

Article

Electrical Power Prediction Using RS-485 Power Meter: A PSO-Optimized XGBoost Approach for Industrial Smart Manufacturing

Mulki Indana Zulfa ¹, Adhe Akbar Azanni ¹, Muhammad Syaiful Aliim ¹, Ari Fadli ², Waleed Ali ^{3,*} and Talal A. A. Abdullah ⁴

¹ Department of Electrical Engineering, Jenderal Soedirman University, Purbalingga 53371, Indonesia; mulki_indanazulfa@unsoed.ac.id (M.I.Z.)

² Department of Computer Science, Gajah Mada University, Yogyakarta 55281, Indonesia

³ Information Technology Department, Faculty of Computing and Information Technology, King Abdulaziz University, Rabigh 21911, Saudi Arabia

⁴ Interdisciplinary Research Center for Finance and Digital Economy, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia; talal.abdullah@kfupm.edu.sa

* Correspondence: waabdullah@kau.edu.sa

Abstract

Accurate electrical power prediction is increasingly critical in industrial smart manufacturing environments, where energy fluctuations and demand variability pose significant operational challenges under the industry 4.0 paradigm. Existing approaches often rely on simulated or secondary data and lack integration with industrial-grade communication protocols, limiting their practical applicability. Incorporating machine learning with real-time data collection is essential for progressing industrial predictive monitoring. This research presents a framework to forecast electrical power usage by utilizing the RS-485 protocol to enhance smart manufacturing processes. The dataset used was obtained from a power meter, recorded over a period of 135 min, resulting in 3100 data. Three learning methods—Random Forest, Extra Trees, and XGBoost—were analyzed, with XGBoost being further refined through PSO for tuning hyperparameters. The models were trained on datasets that included voltage, current, frequency, and power factor, and their effectiveness was evaluated using time-based predictions, standard metrics, and error distributions through cross-validation. The findings illustrate that the PSO-XGBoost consistently surpasses the default XGBoost baseline R^2 of 0.5746, achieving MAE of 0.14 W, RMSE of 0.21 W, and R^2 of 0.8355, representing improvements of 41.67% in MAE, 38.24% in RMSE, and 45.40% in R^2 . The RS-485 protocol enables seamless integration with existing industrial infrastructure, supporting anomaly detection and energy optimization aligned with Industry 4.0 interoperability objectives.

Keywords: machine learning; power prediction; RS-485; XGBoost; PSO; hyperparameter optimization; smart manufacturing



Academic Editors: Valentina De Simone, Valentina Di Pasquale, Raffaele Iannone and Fabio Fruggiero

Received: 21 January 2026

Revised: 26 February 2026

Accepted: 28 February 2026

Published: 3 March 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

The rapid pace of development in Industry 4.0 has enabled new operational methodologies across industrial sectors such as the automotive, electronics, chemicals, and energy sectors, as new processing techniques are made possible through the integration of cyber-physical systems, the Internet of Things (IoT), and advanced data processing [1]. The

value and frequency of the operational dataset drive the profound changes in productivity, resource sustainability, and the reliability of operational systems. This has led to the emergence of smart manufacturing as one of the most significant applications, with the use of real-time data and predictive analytics being employed to improve manufacturing processes and reduce unplanned system downtimes [2].

The electric power industry is developing systems that incorporate renewables and advanced grid technologies. Such systems need energy management and grid stability, requiring more accurate models for forecasting power [3]. Forecasting energy consumption provides utility companies with knowledge to improve load balancing, cut costs, and make the grid system more reliable [4]. In industrial and commercial settings, energy waste needs to be reduced, and real-time analytics and monitoring support demand-responsive management, especially with considerable energy/power fluctuations [5]. In this respect, Industry 4.0 has underscored the importance of processed sensor data obtained through various communication standards, including RS-485. These data acquisition systems obtain accurate data from power meters, facilitating advanced data processing. In this paper, we foster the creation of machine learning models to forecast electrical power consumption from primary data obtained from a custom RS-485 power meter. Random Forest (RF), Extra Trees (ET), and XGBoost are some of the algorithms we use, while PSO is applied to improve their performance through optimization of the parameters.

Even though machine learning has made progress in estimating power, there are still several challenges that need to be studied further. A lot of research does not match up with how individuals really utilize power, because it is based on simulations or secondary data [6]. Kahraman et al. [7] emphasized the necessity of recognizing idea drift, concentrating exclusively on deep learning models. Nonetheless, due to the absence of RS-485-based source data or the implementation of typical ensemble methods, the majority of the results remain predominantly theoretical and lack practical applicability. Olcay et al. [8] compared LSTM and RF for solar energy forecasting; however, their study was confined to renewable datasets, excluded industrial communication protocols, and was inapplicable to conventional industrial power systems. Chahine [9] showed that tree-based algorithms yield accurate outcomes; nevertheless, their emphasis was on fault classification instead of power prediction utilizing industrial-grade primary data. Noppakant and Plangklang [10] utilized RS-485 for load monitoring in tiny buildings but restricted this to demand-side control, even without predictive machine learning.

Chen et al. [11] showed SVR's ability to understand nonlinear relationships by applying it to understand demand response baselines for offices. However, this study did not incorporate any industrial data acquisition and hyperparameter tuning. Simaiya et al. [12] applied LightGBM with transfer learning for smart grid short-term load predictions where they forecasted energy quite successfully; however, they did not use industrial-grade communication systems such as RS-485 for real-time data collection. Dong et al. [13] showed reasonable accuracy of pattern recognition with a K-Nearest Neighbors-based deep learning model for load forecasting; however, they had a lot of unaddressed integration issues for industrial environments. These examples illustrate the potential of adopting a diverse set of machine learning techniques that include SVR, KNN, and LightGBM for power forecasting, but most of them do not provide structured hyperparameter tuning and integration with industrial-grade primary data acquisition systems such as RS-485. Wu et al. [14] used PSO hyperparameter tuning for XGBoost for power forecasting using publicly available electricity market data. Our study aims to bridge this gap by applying PSO-optimized ensemble techniques to primary data from RS-485 industrial power meters, thereby integrating algorithmic optimization with real industrial systems.

This paper presents several key contributions to the field of electrical power prediction. First, it demonstrates the use of a custom RS-485 power meter to obtain genuine primary data, as it collects data relevant to industrial processes. Second, it analyzes data from five common ML algorithms in comparison to XGBoost enhanced by PSO-based hyperparameter tuning and demonstrates the predictive power of XGBoost. The analysis takes into account primary data [15] and is guided by the competing interests of accuracy, robustness, efficiency, and resource allocation. Third, it presents a framework to seamlessly combine RS-485-based data acquisition and optimized machine learning, thereby providing a pathway for energy management systems in industrial settings. This work addresses the knowledge gap by combining PSO-optimized ensemble machine learning models with RS-485-based industrial primary data and demonstrating notable improvements compared to the default setup. Unlike Wu et al. [14], who validated PSO-XGBoost on publicly available data from public datasets, this work combines industrial-level hardware data acquisition, intelligent hyperparameter tuning, and real-time deployment. We summarize similar relevant research in Table 1.

Table 1. Similar works.

Research	Methods	Key Finding	Gap
Kahraman et al. [7]	Deep learning with concept drift detection.	Identified importance of dynamic modeling for real-time energy consumption	No RS-485 integration; focused solely on deep learning
Olcay et al. [8]	LSTM, Random Forest	LSTM outperformed RF for solar energy forecasting	Limited to renewable datasets; no industrial communication protocol
Chahine [9]	Tree-based algorithms	High accuracy in fault classification using ensemble methods	Focused on fault detection, not power prediction; no primary data acquisition
Noppakant and Plangklang [10]	Rule-based demand response	Effective load monitoring for small buildings	No predictive ML; limited to demand-side control only
Chen et al. [11]	SVR	SVR captures nonlinear demand response baselines	No industrial data acquisition; no hyperparameter optimization
Simaiya et al. [12]	LightGBM + Transfer Learning	Effective short-term load prediction	No RS-485 or industrial-grade real-time data collection
Dong et al. [13]	KNN-based deep learning	Reasonable pattern recognition for load forecasting	Unresolved integration challenges for industrial environments
Wu et al. [14]	PSO-XGBoost	PSO-XGBoost effectively optimizes power price forecasting	Validated on public data only; no hardware-level data acquisition
This study	PSO-XGBoost + RF + ET	PSO-XGBoost achieves MAE 0.14 W, RMSE 0.21 W, R^2 0.8355	Bridges algorithmic optimization with real-time industrial primary data acquisition

2. Materials and Methods

This study develops a robust methodology to leverage high-resolution primary data from a custom RS-485 power meter for electrical power prediction using advanced ensemble machine learning algorithms. The methodology is divided into five key phases: data acquisition, data preprocessing, model development, framework integration, and performance evaluation.

2.1. Data Acquisition

The server acts as the main data processing and storage unit, as all electricity meter readings are booked into a central database, as depicted in Figure 1. The power meter measures voltage, current, power factor, and active power. Real-time data communication from the power meter to a central data logger is facilitated through the reliable, industrial-grade RS-485 communication protocol [16]. Electricity meter data are transmitted using the RS-485 communication protocol and then converted into IP-based signals via an RS-485 converter, enabling integration into the factory computer network [17]. Once converted, data from each distribution substation are forwarded to the network via a Cisco switch (Cisco Systems, Inc., San Jose, CA, USA) and connected to the central office server. The IP-formatted data are then processed using Node-RED [18], which performs initial data handling between the RS-485 converter and the MySQL database. Specifically, Node-RED performs the following preprocessing steps: data parsing, timestamp assignment, and data formatting to being written into the MySQL database. In addition, the availability of real-time data is an advantage, because it enables the development of prediction systems that can be directly applied to actual monitoring systems and time series-based systems [19].

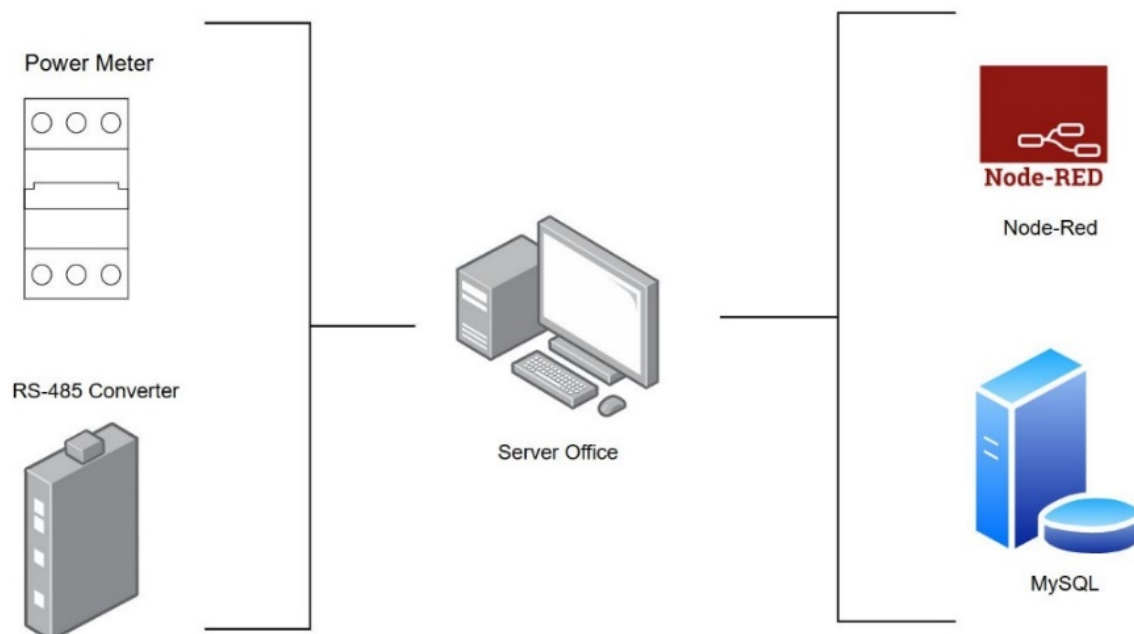


Figure 1. The power meter monitoring global architecture used Node-RED and the MySQL database for data handling.

Figure 2 provides an RS-485 wiring diagram. The system starts with the power meter PM5350 (U1), which monitors and collects data on voltage, current, and power and calculates frequency. This information is sent through an RS-485 module (U2), which processes the data and prepares it into the differential signal format (A+, B−) for proper transmission. For the RS-485 data stream to be accepted by current industrial communication standards, it has to be converted to Ethernet, allowing communication with industrial servers and data processors. While the communication architecture follows standard industrial practice, its integration with a real-time machine learning prediction pipeline in an active manufacturing facility without requiring modification of the existing infrastructure represents the practical deployability contribution of this framework.

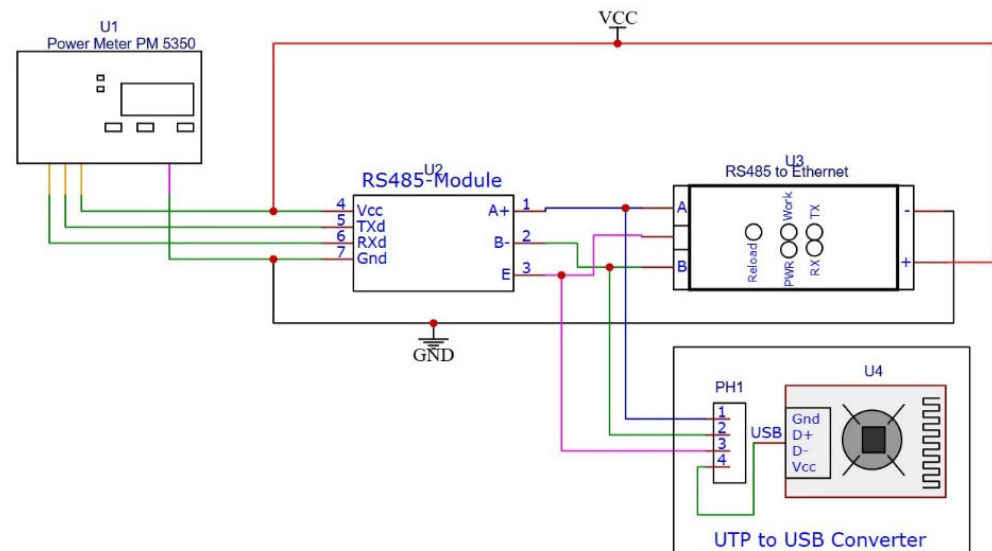


Figure 2. The RS-485 wiring diagram used the physical connections between power meter and Ethernet converter for data transmission.

2.2. Data Preprocessing

The dataset used was taken from one of the power meters on one day in January 2025. Considering that the proof of concept was carried out in a production environment, the measurements only took 135 min and produced 3100 data records. The dataset described in Table 2 needed to be cleaned, checked for data types, ensured for consistency, and prepared so the model could use it optimally [20]. This stage includes column conversion, statistical exploration to understand data distribution, feature transformation and selection, and spitting the dataset into two parts, features and target. This process ensures that the model to be tested on the data has never been seen before in order to measure its capabilities objectively [21]. It is important to note that the training and testing processes use different data for chronological train-split (shuffle = false), so the model does not learn the same data during evaluation to avoid overfitting and ensure good generalization [22].

Table 2. Dataset characteristics.

Variable	Description	Unit/Format	Example
id	Serial number of each data stored in the database	Integer	1, 2, 3
date	Indicates when the data was recorded	yyyy-mm-dd hh:mm:ss	2025-01-28 07:41:46
meterid	Indicates from which power meter was obtained	Integer	meter_2
power	Measured active electrical power value	Watt (W)	20.58
voltage	Measured electrical voltage value	Volt (V)	380.78
current	Measured electrical current value	Ampere (A)	32.03
freq	Measured electrical frequency value	Hertz (Hz)	49.98
pwrfactor	Measured electrical power factor value	[0–1]	0.97

2.3. Proposed Prediction of Electrical Power Consumption

Considering the challenges of predicting electrical power in industrial settings, this study analyzes various ensemble learning techniques, culminating in the tailored optimized gradient boosting solution. The framework includes three baseline tree-based ensemble methods—RF, ET and XGBoost—as well as a PSO-optimized XGBoost model. There are two reasons for choosing those ensemble techniques: (1) they are all tree-based ensembles

and can capture the nonlinear characteristics of the data generated by industrial power consumption; (2) there is a stepwise increase in the level of complexity of the algorithms, from bagging with optimal splits (RF) to randomized split thresholds for further reduction of variance (ET) to sequential gradient boosting with regularization (XGBoost). This provides a clear framework to examine the effect of learning algorithms on the predictive ability for real industrial data. Meanwhile, tree ensemble methods were configured as follows: Random Forest used $n_estimators=100$, $max_features = 'sqrt'$, and bootstrap sampling enabled; Extra Trees used $n_estimators = 100$, $max_features = 'sqrt'$, and bootstrap disabled to maximize randomization; default XGBoost used $learning_rate = 0.3$, $max_depth = 6$, and $n_estimators = 100$ with reg: squarederror as the objective function.

- Random Forest Regression

Random Forest (RF) is an ensemble learning method that constructs multiple decision trees during training and outputs the average prediction across all trees [23]. The algorithm employs bootstrap aggregating (bagging) and random feature selection at each split to reduce overfitting, as shown in Equation (1). Variable $\hat{y}(x)$ is the ensemble prediction, B is the total number of decision trees, and $T_b(x)$ is the prediction of the b -th decision tree trained. The averaging mechanism is a cornerstone of RF's variance reduction ability: by combining predictions from many decorrelated trees trained on bootstrapped samples, the ensemble mitigates the effects of errors of individual trees.

$$\bar{y}_{(x)} = \frac{1}{B} \sum_{b=1}^B T_b(x) \quad (1)$$

- Extra Trees Regression

Extra Trees (ET) takes Random Forest and incorporates more randomness when creating trees [24]. Instead of using specific samples of training data and splitting thresholds based on the best value, Extra Trees uses the entire training dataset and picks split points at random. This reduces the on training time a lot while keeping the accuracy of the predictions. This is an important benefit for industrial applications that need to be done quickly.

- XGBoost Regression

XGBoost is a specific implementation of gradient boosting that takes a pretty good approach to build trees sequentially [25], where a new tree corrects errors from the previous tree. XGBoost uses an additive approach with a built-in regularization to avoid overfitting [26]. As can be seen in Equation (2), the prediction is the sum of the predictions of all the trees, where K is the total number of trees and f_k is the k -th tree. This mechanism simultaneously reduces bias and controls model complexity, which is particularly valuable when working with industrial datasets of limited temporal span.

$$\bar{y}_i = \sum_{k=1}^K f_k(x_i) \quad (2)$$

- PSO-optimized XGBoost

A swarm of candidate solutions is created in the hyperparameter search space using the population-based metaheuristic algorithm known as PSO [27]. Each individual in the swarm adjusts its position based on its own experience and the knowledge of the entire swarm. The updates made in position and velocity are stated in Equations (3) and (4). W is the inertia weight, v_i is the particle velocity, r_1 and r_2 are independent random numbers uniformly, c_1 and c_2 are acceleration coefficient, p_i represents the individual best position of the particle, and g refers to the best position of the entire swarm. The

PSO algorithm was implemented using the pyswarm library with a swarm size of 20 particles over 30 iterations, using inertia weight $w = 0.5$ and cognitive and social acceleration coefficients $c_1 = c_2 = 0.5$. At each iteration, each particle's candidate hyperparameter set was evaluated by training an XGBoost model and computing RMSE via 5-fold cross-validation on the training set, implemented through scikit-learn's `cross_val_score` with `neg_mean_squared_error` scoring.

$$v_i(t+1) = w \cdot v_i(t) + c_1 \cdot r_1 \cdot (p_i - x_i(t)) + c_2 \cdot r_2 \cdot (g - x_i(t)) \quad (3)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (4)$$

2.4. Proposed Framework of Integrating RS-485-Based Data Acquisition with Machine Learning for Industrial Energy Management

The proposed study builds a three-tier structure combining RS-485-based data acquisition and machine learning for the management of industrial energy, presented in Figure 3. At the data acquisition layer, an RS-485 power meter records real-time measurements (voltage, current, power, frequency, and power factor) for the measurements to be converted to IP-based protocols via an RS-485 converter for incorporation into the network [28]. At the processing layer, Node-RED is used for data filtering and formatting, and the preprocessed data are stored in a MySQL database to ensure accessibility in a structured manner [29]. At the prediction layer, coupled with PSO-based hyperparameter optimization, XGBoost and other ensemble methods synthesize a prediction of future industrial power consumption and assist energy management and operational decision making for industrial purposes [30]. This modular structure facilitates the addition of other industrial IoT systems using standard communication protocols, and it serves as a tangible example for the use of predictive models in an industrial environment to support load forecasting, predictive maintenance, and energy optimization, among others [31].

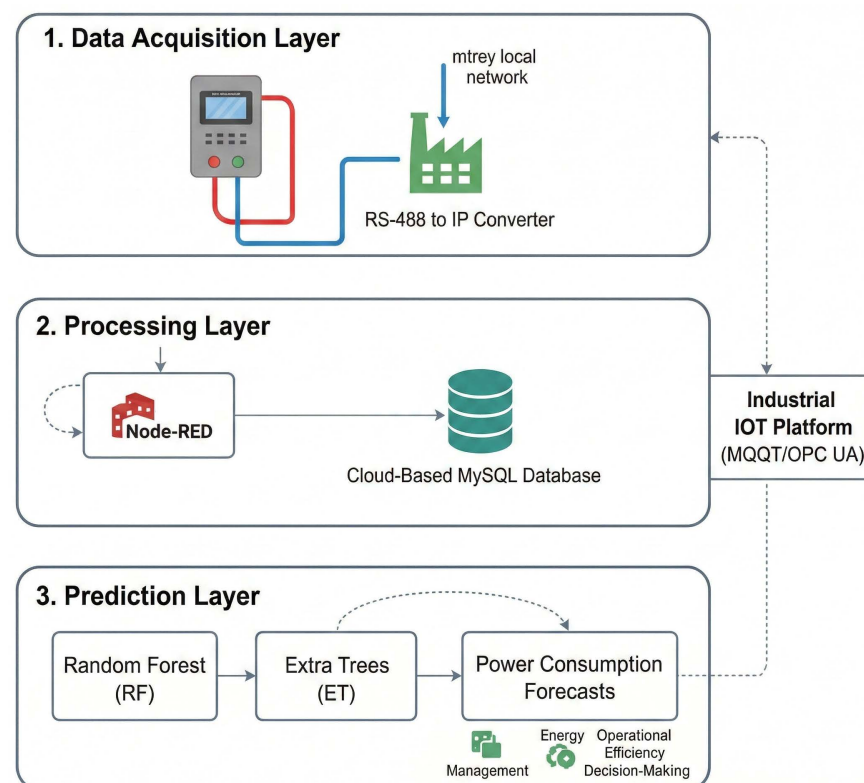


Figure 3. RS-485 framework for machine learning integration.

2.5. Performance Evaluation

To comprehensively assess the predictive performance of the proposed models, three standard regression metrics were employed: MAE, RMSE, and coefficient of determination (R^2) [32]. MAE measures the average magnitude of prediction errors without considering their direction, following Equation (5). The n variable is the number of samples, y_i is the actual value, and $\bar{y}_i$ is the predicted value. Lower MAE values indicate better prediction accuracy.

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \bar{y}_i| \quad (5)$$

RMSE quantifies the standard deviation of prediction errors, giving more weight to larger errors through squaring, following Equation (6). RMSE serves as a critical complementary metric that reveals the model's sensitivity to extreme prediction failures beyond what average error measures alone can capture. RMSE is particularly sensitive to outliers and provides error magnitude in the same units as the target variable.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \bar{y}_i)^2} \quad (6)$$

The coefficient of determination (R^2) measures the proportion of variance in the dependent variable that is predictable from the independent variables, as shown in Equation (7). The R^2 ranges from 0 to 1, with values closer to 1 indicating superior model fit.

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \bar{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (7)$$

3. Results

3.1. Experiment Setup

Python version 3.8 was used for implementation of ET and RF in Scikit-learn and gradient boosting in the XGBoost library v3.2.0. Data collection was carried out using custom written Python scripts that interfaced with an RS-485 power meter via Modbus and stored data in a MySQL database for optimized querying and storage scalability. PSO optimization for tuning eight XGBoost hyperparameters was done using the pyswarm library and five-fold cross-validation. The prediction model was served using the Flask framework to allow real-time communications [33] and was integrated with the company's ERP system over the local network to provide operational teams and management with centralized system alerts and predictive insights. It should be noted that the prediction task in this study is formulated as a simultaneous regression problem, where active power at each time step is predicted from electrical parameters measured at the same instant: voltage, current, frequency, and power factor.

3.2. Data Analysis

Figure 4 shows an analysis of the electrical power consumption dataset from four different angles. It allows for the identification of characteristics, relational attributes, and time attributes that could assist in future modeling activities. In Figure 4a, the box plot data distributions in Power vs. Current data measure construction verify that average power consumption is 18.53 W with a 1.26 W standard deviation, while current average is 28.39 A with a standard deviation of 2.16 A. Power (IQR of approximately 2.5 W) is less variable compared to current (IQR of approximately 5 A). In Figure 4b, a matrix of correlations is created to show the linear correlation of data for electrical parameters, which is the premise of the correlation equations for each set. Figure 4c shows the Power vs. Current chart. The

slope of the line is 0.57, meaning that, in an average case, an increase of 0.57 watts can be expected for each increase in power of 1 amp, caused by the voltage and power factor. Figure 4d time series displays dynamics voltage in steady industrial grid conditions.

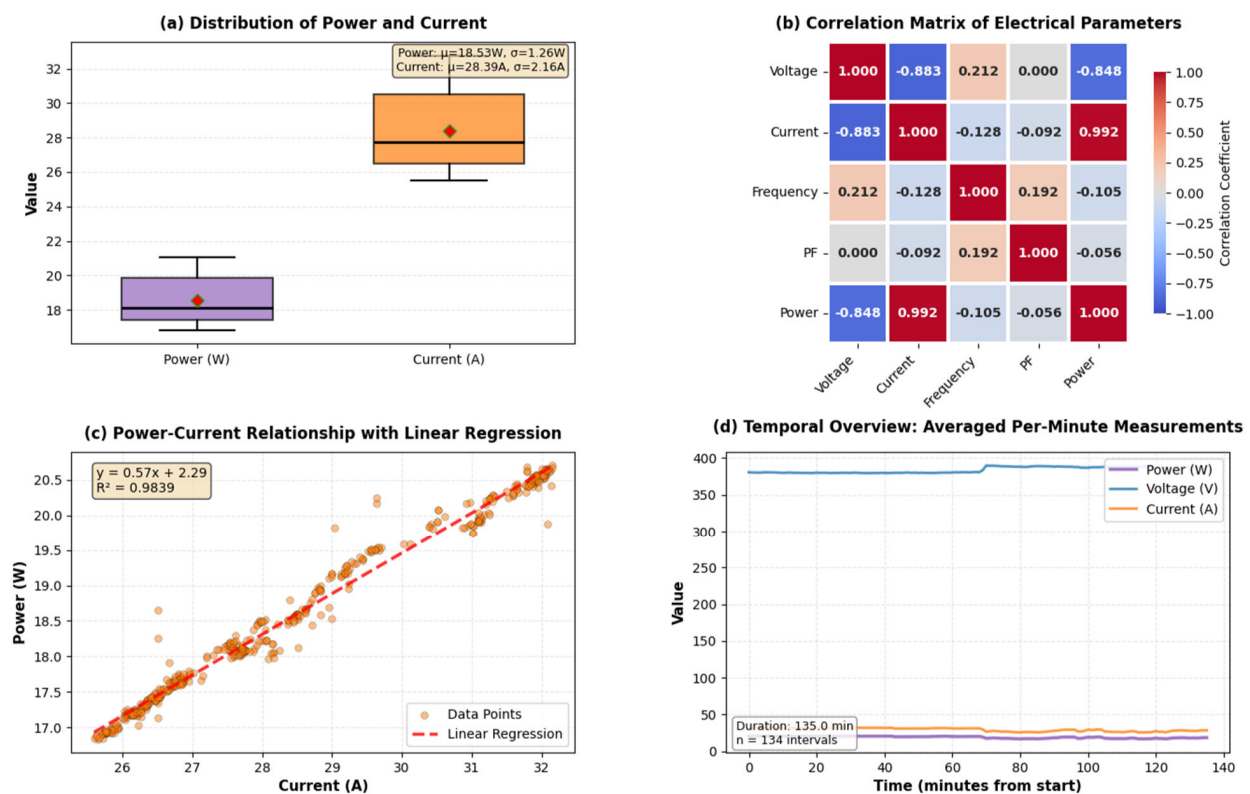


Figure 4. Electrical power consumption dataset.

3.3. Predictive Performance of Models for Electrical Power Consumption

3.3.1. Comparison of Models Performances

Figure 5a shows RF predictions over 650 test samples. The x-axis represents the sample index of the test set sorted by ascending actual power value (rather than temporal order) to facilitate visual comparison of prediction accuracy across models, while the y-axis represents active power in watts. The model has MAE = 0.33 W, RMSE = 0.40 W, and $R^2 = 0.4045$. This means RF accounts for only 40.5% of power variance. The predicted line is quite zigzagged which means it has likely overfitted to the noise of the training data and not the signal. The moderate R^2 signal shows that, with the right hyperparameter optimization, it could be made much better. For ET, its performance in Figure 5b is evidently better, with MAE = 0.27 W, RMSE = 0.33 W, and $R^2 = 0.5915$. This shows 18.2% less MAE and 15.2% more R^2 when compared to Random Forest, which is a notable improvement. The reason ET performs better than RF is that it uses a random splitting threshold rather than optimal splits at each node, and that is more random, which serves as a sort of regularization to help the model generalize better. ET decreases segregation for a little more bias.

The default hyperparameter settings of the XGBoost model in Figure 5c show further improvement in performance with scores of MAE = 0.24 W, RMSE = 0.34 W, and $R^2 = 0.5746$. Although the RMSE of the current XGBoost model is slightly higher than the ET model at 0.34 W compared to 0.33 W, and the 11.1% MAE reduction provides evidence of improved performance, especially in ordinary predictions with no severe errors. XGBoost outperforms ET and RF due to important differences in their underlying algorithms. Whereas ET and RF train independent trees and then average the predictions from the trees, XGBoost uses sequential boosting, in which each new tree is specifically designed to target the residual errors from the current ensemble.

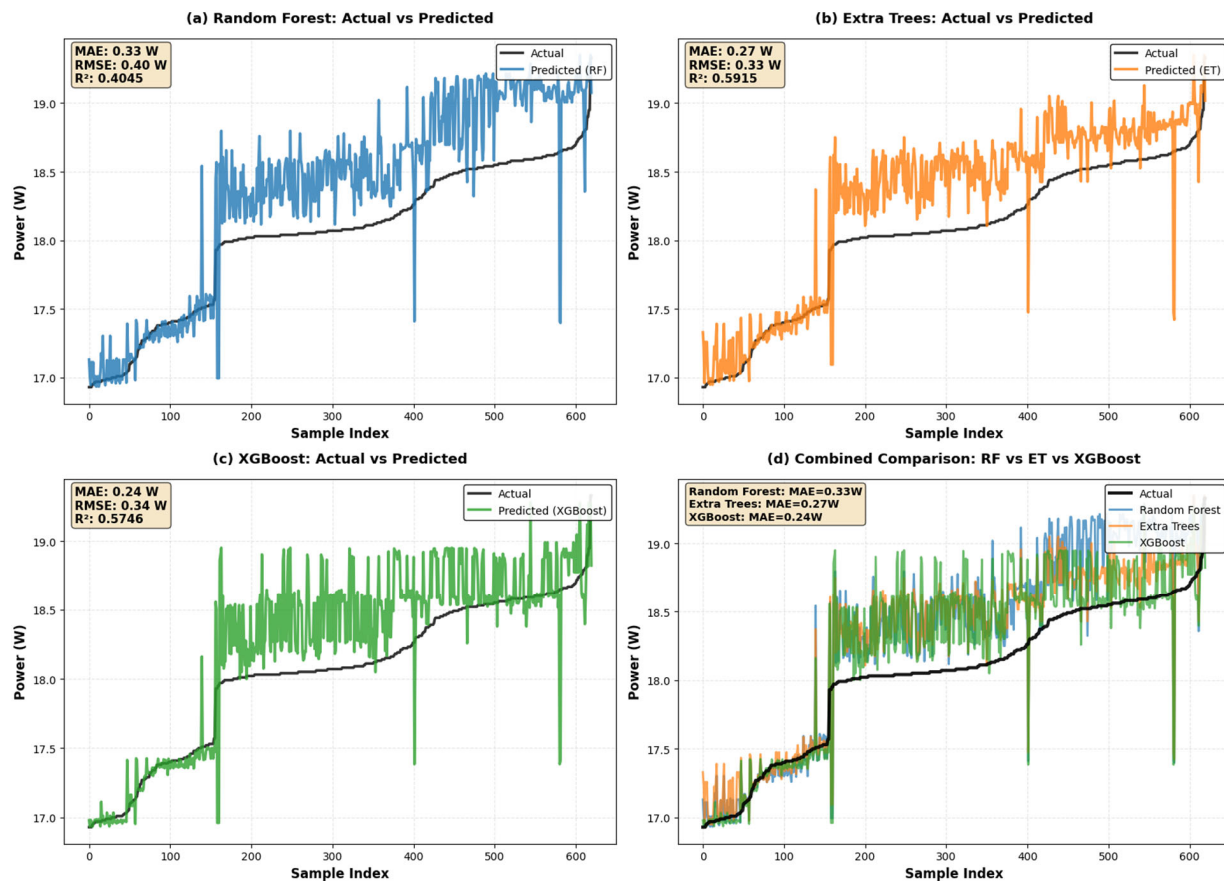


Figure 5. Actual and power prediction comparison.

The learning rate of 0.3 and six levels of depth and 100 estimators of XGBoost demonstrate a good combination of models to allow for hyperparameter adjustments to be made. The consolidated view of all models in Figure 5d presents all predicted power trajectories in a single graph. The primary insights from the baseline comparison indicate that XGBoost's gradient boosting framework consistently exceeds the performance of bagging-based methods, which justifies its selection as a basis for further optimization.

3.3.2. Performance of PSO-Optimized XGBoost

Figure 6 shows the success of PSO-XGBoost hyperparameter tuning using two distinct analyses: PSO convergence behavior and error distribution. Figure 6a shows the PSO convergence trajectory for three different metrics for all 30 iterations. The Mean Absolute Error (MAE) denoted by the blue line shows a significant drop of 0.20 W in the first five iterations, from 0.35 W to 0.15 W. This convergence shows PSO's parallel search approach, as 20 particles are assessing different hyperparameter sets, sharing information about the hyperparameter sets the swarm should investigate further. The R^2 score illustrates positive development, increasing from roughly 0.80 at iteration 0 to 0.84 at iteration 5. It then levels off at 0.8355. The overall increase from PSO optimization has R^2 values increasing the power of explained variance from 80 to 83.6, decreasing the unexplained variance by 17.8% from the initial default position.

The convergence characteristics highlight two distinct features of the PSO optimization process: (1) The early period of improvement is from 0 to 5 iterations. This period shows how effective PSO is by demonstrating how quickly it is able to find the promising hyperparameters for the swarm by doing a parallel search and sharing the information. (2) The long plateau phase from iterations 5 to 30 signifies a complete true exploitation of a given status. This is the gray box for the final optimized results (0.14 W MAE, 0.21 W RMSE, and R^2 0.8355), which should be considered for a new benchmark performance

setting. It shows 41.7%, 38.2%, and 45.4% improvement, respectively. PSO is able to find a hyperparameter set that definitively enhances predictive accuracy and stays within the parameters of what is computationally feasible for actual industrial use.

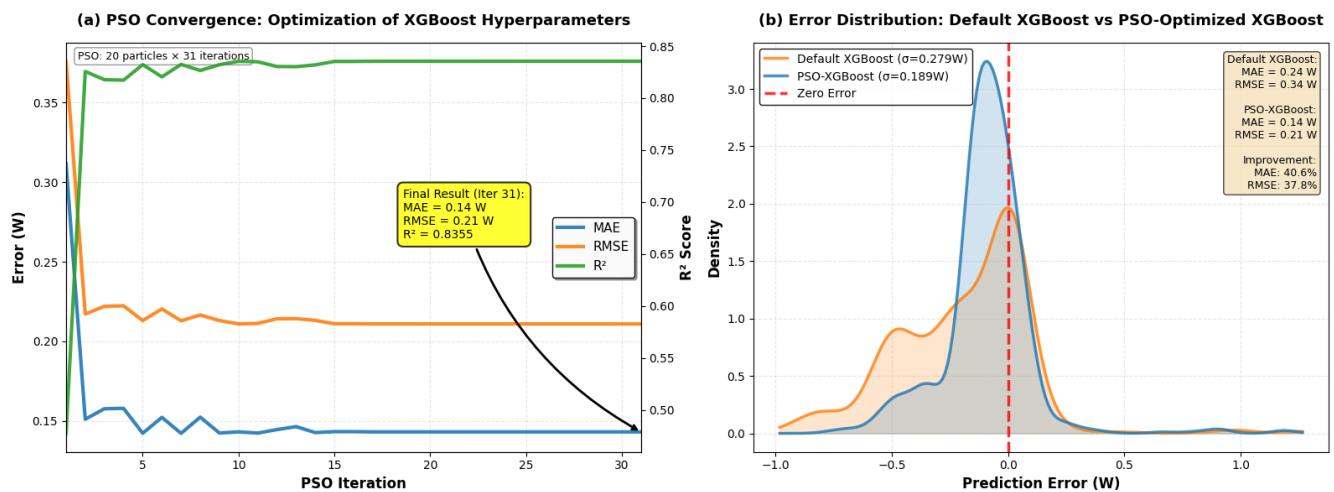


Figure 6. Performance improvement with PSO optimization. (a) PSO convergence. (b) Error distribution comparison.

Figure 6b indicates distributions of error for default XGBoost and PSO-optimized XGBoost. The error distribution for default XGBoost (the orange curve) is widespread, centered around negative values, with a standard deviation of $\sigma = 0.279 W$, indicating a notable range of variability of the predictions. The PSO-optimized XGBoost distribution, on the other hand, has error metrics that are significantly better in two important respects: (1) a 32.3% reduction in error variability, as evidenced by the more tightly concentrated distribution around zero, and (2) a more defined peak that is centered around zero, which indicates that the optimized predictions are, in fact, more closely scattered to the values that are true given no systematic forecasting bias.

Table 3 presents the optimal hyperparameter values identified by PSO alongside the default values and performance impact interpretations. The learning_rate decreased dramatically from 0.3 to 0.0847, implementing slower but more stable gradient descent steps that enable finer-grained error minimization without overshooting optimal configurations. The max_depth maintained its default value of 6, suggesting this tree depth already represents an optimal balance between model complexity and specific prediction task.

Table 3. Optimal hyperparameters identified by PSO.

Hyperparameter	Default Value	PSO-Optimized Value	Impact on Performance
learning_rate	0.3	0.0847	Slower more stable learning
max_depth	6	6	Maintained optimal complexity
n_estimators	100	487	Increased ensemble robustness
subsample	1	0.78	Enhanced generalization
colsample_bytree	1	0.85	Reduced feature correlation
gamma	0	1.23	Conservative splitting
reg_alpha (L1)	0	0.34	Moderate L1 regularization
reg_lambda (L2)	1	0.56	Balanced L2 penalty

Table 4 demonstrates that PSO optimization yields substantial quantitative improvements over the default XGBoost baseline: a 41.67% reduction in MAE (from 0.24 W to 0.14 W), a 38.24% reduction in RMSE (from 0.34 W to 0.21 W), and a 45.40% improvement in R^2 (from 0.5746 to 0.8355). These gains confirm the effectiveness of PSO-based hyperparameter tuning in enhancing predictive accuracy for industrial power consumption prediction. This significant improvement validates the compound benefits of: (1) algorithmic sophistication, (2) built-in regularization, and (3) systematic hyperparameter optimization (PSO).

Table 4. Performance comparison PSO-optimized XGBoost.

Model	MAE (W)	RMSE (W)	R^2	Improvement
Random Forest	0.33	0.40	0.4045	-
Extra Trees	0.27	0.33	0.5915	-
XGBoost	0.24	0.34	0.5746	Baseline
PSO-XGBoost	0.14	0.21	0.8355	+41.67% MAE +38.24% RMSE +45.40% R^2

3.3.3. Error Distribution Analysis

Figure 7 presents an analysis of the error distributions of the four prediction models: RF, ET, XGBoost, and PSO-XGBoost. In Figure 7a, RMSE = 0.40 W and standard deviation = 0.2719 W for RF show that the predictions can vary quite a bit, and the error distribution for RF is almost symmetric. This shows that there is a small bias regarding underprediction, as the average error is -0.2952 W. The RMSE is 0.40 W, with standard deviation of 0.2719 W, and ET has an RMSE of 0.33, which is better than RF, but the average error is -0.2431 , so there is still a small bias toward underprediction. However, ET is a more consistent predictor with fewer large errors. Yes, I have moved on to the next step.

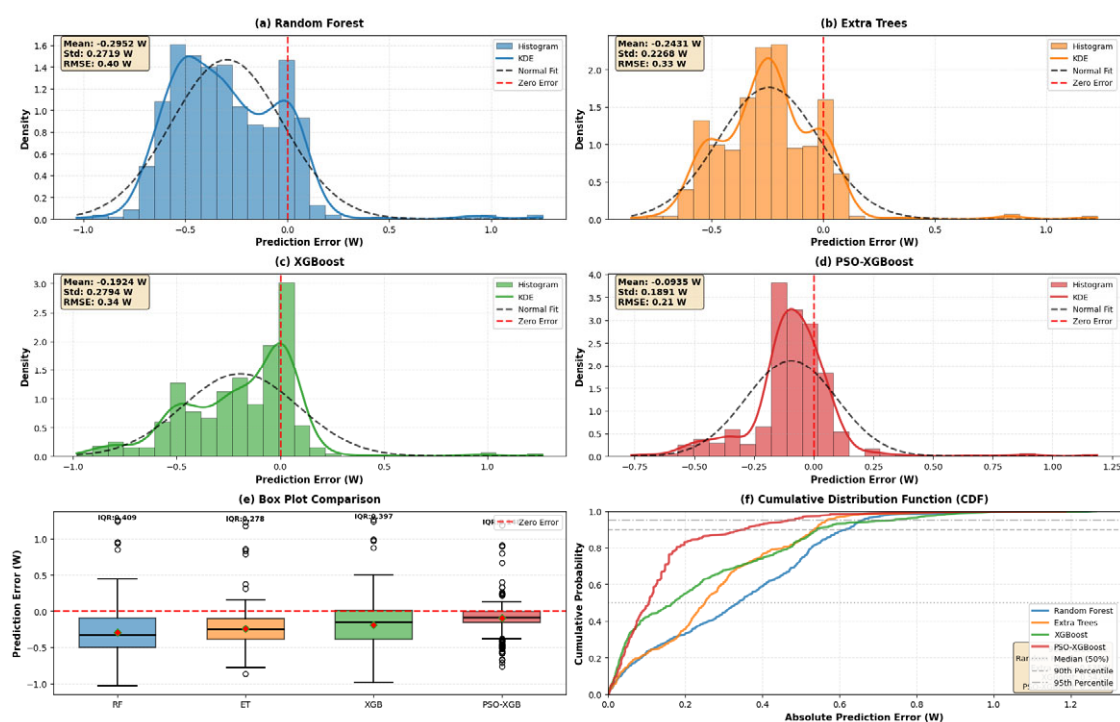


Figure 7. Error distribution analysis: comprehensive comparison.

Figure 7c shows XGBoost achieving additional improvements, with an average error of -0.1924 W and standard deviation of 0.2794 W. The error distribution is almost normal, and the model has a better metric of error balance. Figure 7d presents PSO-XGBoost as the highest achiever, resulting in an average error of -0.0935 W and the lowest standard deviation of 0.1891 W and the best RMSE of 0.21 W. Figure 7e shows the statistical summary provided by the box plots. PSO-XGBoost having the least Interquartile Range (IQR) suggests that its predictions are the most concentrated. Lastly, Figure 7f presents cumulative distribution functions (CDFs) for the absolute prediction error.

3.4. Comparison with Other Common Machine Learning

Table 5 summarizes the total quantitative performance metrics of the seven models evaluated in the study, while Figure 8 illustrates PSO-XGBoost alongside the traditional ML algorithms SVR, LightGBM, and KNN. In SVR's case, predictive performance was reasonable, reflected by a MAE of 0.20 W, RMSE of 0.34 W, and an R^2 of 0.5649 . Figure 8a shows that SVR accounted for 56.5% of the variation in power consumption. Among the additional algorithms, LightGBM, featured in Figure 8b, surpassed SVR, having a MAE of 0.25 W, RMSE of 0.32 W, and an R^2 of 0.6174 . LightGBM is the best-performing baseline of the additional algorithms and second best overall, after default XGBoost.

Table 5. Performance comparison of PSO-XGBoost with other ML models.

Model	MAE (W)	RMSE (W)	R^2
XGBoost	0.24	0.34	0.5746
PSO-XGBoost	0.14	0.21	0.8355
Random Forest	0.33	0.40	0.4045
Extra Trees	0.27	0.33	0.5915
SVR	0.20	0.34	0.5649
LightGBM	0.25	0.32	0.6174
KNN	0.40	0.52	-0.0135

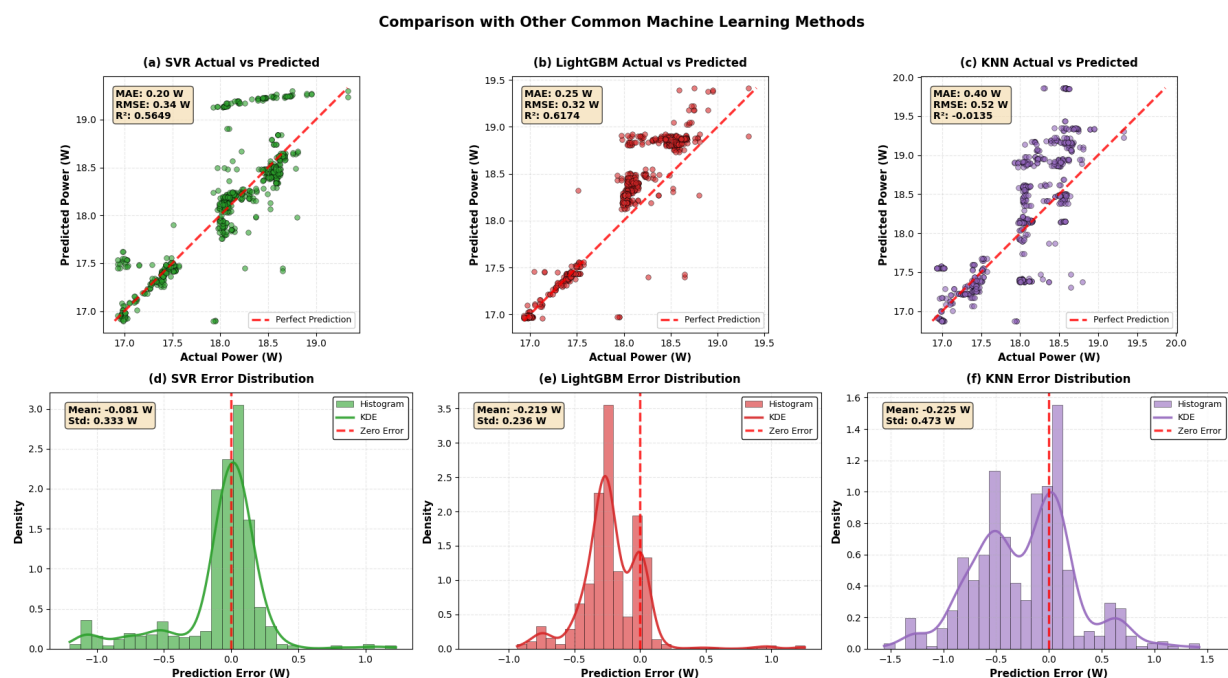


Figure 8. Performance comparison with common ML methods.

Figure 8c shows that KNN demonstrates the least performance, with a MAE of 0.40 W, RMSE of 0.52 W, and a negative R^2 of -0.0135 . This negative R^2 implies that KNN's predictions are less useful than predicting the mean of the power consumption values. Among the additional methods, LightGBM has an error distribution with the least mean of -0.219 W and smallest standard deviation of $\sigma = 0.236$ W, which is a measure of error dispersion. In the least favorable error distribution of the three methods, KNN displayed a mean of -0.225 W and the greatest variability in standard deviation with $\sigma = 0.473$ W, which is in Figure 8f. Overall, PSO-XGBoost is compared to SVR, LightGBM, and KNN, which scales to an overall evaluation of PSO-XGBoost's power consumption modeling.

4. Discussion

The value of R^2 of 0.8355 is the highest value reported in the literature for power prediction using ensemble-based techniques. For example, Simaiya et al. [12] reported an accuracy comparable to that of using LightGBM with transfer learning on smart grid datasets, but their methodology is limited, as they use public data with no industrial-grade primary data collection. Similarly, Wu et al. [14] also used PSO-XGBoost for forecasting electricity prices but with public market data and reported results comparable to our baseline XGBoost. However, we went further by confirming PSO-XGBoost using real-time RS-485 primary data in a manufacturing setting. In our work, the SVR model achieved R^2 of 0.5649, which is comparable to Chen et al. [11], who also used SVR for the baseline demand response for office buildings and acknowledged the model's ability to capture nonlinearities, though it is also highly dependent on the tuning of the hyperparameter. The R^2 of -0.0135 of KNN was very low and is consistent with the findings of Dong et al. [13], who reported the difficulties in using KNN in heterogeneous industrial systems. The findings from these studies further confirm that gradient boosting with hyperparameter optimization is the best approach for industrial power prediction problems with moderate nonlinearity and variability in operations. The Flask-based REST API architecture enables seamless integration with ERP systems for real-time alerts, while the modular three-tier design allows independent scaling of each layer. This capability is especially important for smart manufacturing, which is largely concerned with the elimination of downtime and the optimization of energy use.

This study is primarily limited by the time frame and single site data collection. One manufacturing site means limited operational variability and even more restricted longitudinal load variation. Additionally, the current version of this framework is not able to self-adapt to the predictive concepts from equipment changes or changes in the production schedule, which means predictive performance for long-term deployments will gradually decline.

5. Conclusions

This paper describes the effective framework for predicting electrical power consumption using an RS-485 power meter integrated with a machine learning algorithm optimized by PSO. The experimental results show that the PSO-optimized XGBoost model outperforms all baseline models, achieving a MAE of 0.14 W, RMSE of 0.21 W, and R^2 of 0.8355. Compared to the default XGBoost, these results show 41.67% MAE improvement, 38.24% RMSE improvement, and 45.40% improvement in R^2 . The PSO optimization process proved to be successful by having optimum hyperparameters within the first five iterations and remaining within realistic constraints for industrial use. However, this study has a number of shortcomings that need to be addressed. The dataset is based on a single manufacturing site and one operational period. Therefore, the model was not able to learn the seasonal and operational changes that would occur over the long term within and

across manufacturing sites. Furthermore, the framework currently only delivers predictive analytics and therefore does not provide recommendations on the timing of maintenance or the procurement of energy. Future efforts should explore ways to integrate the framework with digital twin systems and a hybrid deep learning ensemble approach to transform the framework from a predictive monitoring system into a fully autonomous industrial energy management system.

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

Funding: This project was funded by the Deanship of Scientific Research (DSR) at King Abdulaziz University, Jeddah, Saudi Arabia under grant no. (IPP: 107-830-2025).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author due to company sensitive data.

Acknowledgments: The authors, therefore, acknowledge with thanks DSR for technical and financial support. We also thank PT Indocement Indonesia for their support in validating this framework in an industrial setting.

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

Abbreviations

The following abbreviations are used in this manuscript:

ML	Machine Learning
RF	Random Forest
ET	Extra Trees
LightGBM	Light Gradient Boosting Machine
SVR	Support Vector Regression
KNN	K-Nearest Neighbors
PSO	Particle Swarm Optimization
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
RMSE	Root Mean Squared Error
ERP	Enterprise Resource Planning
KDE	Kernel Density Estimate
IQR	Interquartile Range

References

1. Kiasari, M.; Ghaffari, M.; Aly, H.H. A Comprehensive Review of the Current Status of Smart Grid Technologies for Renewable Energies Integration and Future Trends: The Role of Machine Learning and Energy Storage Systems. *Energies* **2024**, *17*, 4128. [\[CrossRef\]](#)
2. Ren, L.; Dong, J.; Liu, S.; Zhang, L.; Wang, L. Embodied Intelligence Toward Future Smart Manufacturing in the Era of AI Foundation Model. *IEEE/ASME Trans. Mechatron.* **2025**, *30*, 2632–2642. [\[CrossRef\]](#)
3. Arumugham, V.; Ghanimi, H.M.A.; Pustokhin, D.A.; Pustokhina, I.V.; Ponnamm, V.S.; Alharbi, M.; Krishnamoorthy, P.; Sengan, S. An Artificial-Intelligence-Based Renewable Energy Prediction Program for Demand-Side Management in Smart Grids. *Sustainability* **2023**, *15*, 5453. [\[CrossRef\]](#)

4. Attiwal, P.; Indora, S. A Comprehensive Review of Modern Methods for Load Prediction in the Smart Grid. *Recent Patents Eng.* **2024**, *18*, 40–51. [\[CrossRef\]](#)
5. He, C.-L. Real-time power management in virtualized data centers using predictive analytics. *Int. J. Cloud Comput. Database Manag.* **2024**, *5*, 21–24. [\[CrossRef\]](#)
6. da Silva, R.F.; Casanova, H.; Orgerie, A.-C.; Tanaka, R.; Deelman, E.; Suter, F. Characterizing, Modeling, and Accurately Simulating Power and Energy Consumption of I/O-intensive Scientific Workflows. *J. Comput. Sci.* **2020**, *44*, 101157. [\[CrossRef\]](#)
7. Kahraman, A.; Kantardzic, M.; Kotan, M. Dynamic Modeling with Integrated Concept Drift Detection for Predicting Real-Time Energy Consumption of Industrial Machines. *IEEE Access* **2022**, *10*, 104622–104635. [\[CrossRef\]](#)
8. Olcay, K.; Tunca, S.G.; Özgür, M.A. Forecasting and Performance Analysis of Energy Production in Solar Power Plants Using Long Short-Term Memory (LSTM) and Random Forest Models. *IEEE Access* **2024**, *12*, 103299–103312. [\[CrossRef\]](#)
9. Chahine, K. Tree-Based Algorithms and Incremental Feature Optimization for Fault Detection and Diagnosis in Photovoltaic Systems. *Eng* **2025**, *6*, 20. [\[CrossRef\]](#)
10. Noppakant, A.; Plangklang, B. Improving Energy Management through Demand Response Programs for Low-Rise University Buildings. *Sustainability* **2022**, *14*, 14233. [\[CrossRef\]](#)
11. Chen, Y.; Xu, P.; Chu, Y.; Li, W.; Wu, Y.; Ni, L.; Bao, Y.; Wang, K. Short-term electrical load forecasting using the Support Vector Regression (SVR) model to calculate the demand response baseline for office buildings. *Appl. Energy* **2017**, *195*, 659–670. [\[CrossRef\]](#)
12. Simaiya, S.; Dahiya, M.; Tomar, S.; Faujdar, N.; Sharma, Y.K.; Nuristani, N. A transfer learning-based hybrid model with LightGBM for smart grid short-term energy load prediction. *Energy Explor. Exploit.* **2024**, *42*, 1853–1876. [\[CrossRef\]](#)
13. Dong, Y.; Ma, X.; Fu, T. Electrical load forecasting: A deep learning approach based on K-nearest neighbors. *Appl. Soft Comput.* **2021**, *99*, 106900. [\[CrossRef\]](#)
14. Wu, K.; Chai, Y.; Zhang, X.; Zhao, X. Research on Power Price Forecasting Based on PSO-XGBoost. *Electronics* **2022**, *11*, 3763. [\[CrossRef\]](#)
15. Tsokeng, T.L.B.; Phan, R.L.; Atindana, C.; Anastasopoulos, P.C. Testing the Efficacy of Hyperparameter Optimization Algorithms in Short-Term Load Forecasting. *arXiv* **2024**. [\[CrossRef\]](#)
16. Bailey, B. Power system protection-power monitoring, data logging, remote interrogation system. In *Conference Record, Proceedings of the Industrial and Commercial Power Systems Technical Conference*; IEEE: Piscataway, NJ, USA, 2022; pp. 109–112. [\[CrossRef\]](#)
17. Sung, G.-M.; Lee, C.-T.; You, Z.-L.; Yu, C.-P. Ethernet Packet-to-RS485 Data Transfer Bridge Application-Specific Integrated Circuit Incorporating Transmission Control Protocol and Static Random-Access Memory. *IEEE Access* **2024**, *12*, 106084–106095. [\[CrossRef\]](#)
18. Oluwaleye, S.; Idachaba, F. Design And Development of a Remote Monitoring Agent For Energy Service Companies In The Telecommunication Industry. *F1000Research* **2022**, *11*, 1106. [\[CrossRef\]](#)
19. Kratsch, W.; Manderscheid, J.; Röglinger, M.; Seyfried, J. Machine Learning in Business Process Monitoring: A Comparison of Deep Learning and Classical Approaches Used for Outcome Prediction. *Bus. Inf. Syst. Eng.* **2021**, *63*, 261–276. [\[CrossRef\]](#)
20. Conti, M.; Donadel, D.; Turrin, F. A Survey on Industrial Control System Testbeds and Datasets for Security Research. *IEEE Commun. Surv. Tutorials* **2021**, *23*, 2248–2294. [\[CrossRef\]](#)
21. Güney, H. Preprocessing Impact Analysis for Machine Learning-Based Network Intrusion Detection. *Sak. Univ. J. Comput. Inf. Sci.* **2023**, *6*, 67–79.
22. Kernbach, J.M.; Staartjes, V.E. Foundations of Machine Learning-Based Clinical Prediction Modeling: Part II—Generalization and Overfitting. In *Proceedings of the Machine Learning in Clinical Neuroscience*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 15–21. [\[CrossRef\]](#)
23. Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5–32. [\[CrossRef\]](#)
24. Geurts, P.; Ernst, D.; Wehenkel, L. Extremely randomized trees. *Mach. Learn.* **2006**, *63*, 3–42. [\[CrossRef\]](#)
25. Sahin, E.K. Assessing the predictive capability of ensemble tree methods for landslide susceptibility mapping using XGBoost, gradient boosting machine, and random forest. *SN Appl. Sci.* **2020**, *2*, 1308. [\[CrossRef\]](#)
26. Chen, T.; Guestrin, C. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*; ACM: New York, NY, USA, 2016. [\[CrossRef\]](#)
27. Kennedy, J.; Eberhart, R. Particle Swarm Optimization. In *Proceedings of the IEEE International Conference on Neural Networks*; IEEE: Piscataway, NJ, USA, 1995; pp. 1942–1948.
28. Mary, J.S.N.; Raj, A.P.A.; Remy, A.I.; Bharathy, S.S. Design of Digital Meters for Industries using Data Acquisition System. *IOP Conf. Ser. Mater. Sci. Eng.* **2022**, *1258*, 012047. [\[CrossRef\]](#)
29. Mountasser, I.; Ouhbi, B.; Hdioud, F.; Frikh, B. Semantic-based Big Data integration framework using scalable distributed ontology matching strategy. *Distrib. Parallel Databases* **2021**, *39*, 891–937. [\[CrossRef\]](#)
30. Dai, Z.; Huang, W. Improving energy management practices through accurate building energy consumption prediction: Analyzing the performance of LightGBM, RF, and XGBoost models with advanced optimization strategies. *Electr. Eng.* **2025**, *107*, 12583–12605. [\[CrossRef\]](#)

31. Janjua, J.I.; Ahmad, R.; Abbas, S.; Mohammed, A.S.; Khan, M.S.; Daud, A.; Abbas, T.; Khan, M.A. Enhancing smart grid electricity prediction with the fusion of intelligent modeling and XAI integration. *Int. J. Adv. Appl. Sci.* **2024**, *11*, 230–248. [[CrossRef](#)]
32. Plevris, V.P.; Solorzano, G.S.; Bakas, N.B.; Ben Seghier, M. Investigation of performance metrics in regression analysis and machine learning-based prediction models. In *Proceedings of the 8th European Congress on Computational Methods in Applied Sciences and Engineering*; CIMNE: Barcelona, Spain, 2022. [[CrossRef](#)]
33. Amakrishna, G.; Ahesh, D. Real-Time Object Detection using Yolov9c and Flask Web Application. *Int. J. Eng. Technol. Manag. Sci.* **2025**, *9*, 699–703. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.