = \begin{bmatrix} -0.1 & 0 & -0.16 \\ 0 & -0.1 & 0.16 \\ -0.01 & 0 & -0.43 \\ -0.01 & -0.01 & 0.3 \\ 0 & 0 & 0.3 \\ 0 & -0.01 & -1.3 \\ -1.3 & 0 & 1.3 \\ -1.3 & -0.3 & -20.7 \end{bmatrix}, G_l = \begin{bmatrix} 0 & 0 & 0.04 \\ -0.01 & -0.01 & 0 \\ -0.45 & -2.15 & 0 \\ -0.69 & -0.97 & 0 \\ -14.26 & -938.31 & 0 \\ 2.77 & 239.93 & 0 \\ -0.74 & -2.77 & 0 \\ 0 & 0 & -0.07 \end{bmatrix}.$$

The parameters ε and ρ are selected as 0.1 and 15, respectively. Then, choosing $\xi = 1 \times 10^{-3}$, the matrix W is calculated as $W = [-0.651, -1.923, 0.309]$. In Figures 3–5, the performance of the proposed robust SFR is illustrated in the presence of the disturbances/uncertainties.

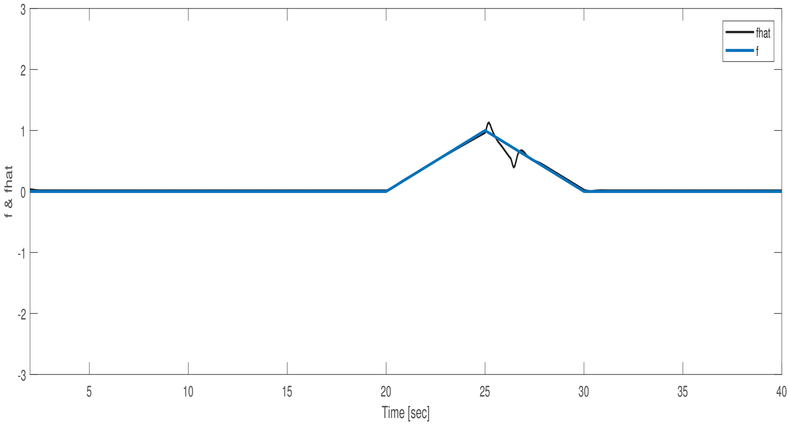


Figure 3. Illustration of robust sensor fault reconstruction (pitch angle sensor fault).

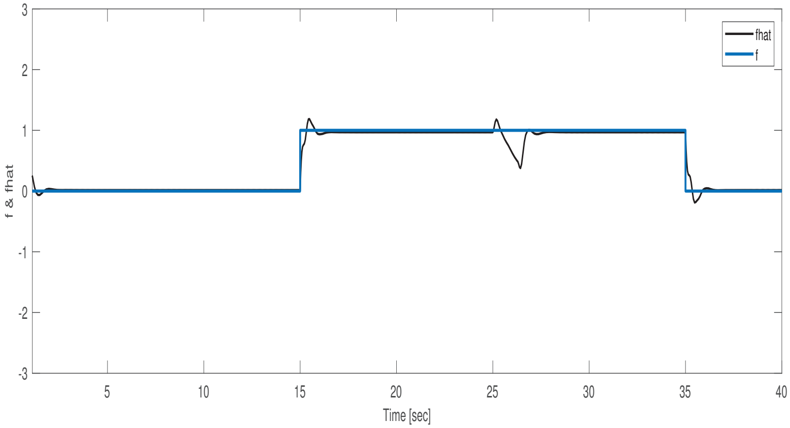


Figure 4. Illustration of robust sensor fault reconstruction (rotor speed sensor fault).

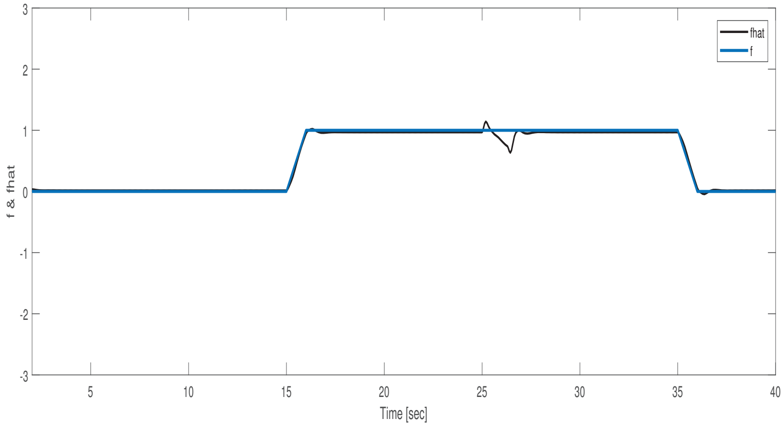


Figure 5. Illustration of robust sensor fault reconstruction (generator speed sensor fault).

6.3. Simultaneous Actuator and Sensor Faults

First, using the transformation in (49), the system is decomposed as in (50) and (51). Then, using the results in Theorem 5, the LMIs (60) and (61) are solved, and the observer gains are obtained from (64) and (65):

$$G_n = \begin{bmatrix} 0.41 & -0.43 & -1.25 \\ -0.27 & 0.12 & -0.16 \\ -1.16 & -0.24 & -2.25 \end{bmatrix}, G_l = \begin{bmatrix} 8.87 & -66.57 & 2.22 \\ 10.45 & -0.6 & 0.52 \\ -2.45 & 7.92 & -0.52 \end{bmatrix}.$$

Finally, the simultaneous actuator and sensor faults are reconstructed using (69) and (74). In this case, the parameters are chosen as given in the previous part. Figures 6 and 7 show the comparison of the simultaneous actuator and sensor fault reconstruction of the proposed method with [32] in the presence of $\partial(t, y, u) = [0.3, -0.5, 0],\ y = 0.3z_5 - 0.5z_6$, and $d(t) = u(t - 20)$. The results verify that despite the existence of unknown disturbance and uncertainty, the proposed method performs well in the reconstruction of both sensor and actuator faults.

Considering the dynamics of disturbance in the sliding mode observer design, there was a reduced impact of disturbance in fault reconstruction in comparison with the approach presented in [32]. In other words, the proposed approach in Theorem 5 has the quick response in the fault reconstruction process when disturbance is entered to the system. In order to measure and investigate the performance of the proposed methods, it is required to use quantitative criteria. In Table 1, the norm specifications of the sensor and actuator fault detection errors for the proposed approach and the method represented in [32] are calculated. As can be seen in Table 1, the proposed approach improves the accuracy of the actuator fault reconstruction more than 10% and the accuracy of the sensor fault reconstruction more than 4%.

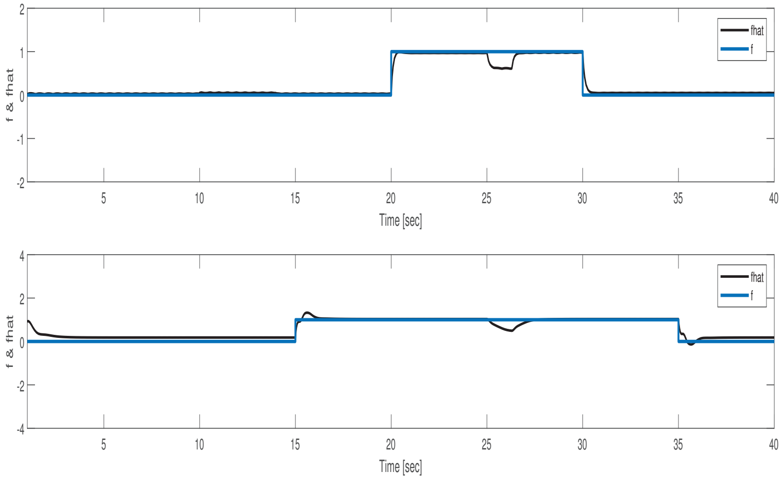


Figure 6. The simultaneous actuator and sensor fault reconstruction using the approach proposed in Theorem 5.

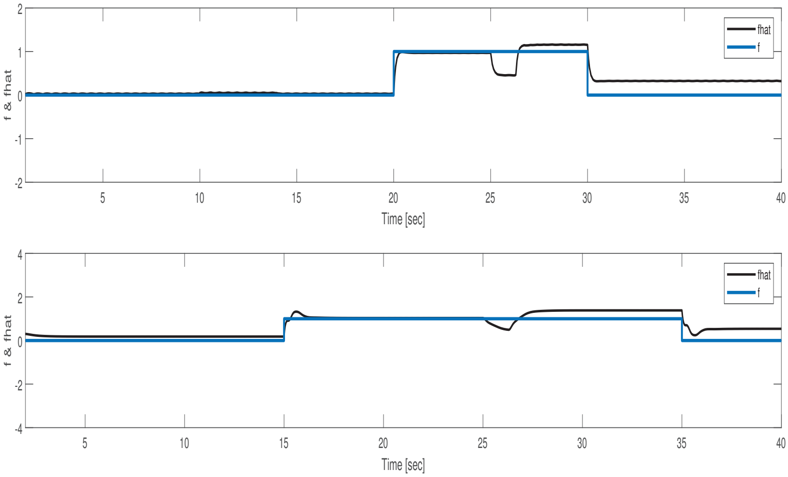


Figure 7. The simultaneous actuator and sensor fault reconstruction using the approach presented in [32].

Table 1. The comparison of the norm specification of the simultaneous faults reconstruction.

	$\|error\|_2(Actuator)$	$\|error\|_2(Sensor)$
Theorem 5	29.04	99.07
[32]	32.61	103.87
Improvement (%)	10.9	4.6

7. Conclusions

In this paper, an efficient robust approach is proposed for simultaneous sensor and actuator faults reconstruction in the presence of both unknown disturbance and uncertainty. First, an SMO-based method was proposed, and the observer gains were derived utilizing an LMI-based method. Then, considering that the system is subject to both disturbance and uncertainty, a robust reconstruction method is proposed, and incorporating the concept of BRL, the fault reconstruction problem is represented as an LMI problem and solved using the available tools. Furthermore, utilizing a wind turbine system, the performance and robustness of the proposed method were demonstrated. The proposed method can be robust against of disturbances and uncertainties, which is the most important advantage of our work. In contrast, the reconstruction of the faults is under the bounded disturbance, which can be our work's disadvantage. Finally, it is noted that although in many nonlinear systems, the nonlinearity and the effects of linearization error can be captured by the disturbances/uncertainties as considered in this paper, as an ongoing future work, it is quite beneficial to extend the proposed approach for pure nonlinear models in the presence of disturbances/uncertainties.

Author Contributions: Conceptualization, A.T. and F.B.; methodology, A.T. and K.H.; software, A.T. and A.B.; validation, F.B. and K.H.; formal analysis, A.T. and F.B.; investigation, K.H. and A.B; resources, A.T. and K.H.; data curation, A.T.; writing—original draft preparation, A.T. and F.B.; writing—review and editing, A.T and K.H.; visualization, A.T. and A.B.; supervision, F.B. and K.H.; project administration, F.B.; funding acquisition, A.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Saif, M.; Guan, Y. A new approach to robust fault detection and identification. *IEEE Trans. Aerosp. Electron. Syst.* **1993**, *29*, 685–695. [CrossRef]
2. Zhang, A.A.; Lv, B.C.; Zhang, C.Z.; She, D.Z. Finite time fault tolerant attitude control-based observer for a rigid satellite subject to thruster faults. *IEEE Access* **2017**, *5*, 16808–16817. [CrossRef]
3. Ghanbarpour, K.; Bayat, F.; Jalilvand, A. Wind turbines sustainable power generation subject to sensor faults: Observer-based MPC approach. *Int. Trans. Electr. Energy Syst.* **2020**, *30*, e12174. [CrossRef]
4. Taherkhani, A.; Bayat, F. Wind turbines robust fault reconstruction using adaptive sliding mode observer. *IET Gener. Transm. Distrib.* **2019**, *13*, 3096–3104. [CrossRef]
5. Yin, S.; Huang, Z. Performance monitoring for vehicle suspension system via fuzzy positivistic C-means clustering based on accelerometer measurements. *IEEE/ASME Trans. Mechatron.* **2014**, *20*, 2613–2620. [CrossRef]
6. Zhang, B.; Lu, S. Fault-tolerant control for four-wheel independent actuated electric vehicle using feedback linearization and cooperative game theory. *Control Eng. Pract.* **2020**, *101*, 104510. [CrossRef]
7. Sakthivel, R.; Selvaraj, P.; Mathiyalagan, K.; Park, J.H. Robust fault-tolerant Hinf control for offshore steel jacket platforms via sampled-data approach. *J. Frankl. Inst.* **2015**, *352*, 2259–2279. [CrossRef]
8. Gonzalez-Prieto, I.; Duran, M.J.; Rios-Garcia, N.; Barrero, F.; Martin, C. Open-switch fault detection in five-phase induction motor drives using model predictive control. *IEEE Trans. Ind. Electron.* **2017**, *65*, 3045–3055. [CrossRef]

9. Su, X.; Liu, X.; Song, Y.D. Fault-tolerant control of multiarea power systems via a sliding-mode observer technique. *IEEE/ASME Trans. Mechatron.* **2017**, *23*, 38–47. [CrossRef]
10. Ghanbarpour, K.; Bayat, F.; Jalilvand, A. Dependable power extraction in wind turbines using model predictive fault tolerant control. *Int. J. Electr. Power Energy Syst.* **2020**, *118*, 105802. [CrossRef]
11. Guzman, J.; Lopez-Estrada, F.R.; Estrada-Manzo, V.; Valencia-Palomo, G. Actuator fault estimation based on a proportional-integral observer with nonquadratic Lyapunov functions. *Int. J. Syst. Sci.* **2021**, *52*, 1938–1951. 2021.1873451. [CrossRef]
12. Tan, C.P.; Edwards, C. Sliding mode observers for robust detection and reconstruction of actuator and sensor faults. *Int. J. Robust Nonlinear Control* **2003**, *13*, 443–463. [CrossRef]
13. Sharma, R.; Aldeen, M. Fault detection in nonlinear systems with unknown inputs using sliding mode observer. In Proceedings of the 2007 American Control Conference, New York, NY, USA, 9–13 July 2007; pp. 432–437.
14. Rayankula, V.; Pathak, P.M. Fault Tolerant Control and Reconfiguration of Mobile Manipulator. *J. Intell. Robot. Syst.* **2021**, *101*, 1–18. [CrossRef]
15. Van Nguyen, T.; Ha, C. Experimental study of sensor fault-tolerant control for an electro-hydraulic actuator based on a robust nonlinear observer. *Energies* **2019**, *12*, 4337. [CrossRef]
16. Selvaraj, P.; Kaviarasan, B.; Sakthivel, R.; Karimi, H.R. Fault-tolerant SMC for Takagi–Sugeno fuzzy systems with time-varying delay and actuator saturation. *IET Control. Theory Appl.* **2017**, *11*, 1112–1123. [CrossRef]
17. Vaidyanathan, S.; Dolvis, L.G.; Jacques, K.; Lien, C.H.; Sambas, A. A new five-dimensional four-wing hyperchaotic system with hidden attractor, its electronic circuit realisation and synchronisation via integral sliding mode control. *Int. J. Model. Identif. Control* **2019**, *32*, 30–45. [CrossRef]
18. Mobayen, S.; Fekih, A.; Vaidyanathan, S.; Sambas, A. Chameleon Chaotic Systems With Quadratic Nonlinearities: An Adaptive Finite-Time Sliding Mode Control Approach and Circuit Simulation. *IEEE Access* **2021**, *9*, 64558–64573. [CrossRef]
19. Hou, Y.Y.; Fang, C.S.; Lien, C.H.; Vaidyanathan, S.; Sambas, A.; Mamat, M.; Johansyah, M.D. Rikitake dynamo system, its circuit simulation and chaotic synchronization via quasi-sliding mode control. *Telkomnika* **2021**, *19*, 1291–1301. [CrossRef]
20. Zhu, Q.; Li, Z.; Tan, X.; Xie, D.; Dai, W. Sensors fault diagnosis and active fault-tolerant control for PMSM drive systems based on a composite sliding mode observer. *Energies* **2019**, *12*, 1695. [CrossRef]
21. Yang, J.; Zhu, F. FDI design for uncertain nonlinear systems with both actuator and sensor faults. *Asian J. Control.* **2015**, *17*, 213–224. [CrossRef]
22. Taherkhani, A.; Bayat, F.; Mobayen, S.; Bartoszewicz, A. Dependable Sensor Fault Reconstruction in Air-Path System of Heavy-Duty Diesel Engines. *Sensors* **2021**, *21*, 7788. [CrossRef] [PubMed]
23. Zhang, X. Sensor bias fault detection and isolation in a class of nonlinear uncertain systems using adaptive estimation. *IEEE Trans. Autom. Control* **2011**, *56*, 1220–1226. [CrossRef]
24. Defoort, M.; Veluvolu, K.C.; Rath, J.J.; Djemai, M. Adaptive sensor and actuator fault estimation for a class of uncertain Lipschitz nonlinear systems. *Int. J. Adapt. Control. Signal Process.* **2016**, *30*, 271–283. [CrossRef]
25. You, F.; Li, H.; Wang, F.; Guan, S. Robust fast adaptive fault estimation for systems with time-varying interval delay. *J. Frankl. Inst.* **2015**, *352*, 5486–5513. [CrossRef]
26. Selvaraj, P.; Sakthivel, R.; Kwon, O.M. Synchronization of fractional-order complex dynamical network with random coupling delay, actuator faults and saturation. *Nonlinear Dyn.* **2018**, *94*, 3101–3116. [CrossRef]
27. Selvaraj, P.; Sakthivel, R.; Ahn, C.K. Observer-based synchronization of complex dynamical networks under actuator saturation and probabilistic faults. *IEEE Trans. Syst. Man Cybern. Syst.* **2018**, *49*, 1516–1526. [CrossRef]
28. Pinto, H.L.D.C.P.; Oliveira, T.R.; Hsu, L. Sliding mode observer for fault reconstruction of time-delay and sampled-output systems—A Time Shift Approach. *Automatica* **2019**, *106*, 390–400. [CrossRef]
29. Yang, C.; Liu, J.; Zeng, Y.; Xie, G. Real-time condition monitoring and fault detection of components based on machine-learning reconstruction model. *Renew. Energy* **2019**, *133*, 433–441. [CrossRef]
30. Harkat, M.F.; Mansouri, M.; Abodayeh, K.; Nounou, M.; Nounou, H. New sensor fault detection and isolation strategy–based interval-valued data. *J. Chemom.* **2020**, *34*, e3222. [CrossRef]
31. Lee, D.J.; Park, Y.; Park, Y.S. Robust H_∞ Sliding Mode Descriptor Observer for Fault and Output Disturbance Estimation of Uncertain Systems. *IEEE Trans. Autom. Control* **2012**, *57*, 2928–2934. [CrossRef]
32. Ben Brahim, A.; Dhahri, S.; Ben Hmida, F.; Sellami, A. Simultaneous actuator and sensor faults reconstruction based on robust sliding mode observer for a class of nonlinear systems. *Asian J. Control* **2017**, *19*, 362–371.
33. Brahim, A.B.; Dhahri, S.; Hmida, F.B.; Sellami, A. Simultaneous actuator and sensor faults estimation design for LPV systems using adaptive sliding mode observers. *Int. J. Autom. Control* **2021**, *15*, 1–27. [CrossRef]
34. You, G.; Xu, T.; Su, H.; Hou, X.; Li, J. Fault-tolerant control for actuator faults of wind energy conversion system. *Energies* **2019**, *12*, 2350. [CrossRef]
35. Tan, C.P.; Edwards, C. An LMI approach for designing sliding mode observers. *Int. J. Control* **2001**, *74*, 1559–1568. [CrossRef]
36. Alwi, H.; Edwards, C.; Tan, C.P. *Fault Detection and Fault-Tolerant Control Using Sliding Modes*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2011.

37. Bhat, S.P.; Bernstein, D.S. Finite-time stability of continuous autonomous systems. *SIAM J. Control. Optim.* **2000**, *38*, 751–766. [CrossRef]
38. Mahmoud, C.; Pascal, G. H_∞ design with pole placement constraints: An LMI approach. *IEEE Trans. Autom. Control* **1996**, *41*, 358–367.

energies

MDPI

Article

Electricity Pattern Analysis by Clustering Domestic Load Profiles Using Discrete Wavelet Transform

Senfeng Cen, Jae Hung Yoo and Chang Gyoon Lim *

Department of Computer Engineering, Chonnam National University, Yeosu 59626, Korea;
jasoncsf.7@gmail.com (S.C.); jhy@jnu.ac.kr (J.H.Y.)
* Correspondence: cglim@jnu.ac.kr; Tel.: +82-61-659-7254

Abstract: Energy demand has grown explosively in recent years, leading to increased attention of energy efficiency (EE) research. Demand response (DR) programs were designed to help power management entities meet energy balance and change end-user electricity usage. Advanced real-time meters (RTM) collect a large amount of fine-granular electric consumption data, which contain valuable information. Understanding the energy consumption patterns for different end users can support demand side management (DSM). This study proposed clustering algorithms to segment consumers and obtain the representative load patterns based on diurnal load profiles. First, the proposed method uses discrete wavelet transform (DWT) to extract features from daily electricity consumption data. Second, the extracted features are reconstructed using a statistical method, combined with Pearson's correlation coefficient and principal component analysis (PCA) for dimensionality reduction. Lastly, three clustering algorithms are employed to segment daily load curves and select the most appropriate algorithm. We experimented our method on the Manhattan dataset and the results indicated that clustering algorithms, combined with discrete wavelet transform, improve the clustering performance. Additionally, we discussed the clustering result and load pattern analysis of the dataset with respect to the electricity pattern.

Keywords: demand response; discrete wavelet transform; Pearson's correlation coefficient; principal component analysis; clustering

Citation: Cen, S.; Yoo, J.H.; Lim, C.G. Electricity Pattern Analysis by Clustering Domestic Load Profiles Using Discrete Wavelet Transform. *Energies* **2022**, *15*, 1350. https://doi.org/10.3390/en15041350

Academic Editors: Sergio Nesmachnow and Islam Safak Bayram

Received: 10 December 2021
Accepted: 11 February 2022
Published: 13 February 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

1. Introduction

Smart grid technologies and applications capable of adaptive, resilient, and sustainable self-healing, with foresight for prediction under different uncertainties, improve the reliability of the power system [1]. Furthermore, the smart grid allows bidirectional communication that supports the demand response (DR) programs [2]. Demand response technologies are widely applied and are constantly improving. The most common DR programs can be categorized into the following two classes: price-based programs and incentive-based programs. Price-based programs contain time of use (ToU), real time pricing (RTP) and critical peak pricing (CPP), which aim to motivate the end-user to change their consumption behavior [3]. On the other hand, incentive-based programs reach a consensus with consumers to reduce electricity consumption. Examples of these schemes are direct-load control (DLC), interruptible/curtailable service (I/C), demand bidding/buy (DB), etc. [4]. Considering various end-user consumption behaviors, it required the utility companies to design reasonable strategies. Therefore, it is necessary to analyze end-users' consumption data to acquire the load patterns.

Advanced metering infrastructure (AMI) and smart meters have been adopted to automatically collect energy consumption data at a fine granular interval, which is usually in intervals of 1 h, 30 min, or even 30 s [5]. Most countries have vigorously deployed smart meters because of the potential value of consumption data [6]. The massive amount of data sampled by smart meters could be used for research, typically load forecasting, customer segmentation, pricing/incentive mechanism, scheduling and control [7].

However, the extracted load consumption data lack labels, hence, the need of clustering techniques to segment the electricity consumption data. In addition, with the high time resolution advanced smart meter implemented in the household, the massive data will increase the complexity of the clustering method, called the "curse of dimensionality" [8]. This is a problem for implementing clustering algorithms because most clustering algorithms become intractable to process high-dimensional data input. To deal with the issue of the curse of dimensionality, the load consumption data needs preprocessing i.e., dimensionality reduction.

This study proposed clustering for segment residential customer daily power data, using discrete wavelet transform to extract features and reduce dimension by statistical methods and principal component analysis (PCA). The dataset, named Multifamily Residential Electricity Dataset (MFRED), contains 10-s resolution daily power data for 26 apartment groups, collected over 365 days in Manhattan, New York, 2019 [9]. First, data cleansing and multi-level one-dimensional (1D) discrete wavelet transform were applied on 8640-value daily load curves. Second, we reduced extracted feature dimensions. Finally, clustering algorithms were implemented, and the evaluation of the methods was carried out. Our main contributions of this work include the following: (1) a proposed method to vastly reduce the daily load profile dimensionality, to accelerate the clustering, and (2) the three cluster validity indices (CVI) imply that our proposed method to extract features outperforms the clustering original data, especially on hierarchical clustering.

The paper is structured as follows: Section 2 briefly discusses the related works. Section 3 describes the MFRED data. Section 4 explains the methodology in the study. Analysis and results are presented in Section 5, with conclusions in Section 6.

2. Related Works

Clustering is unsupervised learning, which could group similar data with no label attached to them [10]. Clustering algorithms can be classified into partitioning algorithms, hierarchical algorithms, density-based algorithms, and grid-based algorithms [11]. The authors of [12] implemented an improved K-means clustering method on load curves and verified that it performed better than the original K-means algorithm. The authors in [13] used modified fuzzy c-means (FCM) to extract representative load profiles of the customers. Ordering points to identify the clustering structure (OPTICS) is one of the density-based clustering models used to analyze consumer bid-offers in [14]. Gaussian mixture model (GMM) clustering is widely used to segment households' load profiles for demand response [15].

Additionally, most clustering algorithms cannot properly process high dimensionality data [16]. Most of the aforementioned works extracted consumption load patterns in terms of hourly, 30-min, 15-min load data. However, the advanced high-frequent smart meter could extract load data in intervals of 1-min, 30-s, and even 1-s, leading to large-scale consumption data that increases computational complexity. Most clustering algorithms evaluating the belonged cluster are calculated by distance. High dimensionality data would consume more computational complexity in each iteration, resulting in more time consumption. Hence, there are numerous studies about dimensionality reduction on load curve clustering, using feature extraction, feature construction and feature selection. In [17], the authors developed electricity price schemes based on demand patterns, using k-means combined with PCA. In [18], the authors proposed singular value decomposition to extract features before k-means clustering and evaluate the error sum of squares (SSE) index to compare with direct clustering. In [19], they used a fused load curve k-means algorithm, based on "Haar" discrete wavelet transform for reduce dimension, to obtain the load patterns of consumers from China and the United States and evaluate clustering performance by four CVI [20]. Xiao et al. [21] proposed a fusion clustering algorithm to obtain the consumption characteristics, using load curve clustering, based on discrete wavelet transform (CC-DWT).

In this study, we implemented clustering to segment 10-s interval daily electricity consumption data, using multi-level discrete wavelet transform, Pearson correlation coefficient, and PCA techniques to preprocess the daily load profiles. The clustering evaluation result shows our proposed method outperformed the conventional methods, without reducing dimension.

3. Data

In this study, we used the Multifamily Residential Electricity Dataset (MFRED) [9], which consisted of 390 apartments, from 1 January to 31 December 2019. This dataset was collected by real-time metering and contained 246 million data from residential buildings in Manhattan, New York, USA. The resolution of data was one sample per 10-s, providing 8640 data points in each daily profile. During the one-year period, some advanced meters were offline due to various reasons (e.g., smart meters offline). Therefore, some electricity data were not recorded in MFRED.

In the MFRED, the percentages of building stock prior to 1940, between 1940–1980, post-1980 were 79%, 7%, and 14%, respectively. The ratios of the entire Manhattan building stock prior to 1940, between 1940–1980, post-1980 were 86%, 6%, 8%, respectively, which means the residential structure in our research is very similar to that of the whole of Manhattan. In addition, considering the privacy leakage, the 390 apartments' data were reconstructed into 26 groups, called apartment groups (AG), which means each AG is made up of 15 apartments that are more representative. Hence, the dataset recorded the average real power (kW), reactive power (kVAR) and consumption (kWh), over 15 apartments, from 26 apartment groups, every 10 s for 365 days. Here, we used one channel real power data for our research. Figure 1 shows the distribution of daily energy consumption, and the black dashed line represents the mean electricity consumption (8.21 kWh).

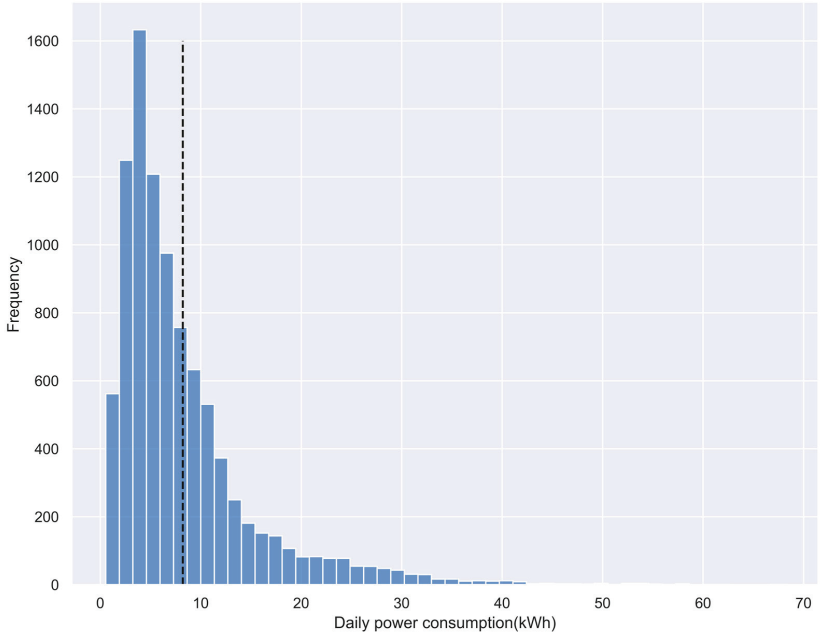


Figure 1. Daily energy consumption distribution.

4. Methodology

Our proposed method consists of the following four major stages: data cleansing, feature extraction, dimensionality reduction and clustering. Daily real power data are obtained from MFRED, and the data are cleansed for the missing value. Multi-level discrete wavelet transform is then applied to extract the features. In the dimensionality reduction stage, we implement the following two methods to decrease the dimension: statistical method combined with Pearson correlation and PCA. Finally, clustering algorithms were applied to segment daily load curves by using selected features. The proposed method is as shown in Figure 2.

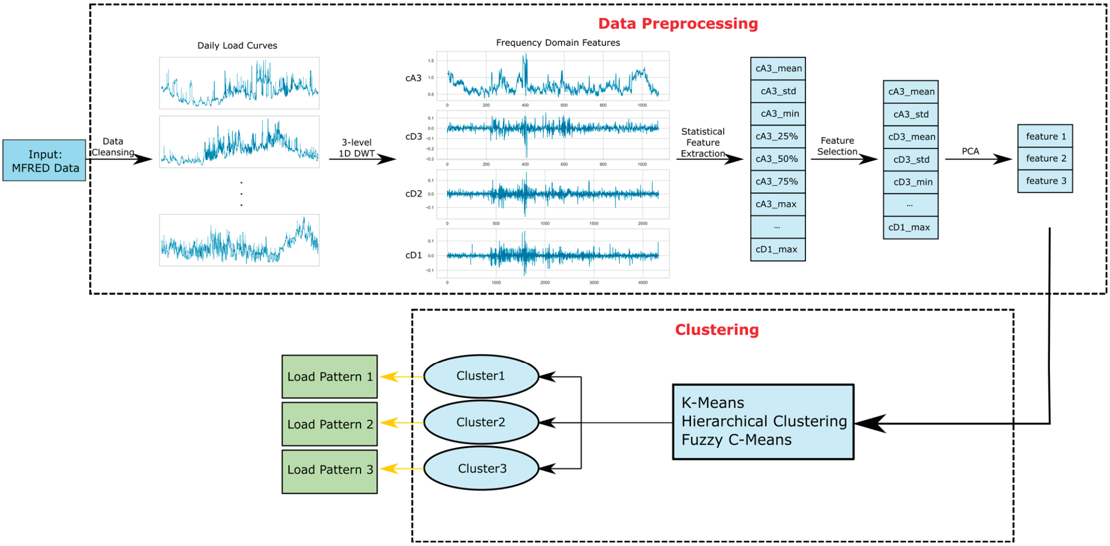


Figure 2. The proposed system diagram for electricity pattern analysis by clustering domestic load profiles.

4.1. Data Cleansing

Real and reactive power data were recorded in MFRED, where the real power data is reserved for the purpose of clustering. The primary issue with real power data is the missing values and anomalous values. Missing values are filled by averaging the previous and post 10-s values. However, tens of thousands of continuous data were missed because of the long-time breakdown of all meters on 09 July 2019, from 14:30 to 21:30 UTC. Therefore, this day is not taken into consideration due to the large amount of missing data. Anomalous values may be caused by the real-time meters (RTM) data collection accuracy, detected by the following five-number summary: the minimum, the maximum, the sample median, and the first and third quartiles. The single outlier was replaced by the average, the maximum and itself. After data cleansing, the reconstructed subset consisted of 8640 ten-second interval real power data (kW) in 364 days and 26 AGs. Thus resulting input data matrix dimension is 9464 × 8640. Figure 3 illustrates the 8640-value diurnal load curves from different AGs.

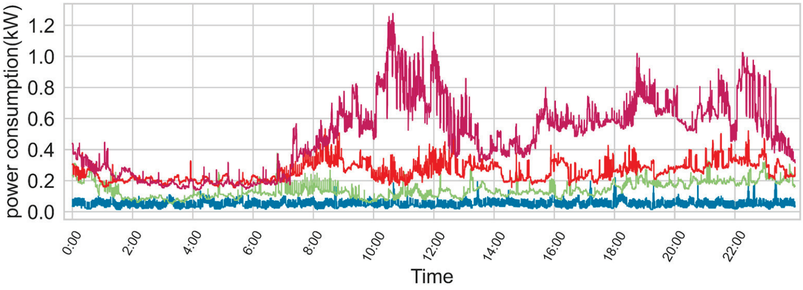


Figure 3. Daily load curves from different AGs. Each daily load curve consisted of 8640 values from meters every 10 s from 00:00:00 to 23:59:50.

4.2. Discrete Wavelet Transform

Wavelet transform contains continuous wavelet transform (CWT) and discrete wavelet transform (DWT). Discrete wavelet transform is widely used in waveform processing, including feature extraction in electroencephalography (EEG) [22], electromyography (EMG) [23], time-series load curves [24,25], etc. DWT decomposes the signal into various sets by passing through the low-pass filter and high-pass filter. The DWT and DWT coefficients are given by Equations (1) and (2), respectively, as follows:

$$\psi_{j,k}(t) = 2^{-\frac{j}{2}}\psi\left(2^{-j}t - k\right) \tag{1}$$

$$W_{j,k} = W\left(2^{j}, k2^{j}\right) = 2^{-j/2}\int_{-\infty}^{\infty}\overline{\psi(2^{-j}t - k)}dt \tag{2}$$

where k is a signal index and j is the scale index.

The detailed coefficients are obtained from a high-pass filter, while approximation coefficients are extracted from a low-pass filter, which could continue to decompose into a high-pass filter and low-pass filter. Figure 4 shows the decomposition of the 3-level 1-D discrete wavelet transform that we used in our research.

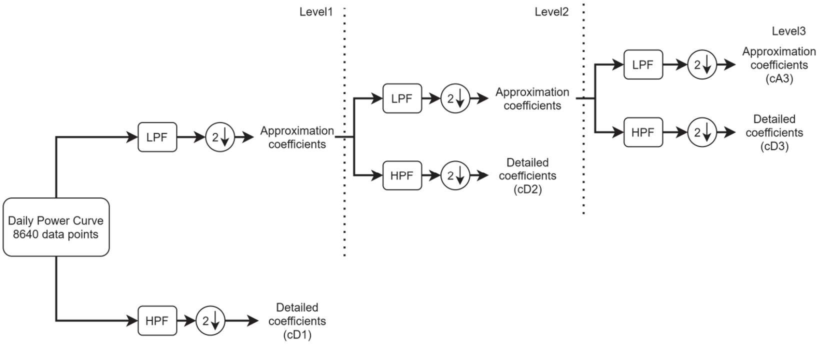

Level1
Level2
Level3
Daily Power Curve
8640 data points
LPF
2↓
Approximation coefficients
HPF
2↓
Detailed coefficients (cD1)
LPF
2↓
Approximation coefficients
HPF
2↓
Detailed coefficients (cD2)
LPF
2↓
Approximation coefficients (cA3)
HPF
2↓
Detailed coefficients (cD3)

Figure 4. Diagram of the multi-level 1-D discrete wavelet transform.

To extract features from daily load curves, we implemented the three-level 1-D Daubechies 4 (db4) discrete wavelet transform. Three-level means it will repeat one-level 1-D discrete wavelet transform three times based on the previous approximation coefficients. The Daubechies wavelet is preferred for feature extraction compared with Haar wavelet which is the special case of Daubechies noted as db1. Haar wavelet is the simplest and first wavelet transform which decomposes the discrete data using the two-length filter.

Eight of filter length in db4 wavelet contains more details but it involves slightly higher computational processes [19]. Thus, we employed db4 to compute the detailed coefficients and approximation coefficients. Three detailed coefficient sets and one approximation coefficient set are denoted as cD1, cD2, cD3, cA3, respectively. Figure 5 shows the four components of the daily load curve while using a three-level 1-D db4 discrete wavelet transform. The cA3 coefficients curve reflects a similar variation with the original load curve, while the value of cD3, cD2, cD1 components is very close to 0, which contains detailed information of daily load curve. For each daily power curve, the number of detailed coefficients (cD1, cD2, cD3) and approximation coefficients (cA3) were 4323, 2165, 1086 and 1086, respectively.

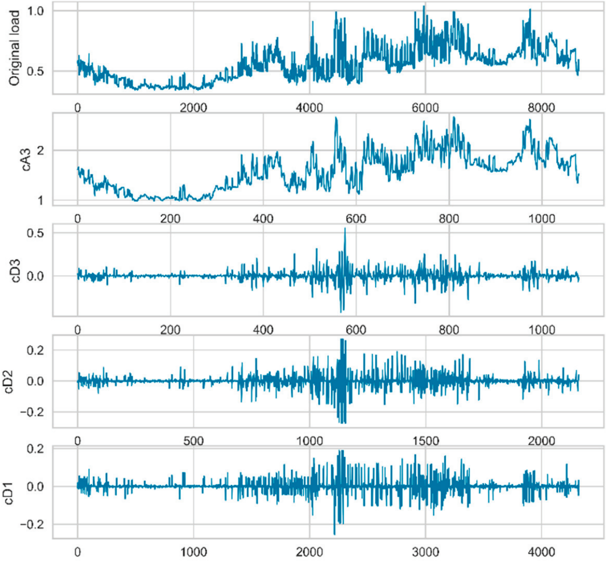


Figure 5. The curves of the components using the db4 wavelet.

4.3. Dimensionality Reduction

This phase aims to reduce the dimensions from extracted features (cA3, cD3, cD2, cD1). First, we used the statistical method to get the statistical variables (mean, std, min, 25%, 50%, 75%, max) from each daily coefficient (cA3_mean, cA3_std, cA3_min, cA3_25%, cA3_50%, cA3_75%, cA3_max, cD3_mean, etc.). There were 28 features extracted from the approximation and detailed coefficients. Second, we calculated Pearson's correlation coefficient, which measures the correlation of each two features. The correlation coefficient values are between −1 and 1, the value close to 1 represents a high positive correlation while the value close to −1 represents a high negative correlation [26]. High correlation features can be replaced by other features with similar characteristics. The correlation coefficient value is calculated from Equation (3), as follows:

$$r_{XY} = \frac{\sum_{i=1}^{n}\left(X_i - \overline{X}\right)\left(Y_i - \overline{Y}\right)}{\sqrt{\sum_{i=1}^{n}\left(X_i - \overline{X}\right)^2}\sqrt{\sum_{i=1}^{n}\left(Y_i - \overline{Y}\right)^2}} \tag{3}$$

where n is the number of samples, X_i, Y_i is the value of data, $\overline{X}$ is the mean value of X, and $\overline{Y}$ is the mean value of Y.

The correlation heatmap that represents the coefficient matrix is shown in Figure 6. According to the correlation heatmap, coefficients close to 1 or −1 imply redundant features. For the purpose of reducing the dimension, we removed one of the features in which the absolute values of correlation coefficients are bigger than 0.95. Figure 7 shows the correlation heatmap after eliminating the high correlation features.

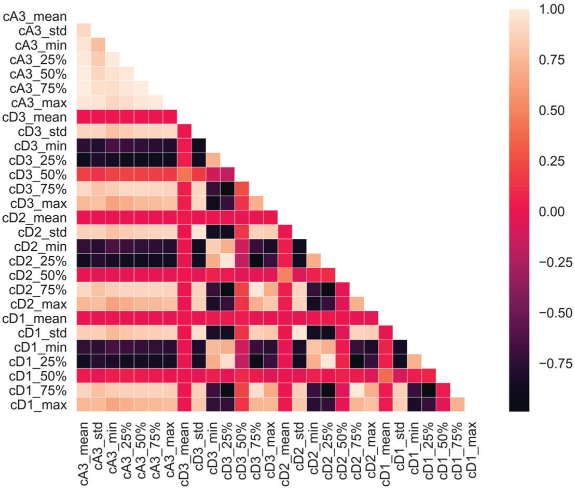


Figure 6. Correlation heatmap representing the coefficient matrix of 28 features.

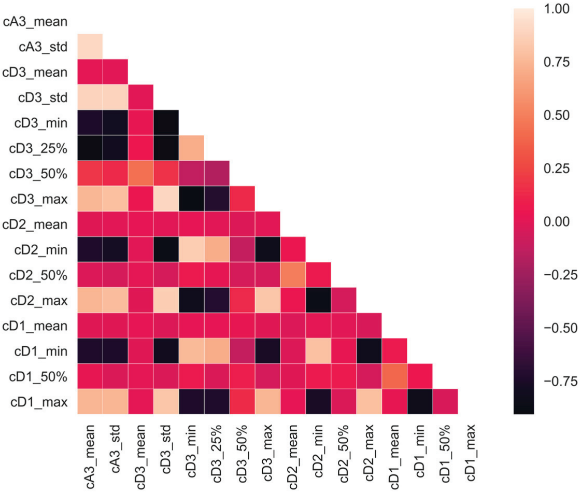


Figure 7. The selected features correlation heatmap after eliminating 12 highly correlated features.

PCA is one of the most appealing techniques that is widely used for dimensionality reduction of large data sets [27]. Given the original high dimensional data, PCA can map the data into k dimensions (k < original dimension) with principal components that are not related to each other and still preserve the original information. The data are normalized by z in the PCA process to obtain the feature vector composed of principal components by finding the covariance, eigenvector, and eigenvalue. In our study, we applied the PCA method to reduce the dimensionality to 3 components and still preserve 99 percent variability. Thus input dimension is reduced to 28 from 8640 using DWT combined with the statistical method and to 16 after correlation analysis. Finally, we have 3 component features applying PCA transform.

4.4. Clustering Method

Clustering load curves into groups is essential to identify load patterns [28]. For comparison purposes, we implemented the following three different clustering methods: k-means, hierarchical, and fuzzy c-means clustering.

4.4.1. K-Means Clustering

K-means is the most popular hard clustering algorithm goal to partition n data into k clusters, with the grouped data close to its centroid [29]. The K-means method is implemented as follows: First, determine the cluster number K then initial K centroids. Second, allocate each sample to the nearest centroids according to the distance. Third, determine the new K centroids that were generated by calculating the mean of the cluster points. Then, repeat the second and third steps until the centroids are completely unchanged.

Determining the number of clusters is one of the major challenges in clustering. The elbow method aims at finding the appropriate number of clusters by calculating the score for a range of values of K [30]. In our study, we determined this parameter by analyzing the following two metrics: distortion and Calinski–Harabasz score. Generally, distortion scoring computes the within cluster sum of squared (WCSS) to select cluster K [31]. Distortion score decreases with K increase. It is computed using Equation (4), as follows:

$$WCSS(K) = \sum_{h=1}^{K} \sum_{x_i \in c_h} \|x_i - \mu_h\|^2 \tag{4}$$

where K is the number of clusters, c_h is the cluster h, μ_h is the hth cluster center, $\|x_i - \mu_h\|^2$ is the Euclidean distance between data point x_i and its belonged centroid μ_h.

We applied Yellowbrick package to visualize the elbow method [32]. Figure 8 illustrates the WCSS value in different K. By applying the elbow method for $1 \leq K \leq 10$, the distortion score reduces rapidly with increase in K until K = 3 and then reduces gradually. We also employed Calinski–Harabasz analysis method in our study. It calculates the ratio of the sum of between-clusters dispersion and inter-cluster dispersion for all clusters, as follows:

$$CH(K) = \frac{\sum_{\text{h}=1}^{K} n_h \|c_h - c\|^2}{\sum_{\text{h}=1}^{K} \sum_{x^{(i)} \in c_h} \|x_i - c_h\|^2} \frac{N-K}{K-1} \tag{5}$$

where N is the total number of data points, K is the number of clusters, n_h and c_h are the number of points and centroids of the hth cluster, respectively, c is the centroid of data points. The higher value of $\sum_{h=1}^{K} n_h \|c_h - c\|^2$ means different cluster centroids are well separated, while the lower value of $\sum_{h=1}^{K} \sum_{x^{(i)} \in c_h} \|x_i - c_h\|^2$ indicates that the points of cluster are well centered. Therefore, the larger the value of the CH index, the more distinct the clusters.

Figure 9 shows the scores according to the change in the value of K, and it has a maximum value when K = 3. Even looking at the graph combined with the distortion and Calinski–Harabasz scores, it proves that it is the optimal solution for k = 3.

4.4.2. Hierarchical Clustering

Hierarchical clustering algorithms are formed by iteratively dividing the groups using bottom-up or top-down methods called agglomerative and divisive hierarchical clustering [33]. In this study, we employed agglomerative hierarchical clustering to segment load curves based on preprocessed features. The agglomerative builds up clusters starting with a single object as a single cluster and then using distance metric to merge the two most similar clusters [34]. Repeat until all of the objects are finally merged into a single cluster. We use "Ward" linkage to compute the distance between the new cluster and the rest of the clusters, minimizing the variance of the merged clusters [35]. Ward linkage criterion can be expressed as follows:

$$\Delta\left(X_i, X_j\right) = \frac{n_i n_j}{n_i + n_j} \left\| c(X_i) - c\left(X_j\right) \right\|^2 \tag{6}$$

where $c(X_i)$ is the centroid of cluster i, n_i denotes the number of points in cluster i.

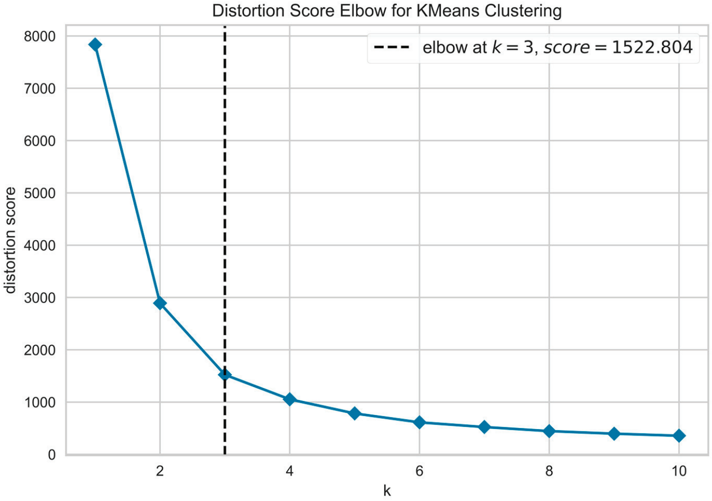


Figure 8. Elbow method estimated by distortion.

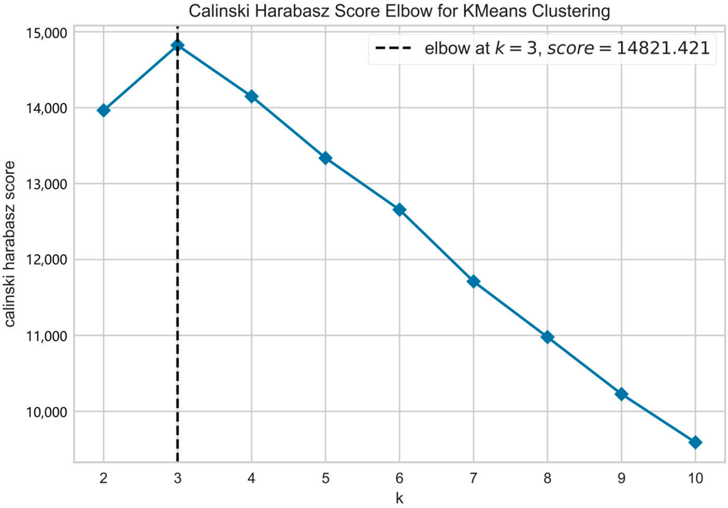


Figure 9. Elbow method estimated by Calinski–Harabasz.

Figure 10 depicts the Ward linkage truncated dendrogram which present a tree structure to vasualize the clusters and the number belonged each cluster. Ward's method dendrogram displays the clustering structure of the data. The numerical data in Figure 10 means the distance between different cluster centers which is calculated by Equation (6). The black dashed line represents the distance threshold which is 50. In addition, we combined the Calinski–Harabasz index with dendrogram to determine the optimal number of clusters (Table 1). According to the result, it can be confirmed that when *K* changes from 2 to 3, the Calinski–Harabasz index increases rapidly and then gradually increases thereafter. The Calinski–Harabasz index and dendrogram indicate that three is the optimal number for the value of *K*.

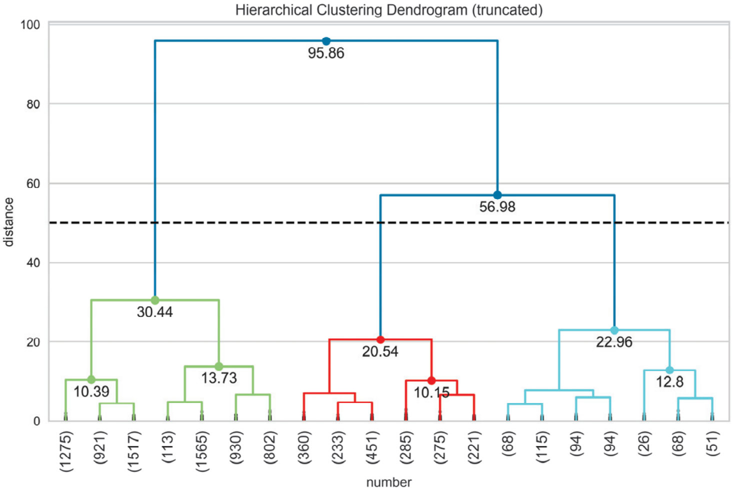


Figure 10. Agglomerative clustering dendrogram using Ward linkage.

Table 1. Calinski–Harabasz Index of agglomerative clustering.

Cluster(K)	Calinski–Harabasz Index
2	13,409.27
3	18,170.62
4	18,235.64
5	18,414.97
6	19,879.79
7	19,482.06
8	19,626.11
9	19,368.50

4.4.3. Fuzzy c-Means Clustering

The fuzzy c-means (FCM) algorithm is one of the soft clustering algorithms, also known as "soft K-means," where each data object can belong to multiple clusters. The fuzzy c-means algorithm has been widely used in many applications, such as consumer behavior and market segmentation [36]. FCM aims to minimize the objective function, as follows:

$$J_m = \sum_{i=1}^{N} \sum_{j=1}^{C} u_{ij}^{m} \|x_i - c_j\|^2 \tag{7}$$

where m is the fuzziness parameter in the range of [1,+∞), u_{ij} is the degree of membership of x_i in cluster j, c_j is the centroid of cluster j. The membership degree and cluster center will be updated iteratively until the objective function value is smaller than the error. The cluster center c_j and membership degree u_{ij} and can be obtained as follows:

$$c_j = \frac{\sum_{i=1}^{N} u_{ij}^m x_i}{\sum_{i=1}^{N} u_{ij}^m} \tag{8}$$

$$u_{ij} = \frac{1}{\sum_{k=1}^{C} \left(\frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{\frac{2}{m-1}}} \tag{9}$$

The algorithm comprises the following steps:

Step 1: Determine the number of clusters, fuzziness parameter m and the error ε.

Step 2: Initialize the membership matrix $U^{[0]}$ using $\sum\limits_{j=1}^{c} \mu_j(x_i) = 1$.

Step 3: At k step, compute the centroid c_k with equation (8).

Step 4: Update the new membership matrix $U^{[k]}, U^{[k+1]}$ with Equation (9).

Step 5: If $\|U^{[k+1]} - U^{[k]}\| < \varepsilon$, stop, else, return to step 3.

The main advantage of FCM is its suitability for overlapped data, its scalability and simplicity, and accuracy. However, the time complexity of fuzzy c-means is more than *k-means*. In our study, we selected the fuzziness index and error ε by grid search. The optimal fuzziness index was determined as m = 1.25, and the error as $\varepsilon = 1 \times 10^{-5}$. Figure 11 shows the clustering result based on three principal components. The points from Cluster 1 and Cluster 2 are relatively compact, but Cluster 3 is more dispersed.

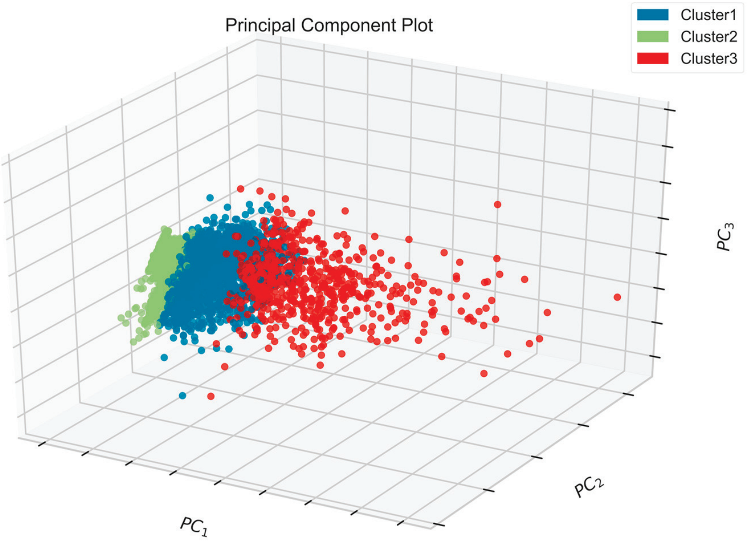


Figure 11. Clustering result with FCM applied when c is set to 3.

5. Experiment Results and Analysis

In the clustering phase, we employed three different clustering algorithms to segment different daily load curves. Considering the diversity of clustering performance evaluations, we selected three indices for validation, namely the silhouette coefficient, Calinski–Harabasz index, and Davies–Bouldin index (DBI) [37], which are internal clustering criteria. The Calinski–Harabasz index has been described in Section 4. The silhouette coefficient combines cohesion and separation. Cohesion indicates the similarity of points in the same cluster. On the contrary, separation indicates the object compared to other clusters. Specifically, the silhouette coefficient is calculated as follows:

$$SC = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{10}$$

where *a(i)* indicates that cohesion is the mean distance between a sample and all other points in the same cluster, and *b(i)* is the minimum value of the mean distance between an object and all other objects in the nearest cluster, then, the equations of *a(i)* and *b(i)* are as follows:

$$a(i) = \frac{1}{|C_i| - 1} \sum_{j \in C_i, i \neq j} d(i,j) \tag{11}$$

$$b(i) = \min_{k \neq i} \frac{1}{|C_k|} \sum_{j \in C_k} d(i,j) \tag{12}$$

The value of silhouette is in the range of $[-1{,}1]$. If the silhouette coefficient is close to 1, it means that the model is suitable; a negative value indicates incorrect clustering. Higher values of the silhouette coefficient imply that the model clustered well. Davies–Bouldin index measures the average similarity between clusters, where the similarity compares the distance between clusters with the size of the clusters themselves. For a given set of clusters $C = \{c_1, c_2, \ldots, c_k\}$, c_i is the most similar with c_j. Davies–Bouldin index is defined as follows:

$$DB = \frac{1}{k} \sum_{i=1}^{k} \max_{i \neq j} \frac{s_i + s_j}{d_{ij}} \tag{13}$$

where k is the cluster number, s_i is the average distance between all objects in cluster i and cluster i centroid, d_{ij} is the distance between *ith* and *jth* cluster centroids. The smaller value of the Davies–Bouldin index implies that the clusters are separated properly.

We compared our proposed method with the original clustering algorithm without reducing the dimension. Table 2 compares the three clustering results, presented by calculating cluster validity indexes. The name of clustering methods that include 'Original' denotes the daily load data without reducing dimensionality. N denotes that the daily load data were normalized by min–max normalization to rescale the data to fit in the range 0 to 1. Generally, normalizing the data before clustering could ignore the distance difference between different variables. Equation (14) presents the formula for min–max normalization, as follows:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{14}$$

According to the evaluation index, our wavelet-based preprocessing method slightly improves clustering performance compared to the original method. However, the performance of wavelet-based hierarchical clustering is better than hierarchical clustering without dimensionality reduction. Compared with the three wavelet-based clustering algorithms, the performance of k-means and FCM were similar, the silhouette coefficient and Davies–Bouldin index of FCM were better than k-means. For hierarchical clustering, the silhouette coefficient is the best, but the other two indices are worse than those of k-means and FCM. In addition, the proposed method significantly saves the computation time by dimensionality reduction.

Table 2. Clustering evaluation comparison results of the proposed methods.

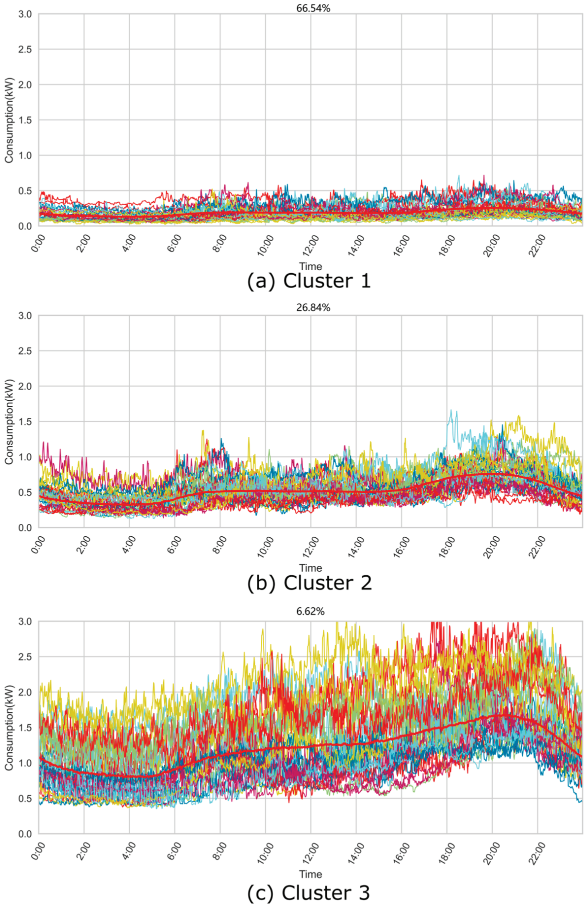
Methods	SC	CH	DBI
Wavelet based K-means	0.5101	13,643.7	0.7741
Wavelet based HC	0.5351	12,795.9	0.7909
Wavelet based FCM	0.5105	13,642.4	0.7736
Original_N_K-means	0.5103	13,642.4	0.7758
Original_N_HC	0.4105	11,431.5	0.8367
Original_N_FCM	0.5058	13,609.7	0.7759

Note: SC is Silhouette Coefficient, CH is Calinski–Harabasz Index, and DBI is Davies–Bouldin index. The larger the SC and CH values, the better. Conversely, the smaller the DBI values, the better.

Based on our comparison, we adopt the wavelet-based fuzzy c-means method. In three clusters, the first, second, and third clusters represent 66.54%, 26.84%, and 6.62% of the daily load curves, respectively. Figure 12 shows the load patterns of the three clusters and daily load curves. Cluster 1 and 3 represent the lowest and highest power consumption, respectively. In each cluster figure, the bold red line represents the representative load pattern, while the other curves represent the daily power usage in the cluster. Cluster 1 contains 6297 daily load curves, with stable power consumption; the average power usage and average peak power were 0.187 kW and 0.438 kW, respectively. Cluster 2 contains 2540 daily load curves; the average power usage and average peak power were 0.517 kW and 1.056 kW, respectively. Cluster 3 is composed of 627 daily load curves, which is the highest power usage group and has the highest variability. For Cluster 3, the average power usage and average peak power were 1.212 kW and 2.209 kW, respectively.

Figure 13 illustrates the average daily power usage box and whisker plot of three clusters. Boxplot could present data distribution based on a five-number summary, including minimum, first quartile, median, third quartile and maximum. There are some outliers (data point in Figure 13) in cluster 2, while in cluster 3, many outliers fall beyond the maximum value. As the power usage increases from cluster 1 to 3, the variation in power also increases. The standard deviation of cluster 1 is 0.0829 kW, cluster 2 is 0.1329 kW, and cluster 3 is 0.3193 kW.

Figure 14 shows the average power load pattern in four seasons of three clusters. It appears that the three clusters have similar power usage characteristics in the four seasons, i.e., the average power usage valley and peak at the same time every season, around 4 am and 8 pm, respectively. Moreover, the household generally needs to use air conditioners to control the indoor temperature during the summer; therefore, electricity usage is higher. The winter consumption in the three clusters is less than that of the summer, insinuating that most apartments have installed a heating system that is not taken into account in the electricity data. Looking at all four seasons, electricity demand is stably required between 8 am and 2 pm in Cluster 1 (a) and Cluster 2 (b). The section that consumes the most power is Cluster 3, and it can be seen that the power demand increases over time during the same period.



(a) Cluster 1

(b) Cluster 2

(c) Cluster 3

Figure 12. Daily load curves and load patterns of each cluster, (**a**) low load consumption group, (**b**) middle load consumption group, and (**c**) high load consumption and instability group.

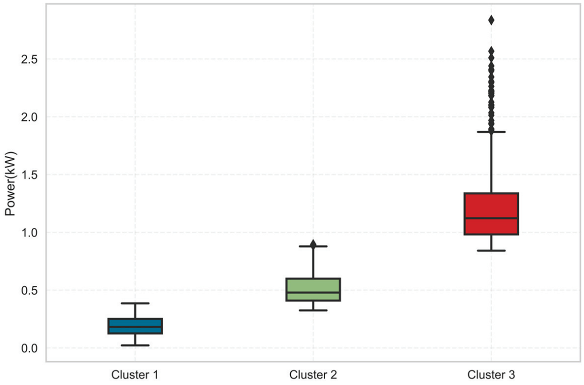
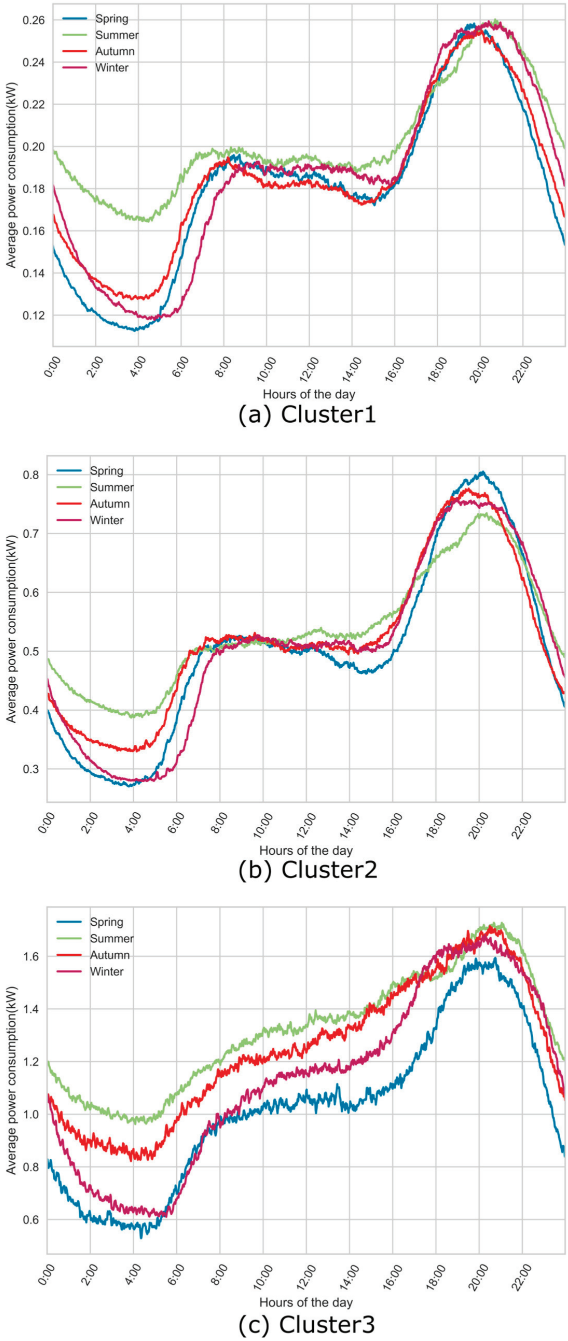


Figure 13. Box and whisker plot of average daily power (kW) usage in three clusters.



(a) Cluster1

(b) Cluster2

(c) Cluster3

Figure 14. Seasonal average load patterns in three clusters, (**a**) low load consumption group, (**b**) middle load consumption group, and (**c**) high load consumption and instability group in four seasons.

6. Conclusions

High-frequency smart meters have been broadly deployed for collecting electricity data. Our proposed method implements discrete wavelet transform to convert time-domain data to frequency domain. We extracted detailed and approximate signals using a statistical approach, then used Pearson's correlation coefficients to filter the high correlation features. To further reduce dimensionality, we applied PCA to preserve three features. The rest of the three features were used to achieve the clustering algorithm. Our study aimed at obtaining the representative load patterns from high time resolution daily load curves in Manhattan. Our method reduces the large dimensions to increase efficiency in clustering. In addition, it improves the clustering result slightly by estimating the silhouette coefficient, Calinski–Harabasz index, and Davies–Bouldin index, then comparing the clustering without discrete wavelet transform. From representative load patterns, the utility policymaker could design a reasonable demand response scheme to maintain the power system stability and help the utility maximize the profit and even reduce consumers' electricity fees. Based on Figure 14, policymakers could design three different advanced time of use tariffs, according to electricity consumption volume and representitive load curves from the three clusters. To each cluster, the electricity demand increases apparently from 4 pm to 8 pm, which could influence the power system stability. It means the appropriate DR scheme is significant during this period, such as load shifting/shedding. For future work, we suggest exploring the sub-cluster from the previous clusters to get more detailed load patterns based on our method.

Author Contributions: Conceptualization, S.C. and C.G.L.; methodology, S.C.; validation, S.C., C.G.L. and J.H.Y.; formal analysis and investigation, S.C.; writing—original draft preparation, S.C.; writing—review and editing, S.C., C.G.L. and J.H.Y.; visualization, S.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research is supported by Jeollannam-do (2021 R&D supporting program operated by Jeonnam Technopark) and financially supported by the Ministry of Trade, Industry and Energy (MOTIE) and Korea Institute for Advancement of Technology (KIAT) under the research project: "National Innovation Cluster R&D program" (Grant number: P0016223).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data is available in a publicly accessible repository. The data used in this study are openly available from the Scientific Data portal in https://www.nature.com/articles/s41597-020-00721-w, accessed on 13 January 2022.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Dileep, G. A survey on smart grid technologies and applications. *Renew. Energy* **2020**, *146*, 2589–2625. [CrossRef]
2. Zhou, X.; Ma, Y.; Gao, Z.; Wang, H. Summary of smart metering and smart grid communication. In Proceedings of the IEEE International Conference on Mechatronics and Automation, Takamatsu, Japan, 6–9 August 2017. [CrossRef]
3. Ahmed, Z.G.; Ahmed, A.H.; Nabil, H.A. Dynamic Pricing; Different Schemes, Related Research Survey and Evaluation. In Proceedings of the International Renewable Energy Congress, Hammamet, Tunisia, 20–22 March 2018.
4. Nojavan, S.; Ajoulabadi, V.; Khalili, T.; Member, S.; Bidram, A.; Member, S. Optimal Power Flow Considering Time of Use and Real-Time Pricing Demand Response Programs. *arXiv* **2021**, arXiv:2102.07828.
5. Liu, X.; Golab, L.; Golab, W.; Ilyas, I.F. Benchmarking smart meter data analytics. In Proceedings of the EDBT 2015 18th International Conference on Extending Database Technology, Brussels, Belgium, 23–27 March 2015; pp. 385–396. [CrossRef]
6. Zhou, S.; Brown, M.A. Smart meter deployment in Europe: A comparative case study on the impacts of national policy schemes. *J. Clean. Prod.* **2017**, *144*, 22–32. [CrossRef]
7. Antonopoulosa, I.; Robu, V.; Couraud, B.; Kirli, D.; Norbu, S.; Kiprakis, A.; Flynn, D.; Elizondo-Gonzalez, S.; Wattame, S. Artificial intelligence and machine learning approaches to energy demand-side response: A systematic review. *Renew. Sustain. Energy Rev.* **2020**, *130*, 109899. [CrossRef]

8. Molchanov, V.; Linsen, L. Overcoming the curse of dimensionality when clustering multivariate volume data. In Proceedings of the VISIGRAPP 2018-Proceedings of the 13th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, Funchal, Madeira, Portugal, 27–29 January 2018. [CrossRef]
9. Meinrenken, C.J.; Rauschkolb, N.; Abrol, S.; Chakrabarty, T.; Decalf, V.C.; Hidey, C.; McKeown, K.; Mehmani, A.; Modi, V.; Culligan, P.J. MFRED, 10 second interval real and reactive power for groups of 390 US apartments of varying size and vintage. *Sci. Data* **2020**, *7*, 375. [CrossRef]
10. Saxena, A.; Prasad, M.; Gupta, A.; Bharill, N.; Pate, P.O.; Tiwari, A.; Joo, E.M.; Weiping, D.; Chin-Teng, L. A review of clustering techniques and developments. *Neurocomputing* **2017**, *267*, 664–681. [CrossRef]
11. Rajabi, A.; Li, L.; Zhang, J.; Zhu, J.; Ghavidel, S.; Ghadi, M.J. A review on clustering of residential electricity customers and its applications. In Proceedings of the 2017 20th International Conference on Electrical Machines and Systems (ICEMS), Sydney, Australia, 11–14 August 2017. [CrossRef]
12. Zhang, M.; Sun, S.; Cao, G.; Kong, X.; Zhao, X.; Zong, S. Load characteristics analysis based on improved k-means clustering algorithm. In Proceedings of the 2019 IEEE 2nd International Conference on Automation, Electronics and Electrical Engineering (AUTEEE), Shenyang, China, 22–24 November 2019; pp. 510–515. [CrossRef]
13. Panapakidis, I.; Asimopoulos, N.; Dagoumas, A.; Christoforidis, G.C. An improved fuzzy c-means algorithm for the implementation of demand side management measures. *Energies* **2017**, *10*, 1407. [CrossRef]
14. Luo, Z.; Hong, S.H.; Ding, Y.M. A data mining-driven incentive-based demand response scheme for a virtual power plant. Appl. *Energy* **2019**, *239*, 549–559. [CrossRef]
15. Stephen, H.; Colin, S.; Peter, G. Analysis and Clustering of Residential Customers Energy Behavioral Demand Using Smart Meter Data. *IEEE Trans. Smart Grid* **2016**, *7*, 135–144.
16. Bouveyron, C.; Girard, S.; Schmid, C. High-dimensional data clustering. *Comput. Stat. Data Anal.* **2007**, *52*, 502–519. [CrossRef]
17. Chen, T.; Qian, K.; Mutanen, A.; Schuller, B.; Jarventausta, P.; Su, W.W. Classification of electricity customer groups towards individualized price scheme design. In Proceedings of the 2017 North American Power Symposium (NAPS), Morgantown, WV, USA, 17–19 September 2017; pp. 4–7. [CrossRef]
18. Wang, J.; Wang, K.; Jia, R.; Chen, X. Research on Load Clustering Based on Singular Value Decomposition and K-means Clustering Algorithm. In Proceedings of the 2020 Asia Energy and Electrical Engineering Symposium (AEEES), Chengdu, China, 29–31 May 2020; pp. 831–835. [CrossRef]
19. Jiang, Z.; Lin, R.; Yang, F.; Wu, B. A fused load curve clustering algorithm based on wavelet transform. *IEEE Trans. Ind. Inform.* **2018**, *14*, 1856–1865. [CrossRef]
20. Vij, A.; Khandnor, P. Validity of internal cluster indices. In Proceedings of the 2016 International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS), Bengaluru, India, 6–8 October 2016; pp. 388–395. [CrossRef]
21. Li, F.; Tian, Y.; Wu, Y.; Liu, Y. A method of mining electricity consumption behaviour based on CC-DWT. In *IOP Conference Series: Earth and Environmental Science*; IOP Publishing: Bristol, UK, 2021; p. 012133. [CrossRef]
22. Aliyu, I.; Lim, C.G. Selection of optimal wavelet features for epileptic EEG signal classification with LSTM. *Neural Comput. Appl.* **2021**, 1–21. [CrossRef]
23. Phinyomark, A.; Limsakul, C.; Phukpattaranont, P. Application of wavelet analysis in EMG feature extraction for pattern classification. *Meas. Sci. Rev.* **2011**, *11*, 45–52. [CrossRef]
24. Rhif, M.; Ben Abbes, A.; Farah, I.R.; Martínez, B.; Sang, Y. Wavelet transform application for/in non-stationary time-series analysis: A review. *Appl. Sci.* **2019**, *9*, 1345. [CrossRef]
25. Cugliari, J.; Goude, Y.; Poggi, J.-M. Disaggregated Electricity Forecasting using Wavelet-Based Clustering of Individual Consumers. In Proceedings of the 2016 IEEE International Energy Conference (ENERGYCON), Leuven, Belgium, 4–8 April 2016. [CrossRef]
26. Gogtay, N.J.; Thatte, U.M. Principles of correlation analysis. *J. Assoc. Physicians India* **2017**, *65*, 78–81. [PubMed]
27. Das, S.; Rao, P.S.N. Principal Component Analysis based Compression Scheme for Power System Steady State Operational Data. In Proceedings of the ISGT2011-India, Kollam, India, 1–3 December 2011. [CrossRef]
28. Rajabi, A.; Eskandari, M.; Ghadi, M.J.; Li, L.; Zhang, J.; Siano, P. A comparative study of clustering techniques for electrical load pattern segmentation. *Renew. Sustain. Energy Rev.* **2020**, *120*, 109628. [CrossRef]
29. Hartigan, J.A.; Wong, M.A. A K-means Clustering Algorithm. *J. R. Stat. Society. Ser. C (Appl. Stat.)* **1979**, *28*, 100–108.
30. Liu, F.; Deng, Y. Determine the Number of Unknown Targets in Open World Based on Elbow Method. *IEEE Trans. Fuzzy Systems.* **2021**, *29*, 986–995. [CrossRef]
31. Thinsungnoen, T.; Kaoungku, N.; Durongdumronchai, P.; Kerdprasop, K.; Kerdprasop, N. The Clustering Validity with Silhouette and Sum of Squared Errors. In Proceedings of the International Conference on Industrial Application Engineering, Kitakyushu, Japan, 28–31 March 2015; pp. 44–51. [CrossRef]
32. Bengfort, B.; Bilbro, R. Yellowbrick: Visualizing the Scikit-Learn Model Selection Process. *J. Open Source Softw.* **2019**, *4*, 1075. [CrossRef]
33. Chicco, G. Overview and performance assessment of the clustering methods for electrical load pattern grouping. *Energy* **2012**, *42*, 68–80. [CrossRef]
34. Ma, Z.; Yan, Y.; Li, K.; Nord, N. Building energy performance assessment using volatility change based symbolic transformation and hierarchical clustering. *Energy Build.* **2018**, *166*, 284–295. [CrossRef]

35. Vijaya, V.; Sharma, S.; Batra, N. Comparative Study of Single Linkage, Complete Linkage, and Ward Method of Agglomerative Clustering. In Proceedings of the of the International Conference on Machine Learning, Big Data, Cloud and Parallel Computing: Trends, Perspectives and Prospects (COMITC), Faridabad, India, 14–16 February 2019; pp. 568–573. [CrossRef]
36. Zhou, Z.; Yang, S.; Shao, Z. Household monthly electricity consumption pattern mining: A fuzzy clustering-based model and a case study. *J. Clean. Prod.* **2017**, *141*, 900–908. [CrossRef]
37. Vendramin, L.; Jaskowiak, P.A.; Campello, R.J.G.B. On the combination of relative clustering validity criteria. In Proceedings of the 25th International Conference on Scientific and Statistical Database Management (SSDBM), Baltimore, MA, USA, 29–31 July 2013; pp. 733–744. [CrossRef]

applied sciences

MDPI

Article

Intelligent Reasoning Rules for Home Energy Management (IRRHEM): Algeria Case Study

Djamel Saba [1,*], Omar Cheikhrouhou [2,*], Wajdi Alhakami [3], Youcef Sahli [1], Abdelkader Hadidi [1] and Habib Hamam [4,5,6]

[1] Unité de Recherche en Energies Renouvelables en Milieu Saharien, URERMS, Centre de Développement des Energies Renouvelables, CDER, Adrar 01000, Alegria; sahli.sofc@gmail.com (Y.S.); hadidiabdelkader@gmail.com (A.H.)

[2] CES Laboratory, National School of Engineers of Sfax, University of Sfax, Sfax 3038, Tunisia

[3] Department of Information Technology, College of Computers and Information Technology, Taif University, P.O. Box 11099, Taif 21944, Saudi Arabia; whakami@tu.edu.sa

[4] Faculty of Engineering, Université de Moncton, Moncton, NB E1A 3E9, Canada; habib.hamam@umoncton.ca

[5] Department of Electrical and Electronic Engineering Science, School of Electrical Engineering, University of Johannesburg, Johannesburg 2006, South Africa

[6] International Institute of Technology and Management, Commune d'Akanda, Libreville P.O. Box 1989, Gabon

* Correspondence: saba_djamel@yahoo.fr (D.S.); omar.cheikhrouhou@isetsf.rnu.tn (O.C.)

Abstract: Algeria is characterized by extreme cold in winter and high heat and humidity in summer. This leads to an increase in the use of electrical appliances, which has a negative impact on electrical energy consumption and its high costs, especially with the high price of electricity in Algeria. In this context, artificial intelligence can help to regulate the daily consumption of electricity, by optimizing the exploitation of natural resources and alerting the individual to avoid energy wasting. This paper proposes a decision-making tool (IRRHEM) for managing electrical energy at smart home. The IRRHEM solution is based on three elements: the use of natural resources, the notification of the inhabitants in case of resources misuse or wasting behavior, and the aggregation of similar activities at same time. Additionally, based on the proposed intelligent reasoning rules, residents' behavior and activities are represented by OWL (Ontology Web Language) and written and executed through SWRL (Semantic Web Rule Language). Finally, the (IRRHEM) solution is tested in a home located in Algiers city inhabited by a family of four persons. The IRRHEM performance evaluation results are very promising and show a 3.60% rate of energy saving.

Keywords: decision-making tool; intelligent reasoning rules; energy saving; energy domain ontology; smart home; protégé software; ontology web language; semantic web rule language

Citation: Saba, D.; Cheikhrouhou, O.; Alhakami, W.; Sahli, Y.; Hadidi, A.; Hamam, H. Intelligent Reasoning Rules for Home Energy Management (IRRHEM): Algeria Case Study. *Appl. Sci.* **2022**, *12*, 1861. https://doi.org/10.3390/app12041861

Academic Editors: Luis Hernández-Callejo, Sara Gallardo Saavedra and Sergio Nesmachnow

Received: 24 October 2021
Accepted: 1 February 2022
Published: 11 February 2022

1. Introduction

A smart home allows its residents to control and manage the different home appliances through the Internet [1]. The first developments in home automation appeared in the 1980s thanks to the reduction of electronic and computer systems [2]. Therefore, the industry has focused its experiments on the development of controllers, interfaces, and tools providing comfort, security, and assistance within a building. Additionally, smart home management systems use several technologies, such as the IoT, cloud computing, Internet, GSM, and GPRS [3,4]. With a smart home, the indoor and outdoor surroundings of the dwelling can be monitored remotely. Recently, thanks to smartphones and the development of new technologies (for instance, apps and connected devices), the installation of smart homes becomes easier as all electrical devices are connected through the Internet.

The term smart energy management has spread in recent years and it is associated with several aspects of life, such as heating, cooling, and lighting systems [5]. This type of management aims to save daily energy consumption through the use of AIT, such as

MAS, and knowledge representation techniques, such as ontology [6–8]. Generally "energy efficiency" is achieved by putting devices into sleep mode or by activating them only when necessary. Electricity bills can be reduced to a much lower level, such as turning off lights when a person can leave the room or adjusting a temperature according to a person's identity or preference. In addition, it becomes possible to track the amount of energy consumed by various appliances at home and obtain forecasts for the future [9]. However, the issue of electric energy has become a priority in most countries due to the increasing need for energy in daily life, as a lot of research has been conducted to find suitable energy management solutions [10]. Some of them have worked to find solutions for energy sources that depend on clean energy sources, such as solar and wind energy [11]. Other research works have taken an interest in moving electricity from the source to the place of storage or consumption, seeking to find the shortest and least wasted electricity distance [12]. Djamel Saba et al. [13] focused on the consumption process by explaining to consumers how to use home appliances, as well as relying on smart solutions for energy consumption.

From an architectural point of view, there are many methods by which we can save more energy at home, whether by insulating the walls and floors, as this process can reduce between 20% and 25% of the heat loss at home [14]. The second method is to use double-glazed windows, as the windows are a major source of heat loss and savings. The third method relates to the use of a shared solar system for heating the water and the house.

In this paper, we focus on the development of an intelligent energy management solution applied to the smart home. This latter is an open and complex system, it includes some geographically distributed elements. In addition, the proposed solution is based on three elements: the first concerns the exploitation of natural resources, the second concerns the correction of occupants' errors and notification of occupants, and the third item concerns the grouping of similar activities at the same period. More precisely, the main contribution of this paper can be summarized in developing a smart solution to choose the most efficient energy sources as well as the best optimization technique that allows obtaining the best configuration of the hybrid energy system.

The remainder of this document includes the energy-saving elements at home in Section 2. Section 3 is reserved to present an IRRHEM design and development. The case study and its simulation are presented in Section 4, followed by the analysis and discussion in Section 5, and, finally, we conclude the paper.

2. IRRHEM Design and Development

In this section, we present the design, the development, and the scenarios on the simulated environment of our IRRHEM proposed method. We begin with a presentation of the used method, followed by a study of the work environment. The results of this study will be used to develop the main elements of ontology (for instance, concepts, relations, and rules of reasoning). The following steps are reserved for editing the ontological data, testing, and validating the ontology. This section ends with the presentation of the scenarios of the simulation of the solution and the display of the results.

2.1. Method Principle

Using the aforementioned elements, such as the building materials or specific electrical devices, energy savings can be obtained, but they remain insufficient, whether due to the random operation of the devices or to errors made by residents. Therefore, it is very important to include automatic or smart solutions to achieve more energy savings. In this context, this research work proposes an intelligent solution based on the ontology of energy and intelligent reasoning rules. Through the interaction between intelligent reasoning rules and the use of information stored in the knowledge base in the form of concepts (objects). It is possible to regulate the use of household appliances, as well as to make the best use of natural resources. Finally, the home occupants can be alerted to organize simultaneously similar activities. To achieve the main objective, which is the ideal saving of electrical energy consumption, a method has been developed, which includes many steps (Figure 1):

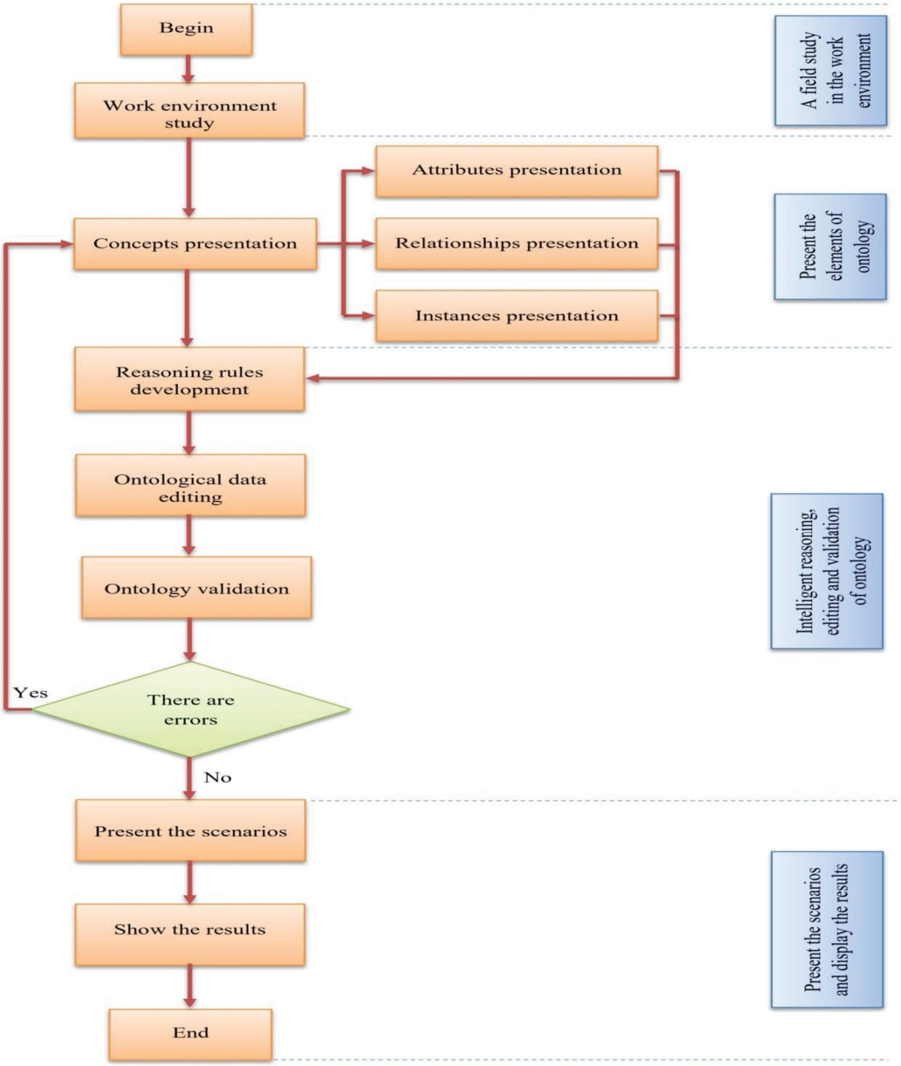

Begin
Work environment study
Concepts presentation
Attributes presentation
Relationships presentation
Instances presentation
Reasoning rules development
Ontological data editing
Ontology validation
There are errors
Yes
No
Present the scenarios
Show the results
End
A field study in the work environment
Present the elements of ontology
Intelligent reasoning, editing and validation of ontology
Present the scenarios and display the results

Figure 1. The proposed IRRHEM method.

2.2. Work Environment Study

In this study, the internal environment of the home is affected by many negative aspects that have a direct relationship with the residents, in particular:

χ Random setting of the electric air conditioner;
χ Leaving the water heater on continuously without setting the timer at night to use it in the morning;
χ Not turning off lights when they are not needed, and not using natural light instead of electricity (at specific times of the day);
χ Not unplugging electrical devices when not in use;
χ Not exploiting renewable energy resources;
χ Not using the refrigerator efficiently to avoid wasting electricity, for example, by avoiding continuously opening and closing it so that cold air does not escape;

χ Not using electric ovens correctly, for example, opening the oven door several times to check the cooking level. It should also be noted that not using utensils that are the same size as the stove, results in heat loss. Finally, a common mistake is not turning off the stove minutes before the end of cooking, although the residual heat after turning it off may be sufficient to complete the food cooking;

χ Putting hot food in the refrigerator causes an increase in energy consumption;

χ Keeping a lot of leftover food in the refrigerator may lead to an increase in power consumption;

χ Cooking food without a "pot" lid, which increases the cooking time and therefore increases the amount of electrical energy consumed;

χ Using the microwave to boil water instead of an electric kettle, although the first appliance consumes a greater amount of electrical energy compared to the second appliance;

χ Using the electric cooker instead of the electric kettle because the cooker needs time to heat up also the cooker stays hot for up to twenty minutes even after the task is completed and it is a waste of energy;

χ Failure to use modern electrical devices that do not require a lot of electrical energy during their operation, as well as not consulting the technical side when purchasing electrical devices (for example, not reading the label of the device, which explains its classification in terms of consumption and energy efficiency);

χ Frequent use of the hot water cycle and not choosing the cold water cycle over hot water (for example, uncertainty as to operating dishwashers at full capacity);

χ Not changing the air conditioning filters, because the accumulation of dust in the filters causes the air conditioner to work twice as hard to push in cold air, and therefore the consumption of a greater quantity of electric energy;

χ Not using LED bulbs that consume less energy;

χ Operating a clothes dryer instead of drying clothes with natural air and sun, which leads to the additional consumption of electrical energy;

χ Opening the curtains when the air conditioner is on, causing heat to spread in the room, resulting in an increase in work for the air conditioner and therefore higher energy consumption;

χ Increasing the temperature of the washing machine, although heating the water does not significantly affect the cleanliness of the clothes;

χ Failure to reduce the brightness of the TV screen, which may cause the TV to consume more electricity;

χ Not using thermal insulation in buildings;

χ Failure to group similar activities at the same time, which causes additional uses of electrical devices and thus the additional consumption of electrical energy;

χ Not using modern technologies to run the house and energy in particular, such as artificial intelligence techniques, knowledge capitalization tools, and rules-based reasoning tools linked to resident behavior.

After analyzing and studying previous work, the following propositions can be suggested:

✓ Air conditioning accounts for about half of the electrical energy consumption in homes. It should be remembered that 24 degrees Celsius is the perfect temperature to balance comfort and energy saving at home [15]. In addition, it is necessary to ensure that the air conditioning system is working efficiently;

✓ Instead of letting the water heater run constantly, it is better to turn it on only half an hour before showering, or set the automatic timer switch to heat the water at night, so that the water is ready in the morning [16];

✓ Individuals should have an awareness of turning off the lights at home when they do not need them. It is also better to replace traditional lamps with modern ones; of course, it is preferable to use natural light rather than bulbs [17];

- ✓ Freezers and refrigerators must be used efficiently to avoid wasting electricity by avoiding the constant opening and closing of the refrigerator door so that the cold air does not escape;
- ✓ The process of opening the oven door several times to check the cooking stage of the food results in heat escaping from inside the oven. When cooking on the stovetop, it is recommended to use pots of the same size as the stovetop to retain heat. It is also a good idea to reduce the heat so that the food is cooked evenly on all sides and the utensils do not burn. Finally, the stove can be turned off a few minutes before the food is fully cooked, as the heat will continue to cook the food for a while after being turned off;
- ✓ It is important to make sure to buy electrical appliances with low power consumption, and they usually carry the consumption class "A" or "A+", because these appliances are energy efficient [18];
- ✓ Dishwashers use the most energy in a home, so it is important to make sure they are at a full capacity before operating them. It is recommended to not use the hot water cycle and to opt for the cold water cycle. Additionally, it is recommended to use the eco cycle option if the devices are equipped with this function;
- ✓ The use of thermal insulation in buildings, where studies have shown in this regard that the use of insulation in walls and balconies can save up to 60% of electricity, in particular in very hot regions;
- ✓ Use of high-efficiency air conditioning systems [19];
- ✓ Avoid continuously opening the refrigerator or freezer during the day;
- ✓ Closing the blinds when the air conditioner is operating state;
- ✓ Avoid opening the lid of the "pot" while cooking food;
- ✓ Disconnecting electrical appliances when not in use;
- ✓ It is better to use the electric kettle instead of the microwave;
- ✓ Avoid leaving the oven door open, because leaving the oven door open during cooking leads to 50% more loss of electricity cooking;
- ✓ It is not necessary to increase the temperature of the washing machine since heating the water practically does not affect the cleanliness of the clothes. It is best if the temperature is between 30 and 40 degrees Celsius, which saves a lot of energy that the washer uses to heat the water [20];
- ✓ It is important to reduce the brightness of the TV screen, where it is possible to reduce the brightness of the image without affecting its quality. In addition, the bigger the TV screen, the more electricity it consumes, and the lower the picture brightness, the less the energy consumption;
- ✓ Take care to cook with the lid on the pot, as this helps reduce energy consumption. In addition, the size of the utensils used should also be suitable for the size of the meal being cooked;
- ✓ Making sure that the cooling degree of the food in the refrigerator is not more than 7 degrees Celsius and the cooling temperature in the freezer is not more than 18 degrees Celsius [21]. It is also recommended to place these devices in a cool place and avoid placing them next to the oven, dishwasher, or heating devices.

All these proposals are really interesting to achieve energy savings without neglecting the comfort of the inhabitants. However, there are other issues related to the characteristics of forgetfulness and neglect that characterize the individual, which make the implementation of the above suggestions really difficult. On this basis, a suggestion was presented to develop automatic (semi-automatic) and intelligent solutions, and conducted to ensure the completion of the previous proposals. The proposed work is an intelligent solution based on the ontology of energy management. This choice is justifiable due to the nature of the system that is open, complex, and distributed. Then, the system concerned by this study is characterized by a significant informational volume that requires flexibility in the presentation of knowledge of the ontology web language (OWL), which will be exploited to create intelligent reasoning using the semantic web rules language (SWRL). The ontology carried out is applied to a house located in the city of Algiers. Finally, following a comparison between two scenarios of energy consumption, the solution shows their importance from the efficient energy saving viewpoint.

2.3. IRRHEM Architecture

IRRHEM offers an intelligent techniques energy management system based on an Intelligent Context-Awareness Management (Intelligent-CAM). Intelligent-CAM uses contextual awareness ontology and intelligent reasoning to provide contextual information to simulate the behaviors of a population and its surroundings.

From an architectural point of view, the architecture for the IRRHEM system is proposed (Figure 2). IRRHEM is capable of better controlling the home environment (occupants, equipment, and environment), all this is to ensure the main objective that concerns energy optimization. Then, to obtain correct information and make appropriate decisions about current situations, the Intelligent-CAM detects various situations related to the home and its surroundings to provide full Context-Awareness to obtain the minimum number of personnel interventions, as well as avoiding the wastage of energy. Finally, the structure of the proposed solution is formed into seven models, namely:

2.3.1. Data Aggregate Model

This unit receives data from devices and sensors, such as temperature sensors and lights. Moreover, it is responsible for modifying the environmental conditions by sending a command to the operator's infrastructures, such as the switch or any existing services. This unit then sends the data collected from the smart building to the Intelligent-CAM model for processing.

2.3.2. Intelligent CAM Model

This module provides Context-Awareness about what is happening in the smart home. This Context-Awareness is useful for energy efficiency inference techniques to provide the best services at the right time and place. Intelligent-CAM is responsible for managing the Context lifecycle in the IRRHEM system. The context life cycle consists of five phases. The first is "Context acquisition", where contexts need to be obtained from different sources, which may be physical (e.g., physical sensors) or virtual (e.g., virtual sensors and software). The second is "Context-Awareness Constructor", where the collected data must be modeled and represented in a meaningful way. In this section, the ontology technique is used. The third is the "Temporal Context", in which the typical contexts are preprocessed. This step improves the thinking stage and leads to a better result. The fourth is "Context reasoning", where the modeled data needs to be processed to extract new information and generate high-level context information from low-level contextual data. Finally, the "Context dissemination" distributes high- and low-level content text to interested consumers (such as, energy-saving logical engines, user interfaces, or any external services).

2.3.3. Energy Reasoning Engine Model

This is the part of the system that generates conclusions and decisions from the available knowledge about the smart building, and plays an important role in implementing the proposed solution by discovering the causes and contexts of energy waste using a set of smart reasoning rules (presented in the next sections). The first step for this unit is to arrange the positions in descending order. The benefit of this arrangement lies in dealing with the most important and most wasted states of energy. It also provides measures to save energy and eliminate energy waste by using SWRL rules, where those rules are represented as conditional logic. Rulesets can also be managed and applied separately to other functions, and each parent clause association rule can be linked to a list of executable actions.

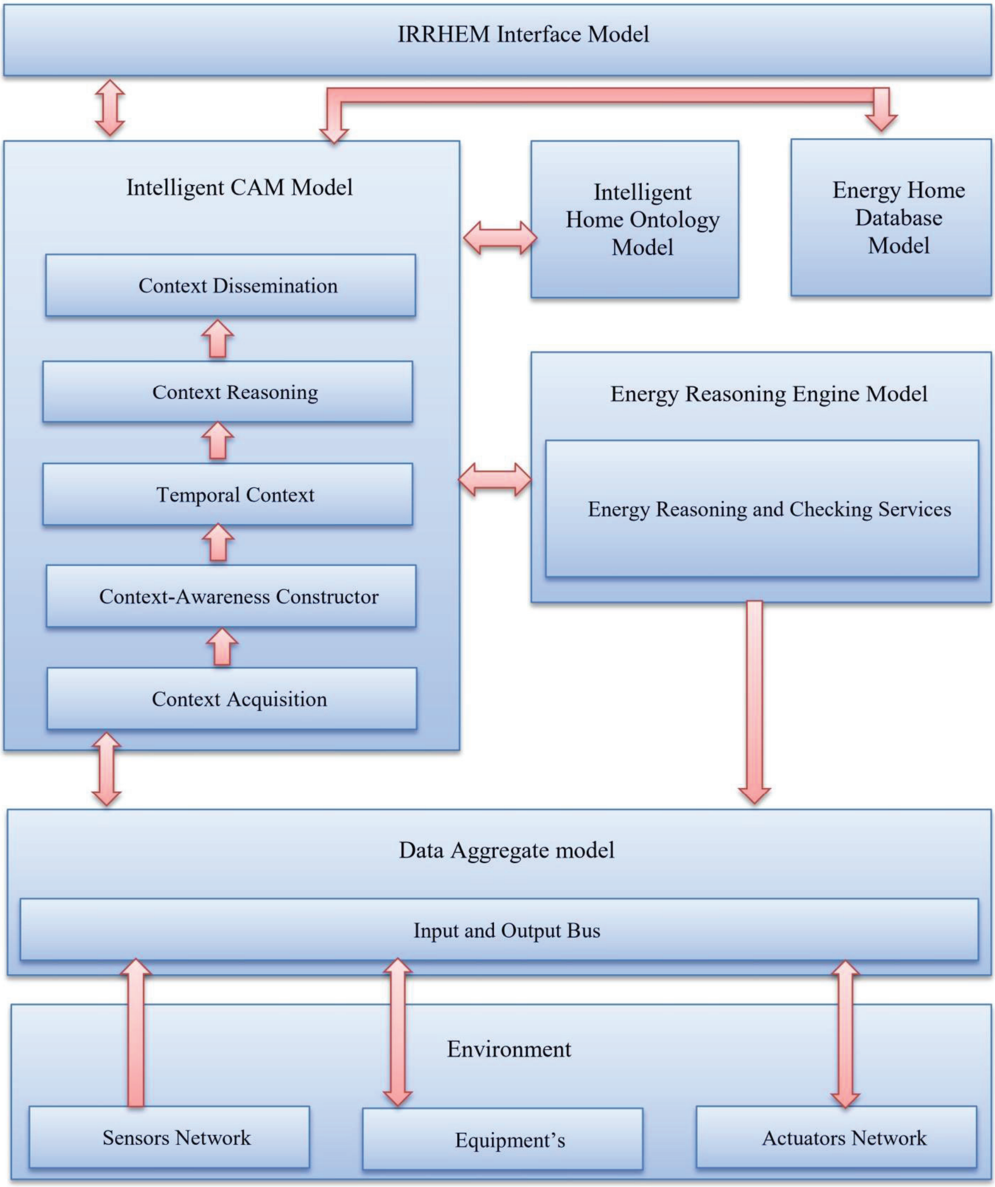


Figure 2. Architecture of IRRHEM.

2.3.4. Intelligent home Ontology Model

Ontology is one of the best tools for representing the field of knowledge, particularly in the management of energy in homes. Many works have been interested in this field, the most important of which are the works of Degha et al., most of which, in their entirety, suggest a structural framework for organizing smart building data [22]. It includes machine-interpretable definitions of the basic concepts of the smart building field and the relationships between them. These works include an important number of concepts, namely human, environment, services, devices, places and Context-Awareness. The ontology-based formal context model can play a vital role in facilitating reasoning by representing the knowledge of the home energy domain. The Semantic Web Rules Language (SWRL) is used, where the rules are applied for different purposes; in addition, the Web Ontology Language (OWL) is used to represent concepts, properties, and relationships. The names of concepts and relationships taken from the ontology (described in detail in the Sections 2.4–2.6) are represented.

2.3.5. Energy Home Database Model

A database is a systematic collection of data to support electronic storage and data processing, and to make data management easier. In this paper, a central database was used to store information about the building, such as hardware statistics, climate information, or any events that occur inside the building. It provides different ways to access the data and history, and it is used to automatically provide services if the same situation is repeated.

2.3.6. IRRHEM Interface Model:

A user interface module is used to interact with users and exchange information. The goal of the user interface design is the ease of use in operating a device or software to achieve the desired result. This generally means that the operator must provide minimum inputs to achieve the desired output in the form of notifications and feedback, and this interface can be a web-based application or the smartphone application.

2.3.7. Home Environment Model

The home environment is a space consisting of a group of objects and programs that interact with each other (for instance, occupants, sensors, actuators, and appliances). The creation of this environment and the selection of its components are carried out according to systematic studies to achieve certain goals, such as saving energy. In general, the elements of this environment belong to the passive category related to constructions and the active category, which pertain to the programs and solutions that apply to the first category of objects.

2.4. IRRHEM Concepts Presentation

An ontology is a formal and explicit representation of knowledge in a given domain (e.g., the home environment), with the aim of efficiently using the data and ensuring flexibility in the sharing and updating of ontologies [23,24]. However, a concept (classes) is the basic element for an ontology; it represents a hardware or software object of the studied environment.

2.4.1. "Call" Concept

There are many similar individuals' activities that can be grouped in a single time (for example, eating meals), so that there is the joint use of devices that consume electricity, and thus benefit from an important energy saving method. It is only a reminder that the process of grouping activities can be carried out without neglecting the comfort of the residents. To ensure the latter, it is necessary to first collect the activities in a unified and short time, and if this process causes harm to an individual, the time is expanded to be slightly longer and therefore electricity consumption can also be obtained, but in a smaller amount than in the first case (Figure 3).

2.4.2. "Sensor" Concept

The "sensor" concept is an essential element of the IRRHEM ontology; it detects the information coming from the physical environment and reacts to it. The information captured can be light, heat, movement, humidity, pressure, or other environmental phenomena (Figure 4). The output is typically a signal that is converted to an operator-readable display at the location of the sensor, or transmitted electronically over a network for reading or processing.

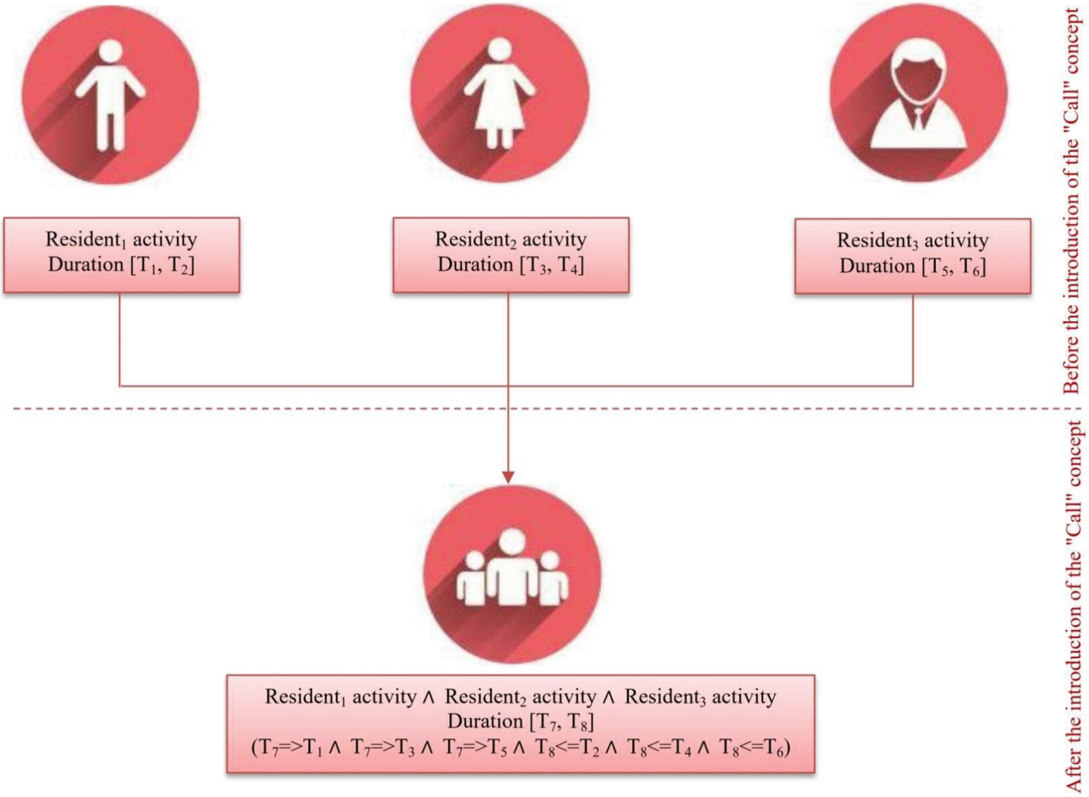

Resident$_1$ activity
Duration [T_1, T_2]
Resident$_2$ activity
Duration [T_3, T_4]
Resident$_3$ activity
Duration [T_5, T_6]
Resident$_1$ activity ∧ Resident$_2$ activity ∧ Resident$_3$ activity
Duration [T_7, T_8]
(T_7=>T_1 ∧ T_7=>T_3 ∧ T_7=>T_5 ∧ T_8<=T_2 ∧ T_8<=T_4 ∧ T_8<=T_6)
Before the introduction of the "Call" concept
After the introduction of the "Call" concept

Figure 3. Impact of the introduction of the "Call" concept in IRRHEM.

2.4.3. Other Concepts

The ontology model includes any hierarchy of concepts rooted in the "Home" concept, which represents the residential environment. The model also includes disjoint secondary hierarchies describing the categorizations of subdomain objects of the application domain (such as, sensor, equipment, and activities) (Table 1).

Table 1. IRRHEM concepts.

Concept	Description
Home	Represents the place of residence
Resident	Represents the home resident
TaskResident	Represents the resident task
EquipmentHome	Represents the electrical equipment
ActionEquipment	Represents an action performed by the equipment
BehaviorResident	Represents the resident behavior
WeatherHomeEnvironment	Represents the climate that characterizes a home environment

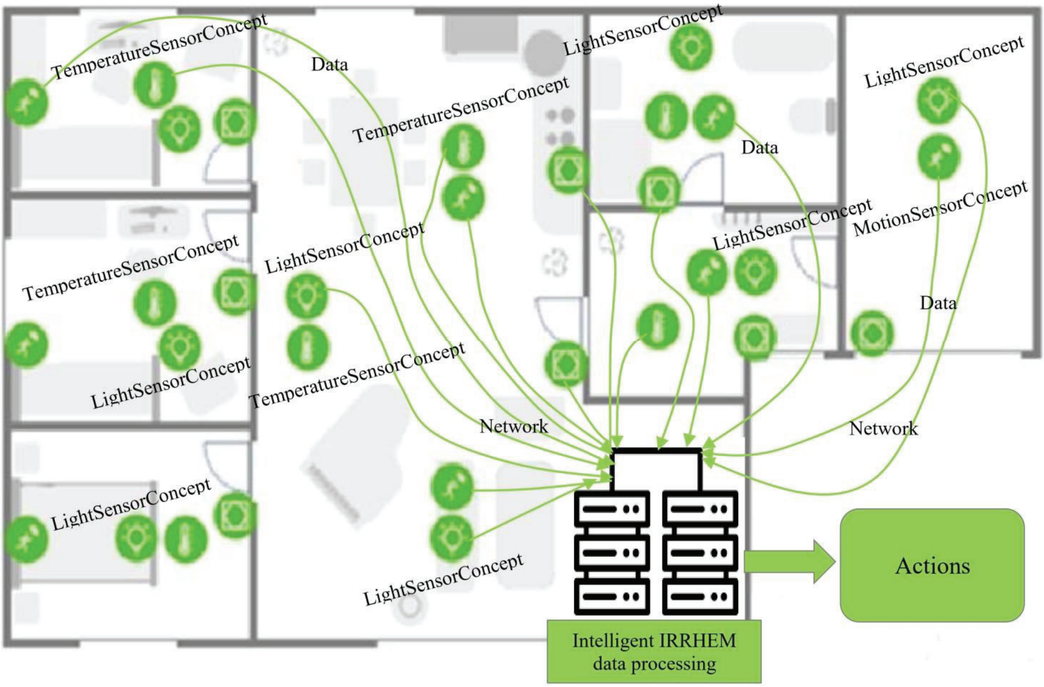


Figure 4. "Sensor" concept architecture in IRRHEM.

2.5. Attributes for IRRHEM Concepts

The attributes or properties are the functionalities, characteristics, or settings that objects can own and share [25]. Each IRRHEM concept includes a set of attributes (Table 2).

Table 2. Some examples of the attributes for IRRHEM concepts.

Attribute	Description	Concept
ResidentName	Resident name	Resident
ResidentProfile	Resident profile	Resident
EquipmentAction	Actions of equipment	EquipmentHome
EquipmentPlace	Place of equipment	EquipmentHome

2.6. IRRHEM Relations

Relationships are the links that objects can have with each other [6]. An ontological relation (hierarchical or descriptive) is a relation linking ontological concepts, constructed from termino-ontological relations, and described in a formal language. A hierarchical relationship expresses an inheritance of the properties of the concept (Table 3).

Table 3. Some examples for IRRHEM relationships.

Relation	Description	Related Concepts
ResidenttHasTask	Everyone at home performs activities	Resident, TaskResident
ResidentHasProfil	Home resident possesses a profile	Resident, ProfileResident
EquipmentHasAction	Equipment can perform actions	EquipmentHome, ActionEquipment
EquipmentHasPlace	Equipment is located in place the of the home	EquipmentHome, PlaceEquipment

2.7. IRRHEM Instances

The use of ontology will depend on the ability to reason about individuals [26]. To achieve this in a beneficial manner, it is necessary to have a mechanism for describing the classes to which individuals belong and the properties they inherit due to their membership in these classes. We can always assert specific properties about individuals, but the great strength of ontologies lies in class-based reasoning. Finally, the extension of the class is the set of individual members of a class (Table 4).

Table 4. Some examples for IRRHEM individuals (instances).

Individual	Concept
TV	EquipmentHome
Refrigerator	EquipmentHome
Living room	Place
Breakfast	TaskResident

2.8. Intelligent Reasoning of IRRHEM

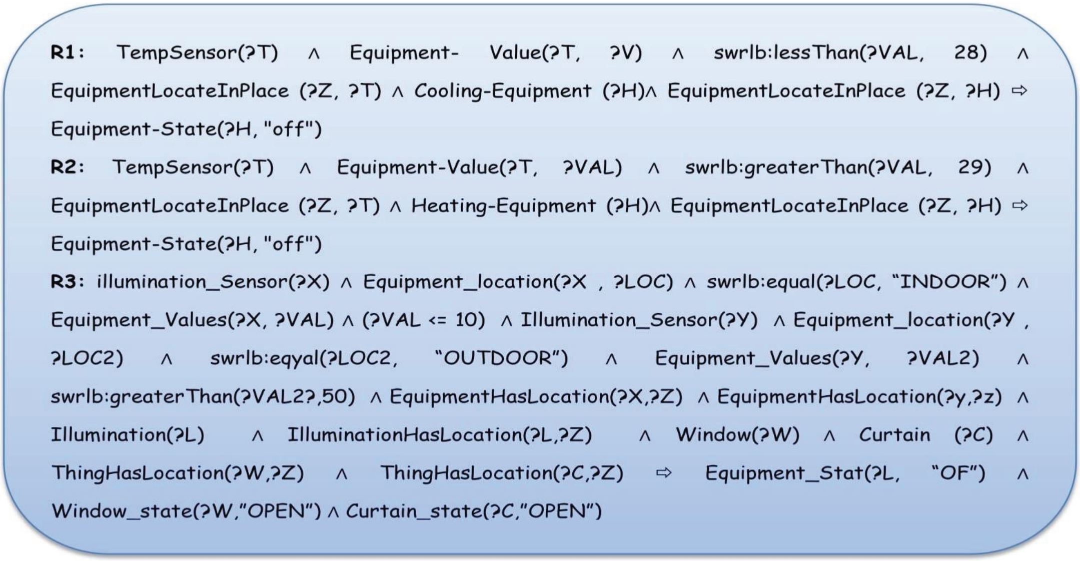
Regardless of the tools, their formalisms, or methods of inference, knowledge-based systems, such as expert systems, are based on a clear separation between knowledge and methods of inference [27]. Thus knowledge, instead of being nested in the structure of the program (such as, the "if . . . then . . . else . . . " commands of a programming language), is considered as interchangeable data, susceptible to change [28]. It can be corrected, updated, and exploited by programs (inference engines), which simulate reasoning mechanisms and provide the interface with users [29]. Then, in any knowledge-based system or expert system, the following components will be found at the base:

- The knowledge base is domain-specific and, unlike databases, is not limited to factual knowledge, or data [30]. It brings together all the types of knowledge relevant to the domain considered, namely, the description of objects and their relationships (for example, in the form of ontology), rules to be applied to make a diagnosis or solve a problem, and meta-knowledge making it possible to choose which rules apply.
- The system remains at all times, in its working memory, a base of known facts (or instances of ontology). Thanks to it, the inference engine can choose the elements of its knowledge base, for example, the rules to be used effeciently and according to the facts available, and adapted to the problem posed [31]. The fact base grows as the inference engine infers new facts by applying knowledge to the already known facts.
- The inference engine is the program that builds reasoning by drawing its materials from the knowledge base and the fact base. By examining the fact base, it detects interesting knowledge that can be applied to certain facts, connects them, and builds a resolution plan. It deduces new facts from those provided at the start or during the interaction by the user. Independent of the domain, the inference engine brings together the reasoning mechanisms that will exploit the knowledge base.
- To these three essential components are added the interface modules, which are also independent of the field of knowledge. Thanks to them, the expert can easily access the knowledge base, modify it by correcting unnecessary or erroneous information, or by adding precision. The user can follow the reasoning of the system in a language that is natural to him, ask questions, and ask for explanations, without having to acquire in-depth knowledge of expert systems or computing [32].

IRRHEM is based on intelligent reasoning rules formulated by predicate logic and facts, whereby all the rules are loaded for the accomplishment of the tasks (Figures 5–7).

- Rule 1 (R1) is concerned with adjusting the temperature in a specific place at home according to the wishes and the desires of residents. The use of this rule allows the cooling to stop when the temperature becomes less than 28 degrees Celsius.

- Rule 2 (R2) is concerned with adjusting the temperature in a specific place at home according to the residents' desire. The use of this rule allows for the heating system to be stopped when the temperature becomes greater than 29 degrees Celsius.
- Rule 3 (R3) is concerned with adjusting the lighting of a specific place inside the home to benefit from the external lighting, and thus turning off the lamps lead to the gain of a large amount of energy. However, by measuring the internal lighting, estimated at (10), and the external lighting, rated at (50), where we notice strong lighting outside, can be used by opening the windows and curtains.
- For rule 4 (R4), in many cases, the light bulbs are left on despite the absence of people from the home. This rule turns off all the light bulbs, except for the refrigerator, because it contains food items, and a home security camera.
- Rule 5 (R5) is concerned with correcting some negative traits of individuals, such as forgetfulness or neglect. For example, in many cases, electrical appliances are left to operate despite the absence of individuals from the home. In this instance, when people are absent from the home, this rule turns off all the electrical appliances, except for the refrigerator, because it contains food items, and a home protection camera.
- Rule 6 (R6) allows the temperature of a particular place in the house to be adapted according to the wishes of the inhabitant.
- Rule 7 (R7) is related to the process of grouping similar activities of the individuals. In this example, the activity of the individual that takes the least time (50 min) is chosen.



R1: TempSensor(?T) ∧ Equipment- Value(?T, ?V) ∧ swrlb:lessThan(?VAL, 28) ∧ EquipmentLocateInPlace (?Z, ?T) ∧ Cooling-Equipment (?H)∧ EquipmentLocateInPlace (?Z, ?H) ⇨ Equipment-State(?H, "off")

R2: TempSensor(?T) ∧ Equipment-Value(?T, ?VAL) ∧ swrlb:greaterThan(?VAL, 29) ∧ EquipmentLocateInPlace (?Z, ?T) ∧ Heating-Equipment (?H)∧ EquipmentLocateInPlace (?Z, ?H) ⇨ Equipment-State(?H, "off")

R3: illumination_Sensor(?X) ∧ Equipment_location(?X , ?LOC) ∧ swrlb:equal(?LOC, "INDOOR") ∧ Equipment_Values(?X, ?VAL) ∧ (?VAL <= 10) ∧ Illumination_Sensor(?Y) ∧ Equipment_location(?Y , ?LOC2) ∧ swrlb:eqyal(?LOC2, "OUTDOOR") ∧ Equipment_Values(?Y, ?VAL2) ∧ swrlb:greaterThan(?VAL2?,50) ∧ EquipmentHasLocation(?X,?Z) ∧ EquipmentHasLocation(?y,?z) ∧ Illumination(?L) ∧ IlluminationHasLocation(?L,?Z) ∧ Window(?W) ∧ Curtain (?C) ∧ ThingHasLocation(?W,?Z) ∧ ThingHasLocation(?C,?Z) ⇨ Equipment_Stat(?L, "OF") ∧ Window_state(?W,"OPEN") ∧ Curtain_state(?C,"OPEN")

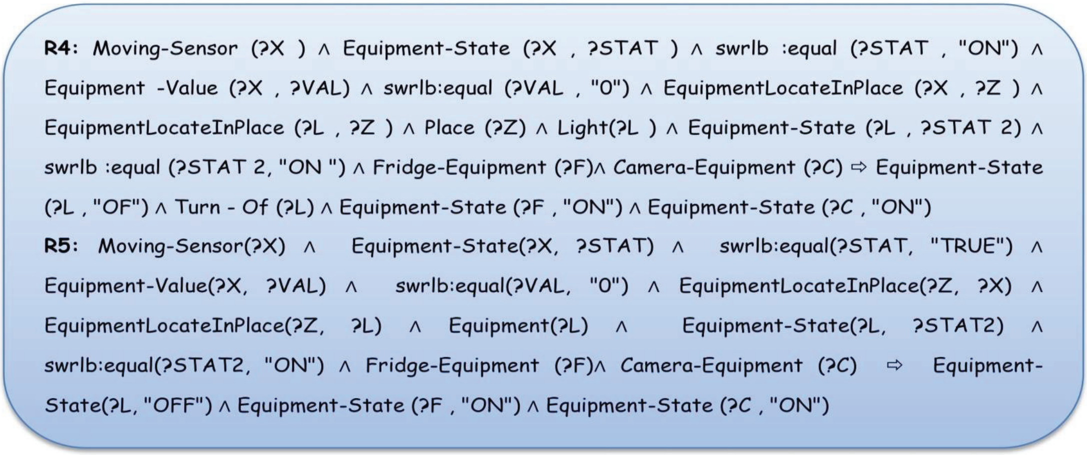
Figure 5. Rules 1–3.

2.9. Ontological Data Editing

In computer science, ontology is a technical term that denotes an artefact designed to be able to model the knowledge of a real or imaginary domain [33]. One of the first objectives of developing ontology is to share the same understanding of the structure of information between people. There are different reasons why it is necessary to develop ontology [34]:

- Allow the reuse of knowledge in a field;
- Make explicit the hypotheses of a domain;
- Separate domain knowledge from operational knowledge;
- Analyze the knowledge of a domain;

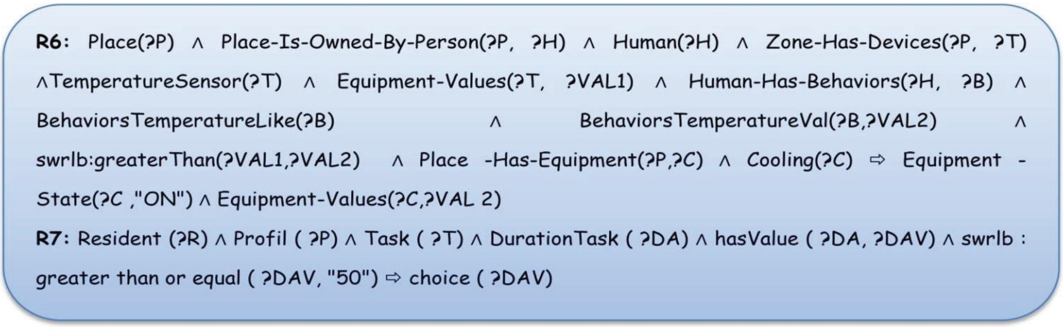
- Facilitate the interoperability between two systems;
- Ensure the reliability of knowledge;
- Facilitate the communication between users.



R4: Moving-Sensor (?X) ∧ Equipment-State (?X , ?STAT) ∧ swrlb :equal (?STAT , "ON") ∧ Equipment -Value (?X , ?VAL) ∧ swrlb:equal (?VAL , "0") ∧ EquipmentLocateInPlace (?X , ?Z) ∧ EquipmentLocateInPlace (?L , ?Z) ∧ Place (?Z) ∧ Light(?L) ∧ Equipment-State (?L , ?STAT 2) ∧ swrlb :equal (?STAT 2, "ON ") ∧ Fridge-Equipment (?F)∧ Camera-Equipment (?C) ⇨ Equipment-State (?L , "OF") ∧ Turn - Of (?L) ∧ Equipment-State (?F , "ON") ∧ Equipment-State (?C , "ON")

R5: Moving-Sensor(?X) ∧ Equipment-State(?X, ?STAT) ∧ swrlb:equal(?STAT, "TRUE") ∧ Equipment-Value(?X, ?VAL) ∧ swrlb:equal(?VAL, "0") ∧ EquipmentLocateInPlace(?Z, ?X) ∧ EquipmentLocateInPlace(?Z, ?L) ∧ Equipment(?L) ∧ Equipment-State(?L, ?STAT2) ∧ swrlb:equal(?STAT2, "ON") ∧ Fridge-Equipment (?F)∧ Camera-Equipment (?C) ⇨ Equipment-State(?L, "OFF") ∧ Equipment-State (?F , "ON") ∧ Equipment-State (?C , "ON")

Figure 6. Rule 4 and Rule 5.



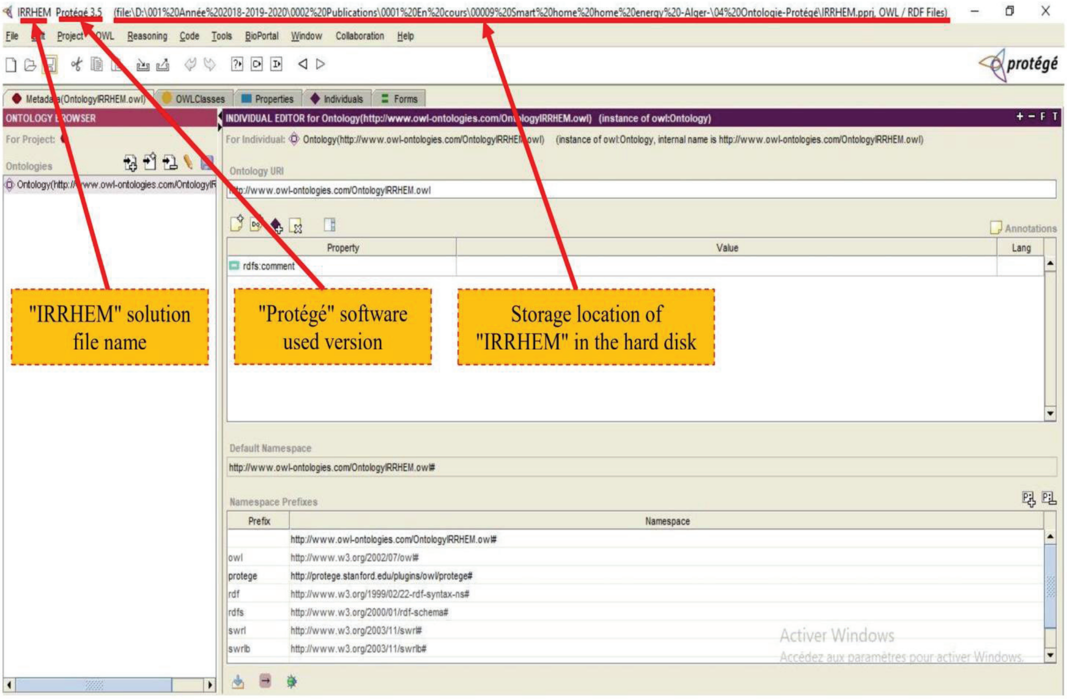
R6: Place(?P) ∧ Place-Is-Owned-By-Person(?P, ?H) ∧ Human(?H) ∧ Zone-Has-Devices(?P, ?T) ∧TemperatureSensor(?T) ∧ Equipment-Values(?T, ?VAL1) ∧ Human-Has-Behaviors(?H, ?B) ∧ BehaviorsTemperatureLike(?B) ∧ BehaviorsTemperatureVal(?B,?VAL2) ∧ swrlb:greaterThan(?VAL1,?VAL2) ∧ Place -Has-Equipment(?P,?C) ∧ Cooling(?C) ⇨ Equipment -State(?C ,"ON") ∧ Equipment-Values(?C,?VAL 2)

R7: Resident (?R) ∧ Profil (?P) ∧ Task (?T) ∧ DurationTask (?DA) ∧ hasValue (?DA, ?DAV) ∧ swrlb : greater than or equal (?DAV, "50") ⇨ choice (?DAV)

Figure 7. Rule 6 and Rule 7.

There are many free editors, such as Protégé 3.5, SWOOP, Ontolingua, KMgen, IsaViz, and DOE. In this category of publishers, the best known and the most used is Protégé. It is an open-source tool developed by Stanford University [35]. Since its first version, it has significantly evolved and continues to evolve rapidly (Figure 8).

The "Protégé" software is employed to edit and read the "IRRHEM" ontology, where all hierarchical categories are created for each concept with its properties and relationships. Intelligent thinking rules are also an important part of implementing the proposed solution, as additional components are used, such as SWRLTab for Protégé 3.5, which provides the SWRL and SQWRL rule execution environment [36]. The next step is to use the debugger in "Protégé" to check the correctness and consistency of the information entered in the previous stages. In addition to that, we relied on previous ontology solutions, through the process of importing them, which allowed us to save a lot of time and effort. Finally, the language "JAVA", can be used to develop IRRHEM modules.



IRRHEM Protégé 3.5 (file:\D:\001%20Année%202018-2019-2020\0002%20Publications\0001%20En%20cours\00009%20Smart%20home%20home%20energy%20-Alger-\04%20Ontologie-Protégé\IRRHEM.pprj, OWL / RDF Files)

Ontology(http://www.owl-ontologies.com/OntologyIR

INDIVIDUAL EDITOR for Ontology(http://www.owl-ontologies.com/OntologyIRRHEM.owl) (instance of owl:Ontology)

For Individual: Ontology(http://www.owl-ontologies.com/OntologyIRRHEM.owl) (instance of owl:Ontology, internal name is http://www.owl-ontologies.com/OntologyIRRHEM.owl)

Ontology URI: http://www.owl-ontologies.com/OntologyIRRHEM.owl

Default Namespace: http://www.owl-ontologies.com/OntologyIRRHEM.owl#

Namespace Prefixes

Prefix	Namespace
	http://www.owl-ontologies.com/OntologyIRRHEM.owl#
owl	http://www.w3.org/2002/07/owl#
protege	http://protege.stanford.edu/plugins/owl/protege#
rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#
rdfs	http://www.w3.org/2000/01/rdf-schema#
swrl	http://www.w3.org/2003/11/swrl#
swrlb	http://www.w3.org/2003/11/swrlb#

Activer Windows
Accédez aux paramètres pour activer Windows.

"IRRHEM" solution file name

"Protégé" software used version

Storage location of "IRRHEM" in the hard disk

Figure 8. Editor "Protégé" graphical interfaces.

In the first step, the concepts of IRRHEM are introduced. The following steps concern editing the relationships and instances. The final step concerns the editing of the intelligent reasoning rules.

3. Case Study and Simulation

It is important to present the elements that are directly related to the consumption of electrical energy in a home, such as climatic data, household appliances, and the activities of residents. Then, to know the importance of the proposed solution (IRRHEM), two energy consumption scenarios are presented (with and without the intervention of the proposed solution).

3.1. Presentation of the IRRHEM Environment

The energy consumption scenarios were carried out on a family house located in the city of Algiers in Algeria. Algiers is located in the north-center of the country and occupies an interesting geostrategic position, both from the point of view of economic flows and exchanges with the rest of the world, and from a geopolitical point of view. It extends over more than 809 km^2. The city of Algiers is bounded to the north by the Mediterranean Sea with a coastline of 80 km, to the south by the Blida city, and to the west and east by the cities of Tipaza and Boumerdes, respectively [37] (Figure 9). It is characterized by a latitude of 36°45′08″ N, a longitude of 3°02′31″ E, and an elevation above sea level of 186 m. The home in the study was inhabited by a family of four members (a father, mother, boy, and girl) (Table 5).

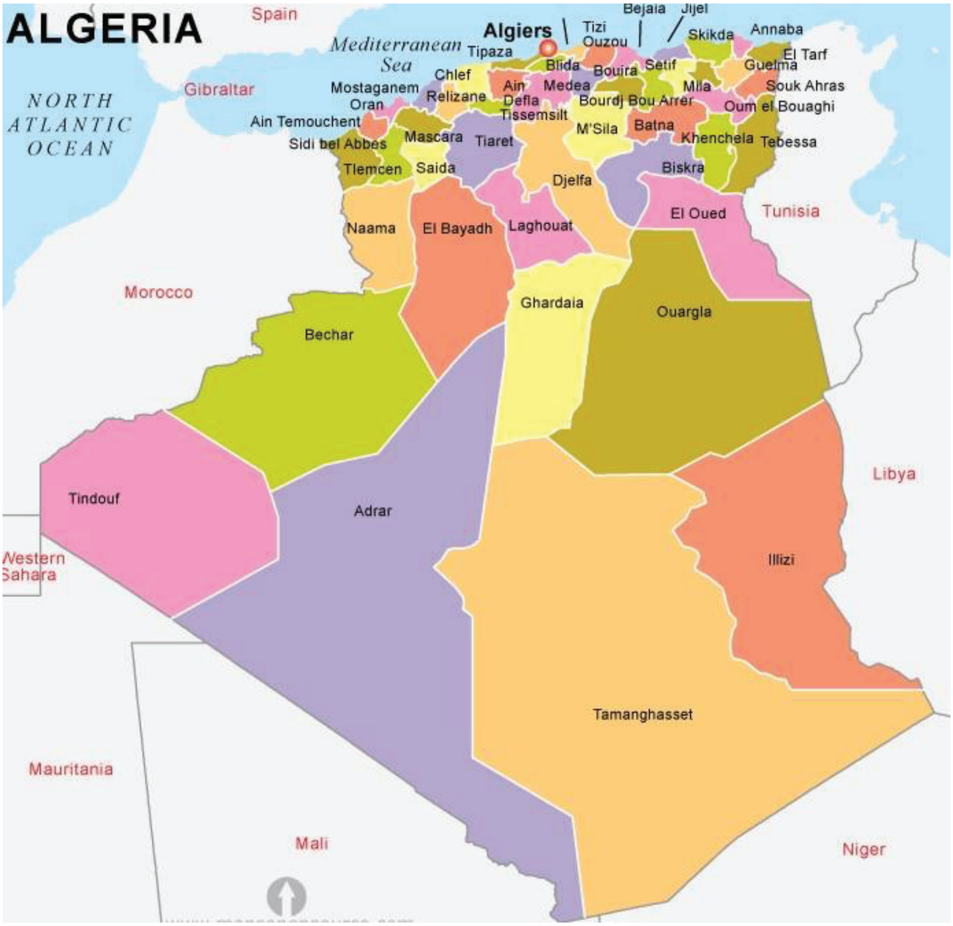


Figure 9. Geographic map of Algiers city [38].

The climate in Algiers is warm and temperate. In winter, the rains are much more important in the city of Algiers than they are in summer. The annual average temperature in the city of Algiers is 18.2 °C. The mean annual precipitation is 615 mm. The driest month is July, with an average of 1 mm of rain. In addition, the record precipitation level is recorded in November with an average of 94 mm of rain. The warmest month of the year is August with an average temperature of 26.7 °C. The coldest month of the year is January with an average temperature of 11.0 °C. The difference in precipitation between the driest and wettest months is 93 mm. A variation of 15.7 °C is recorded over the year [39].

The family home contains a set of appliances that depend on electrical energy to operate. Table 6 presents a number of electrical appliances, their electricity consumption characteristics, and their location.

3.2. IRRHEM Scenarios

To know the effectiveness of IRRHEM, two scenarios were proposed: the first without the intervention of IRRHEM (electricity consumption in a normal state) and the second deals with the electricity consumption with the intervention of IRRHEM. In addition, 1 May

2020, was selected to test IRRHEM. To measure the amount of energy consumed during that day, we divided that day into units of time of one hour.

Table 5. Family members concerned with the study.

Family Member	Age	Function	Preference	Event/Activity
Father	45	Employee	• 25 °C ≤ Temperature ≤ 28 °C • Illumination ≥ 20 Linux	• Sleeping • Waking up from sleep • Going to the bathroom • Going to the bathroom
Mother	40	Housewife	• 21 °C ≤ Temperature ≤ 26 °C Illumination ≥ 20 Linux	• Sleeping • Fajr prayer • Reading the Koran • House cleaning
Boy	16	High school student	• 27 °C ≤ Temperature ≤ 29 °C Illumination ≥ 15 Linux	• Sleeping • At school • Returning from school and going to the dining room
Girl	13	A student in middle school	• 27 °C ≤ Temperature: ≤ 30 °C Illumination ≥ 15 Linux	• Sleeping • At school • Going to the kitchen

Table 6. Electrical equipment.

Place	Electrical Equipment	Number (N)	Power (P_{ap})
Bathroom	Electric water boiler	1	1800
	Washing machine	1	240
	Economic lamp	2	25
Hall	Economic lamp	3	25
Living room	Air conditioner (8000 BTU)	1	900
	Economic lamp	3	25
	Home internet router	1	7
Kitchen	Refrigerator combi (250 L)	1	175
	Economic lamp	2	25
	Microwave	1	1125
Garage	Camera	1	20
	Electric car	1	3000
	Economic lamp	4	25
Room	Air conditioner (6000 BTU)	1	700
	Iron	1	800
	Economic lamp	2	25

3.2.1. First Scenario (Without IRRHEM Intervention)

The amount of energy consumed is directly related to the activities of family members, thanks to which it is possible to know the electrical devices used. What distinguishes the energy consumption in this scenario is the presence of many disadvantages, such as the non-exploitation of natural resources (e.g., sunlight) as well as the random exploitation of electrical devices (Tables 7 and 8). Additionally, to calculate the energy consumption (P_{ap} (W)) by family members over a period (T (hours)), there are mathematical formulas in the literature such as:

$$E_{ap}(Wh) = P_{ap}.T \tag{1}$$

It is also possible to calculate the energy consumed by a number of devices of the same type using the following rule:

$$E_{apt}(Wh) = E_{ap}.N \tag{2}$$

The total power is calculated by the following mathematical rule:

$$E_t(Wh) = \sum_{i=1}^{n} E_{apt}(i) \tag{3}$$

where "i" is the equipment type.

The average values of the climate data in the period from 00 h: 00 m: 00 s to 00 h: 59 m: 59 s are: horizontal radiation (W/m^2) 0, temperature 20 °C, and humidity 65% [40].

The average values of the climate data in the period from 11 h: 00 m: 00 s to 11 h: 59 m: 59 s are: horizontal radiation (W/m^2) 0, temperature 29 °C, and humidity 80% [40].

3.2.2. Second Scenario (With IRRHEM Intervention)

Table 9 presents the results related to the IRRHEM intervention. These results are based on the rules of intelligent thinking.

Table 7. Energy consumption scenario in the period: 01 h: 00 m: 00 s to 01 h: 59 m: 59 s.

Time (h, m, s)	Family Member	Place	Event/Activity	Action	Equipment in Operation	Number (N)	Operating Time (Minutes)	Power (W)	Energy Consumed by Event (W)	Energy Consumed per Hour (W)
00 h:00 m:00 s–00 h:59 m:59 s	Father Mother	Bedroom Bedroom	Sleeping Sleeping	Nothing Nothing	Air conditioner (6000 BTU)	1	60	700	700	2205
	Boy	Room 2	Sleeping	Nothing	Air conditioner (6000 BTU)	1	60	700	700	
	Girl	Room 3	Sleeping	Nothing	Air conditioner (6000 BTU)	1	60	700	700	
	/	/	/	/	Refrigerator combi (250 L)	1	60	170	170	
	/	/	/	/	Camera	1	60	20	20	
	/	/	/	/	Intercom	1	60	40	40	
	/	/	/	/	Light (outside)	2	60	25	50	

- In the periods from 00 h: 00 min: 00 s to 01 h: 00 min: 00 s and from 01 h: 00 min: 00 s to 02 h: 00 min: 00 s, there is no activity for all family members, and therefore it is impossibile to apply the rules of intelligent reasoning (because the execution of the solution rules is linked mainly to the activity of individuals and the mistakes they commit).
- In the periods from 05 h: 00 min: 00 s to 06 h: 00 min: 00 s and from 06 h: 00 min: 00 s to 07 h: 00 min: 00 s, energy savings of 1225 watts and 1341.67 watts, respectively, were observed through the intervention of rule "R1", which relates to switching off refrigeration equipment; rule "R4", which relates to switching off electrical equipment (except for the appliances that are necessary to operate constantly, such as refrigerators and security equipment); rule "R5", which relates to correcting the mistakes committed and forgetfulness that characterize the family members of the home in many cases; rule "R6", which relates to the process of air-conditioning or heating a room in the home according to personal choice; and rule "R7", which relates to the process grouping the similar activities of the family members.

- From 07 h: 00 min: 00 s to 08 h: 00 min: 00 s, there is an energy-saving amount of 1395 watts, through the intervention of rule "R1", the turning off of cooling equipment; rule "R3", the adjusting of indoor lighting according to the availability of light outside the home; rule "R4", the turning off of electrical equipment when a room is absent of family members; rule "R5", related to correcting errors and forgetfulness that characterizes the members of the home; rule "R6", hich relates to adjusting the air conditioning according to individual preference; and rule "R7", which relates to grouping the similar activities of the members of the home.
- In the periods from 09 h: 00 min: 00 s to 10 h: 00 min: 00 s and from 11 h: 00 min: 00 s to 12 h: 00 min: 00 s, energy savings of 16.6 watts and 108.33 watts, respectively, were observed through the intervention of rule "R3", related to adjusting the lighting of a place in the home by taking advantage of external light; rule "R4", related to turning off electrical equipment (except for some equipment); rule "R5", related to correcting some negative characteristics of the family members, such as forgetfulness; and rule "R7", regarding the grouping of similar activities of individuals.
- In the periods from 12 h: 00 min: 00 s to 13 h: 00 min: 00 s, 15 h: 00 min: 00 s to 16 h: 00 min: 00 s, 16 h: 00 min: 00 s to 17 h: 00 min: 00 s, 17 h: 00 min: 00 s to 18 h: 00 min: 00 s, and 18 h: 00 min: 00 s to 19 h: 00 min: 00 s, energy-saving figures of 8.33 watts, 54.17 watts, 66.17 watts, 8.33 watts, and 8.33 watts, respectively, are observed through the intervention of rule "R3", which relates to the optimum use of sunlight for room lighting; rule "R4", which concerns the switching off of electrical equipment in the case of the absence of family members in the room; and rule "R5", related to correcting some errors committed by the family members, such as operating some devices without using them, or forgetting about a device that is in use state.

Table 8. Energy consumption scenario in the period: 11 h: 00 m: 00 s to 11 h: 59 m: 59 s.

Time (h, min, s)	Family Member	Place	Event–Activity	Action	Equipment in Operation	Number (N)	Operating Time (Minutes)	Power (W)	Energy Consumed by Event (W)	Energy Consumed per Hour (W)
11 h:00 m:00 s–11 h:59 m:59 s	Father	Out of the house	/	/	/	0	0	0	0	367.50
	Mother	Kitchen	He is out of the house (at work)	Lunch	Light_Kitchen	2	45	25	37.50	
		Dining room	Going to the kitchen	Lunch	Light_DiningRoom	2	15	25	12.50	
	Boy	Dining room	Returning from school and going to the dining room	Lunch	Light_DiningRoom	2	10	15	8.33	
		Room 2	Going to Room 2	Preparation of homework	Light_Room2 Laptop	2 1	15 35	25 50	12.50 29.17	
	Girl	Dining room	Returning from school and going to the dining room	Lunch	Light_DiningRoom	2	15	25	12.50	
		Room 3	Going to Room 3	Preparation of homework	Light_Room3	2	30	25	25.00	
	/	/	/	/	Refrigerator combi (250 L)	1	60	170	170	
	/	/	/	/	Camera	1	60	20	20	
	/	/	/	/	Intercom	1	60	40	40	
	/	/	/	/	Light (outside)	0	60	25	0	

Table 9. The obtained results using IRRHEM intervention.

Time (h, min, s)	Energy Consumption without IRRHEM Intervention (W)	Energy Consumption with IRRHEM Intervention (W)	Energy Saved (W)	Rules Used by IRRHEM
00 h: 00 min: 00 s–01 h: 00 min: 00 s	2380.00	2380.00	0.00	/
01 h: 00 min: 00 s–02 h: 00 min: 00 s	2205.00	2205.00	0.00	/
...	...	...	...	...
05 h: 00 min: 00 s–06 h: 00 min: 00 s	1505.00	280.00	1225.00	R1, R4–R7
06 h: 00 min: 00 s–07 h: 00 min: 00 s	2365.00	1023.33	1341.67	R1, R4–R7
07 h: 00 min: 00 s–08 h: 00 min: 00 s	1625.00	230.00	1395.00	R1, R3–R7
08 h: 00 min: 00 s–09 h: 00 min: 00 s	396. 67	396.67	0.00	/
09 h: 00 min: 00 s–10 h: 00 min: 00 s	1231.67	1215.00	16.67	R3–R5, R7
10 h: 00 min: 00 s–11 h: 00 min: 00 s	535.00	535.00	0.00	/
11 h: 00 min: 00 s–12 h: 00 min: 00 s	367.50	259.17	108.33	R3–R5, R7
12 h: 00 min: 00 s–13 h: 00 min: 00 s	1138.33	1130.00	8.33	R3–R5
13 h: 00 min: 00 s–14 h: 00 min: 00 s	930.00	930.00	0.00	/
14 h: 00 min: 00 s–15 h: 00 min: 00 s	230.00	230.00	0.00	/
15 h: 00 min: 00 s–16 h: 00 min: 00 s	2796.67	2742.50	54.17	R3–R5
16 h: 00 min: 00 s–17 h: 00 min: 00 s	1221.67	1155.00	66.67	R3–R5
17 h: 00 min: 00 s–18 h: 00 min: 00 s	238.33	230.00	8.33	R3–R5
18 h: 00 min: 00 s–19 h: 00 min: 00 s	238.33	230.00	8.33	R3–R5
...	...	...	...	...
22 h: 00 min: 00 s–23 h: 00 min: 00 s	84,980.00	84,980.00	0.00	/
23 h: 00 min: 00 s–00 h: 00 min: 00 s	2380.00	2380.00	0.00	/

4. Analysis and Discussion

Through the obtained results, it can be seen that the value of the energy consumed by the IRRHEM solution intervention is less than or equal to the value of the energy consumed without the intervention of this solution, which proves the IRRHEM solution efficiency in many cases, including (Figure 10):

- Energy savings of 81.39% and 56.73% are obtained during the periods of 05 h: 00 min: 00 s to 06 h: 00 min: 00 s and from 06 h: 00 min: 00 s to 07 h: 00 min: 00 s, respectively, based on:
  - Turning off the “air conditioner 6000 BTU” in the parent’s room;
  - Turning off the “air conditioner 6000 BTU” in the boy’s room;
  - Turning off the “air conditioner 6000 BTU” in the girl’s room;
  - Switching off all the electrical equipment in the event of the absence of occupants;

  - Contributing to the correction of negative characteristics of the family members, such as making mistakes in the use of electrical equipment, as well as forgetting to turn off equipment that is in a working state and not in use;
  - Adapting the rooms of the home according to the family members' wishes;
  - Grouping similar activities for adapting the rooms of the home according to the wishes of members (for example, "Fajr" prayer in congregation).
- Energy savings of 85.84% were obtained during the period of 07 h: 00 min: 00 s to 08 h: 00 min: 00 s, based on:
  - Turning off the "air conditioner 6000 BTU" in the rooms of the parents, boy, and girl;
  - Turning off the electric light bulbs in the rooms of the home (except in special cases) and taking advantage of the sunlight coming from outside of the house;
  - When the family members are absent from a room, all electrical equipment is turned off (except in special cases, such as the refrigerator or security equipment);
  - Correcting some negative actions of individuals, such as leaving a device running without it being needed, or forgetting to turn off a working device. In addition, through this solution, a room in the home can be adapted according to the personal preferences of the individuals, and also the similar activities of family members can be grouped together to be conducted at the same time (for example, the activity of eating meals together).
- Energy savings of 1.34% and 29.47% were obtained during the periods of 09 h: 00 min: 00 s to 10 h: 00 min: 00 s and from 11 h: 00 min: 00 s to 12 h: 00 min: 00 s, respectively, based on:
  - Exploiting the external lighting by opening windows and curtains and turning off some (or all) of the lights bulbs;
  - Turning off electrical equipment in the absence of family members. This solution also addresses the case of an individual forgetting or operating equipment without exploiting it. Finally, this solution aggregates the similar activities of family members, simultaneously.
- Energy savings of 0.73%, 1.93%, 5.45%, 3.49%, and 3.49% were obtained during the periods of 12 h: 00 min: 00 s to 13 h: 00 min: 00 s, 15 h: 00 min: 00 s to 16 h: 00 min: 00 s, 16 h: 00 min: 00 s to 17 h: 00 min: 00 s, 17 h: 00 min: 00 s to 18 h: 00 min: 00 s, and from 18 h: 00 min: 00 s to 19 h: 00 min: 00 s, respectively, based on:
  - Turning off some or all of the light bulbs by taking advantage of the sunlight;
  - In case of the absence of all peoples from a room, the proposed solution take the turning off of electrical appliances (except for some appliances, such as refrigerators and security equipment);
  - Correcting some errors committed in the exploitation of electrical equipment, as well as cases of forgetfulness.

The employment of the IRRHEM solution in different regions results in energy-saving that differs from one region to another. Even if this solution is applied to the same family, we logically obtain different rates of energy consumption; this is due to three main reasons. The first is the climatic factor, which has a great impact on energy saving, thanks to the optimal exploitation of natural resources, such as temperature, lighting, wind. The second factor focuses on home construction that concerns several points, such as home orientation, the quality of the materials used in the construction, and home architecture. Finally, the factor linked to the habits, cultures, and activities of residents also has a significant impact on energy saving. However, for the IRRHEM solution to be universally applied in different regions, a good design of the following elements is necessary:

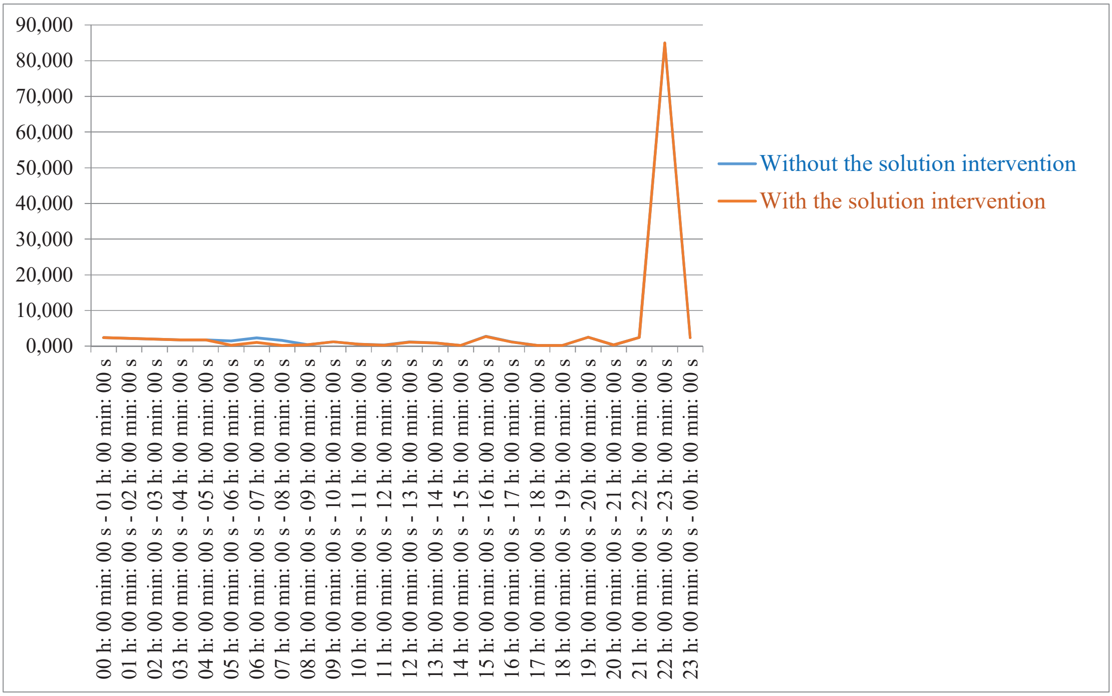


Figure 10. Energy consumption with and without the IRRHEM intervention.

- Profile: this concept is used to provide an explicit representation of many elements of a smart home in a structured information model. In the IRRHEM solution, the profile concept is divided into several subcategories to describe each concept in a particular category as “ResidentProfile” for “Resident”. In addition, this concept is very important and used for several purposes, such as increasing the comfort of the residents and meeting the needs of the residents. Additionally, the category of the profile concerns the “Home” concept that is considered as the starting point for all other concepts, since this concept contains properties or information to describe the smart home, such as: “HomeID”, “HomeLocation”, and “HomeSurfaceSize”.
- Activity scheduling: this concept presents a passive approach to energy efficiency based on determining the optimal schedule for resident activities. It is characterized by several advantages, among which is the optimization of energy consumption.

An important point that characterizes this solution concerns the possibility of extending this solution to future works (the flexibility in updating or exporting the ontology of this solution to develop others), due to the use of the ontological approach, where the latter presents one of the main reasons for choosing this approach.

5. Conclusions

Home automation is the binding between the various devices and systems at home, so that they can all be controlled from anywhere, and the interaction required between them. The intelligent system controls aspects such as lighting, heating, ventilation, air conditioning, safety, security, and energy-saving. The latter is the main objective of the IRRHEM solution.

To accomplish the IRRHEM solution, an energy ontology approach was chosen. This choice is linked to the nature of domestic systems, which are characterized by their possession of a large volume of information, mainly related to family members, household appliances, and climatic data. Therefore, to develop IRRHEM, a series of steps was followed. The first step concerned a detailed study on the home environment; the results of this step were used to build the conceptual model of the data. This model was edited using the "Protégé" software and syntactically checked. The last step concerned the test of the IRRHEM solution using two scenarios (with or without the IRRHEM solution intervention). The results obtained show the importance of IRRHEM by saving 3.60% of energy. In addition, IRRHEM made it possible to perform a set of actions, such as turning off electrical appliances without reducing the comfort of the residents of the home. Additionally, IRRHEM made it possible to:

✓ Group the residents' activities to one time period, which allows for the saving of a large amount of electrical energy through the unified use of electrical appliances;
✓ Switch off the heating systems (or switch on), according to the outside temperature and the preference of the residents;
✓ Switch off the cooling systems (or switch on), according to the outside temperature without neglecting the home residents' comfort;
✓ Open and close the windows, curtains, and doors to take advantage of the outdoor climate (sunlight, heat, and air);
✓ Turn off the lights in a certain area of the house (or all lights) when the residents of the house are away from that area (or the whole house);
✓ Turn off all the appliances in the house, except certain appliances, such as the refrigerator and protective devices, if all the residents are absent from the house.

In addition, among the advantages of the IRRHEM solution is the flexibility to use and update the knowledge base. Finally, the future works can be scheduled as follows:

χ With the development of technology, especially for household appliances, it is important to update the information of the knowledge base;
χ Take into account untreated cases, such as the operation of the cooling and heating system, before entering the house;
χ Integrating renewable energies sources (such as, solar and wind) at home.

Author Contributions: Conceptualization, D.S.; Formal analysis, D.S.; Validation, D.S.; Writing—original draft, D.S.; Software, D.S. and H.H.; Methodology, D.S. and O.C.; Validation, D.S. and O.C.; Funding acquisition, O.C., W.A. and H.H.; Supervision, D.S., O.C. and Y.S.; Visualization, W.A. and H.H.; Writing—Review & editing, Y.S., A.H. and H.H. All authors have read and agreed to the published version of the manuscript.

Funding: Taif University Researchers Supporting Project (TURSP), Taif University, Kingdom of Saudi Arabia under the grant number: TURSP-2020/107.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: This project was supported by the Taif University Researchers Supporting Project number (TURSP-2020/107), Taif University, Taif, Saudi Arabia.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Rahimi, M.; Songhorabadi, M.; Kashani, M.H. Fog-based smart homes: A systematic review. *J. Netw. Comput. Appl.* **2020**, *153*, 102531. [CrossRef]
2. Ryan, J.L. Home Automation. *Electron. Commun. Eng. J.* **1989**, *1*, 185–192. [CrossRef]
3. Thansekhar, M.R.; Balaji, N.; Pandikumar, S.; Vetrivel, R.S. Internet of Things Based Architecture of Web and Smart Home Interface Using GSM. *Int. J. Innov. Res. Sci. Eng. Technol.* **2014**, *3*, 1721–1727.
4. Ijaz, M.; Li, G.; Lin, L.; Cheikhrouhou, O.; Hamam, H.; Noor, A. Integration and Applications of Fog Computing and Cloud Computing Based on the Internet of Things for Provision of Healthcare Services at Home. *Electronics* **2021**, *10*, 1077. [CrossRef]

5. Rafique, M.K.; Khan, S.U.; Zaman, M.S.U.; Mehmood, K.K.; Haider, Z.M.; Bukhari, S.B.A.; Kim, C.-H. An Intelligent Hybrid Energy Management System for a Smart House Considering Bidirectional Power Flow and Various EV Charging Techniques. *Appl. Sci.* **2019**, *9*, 1658. [CrossRef]
6. Saba, D.; Sahli, Y.; Abanda, F.H.; Maouedj, R.; Tidjar, B. Development of new ontological solution for an energy intelligent management in Adrar city. *Sustain. Comput. Inform. Syst.* **2019**, *21*, 189–203. [CrossRef]
7. Raval, M.; Bhardwaj, S.; Aravelli, A.; Dofe, J.; Gohel, H. Smart energy optmization for massive IoT using artificial intelligence. *Internet Things* **2021**, *13*, 100354. [CrossRef]
8. Saba, D.; Laallam, F.Z.; Hadidi, A.E.; Berbaoui, B. Contribution to the Management of Energy in the Systems Multi Renewable Sources with Energy by the Application of the Multi Agents Systems "MAS". *Energy Procedia* **2015**, *74*, 616–623. [CrossRef]
9. Saba, D.; Laallam, F.Z.; Degha, H.E.; Berbaoui, B.; Maouedj, R. Design and Development of an Intelligent Ontology-Based Solution for Energy Management in the Home. In *Machine Learning Paradigms: Theory and Application*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 135–167.
10. Lashkari, Chen and Musilek, Energy Management for Smart Homes—State of the Art. *Appl. Sci.* **2019**, *9*, 3459. [CrossRef]
11. Saba, D.; Laallam, F.Z.; Berbaoui, B.; Fonbeyin, H.A. An Energy Management Approach in Hybrid Energy System Based on Agent's Coordination. In *AISI 2016: Proceedings of the International Conference on Advanced Intelligent Systems and Informatics 2016*; Spinger: Cham, Switzerland, 2016.
12. Saba, D.; Sahli, Y.; Maouedj, R.; Hadidi, A. Energy Management Based on Internet of Things. In *Recent Advances in Technology Acceptance Models and Theories: Studies in Systems, Decision and Control*; Al-Emran, M., Shaalan, K., Eds.; Springer Nature: Cham, Switzerland, 2021; pp. 349–372.
13. Saba, D.; Berbaoui, B.; Degha, H.E.; Laallam, F.Z. A Generic Optimization Solution for Hybrid Energy Systems Based on Agent Coordination. In *Proceedings of the International Conference on Advanced Intelligent Systems and Informatics 2017 AISI 2017*; Advances in Intelligent Systems and Computing; Springer: Cham, Switzerland, 2018.
14. Energy Saving Trust. Save Energy at Home. 2017. Available online: https://www.energysavingtrust.org.uk/sites/default/files/reports/EST_11120_Save%20Energy%20in%20your%20Home_15.6.pdf (accessed on 12 October 2021).
15. Saba, D.; Sahli, Y.; Hadidi, A. An ontology based energy management for smart home. *Sustain. Comput. Inform. Syst.* **2021**, *31*, 100591. [CrossRef]
16. Chantasiriwan, S. Comparative thermo-economic analysis of regenerative Rankine cycles with two feed water heaters. *Case Stud. Therm. Eng.* **2021**, *28*, 101476. [CrossRef]
17. Chen, H.-Y.; Whang, A.J.-W.; Chen, Y.-Y.; Chou, C.-H. The hybrid lighting system with natural light and LED for tunnel lighting. *Optik* **2020**, *203*, 163958. [CrossRef]
18. García-Fuentes, M.Á.; Serna, V.; Hernández, G.; Meiss, A. An Evaluation Framework to Support Optimisation of Scenarios for Energy Efficient Retrofitting of Buildings at the District Level. *Appl. Sci.* **2019**, *9*, 2448. [CrossRef]
19. Saba, D.; Maouedj, R.; Berbaoui, B. Contribution to the development of an energy management solution in a green smart home (EMSGSH). In Proceedings of the 7th International Conference on Software Engineering and New Technologies, Hammamet, Tunisia, 26–28 December 2018.
20. Stamminger, R.; Bues, A.; Alfieri, F.; Cordella, M. Durability of washing machines under real life conditions: Definition and application of a testing procedure. *J. Clean. Prod.* **2020**, *261*, 121222. [CrossRef]
21. Harrington, L.; Aye, L.; Fuller, B. Impact of room temperature on energy consumption of household refrigerators: Lessons from analysis of field and laboratory data. *Appl. Energy* **2018**, *211*, 346–357. [CrossRef]
22. Degha, H.E.; Laallam, F.Z.; Said, B.; Saba, D. Onto-SB: Human Profile Ontology for Energy Efficiency in Smart Building. In Proceedings of the 2018 3rd International Conference on Pattern Analysis and Intelligent Systems (PAIS), Tebessa, Algeria, 24–25 October 2018.
23. Saba, D.; Degha, H.E.; Berbaoui, B.; Maouedj, R. Development of an Ontology Based Solution for Energy Saving Through a Smart Home in the City of Adrar in Algeria. In Proceedings of the International Conference on Advanced Machine Learning Technologies and Applications (AMLTA2018), Cairo, Egypt, 22–24 February 2018; Advances in Intelligent Systems and Computing. Springer: Cham, Switzerland, 2018; pp. 531–541.
24. Kaur, P.; Nand, P.; Naseer, S.; Gardezi, A.A.; Alassery, F.; Hamam, H.; Cheikhrouhou, O.; Shafiq, M. Ontology-Based Semantic Search Framework for Disparate Datasets. *Intell. Autom. Soft Comput.* **2022**, *32*, 1717–1728. [CrossRef]
25. Saba, D.; Laallam, F.Z.; Belmili, H.; Abanda, F.H.; Bouraiou, A. Development of an ontology-based generic optimisation tool for the design of Hybrid energy systems. *Int. J. Comput. Appl. Technol.* **2017**, *55*, 232–243. [CrossRef]
26. Villamar Gómez, L.; Miura, J. Ontology-based knowledge management with verbal interaction for command interpretation and execution by home service robots. *Robot. Auton. Syst.* **2021**, *140*, 103763. [CrossRef]
27. Jung, S.; Lee, S.; Yu, J. Ontological Approach for Automatic Inference of Concrete Crack Cause. *Appl. Sci.* **2020**, *11*, 252. [CrossRef]
28. Huitzil, I.; Molina-Solana, M.; Gómez-Romero, J.; Bobillo, F. Minimalistic fuzzy ontology reasoning: An application to Building Information Modeling. *Appl. Soft Comput.* **2021**, *103*, 107158. [CrossRef]
29. Ren, G.; Li, H.; Liu, S.; Goonetillake, J.; Khudhair, A.; Arthur, S. Aligning BIM and ontology for information retrieve and reasoning in value for money assessment. *Autom. Constr.* **2021**, *124*, 103565. [CrossRef]
30. Hai, N.; Gong, D.; Liu, S. Ontology knowledge base combined with Bayesian networks for integrated corridor risk warning. *Comput. Commun.* **2021**, *174*, 190–204. [CrossRef]

31. Berman, J.J. Drawing inferences from classifications and ontologies. In *Logic and Critical Thinking in the Biomedical Sciences*; Elsevier: Amsterdam, The Netherlands, 2020; pp. 187–208.
32. Muhammad, L.J.; Garba, E.J.; Oye, N.D.; Wajiga, G.M.; Garko, A.B. Fuzzy rule-driven data mining framework for knowledge acquisition for expert system. *Transl. Bioinform. Healthc. Med.* **2021**, *13*, 201–214.
33. Nandhinidevi, S.; Saraswathi, K.; Thangamani, M.; Ganthimathi, M. Design and development of bird ontology using protégé. *Mater. Today Proc.* **2021**, in press. [CrossRef]
34. Degha, H.E.; Laallam, F.Z.; Said, B. Intelligent context-awareness system for energy efficiency in smart building based on ontology. *Sustain. Comput. Inform. Syst.* **2019**, *21*, 212–233. [CrossRef]
35. Chen, C.; Chen, K.; Hsu, C.-Y.; Li, Y.-C. Developing guideline-based decision support systems using protégé and jess. *Comput. Methods Programs Biomed.* **2011**, *102*, 288–294. [CrossRef]
36. Dadkhah, M.; Araban, S.; Paydar, S. A systematic literature review on semantic web enabled software testing. *J. Syst. Softw.* **2020**, *162*, 110485. [CrossRef]
37. D'Alger, W. présentation de la Wilaya d'Alger. Available online: http://www.wilaya-alger.dz/fr/presentation-de-la-wilaya/ (accessed on 15 May 2021).
38. Bouraiou, A.; Necaibia, A.; Boutasseta, N.; Mekhilef, S.; Dabou, R.; Ziane, A.; Sahouane, N.; Attoui, I.; Mostefaoui, M.; Touaba, O. Status of renewable energy potential and utilization in Algeria. *J. Clean. Prod.* **2020**, *246*, 119011. [CrossRef]
39. Available online: https://fr.climate-data.org/afrique/algerie/alger/alger-centre-59540/ (accessed on 26 April 2021).
40. Available online: https://www.historique-meteo.net/afrique/algerie/alger/2020/05/ (accessed on 15 May 2021).

energies

MDPI

Article

Learning-Aided Optimal Power Flow Based Fast Total Transfer Capability Calculation

Ji'ang Liu [1,*], Youbo Liu [1], Gao Qiu [1,*] and Xiao Shao [2]

[1] College of Electrical Engineering Technology, Sichuan University, Chengdu 610065, China; liuyoubo@scu.edu.cn

[2] State Grid Tianfu New Area Electric Power Supply Company, Chengdu 610041, China; shaoxiao@163.com

[*] Correspondence: ymm_liujiang@163.com (J.L.); qiugscu@scu.edu.cn (G.Q.)

Abstract: Total transfer capability (TTC) is a vital security indicator for power exchange among areas. It characterizes time-variants and transient stability dynamics, and thus is challenging to evaluate efficiently, which can jeopardize operational safety. A leaning-aided optimal power flow method is proposed to handle the above challenges. At the outset, deep learning (DL) is utilized to globally establish real-time transient stability estimators in parametric space, such that the dimensionality of dynamic simulators can be reduced. The computationally intensive transient stability constraints in TTC calculation and their sensitivities are therewith converted into fast forward and backward processes. The DL-aided constrained model is finally solved by nonlinear programming. The numerical results on the modified IEEE 39-bus system demonstrate that the proposed method outperforms several model-based methods in accuracy and efficiency.

Keywords: total transfer capability; surrogate assisted method; transient stability; deep learning; interior point method

Citation: Liu, J.; Liu, Y.; Qiu, G.; Shao, X. Learning-Aided Optimal Power Flow Based Fast Total Transfer Capability Calculation. *Energies* **2022**, *15*, 1320. https://doi.org/10.3390/en15041320

Academic Editors: Luis Hernández-Callejo, Sergio Nesmachnow and Sara Gallardo Saavedra

Received: 30 December 2021
Accepted: 7 February 2022
Published: 11 February 2022

1. Introduction

Power systems are currently operated near their stability boundary with the significant proliferation of interconnected grids and renewable penetration [1]. Therefore, online monitoring to transfer security margin of inter-area power transfer is in urgent demand. In the electric industry, total transfer capability (TTC), defined as maximum power exchange allowed to withstand multifarious security contingencies, is a widespread metric to quantify such a security margin. Limited by this issue, dispatchers generally use a conservative constant of offline TTC to decide online operations. Undoubtedly, such TTC values can incur the unwanted waste of line capacity and incorrect estimation to security margin. To untie these knots, the essence is to accelerate TTC calculation.

Thus far, several approaches have been proposed to model TTC calculation [2–4]. Among them, methods with only steady-state considered are inapplicable for TTC evaluation involving transient stability (TS) [5]. To enable TS assessment (TSA), TTC is preferred to be modeled as TS constrained (TSC) programming problem. As the models shown in [6–10], differential-algebraic equations (DAEs) representing system dynamics and TS constraints are discretized throughout the time domain simulation period. And the resulting differential equations are incorporated into the optimal power flow (OPF) model. Nevertheless, as mentioned before, solving such models is quite computationally expensive due to the high-dimensional and nonlinear DAEs involved. In light of this, under current time-varying power grids, inefficient physics-dominated methods can be problematic for fast TTC monitors.

Data-driven approaches have become mainstream to increase calculation speed for security assessment in large-scale power systems [11–13]. Reference [11] proposed an online measurement-based TTC estimator using the nonparametric estimation. Sun et al. developed an automatic learning technique based on the linear least-squares fitting method to extract the TTC operating rules [12]. Unfortunately, these methods own two critical drawbacks. One is that they are hard to capture nonlinear patterns. An empirical and heuristic TTC calculation in the stage of prior sample production is the other problem. It cannot ensure finding the most extreme operating conditions, leading to low fidelity against true modes.

To overcome the first drawback, machine learning (ML) is a promising alternative thanks to its strong nonlinearity learning ability. Reference [13] introduced a hierarchical deep learning machine (HDLM) to successfully achieve real-time TSA, won over other physics-based methods with respect to speed, and beat linear data-driven methods on precision. But sustainable energy is under-investigated. In [14], a TSA framework based on a long short-term memory network was proposed; it improved assessment accuracy by learning from post-fault temporal PMU data dependencies. These applications manifest that ML is a better choice than linear learning methods in nonlinearity modeling tasks. A comparison table with the advantages and disadvantages of the above references is listed in Table 1.

Table 1. A comparison table with the advantages and disadvantages of each reference.

References	Type	Advantages	Disadvantages
[2–4]	Physical-driven model	Focus on steady-state; easy to solve	Transient stability is out of consideration
[6–10]	Physical-driven model with transient stability constraints	Involved transient stability constraints	Computationally expensive
[11–13]	Data-driven model	Faster calculation speed	Hard to capture nonlinear patterns; or sustainable energy is out of consideration

On the other hand, ML can substitute the most time-consuming TSA modules and partially participate in TTC calculation to deal with the second deficiency. This idea follows the classical roadmap of using optimal power flow (OPF) to approach extreme operations but tactfully bypasses high-dimensional modeling such that optimizers can quickly solve TTC. It is technically termed as a learning-aided (also known as surrogate-assisted) method (LAM) [15–18], which utilizes ML algorithms to surrogate the most complex and computationally intensive parts in optimization problems. Reference [18] proposed a method that makes a fusion between surrogates and the evolutionary algorithm to improve the efficiency of optimizing high-dimensional expensive problems. In [1], LAM is also utilized to solve the TTC constrained operation planning problem. The above studies show that LAM can speed up solving optimization problems. At the same time, because it is a data-mechanism hybrid-driven method rather than an utterly data-driven method, it performs better in terms of fidelity.

By prioritizing both merits of physics- and data-driven modeling, this paper proposed a learning-aided optimal power flow based fast TTC calculation methods with the following features:

Deep belief network (DBN) is advocated to surrogate computationally intensive and high-dimensional time-domain based transient stability modelling. This learning-aided scheme allows us to significantly reduce complexity of TTC calculation.

- DBN backwards process is conducted to derive sensitivity of transient stability margin. This sensitivity supports fast and accurate decision for the most extreme growth path of generation and load. The TTC solved under such path is conservative and robust to account for a reliable security indicator.
- Thanks to the above merits, interior point method (IPM) is then introduced to fast calculate TTC. Specifically, DBN forwards and backwards processes respectively provide fast and accurate transient stability inference and gradient information for

IPM. This scheme is firstly used in OPF-based TTC calculation, and numerical studies justified its merits of compromising calculation efficiency and accuracy.

- A comprehensive comparative study is constructed. Numbers of traditional methods, such as the TSCOPF method [9], the sensitivity-based method [19], the repeated power flow (RPF) method [20], and the direct data-driven method [21], are used to demonstrate the superiority of our method.

The organization of this paper is as follows: Section 2 introduces TS constrained optimal power flow (TSCOPF), adopted to model TTC calculation. The learning-aided model for the TS constraints is introduced in Section 3. Section 4 details the proposed solving scheme method, where the Jacobin and Hessian matrices of the learning model are deduced to analytical form to enable combination with nonlinear programming. Section 5 illustrates the numerical study. Finally, the conclusion is presented in Section 6.

2. TTC Calculation with TSCOPF

We believe the OPF method is a brilliant choice because the optimization procedure enables a theoretical search for extreme operating conditions representing TTC. Therefore, TSCOPF is adopted to model TTC calculation problem in this section. According to [22], the generic OPF method for calculating TTC can be formulated as follows:

$$\begin{gathered}\text{Max}\, f(\boldsymbol{y},\boldsymbol{u}) \\ \text{s.t. } g(\boldsymbol{y},\boldsymbol{u}) = 0 \\ h(\boldsymbol{y},\boldsymbol{u}) \leq 0\end{gathered} \tag{1}$$

where $\boldsymbol{y}$,$\boldsymbol{u}$ are the state and control variable vector of the system; and $g(\cdot)$, $h(\cdot)$ are the set of equality and inequality constraints, respectively.

(1) Objective function: It aims to maximize the sum of the active power output of all generators in the source area, i.e.,

$$\text{Max}\, f(\boldsymbol{y},\boldsymbol{u}) = \sum_{k\in Ssou} P_{Gk}, \tag{2}$$

where P_{Gk} is generator active power output at bus k; and S_{sou} means the source area bus set.

(2) Static equality constraints: Power flow equations are formed under polar coordinates, shown below:

$$\begin{gathered}P_{Gi} - P_{Di} - V_i\sum^{n}{}_{j\,=\,1} V_j(G_{ij}\cos\theta_{ij} + B_{ij}\sin\theta_{ij}) = 0, \\ Q_{Gi} - Q_{Di} - V_i\sum^{n}{}_{j\,=\,1} V_j(G_{ij}\sin\theta_{ij} - B_{ij}\cos\theta_{ij}) = 0\end{gathered} \tag{3}$$

where P_{Gi}, P_{Di} represent active generation and demand for bus i; Q_{Gi}, Q_{Di} are reactive generation and demand for bus i, respective; V_i and θ_i are the voltage magnitude and phase angle of bus i, and $\theta_{ij} = \theta_i - \theta_j$; $G_{ij} + jB_{ij}$ is the driving point admittance and the transfer admittance; n is the number of buses.

(3) Static inequality constraints:

$$\begin{gathered}P_{Gi}{}^{\min} \leq P_{Gi} \leq P_{Gi}{}^{\max},\, Gi{\in}S_G{\cup}S_W \\ Q_{Gi}{}^{\min} \leq Q_{Gi} \leq Q_{Gi}{}^{\max},\, Gi{\in}S_G{\cup}S_W \\ V_i{}^{\min} \leq V_i \leq V_i{}^{\max},\, i{\in}S_n \\ P_{ij} \leq P_{ij}{}^{\max},\, ij{\in}S_l\end{gathered} \tag{4}$$

where $P_{Gi}{}^{\min}$, $P_{Gi}{}^{\max}$, $Q_{Gi}{}^{\min}$, $Q_{Gi}{}^{\max}$ are the lower and upper limits of the generator active and reactive power at bus k, respective; $V_i{}^{\min}$ and $V_i{}^{\max}$ are the lower and upper limits of the voltage at bus i; $P_{ij}{}^{\max}$ is the transmission threshold of line ij; S_G, S_W, S_n, S_l are the sets of generators, wind farms, buses, and lines.

(4) Transient stability constraints: This paper adopts the classical generator model to analyze transient stability. During the dynamic process, loads are modeled as constant impedance. Hence, generic TS models can be simplified as follows:

$$\begin{gathered} \boldsymbol{x}'(t) = \rho_c(\boldsymbol{x}(\mathrm{t}), \boldsymbol{y}(t), \boldsymbol{u}), \\ \psi_c(\boldsymbol{x}(t), \boldsymbol{y}(t), \boldsymbol{u}) \geq 0, c \in S_c, t \in (t_0, t_{\mathrm{end}}] \end{gathered} \tag{5}$$

where $\boldsymbol{x}$, $\boldsymbol{y}$ are the algebraic and state variables; $[\boldsymbol{x}(t), \boldsymbol{y}(\mathrm{t})]$ refers to the operating condition during the transient period $(t_0, t_{\mathrm{end}}]$; S_c is a set of pre-contingencies; $\psi_c(\cdot)$ is the transient stability criterion used in this paper [23], and it is shown as follows:

$$\begin{gathered} |\delta_i(t) - \delta_{COI}(t)| \leq \delta_{thr}, t \in (t_0, t_{\mathrm{end}}] \\ \delta_{COI}(t) = (\textstyle\sum_i M_i \cdot \delta_i(t)) \ / \ (\sum_i M_i), i \in \{1, \ldots, n_G\} \end{gathered} \tag{6}$$

where $\delta_i(t)$ is the rotor angles of generator i; $\delta_{COI}(t)$ is the rotor angle under the center of inertia (COI); δ_{thr} is the instability threshold that is usually set as 180 degree [23]; M_i represents the inertia constant of the ith generator; and n_G denotes the number of generators.

It should be mentioned the DAEs Equation (5) encompasses numerous time-domain variables, i.e., $\delta_i(t)$. With more precise timestep and more contingencies to be checked, the dimensionality of Equation (5) will be of exponential growth.

3. Proposed Surrogate Model

In TSC programming problem, exact state and parameter estimation for dynamic components (e.g., synchronous generators) must be conducted to truly model the transient process. This is difficult because large-scale state estimation is challenging regarding efficiency and precision. A sensible alternative is to directly encapsulate transient stability dynamics in a parameterized model so as to bypass state estimation. Rich data is needed, fortunately, it can be easily gathered nowadays in smart grid.

As reported before, TSA significantly increases the computational burden of solving the OPF model. To reduce the massive time-domain variables, a data-driven learning-aided model is proposed. This model allows us to map Equation (5) into a parametric space, such that the time-domain variables can be surrogated by ML structural parameters independent of optimization, and few parallel forwards processes of ML are enabled to circumvent quantities of DAEs.

3.1. Data Sample Generation

The first step of training such learning-aided models is data generation. To this end, random operations are sampled and simulated under prior distributions of power systems. To simplify illustrations, we respectively denote the input features and the target features as $\boldsymbol{X}$ and $\boldsymbol{Y}$. $\boldsymbol{X}$ covers almost all variables that SCADA can measure, while $\boldsymbol{Y}$ is the TS margin index. Equation (7) details the data structure:

$$\begin{gathered} \boldsymbol{X} = \{\boldsymbol{P}_G, \boldsymbol{V}_G, \boldsymbol{P}_D, \boldsymbol{Q}_D, \boldsymbol{V}_b\}, G \in S_G \cup S_W, D \in S_D \\ \boldsymbol{Y} = \{\Gamma_c\}, c \in S_c \end{gathered} \tag{7}$$

where $\boldsymbol{P}_G$ and $\boldsymbol{P}_D$ are the characteristic vector of active generator output and active load, respectively; $\boldsymbol{Q}_D$ is the vector of reactive load; $\boldsymbol{V}_G$ represents the voltage of buses where generators are located; and $\boldsymbol{V}_b$ means the voltage of other buses. Γ_c represents the TS margin of the corresponding operation. In this paper, TS index (TSI) is adopted to quantify TS margin, which can be formulated as:

$$\begin{gathered} \mathrm{TSI} = 100 \times (\delta_{thr} - |\delta_{\max}|)/(\delta_{thr} + |\delta_{\max}|), \\ \delta_{\max} = \max(|\delta_{Gi} - \delta_{Gj}|), Gi, Gj \in S_G \end{gathered} \tag{8}$$

where $\delta_{\max}$ is the maximum power angle difference during the post-fault duration.

Now turning to introduce calculation for Equation (7). To attain $\boldsymbol{X}$, we firstly sample controllable variables under their prior limits by Equation (9), and loads under historical distributions by Equation (9):

$$\begin{aligned}\boldsymbol{X}_{gen} &= \{\boldsymbol{X}^1{}_{gen}; \ldots ; \boldsymbol{X}^n{}_{gen}\} = \{\boldsymbol{P}^1{}_G, \boldsymbol{V}^1{}_G; \ldots ; \boldsymbol{P}^n{}_G, \boldsymbol{V}^n{}_G\},\\ \boldsymbol{X}_{load} &= \{\boldsymbol{X}^1{}_{load}; \ldots ; \boldsymbol{X}^n{}_{load}\} = \{\boldsymbol{P}^1{}_D, \boldsymbol{Q}^1{}_D; \ldots ; \boldsymbol{P}^n{}_D, \boldsymbol{Q}^n{}_D\}\end{aligned} \tag{9}$$

where $\boldsymbol{X}_{gen}$, $\boldsymbol{X}_{load}$ are the control and load variables subsets of $\boldsymbol{X}$, respectively; n is the number of samples.

The power flow program is then performed to get equilibrium points to determine the state variables $\boldsymbol{V}_b$. Notably, samples should be evenly distributed over operational space to ensure the generalization ability of the learning-aided model. Therefore, Latin hypercube sampling (LHS) is adopted to generate samples in this paper [24].

Afterward, we impose disturbances in contingencies for one equilibrium point of $\boldsymbol{X}$ to obtain post-fault trajectories to compute TS margin Γ_c. Via traversing each point in $\boldsymbol{X}$, $\boldsymbol{Y}$ can be collected. Supervised learning can herewith be utilized to learn the learning-aided model.

3.2. Deep Belief Network Based TSA Learning-Aided Model

According to the data structure, the deep belief network (DBN) is an advisable alternative for our goal. DBN is a probability generation model that stacks multiple restricted Boltzmann machines (RBMs) and a fully connected layer. RBM is an unsupervised network composed of a visible and hidden layer, and it can probabilistically reconstruct input features by two-way connections between the two layers.

As an energy-based model, the energy function of RBM is calculated by [25]:

$$E(\boldsymbol{v},\boldsymbol{h}) = -\,\boldsymbol{v}^T\boldsymbol{w}\boldsymbol{h} - \boldsymbol{a}^T\boldsymbol{v} - \boldsymbol{b}^T\boldsymbol{h},\ \boldsymbol{w}\in\mathrm{R}^{nh\times nv},\ \boldsymbol{a}\in\mathrm{R}^{nv},\ \boldsymbol{b}\in\mathrm{R}^{nh} \tag{10}$$

where $\boldsymbol{v}$,$\boldsymbol{h}$ are the visible and hidden layer matrices; $\boldsymbol{a}$,$\boldsymbol{b}$ are the bias matrices of $\boldsymbol{v}$,$\boldsymbol{h}$ respectively; and $\boldsymbol{w}$ is the weight matrix between two layers. The joint probability distribution $P(\boldsymbol{v},\boldsymbol{h})$ of $\boldsymbol{v}$ and $\boldsymbol{h}$ is formulated by:

$$P(\boldsymbol{v},\boldsymbol{h}) = Z^{-1}\, e^{-E(\boldsymbol{v},\boldsymbol{h})},\ Z = \sum\nolimits_{\boldsymbol{v},\boldsymbol{h}} e^{-E(\boldsymbol{v},\boldsymbol{h})} \tag{11}$$

where Z is the normalization factor that ensures the sum of the probability distribution is 1. The marginal probability of $\boldsymbol{v}$ and $\boldsymbol{h}$, which are also called the likelihood functions, can be formulated as:

$$P(\boldsymbol{v}) = Z^{-1}\sum\nolimits_{\boldsymbol{h}} e^{-E(\boldsymbol{v},\boldsymbol{h})},\ P(\boldsymbol{h}) = Z^{-1}\sum\nolimits_{\boldsymbol{v}} e^{-E(\boldsymbol{v},\boldsymbol{h})} \tag{12}$$

Due to the lack of intra-layer connections in RBM, the activations of units in the visible and hidden layers are independent. Therefore, when the visible layer (or hidden layer) units state is given, we can deduce the formulation of the conditional probability that an individual unit of the hidden layer (or visible layer) is activated as:

$$\begin{aligned}P(h_i = 1 \mid \boldsymbol{v}) &= M(b_i + \textstyle\sum_i w_{ij}\cdot v_j),\ h_i\in\boldsymbol{h},\ v_j\in\boldsymbol{v},\ w_{ij}\in\boldsymbol{w}\\ P(v_i = 1 \mid \boldsymbol{h}) &= M(a_j + \textstyle\sum_j w_{ij}\cdot h_i),\ a_j\in\boldsymbol{a}\in\mathrm{R}^{nv},\ b_i\in\boldsymbol{b}\in\mathrm{R}^{nh}\end{aligned} \tag{13}$$

where $M(\cdot)$ is the activation function, and in the paper, it is the *Sigmoid* function. Then, the conditional probability of $\boldsymbol{h}$ (or $\boldsymbol{v}$) given $\boldsymbol{v}$ (or $\boldsymbol{h}$) can be obtained:

$$P(\boldsymbol{h} \mid \boldsymbol{v}) = \prod\nolimits^{nh}{}_{i\,=\,1}P(h_i \mid \boldsymbol{v}),\ P(\boldsymbol{v} \mid \boldsymbol{h}) = \prod\nolimits^{nv}{}_{j\,=\,1}P(v_j \mid \boldsymbol{h}),\ h_i\in\boldsymbol{h},\ v_j\in\boldsymbol{v} \tag{14}$$

where nh,nv are the number of units in the hidden and visible layer, respectively.

Training RBM is to maximize the following likelihood L:

$$\ln L = \ln\prod\nolimits_{v\in Strain}P(\boldsymbol{v}), \tag{15}$$

where S_{train} is the training sample set. The commonly used numerical method for maximizing (15) is gradient ascent, which iteratively updates the parameters. Take $\boldsymbol{w}$ as an example, and the weight w_{ij} is updated via Equations (16) and (17):

$$w_{ij} = w_{ij} + \eta \cdot (\partial \ln(P(\boldsymbol{v})))/(\partial w_{ij}),\ w_{ij} \in \boldsymbol{w} \tag{16}$$

$$(\partial \ln(P(\boldsymbol{v})))/(\partial w_{ij}) = P(h_i = 1 \mid \boldsymbol{v})v_j - \sum\nolimits_{\boldsymbol{v}} P(\boldsymbol{v})P(h_i = 1 \mid \boldsymbol{v})v_j,\ h_i \in \boldsymbol{h},\ v_j \in \boldsymbol{v} \tag{17}$$

where η is the learning rate.

DBN training consists of two parts: pre-training and fine-tuning. In the pre-training part, any two connected layers except the fully connected layer can be regarded as an RBM. These RBMs are trained to obtain better initial weights and to alleviate the gradient disappearance problem. In the fine-tuning part, the trained RBMs are connected with the fully connected layer. The sample sets $[\boldsymbol{X}, \boldsymbol{Y}]$ and the global learning algorithm are then used for supervised fine-tuning of the DBN, learning the mapping between input data and labels. Thence, the mathematical model of an l-layer DBN can be simplified by Equation (18):

$$\Psi(\boldsymbol{X}) = O(M(D_{l\,-\,1}(\ldots M(D_1(\boldsymbol{X}))\ldots))), \tag{18}$$

$$D_i(\boldsymbol{x}_i) = \boldsymbol{w}_i\boldsymbol{x}_i + \boldsymbol{b}_i,\ w_i \in \mathrm{R}^{ni\,\times\,n(i-1)},\ i = 1, \ldots, l-1 \tag{19}$$

$$\begin{array}{c} \boldsymbol{w}_i = [\boldsymbol{w}_i{}^1, \ldots, \boldsymbol{w}_i{}^{ni}],\ \boldsymbol{w}_i{}^{ni} \in \mathrm{R}^{1\,\times\,n(i-1)} \\ \boldsymbol{b}_i = [b_i{}^1, \ldots, b_i{}^{ni}] \end{array} \tag{20}$$

where $O(\cdot)$ is the output function of the fully connected layer, and $O(\boldsymbol{x}) = D_l(\boldsymbol{x}_l)$. The loss function can be defined as the weighted sum of the estimated error and L_2 norm, i.e.:

$$\mathrm{Min}\ \alpha\|\boldsymbol{Y} - \Psi(\boldsymbol{X})\|_2{}^2 + (1-\alpha)\sum\nolimits^{n}{}_{i\,=\,1}\ \|\boldsymbol{w}_i\|_2{}^2, \tag{21}$$

Equation (21) can be solved by training and fine-tuning the DBN model [24].

After training, the learning-aided model is reformed as Equation (22):

$$\Gamma_c = \Psi_c(\boldsymbol{X}),\ c \in S_c \tag{22}$$

3.3. Learning-Aided OPF for TTC Calculation

The trained learning-aided model is finally forwarded to replace (5)~(6) to mitigate the TTC computational burden. The reformed learning-aided OPF for calculating TTC is given as follows:

$$\begin{array}{c} \text{Maximize (2)} \\ \text{s.t. (3)\textasciitilde(4)} \\ \Gamma_c \geq 0,\ c \in S_c \end{array} \tag{23}$$

In Equation (23), steady-state physics remains the same, but dynamics become a data model. This modeling strategy possesses several merits: (1) it remarkably reduces the solving complexity of the full physics version. (2) it preserves physics to decrease adverse effects from significant learning errors. A common way to solve Equation (23) is gradient-free algorithms [26–28]. However, these algorithms characterize cumbersome stochastic search mechanism. A fast-solving algorithm for such physics and data hybrid model is still under exploitation.

4. Proposed Solution Method

In this paper, the interior point method (IPM) [29] is conducted to solve (25). Towards this end, the *Jacobian* and *Hessian* matrix of the trained DBN model is analyzed.

4.1. Interior Point Method

For the sake of simplification, model Equation (23) is firstly reformed as the following canonical form:

$$\begin{gathered}\text{Max } F(\boldsymbol{x}),\\ \text{s.t. } \boldsymbol{G}(\boldsymbol{x}) = \boldsymbol{0},\\ [\boldsymbol{H}(\boldsymbol{x}) \leq \boldsymbol{0}] = [\boldsymbol{H}_c(\boldsymbol{x}) \leq \boldsymbol{0}, \boldsymbol{H}_S(\boldsymbol{x}) \leq \boldsymbol{0}]\end{gathered} \tag{24}$$

where $F(\boldsymbol{x})$ is the objective function; $\boldsymbol{G}(\boldsymbol{x}) = [G_1(\boldsymbol{x}), \ldots, G_{\mathrm{m}}(\boldsymbol{x})]^T$ is the nonlinear equality constraints; $\boldsymbol{H}(\boldsymbol{x}) = [H_1(\boldsymbol{x}), \ldots, H_{\mathrm{r}}(\boldsymbol{x})]^T$ is the non-linear inequality constraints, and $\boldsymbol{H}_c(\boldsymbol{x})$, $\boldsymbol{H}_S(\boldsymbol{x})$ are the constraints in (4) and the surrogate model in (23b), respectively; r,m are the number of inequality and equality constraints.

Use IPM to solve Equation (24), and the steps are as follows [29]:

1. Add slack variables $\boldsymbol{l} = [l_1, \ldots, l_r]^T$ ($\boldsymbol{l} > \boldsymbol{0}$) and $\boldsymbol{u} = [u_1, \ldots, u_r]^T$ ($\boldsymbol{u} > \boldsymbol{0}$) to transform $\boldsymbol{H}(\boldsymbol{x})$ into equality constraints;
2. Introduce the disturbance factor μ ($\mu > 0$) to transfer $F(\boldsymbol{x})$ into the barrier function, which makes it impossible for the barrier objective function to find an extremal solution on the boundary, and the optimal solution can only be obtained when the constraints are satisfied;
3. Apply Lagrangian multiplier method to solve the transformed model, and the Lagrangian function is formulated as:

$$\begin{gathered}L = F(\boldsymbol{x}) - \zeta^T \boldsymbol{G}(\boldsymbol{x}) - \boldsymbol{z}^T[\boldsymbol{H}(\boldsymbol{x}) - \boldsymbol{l} - \boldsymbol{H}_{\mathrm{min}}] - \boldsymbol{\omega}^T[\boldsymbol{H}(\boldsymbol{x}) + \boldsymbol{u} - \boldsymbol{H}_{\mathrm{max}}]\\ - \mu\sum^{r}{}_{i=1}\log(l_i) - \mu\sum^{r}{}_{i=1}\log(l_i),\end{gathered} \tag{25}$$

where $\boldsymbol{\zeta}$,$\boldsymbol{z}$,$\boldsymbol{\omega}$, are Lagrangian multipliers, respectively.

4. Calculate μ via Equation (26):

$$\mu = \sigma(\boldsymbol{l}^T\boldsymbol{z} - \boldsymbol{u}^T\boldsymbol{\omega})/2\mathrm{r}, \tag{26}$$

where σ denotes the central parameter.

5. Consider the Karush–Kuhn–Tucker (KKT) conditions and adopt the Newton method, the matrix form of the modified equations can be deduced as:

$$\begin{gathered}\boldsymbol{\Lambda} \cdot \Delta\boldsymbol{x} + (\partial\boldsymbol{G}(\boldsymbol{x})/\partial\boldsymbol{x}) \cdot \Delta\boldsymbol{x} = \boldsymbol{\Phi},\\ (\partial\boldsymbol{G}(\boldsymbol{x})^T/\partial\boldsymbol{x}) \cdot \Delta\boldsymbol{x} = \boldsymbol{G},\\ \boldsymbol{\Lambda} = (\partial^2\boldsymbol{G}(\boldsymbol{x})/\partial\boldsymbol{x}^2)\zeta + (\partial^2\boldsymbol{H}(\boldsymbol{x})/\partial\boldsymbol{x}^2)(\boldsymbol{z} + \boldsymbol{\omega}) - (\partial^2F(\boldsymbol{x})/\partial\boldsymbol{x}^2)\\ + (\partial\boldsymbol{H}(\boldsymbol{x})/\partial\boldsymbol{x})(\boldsymbol{u}^{-1}\boldsymbol{\omega} - \boldsymbol{l}^{-1}\boldsymbol{z})(\partial\boldsymbol{H}(\boldsymbol{x})/\partial\boldsymbol{x})^T,\\ \boldsymbol{\Phi} = -L_{\boldsymbol{x}} - (\partial\boldsymbol{H}(\boldsymbol{x})/\partial\boldsymbol{x})[\boldsymbol{L}^{-1}(L_{\boldsymbol{l}}{}^{\mu} + \boldsymbol{Z}L_{\boldsymbol{z}}) + \boldsymbol{U}^{-1}(L_{\boldsymbol{u}}{}^{\mu} + \boldsymbol{W}L_{\boldsymbol{\omega}})]\end{gathered} \tag{27}$$

where $L_{\boldsymbol{x}}$, $L_{\boldsymbol{z}}$, $L_{\boldsymbol{\omega}}$, $L_{\boldsymbol{l}}{}^{\mu}$ and $L_{\boldsymbol{u}}{}^{\mu}$ are the partial derivatives of L to $\boldsymbol{x}$, $\boldsymbol{z}$, $\boldsymbol{\omega}$, $\boldsymbol{l}$ and $\boldsymbol{u}$.

Besides, the corrections of $\boldsymbol{z}$, $\boldsymbol{\omega}$, $\boldsymbol{l}$ and $\boldsymbol{u}$ can be calculated via (28):

$$\begin{gathered}\Delta\boldsymbol{z} = \boldsymbol{L}^{-1}L_{\boldsymbol{l}}{}^{\mu} - \boldsymbol{L}^{-1}\boldsymbol{Z}\Delta\boldsymbol{l},\\ \Delta\boldsymbol{l} = (\partial\boldsymbol{H}(\boldsymbol{x})/\partial\boldsymbol{x})^T\Delta\boldsymbol{x} - L_{\boldsymbol{z}},\\ \Delta\boldsymbol{\omega} = \boldsymbol{U}^{-1}L_{\boldsymbol{u}}{}^{\mu} - \boldsymbol{U}^{-1}\boldsymbol{W}\Delta\boldsymbol{u},\\ \Delta\boldsymbol{u} = -(\partial\boldsymbol{H}(\boldsymbol{x})/\partial\boldsymbol{x})^T\Delta\boldsymbol{x} + L_{\boldsymbol{\omega}}\end{gathered} \tag{28}$$

where $\boldsymbol{Z} = \mathrm{diag}(\boldsymbol{z})$; $\boldsymbol{W} = \mathrm{diag}(\boldsymbol{\omega})$; $\boldsymbol{L} = \mathrm{diag}(\boldsymbol{l})$; and $\boldsymbol{U} = \mathrm{diag}(\boldsymbol{u})$.

6. Use the corrections calculated via Equations (27) and (28) to update the variables as follows:

$$\begin{gathered}\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \alpha_p\Delta\boldsymbol{x},\ \boldsymbol{\zeta}^{(k+1)} = \boldsymbol{\zeta}^{(k)} + \alpha_d\Delta\boldsymbol{\zeta},\\ \boldsymbol{l}^{(k+1)} = \boldsymbol{l}^{(k)} + \alpha_p\Delta\boldsymbol{l},\ \mathbf{z}^{(k+1)} = \mathbf{z}^{(k)} + \alpha_d\Delta\mathbf{z},\\ \boldsymbol{u}^{(k+1)} = \boldsymbol{u}^{(k)} + \alpha_p\Delta\boldsymbol{u},\ \boldsymbol{\omega}^{(k+1)} = \boldsymbol{\omega}^{(k)} + \alpha_d\Delta\boldsymbol{\omega},\end{gathered} \tag{29}$$

where the step size α_p and α_d for each update are shown in Equation (30).

$$\begin{gathered}\alpha_p = 0.9995\min[\min(-\boldsymbol{l}_i/\Delta\boldsymbol{l}_i, \boldsymbol{l}_i < 0; -\boldsymbol{l}_i/\Delta\boldsymbol{u}_i, \boldsymbol{u}_i < 0), 1], \\ \alpha_d = 0.9995\min[\min(-z_i/\Delta z_i, z_i < 0; -\boldsymbol{\omega}_i/\Delta\boldsymbol{\omega}_i, \boldsymbol{\omega}_i > 0), 1], i = 1, \ldots, r\end{gathered} \tag{30}$$

7. Termination condition: if $(\boldsymbol{l}^T\boldsymbol{z} - \boldsymbol{u}^T\boldsymbol{\omega}) < \varepsilon$, the current $\boldsymbol{x}$ is output; else re-execute (4) to (6). Here ε represents the specified threshold.

4.2. Deducing Analytical Surrogate Model for IPM

As shown in Equation (26), the gradient of functions F,$\boldsymbol{G}$ and $\boldsymbol{H}$ are needed in the process of IPM. The gradient of F, $\boldsymbol{G}$ and $\boldsymbol{H}_c$ can be obtained directly. However, since the surrogate model is a "black-box", the gradient of $\boldsymbol{H}_s$ cannot be simply calculated. Based on Equations (22) and (23), the gradient of $\boldsymbol{H}_s$ can be transformed into:

$$\begin{gathered}\triangledown\boldsymbol{H}_s = \triangledown\Gamma_c = \triangledown\Psi_c(\boldsymbol{X}), c\in S_c \\ \triangledown^2\boldsymbol{H}_s = \triangledown^2\Gamma_c = \triangledown^2\Psi_c(\boldsymbol{X})\end{gathered} \tag{31}$$

Next, by DBN backwards process, Equation (31) is deduced to get the Jacobin and Hessian matrix, which are also known as sensitivities of transient stability against optimization variables. See Appendix A for the detailed DBN backwards process. The algorithm flow and implementation of the proposed method are shown in Figure 1.

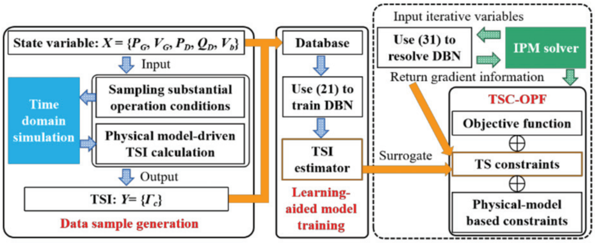

State variable: $X = \{P_G, V_G, P_D, Q_D, V_b\}$
Input
Time domain simulation
Sampling substantial operation conditions
Physical model-driven TSI calculation
Output
TSI: $Y = \{\Gamma_c\}$
Data sample generation
Database
Use (21) to train DBN
TSI estimator
Learning-aided model training
Input iterative variables
Use (31) to resolve DBN
IPM solver
Return gradient information
TSC-OPF
Objective function
⊕
Surrogate
TS constraints
⊕
Physical-model based constraints

Figure 1. The flow chart of the proposed method.

5. Numerical Case Study

5.1. Test System

The proposed method is testified on the modified IEEE 39-bus system. The base power of the system is 100 MW, and the system is divided into the source (Area I) and sink (Area II) areas by four tie-lines 1–39, 2–3, 3–18, and 16–17, as shown in Figure 2. Two wind farms with a total capacity of 500 MW are connected to buses 17 and 21. As shown in Section 3.1, TSI calculated by power angle is adopted to quantify TS margin. So, this paper assumes that the wind farms have sufficient reactive power reserves and low voltage ride-through capability to ensure they do not trip. A three-phase short circuit on each tie-line is pre-selected as the contingencies.

5.2. Learning-Aided Model Construction

As mentioned before, generation prior distribution is assumed to be a uniform distribution over generators' nominal limits. Regarding generation and load balance constrained, the total load is determined as the sum of generation. Nodal load is then acquired by sampling from historical load distributions. As for the settings of time-domain simulation, fault start time is 0.1 s, simulation period is 2 s, and timestep is 0.05 s. Following the above preconditions, 10,000 samples are generated, of which the ratios to the training set, validation set, and test set are 80%, 10%, and 10%.

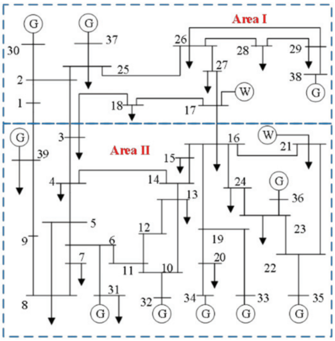


Figure 2. The modified IEEE 39-bus system for case study.

The DBN structure from the input layer to the output layer is {93-40-20-10-5-4}, where the elements stand for neuron quantity. Sigmoid is selected as the activation function. After training the samples, the DBN is forwarded to be tested on out-of-sample sets (i.e., test set). The scatter of estimates vs. actual values and error distribution is shown in Figure 3. The coefficient of determination (R2) is 0.9814, and the mean square error is 5.84 × 10^{-4}. As shown in Figure 3b, the error distribution approximately obeys a normal distribution, and the mean and standard deviation of the estimation error are 0 and 0.023. It can be found that 95% of the samples are in the interval of [−0.042, 0.046] through statistics, and the error with the 95% confidence level of the normal distribution is 0.004. Figure 3 demonstrates that the proposed learning model can render accurate TSA and strongly generalizes.

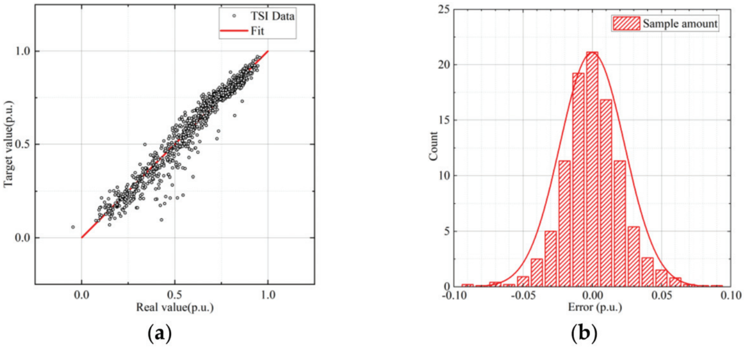


Figure 3. The visualization of testing the trained learning model: (**a**) estimate vs. true; (**b**) error distribution.

To further verify the performance of DBN, comparisons against back propagation neural network (BPNN), support vector regression (SVR), and regression tree (RT) are carried out, and the outcomes are given in Table 2. Mean square error (MSE) and square correlation coefficient (SCC) are used to evaluate the performance. You can see clearly that the DBNs beat other learning methods; thus, it can be concluded that DBN is the best one in TSA tasks.

Table 2. Accuracy comparison of each TSI surrogate model on test sets.

Indicator	2-Layer DBN	3-Layer DBN	BPNN	SVR	RT
MSE/p.u.	0.0054	0.0019	0.0023	0.0346	0.0927
SCC	0.9480	0.9712	0.9627	0.9171	0.8814

5.3. The Results of TTC Fast Calculation

In this section, the proposed method is testified and compared with other methods, such as the TSC-OPF method [9], the sensitivity-based method [19], the repeated power flow (RPF) method [20], and the direct data-driven method [21]. These methods are summarized in Table 3. M1 is to directly incorporate the DAEs into the optimization problem by adopting the implicit integration rule. M2 uses trajectory sensitivity to achieve TTC calculation. M3 gets the TTC by gradually increasing the generator power base on the initial state and repeatedly calculating the power flow until a certain constraint is about to be violated. And, M5 applies NNs to learn the mapping between system state variables and TTC values. Moreover, to manifest the superiority of our methods, we have advanced experiments under single- and multi-contingency conditions, of which the outcomes are respectively visualized in Figure 4a,b.

Table 3. Different methods and pre-contingencies for TTC calculation.

Methods	**TSCOPF**	**The Sensitivity-Based Method**	**The Repeated Power Flow Method**	**TSCOPF with DBN-Assisted**
Symbol	M1	M2	M3	M4
	single contingency		multi contingency	
Line	1–39		1–39, 2–3, 3–18, 16–17	

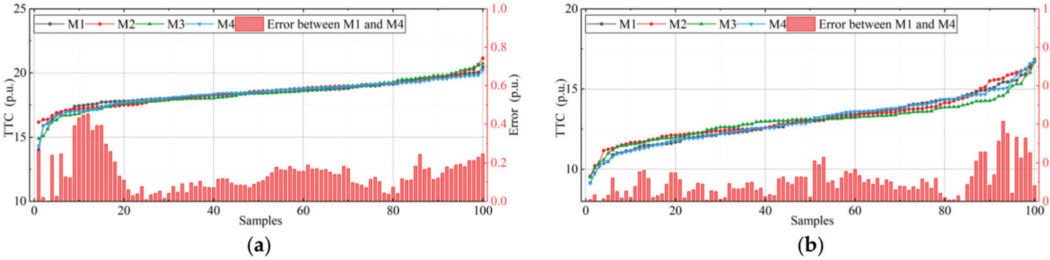


Figure 4. The results of TTC calculation under four different methods: (**a**) Single contingency; (**b**) Multi contingencies.

Figure 4 shows the TTC values calculated by the applied methods under 100 unseen scenarios. The samples are sorted according to the ascending order of the TTC value calculated by M1 to facilitate viewing, and the histogram shows the error between the TTC values calculated by M1 and M4. Taking Figure 4b as an example, the TTC error of M4 is within the acceptable range of [0, 0.5 p.u.], and the average error is 0.1019 p.u. The TTC average errors of M2 and M3 are 0.2308 p.u. and 0.3547 p.u., respectively. Obviously, the TTC value calculated by M4 has the smallest error among several comparison methods. It illustrates that the proposed learning-aided OPF based method can calculate the TTC value more accurately than the RPF and sensitivity-based method. This is because M4, like M1, is modeled based on TSCOPF, which can better describe the system state and has better fidelity than M2 and M3. In addition, it can search the extreme operating point more accurately.

Furthermore, to verify the accuracy of the proposed method, it is compared with the direct data-driven approach (symbol as M5). M5 takes the TTC calculated by M1 as the sample label. Then, it utilizes the DBN model to learn the implicit relationship between the input feature **X** and the target feature $\mathbf{Y}_{\mathrm{TTC}}$ and forms a mapping. Figure 5 shows the comparison results of M4 and M5 when the TTC calculated by M1 is used as the reference value. It can be found that M4 has a smaller average relative error, 0.1019 p.u., than M5, which is 0.3297 p.u., in 100 test samples. It means the proposed method has better fidelity than direct data-driven methods.

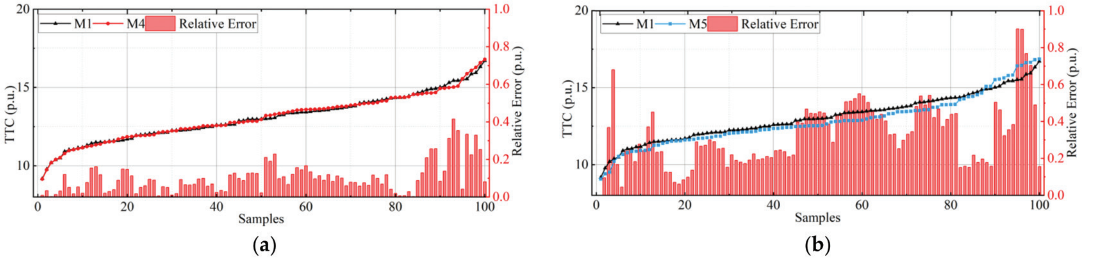


Figure 5. The results of TTC calculation under four different methods: (**a**) M4 compared with M1; (**b**) M5 compared with M1.

In addition, time-domain simulations are performed for each test sample to verify that the operation obtained when calculating the TTC value satisfies the TSCs. The post -fault transient trajectory of rotor angle differences between the individual generators is recorded. The result of a typical sample is shown in Figure 6, where Figure 6a is the transient trajectories after sample initial power flow calculation. Then, utilize the proposed method to calculate the TTC of this sample, and a new operating condition, whose transient trajectories are shown in Figure 6b, can be obtained. It can be observed that the curves have apparent fluctuations. The angle difference between Gen34 (the generator on bus 34) and Gen39 (the generator on bus 39) has the most significant change and is close to the set stability threshold, 180 degrees. It means that the system is operating at its TS boundary at this time. In addition, Figure 6b illustrates that DBN can accurately estimate TSI, and the sensitivity of transient stability margin can help OPF find boundaries of the system. It demonstrates that the learning-aided model can follow the TSCs effectively when it calculates the TTC. Furthermore, the learning-aided model can help the TSCOPF accurately find the most extreme operating condition.

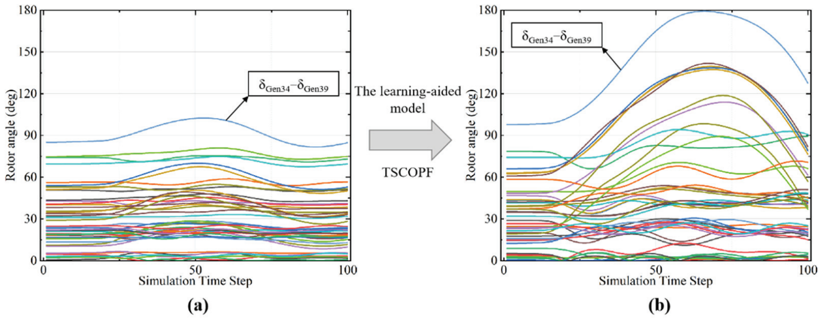


Figure 6. The post-fault transient trajectories of rotor angle differences of the test sample: (**a**) the trajectories after sample initial power flow calculation; (**b**) after TTC calculation.

5.4. Efficiency Comparison

Figure 7 depicts solving time statistics, where the X-axis represents the number of faults in pre-contingency, and the Y-axis is computation time. As shown in Figure 7, the runtime of M1, M2, and M3 are significantly longer than that of M4 under single contingency (i.e., one fault in pre-contingency). And, all algorithms consume more time to compute TTC with more contingencies considered, except for M4. This is because M1, M2, and M3 all need to calculate DAEs associated with TSCs in iterations, and the dimensions of DAEs are higher as more contingencies are considered. However, M4 surrogates the time-consuming part by learning-aided model, and reduces the computation time. The results claim that the proposed method significantly outperforms other comparative methods with respect to efficiency.

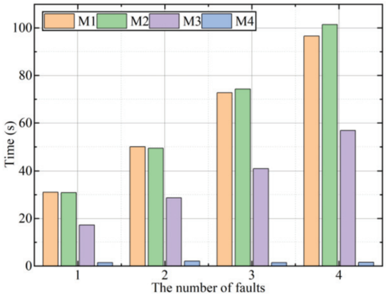


Figure 7. Efficiency analysis.

6. Conclusions

Large-scale wind power penetration has increased the potential insecurity risk of inter-area power exchange. Therefore, rapid and accurate security assessment for inter-corridors is imperative. Towards this end, this paper proposes a learning-aided method for fast TTC calculation. The TTC calculation is firstly modeled as transient stability constrained optimal power flow. Then, to reduce the complexity of the TSCOPF model, DBN-based learning-aided transient stability assessment is introduced to surrogate high-dimensional and time-consuming time-domain constraints. In the end, the Jacobian and Hessian matrix of the trained learning-aided model is derived; thereby, nonlinear programming is allowed to solve the learning-aided TSCOPF model efficiently.

The result of the case study demonstrates that the learning-aided model can achieve TSA with higher accuracy and generalization. Moreover, the learning-aided TSCOPF model proposed in this paper can obtain more accurate TTC values than RPF, sensitivity-based, and direct data-driven methods. This is because the proposed method can both take into account the fidelity and efficiency of physics- and data-driven modeling by combining the learning-aided model with the OPF. And compared with the heuristic search of RPF, the OPF model can search the extreme operating point more accurately. On the other hand, due to the use of the learning-aided model to surrogate the time-consuming TSA, it has higher computational efficiency than other physics-driven methods, which means that it can be applied online after sufficient offline training. Besides, the proposed method is not limited to TTC-oriented research. Because of its high compatibility with other static or dynamic models, it can be extended to other index calculations in the power system that require a large amount of computation but require high efficiency. Other advanced machine learning algorithms will be used to achieve better calculation performance in our future work. And, it would also be meaningful to optimize and control the TTC.

Author Contributions: Conceptualization, G.Q. and J.L.; methodology, G.Q. and J.L.; software, J.L.; validation, J.L.; formal analysis, J.L.; resources, Y.L.; data curation, J.L.; writing—original draft preparation, J.L.; writing—review and editing, G.Q. and Y.L.; visualization, J.L.; supervision, Y.L.; project administration, X.S.; funding acquisition, Y.L. and X.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

According to the chain rule, $\bigtriangledown \boldsymbol{H}_s$ in (31) is:

$$\begin{aligned}\partial\Psi_c(\boldsymbol{X})/\partial\boldsymbol{x}_i &= [\mathrm{d}\Psi_c(\boldsymbol{X})/\mathrm{d}M(D_{l-1}(\ldots(M(D_1(\boldsymbol{X})))\ldots))] \times \ldots \times [\mathrm{d}D_1(\boldsymbol{X})/\mathrm{d}\boldsymbol{X}] \times [\mathrm{d}\boldsymbol{X}/\mathrm{d}\boldsymbol{x}_i] \\ &= \boldsymbol{w}_l\, [\mathrm{d}_x M(D_{l-1}(\ldots(M(D_1(\boldsymbol{X})))\ldots)) \times (\boldsymbol{w}_{l-1}\boldsymbol{e}_i)],\end{aligned} \tag{A1}$$

To simplify (A1), the following functions are defined:

$$\boldsymbol{w}_i{}^1 = \boldsymbol{w}^1\boldsymbol{e}_i \tag{A2}$$

$$A_l(\boldsymbol{X}) = \mathrm{d}_{\boldsymbol{x}} M(D_l(\ldots(M(D_1(\boldsymbol{X})))\ldots)) \times \boldsymbol{w}_l \tag{A3}$$

$$\begin{aligned}\textstyle\prod^{l2}{}_{i=l1} A_i(\boldsymbol{X}) &= A_{l1} \times \ldots \times A_{l2}, && \textit{if } l_2 < l_1 < l, \\ &= A_{l2} \times \ldots \times A_{l1}, && \textit{if } l_1 < l_2 < l, \\ &= 1, && \textit{if } l_2 < l \leq l_1, \\ &= A_{l1}, && \textit{if } l_1 = l_2\end{aligned} \tag{A4}$$

where $\mathrm{d}_{\boldsymbol{x}} M(D_l(\ldots(\boldsymbol{X})\ldots))$ in (A3) can be derived from the following matrix formulation:

$$\begin{aligned}\mathrm{d}_{\boldsymbol{x}} M(D_l(\ldots(\boldsymbol{X})\ldots)) = \{\, &[\mathrm{d}_{Dl\_1(\ldots(\boldsymbol{X})\ldots)} M(D_l{}^1(\ldots(\boldsymbol{X})\ldots)), 0, \ldots, 0]; \\ &[0, \mathrm{d}_{Dl\_2(\ldots(\boldsymbol{X})\ldots)} M(D_l{}^2(\ldots(\boldsymbol{X})\ldots)), \ldots, 0]; \\ &\ldots \\ &[0, 0, \ldots, \mathrm{d}_{Dl\_nl(\ldots(\boldsymbol{X})\ldots)} M(D_l{}^{nl}(\ldots(\boldsymbol{X})\ldots))]\},\end{aligned} \tag{A5}$$

$$\begin{aligned}D_l(\ldots(\boldsymbol{X})\ldots) &= [D_l{}^1(\ldots(\boldsymbol{X})\ldots); \ldots; D_l{}^{nl}(\ldots(\boldsymbol{X})\ldots)] \in \mathrm{R}^{nl\times 1}, \\ D_l{}^k(\ldots(\boldsymbol{X})\ldots) &= \boldsymbol{w}_l{}^k M(D_{l\text{-}1}(\ldots(\boldsymbol{X})\ldots)) + b_l{}^k\end{aligned} \tag{A6}$$

Using (A2)~(A4), the *Jacobian* matrix can be simplified to:

$$\partial\Gamma_c/\partial\boldsymbol{x}_i = \boldsymbol{w}_l[\textstyle\prod^2{}_{i=l-1} A_i(\boldsymbol{X})] \times [\mathrm{d}_{\boldsymbol{x}} M(D_1(\boldsymbol{X})) \times \boldsymbol{w}_1{}^i], \tag{A7}$$

Similar to the derivation process of the *Jacobian* matrix, the *Hessian* matrix can be obtained by the following formulations:

$$\begin{aligned}\mathrm{d}_{xj} A_l(\boldsymbol{X}) &= \mathrm{d}^2{}_{\boldsymbol{x},xj} M(D_l(\ldots(\boldsymbol{X})\ldots)) \times \boldsymbol{w}_l \\ &= \{\,[\mathrm{d}^2{}_{\boldsymbol{x},xj} M(D_l{}^1(\ldots(\boldsymbol{X})\ldots)), 0, \ldots, 0]; \\ &\quad [0, \mathrm{d}^2{}_{\boldsymbol{x},xj} M(D_l{}^2(\ldots(\boldsymbol{X})\ldots)), \ldots, 0]; \\ &\quad \ldots \\ &\quad [0, 0, \ldots, \mathrm{d}^2{}_{\boldsymbol{x},xj} M(D_l{}^{nl}(\ldots(\boldsymbol{X})\ldots))]\,\}\end{aligned} \tag{A8}$$

$$\begin{aligned}\mathrm{d}^2{}_{\boldsymbol{x},xj} M(D_l{}^k(\ldots(\boldsymbol{X})\ldots)) = [&\mathrm{d}^2 M(D_l{}^k(\ldots(\boldsymbol{X})\ldots)) / \mathrm{d}(D_l{}^k(\ldots(\boldsymbol{X})\ldots))^2] \times \boldsymbol{w}_l \times \\ &\textstyle\prod^2{}_{i=l-1} A_i(\boldsymbol{X}) \times [\mathrm{d}_{\boldsymbol{x}} M(D_1(\boldsymbol{X})) \times \boldsymbol{w}_1{}^j],\end{aligned} \tag{A9}$$

where, $\boldsymbol{w}_l = \boldsymbol{w}_1{}^i$ if $l = 1$ in (A8) and (A9). And, defined (A10) as follows:

$$\begin{aligned}\Lambda_k &= [\textstyle\prod^{l-1}{}_{i=k+1} A_i(\boldsymbol{X})] \times \mathrm{d}_{xj} A_k(\boldsymbol{X}) \times [\textstyle\prod^1{}_{i=k\text{-}1} A_i(\boldsymbol{X})], \textit{ if } k \geq 2, \\ &= [\textstyle\prod^{l-1}{}_{i=k+1} A_i(\boldsymbol{X})] \times \mathrm{d}_{xj} A_k(\boldsymbol{X}), \textit{ if } k = 1, \\ &= \mathrm{d}_{xj} A_k(\boldsymbol{X}) \times [\textstyle\prod^1{}_{i=k-1} A_i(\boldsymbol{X})], \textit{ if } k = l-1\end{aligned} \tag{A10}$$

Then, the Hessian matrix can be derived as (A11):

$$\partial^2\Gamma_c/(\partial\boldsymbol{x}_i\partial\boldsymbol{x}_j) = \boldsymbol{w}_l[\Sigma^{l-1}{}_{k=1}\Lambda_k] \tag{A11}$$

References

1. Qiu, G.; Liu, Y.; Zhao, J.; Liu, J.; Wang, L.; Liu, T.; Gao, H. Analytic Deep learning-based surrogate model for operational planning with dynamic TTC constraints. *IEEE Trans. Power Syst.* **2020**, *36*, 3507–3519. [CrossRef]
2. Zhang, X.; Santiago, G. Decentralized total transfer capability evaluation using domain decomposition methods. *IEEE Trans. Power Syst.* **2016**, *31*, 3349–3357. [CrossRef]

3. Tang, L.; Sun, W. An automated transient stability constrained optimal power flow based on trajectory sensitivity analysis. *IEEE Trans. Power Syst.* **2017**, *32*, 590–599. [CrossRef]
4. Min, L.; Ali, A. Total transfer capability computation for multi-area power systems. *IEEE Trans. Power Syst.* **2006**, *21*, 1141–1147. [CrossRef]
5. Qiu, G.; Liu, Y.; Liu, J.; Wang, L.; Liu, T.; Gao, H.; Shafqat, J. Surrogate-assisted optimal re-dispatch control for risk-aware regulation of dynamic total transfer capability. *IET Gener. Transm. Distrib.* **2021**, *15*, 1949–1961. [CrossRef]
6. Gan, D.; Robert, J.T.; Ray, D.Z. Stability-constrained optimal power flow. *IEEE Trans. Power Syst.* **2000**, *15*, 535–540. [CrossRef]
7. La Scala, M.; Trovato, M.; Antonelli, C. On-line dynamic preventive control: An algorithm for transient security dispatch. *IEEE Trans. Power Syst.* **1998**, *13*, 601–610. [CrossRef]
8. Yan, X.; Yin, M.; Zhao, Y.; Zhang, R.; David, J.H.; Zhang, Y. Robust dispatch of high wind power-penetrated power systems against transient instability. *IEEE Trans. Power Syst.* **2018**, *33*, 174–186.
9. Yan, X.; Ma, J.; Zhao, Y.; David, J.H. Robust transient stability-constrained optimal power flow with uncertain dynamic loads. *IEEE Trans. Power Syst.* **2017**, *32*, 3415–3426.
10. Alejandro, P.; Claudio, R.F.; Enrique, A.Z.; Jose, M.L. Directional derivative-based transient stability-constrained optimal power flow. *IEEE Trans. Smart Grid.* **2017**, *8*, 1911–1921.
11. Liu, Y.; Zhao, J.; Xu, L.; Liu, T.; Qiu, G.; Liu, J. Online TTC estimation using nonparametric analytics considering wind power integration. *IEEE Trans. Power Syst.* **2019**, *34*, 494–505. [CrossRef]
12. Sun, H.; Zhao, F.; Wang, H.; Wang, K.; Jiang, W.; Guo, Q.; Zhang, B.; Louis, W. Automatic learning of fine operating rules for online power system security control. *IEEE Trans. Neural Netw. Learn. Syst.* **2016**, *27*, 1708–1719. [CrossRef] [PubMed]
13. Zhu, L.; David, J.H.; Lu, C. Hierarchical deep learning machine for power system online transient stability prediction. *IEEE Trans. Power Syst.* **2020**, *35*, 2399–2411. [CrossRef]
14. James, J.Q.Y.; David, J.H.; Albert, Y.S.; Gu, J.; Victor, O.K.L. Intelligent time-adaptive transient stability assessment system. *IEEE Trans. Power Syst.* **2018**, *33*, 1049–1058.
15. Jin, Y. Surrogate-assisted evolutionary computation: Recent advances and future challenges. *ISwarm Evol. Comput.* **2011**, *1*, 61–70. [CrossRef]
16. Jin, Y.; Wang, H.; Tinkle, C.; Guo, D.; Kaisa, M. Data-driven evolutionary optimization: An overview and case studies. *IEEE Trans. Evol. Comput.* **2019**, *23*, 442–458. [CrossRef]
17. Raphael, T.H.; Diana, V.; Anirban, C. Parallel surrogate-assisted global optimization with expensive functions-a survey. *Struct. Multidiscipl. Optim.* **2016**, *54*, 3–13.
18. Cai, X.; Gao, L.; Li, X. Efficient generalized surrogate-assisted evolutionary algorithm for high-dimensional expensive problems. *IEEE Trans. Evol. Comput.* **2020**, *24*, 365–379. [CrossRef]
19. Hamoud, G. Assessment of available transfer capability of transmission systems. *IEEE Trans. Power Syst.* **2000**, *15*, 27–32. [CrossRef]
20. Piyush, W.P.; Sachin, K.; Sinha, A.K. Total transfer capability calculation using modified repeated power flow method. In Proceedings of the 2015 Annual IEEE India Conference (INDICON), New Delhi, India, 17–19 December 2015; IEEE: Piscataway, NJ, USA, 2015.
21. Panagiotis, P.; Theofilos, P.; Andreas, C.; Jovica, M. Measurement based method for online characterization of generator dynamic behaviour in systems with renewable generation. *IEEE Trans. Power Syst.* **2018**, *33*, 6466–6475.
22. Nattawut, P.; Akihiko, Y.; Yoshiki, N.; Verma, S.C. Improved risk-based TTC evaluation with system case partitioning. *Int. J. Electr. Power Energy Syst.* **2013**, *44*, 530–539.
23. Lukmanul, H.; Junji, K.; Yue, Y.; Tomohisa, M.; Yoshifumi, Z.; Naoto, Y.; Yoshihito, N.; Kimihiko, S.; Akira, T. A study on the effect of generation shedding to total transfer capability by means of transient stability constrained optimal power flow. *IEEE Trans. Power Syst.* **2009**, *24*, 347–355.
24. Pandia, J.; Chanan, S. Reliability constrained multi-area adequacy planning using stochastic programming with sample-average approximations. *IEEE Trans. Power Syst.* **2008**, *23*, 504–513.
25. Zhang, C.; Pin, L.; Qin, A.K.; Kay, C.T. Multiobjective deep belief networks ensemble for remaining useful life estimation in prognostics. *IEEE Trans. Neural Netw. Learn. Syst.* **2017**, *28*, 2306–2318. [CrossRef] [PubMed]
26. Jaber, A.; Yushi, M.; Toshifumi, I. Assessment and optimization methods for microgrid with multiple VSG units. *IEEE Trans. Smart Grid.* **2018**, *9*, 1462–1471.
27. Huang, H.; Chung, C.Y. Coordinated damping control design for DFIG-based wind generation considering power output variation. *IEEE Trans. Power Syst.* **2012**, *27*, 1916–1925. [CrossRef]
28. Antonio, L.B.; Glauco, N.T.; Djalma, M.F. Simultaneous tuning of power system damping controllers using genetic algorithms. *IEEE Trans. Power Syst.* **2000**, *15*, 163–169.
29. Andrei, P.; Iman, S.; Chris, M. Interior Point Differential Dynamic Programming. *IEEE Trans. Control Syst. Tech.* **2021**, *29*, 2720–2727.

MDPI
St. Alban-Anlage 66
4052 Basel
Switzerland
Tel. +41 61 683 77 34
Fax +41 61 302 89 18
www.mdpi.com

MDPI Books Editorial Office
E-mail: books@mdpi.com
www.mdpi.com/journal/books

MDPI

Academic Open
Access Publishing

www.mdpi.com

ISBN 978-3-0365-7647-3