

Article

Fault Detection and Identification with Kernel Principal Component Analysis and Long Short-Term Memory Artificial Neural Network Combined Method

Nahid Jafari ¹  and António M. Lopes ^{2,*} 

¹ Faculty of Electrical Engineering, Shahid Bahonar University of Kerman, Kerman 76169-13439, Iran; jafari_nahid@eng.uk.ac.ir

² LAETA/INEGI, Faculty of Engineering, University of Porto, 4200-465 Porto, Portugal

* Correspondence: aml@fe.up.pt

Abstract: A new fault detection and identification approach is proposed. The kernel principal component analysis (KPCA) is first applied to the data for reducing dimensionality, and the occurrence of faults is determined by means of two statistical indices, T^2 and Q . The K-means clustering algorithm is then adopted to analyze the data and perform clustering, according to the type of fault. Finally, the type of fault is determined using a long short-term memory (LSTM) neural network. The performance of the proposed technique is compared with the principal component analysis (PCA) method in early detecting malfunctions on a continuous stirred tank reactor (CSTR) system. Up to 10 sensor faults and other system degradation conditions are considered. The performance of the LSTM neural network is compared with three other machine learning techniques, namely the support vector machine (SVM), K-nearest neighbors (KNN) algorithm, and decision trees, in determining the type of fault. The results indicate the superior performance of the suggested methodology in both early fault detection and fault identification.

Keywords: fault detection and identification; kernel principal component analysis; artificial neural network

MSC: 62M45; 62H30; 91C20; 94C12



Citation: Jafari, N.; Lopes, A.M. Fault Detection and Identification with Kernel Principal Component Analysis and Long Short-Term Memory Artificial Neural Network Combined Method. *Axioms* **2023**, *12*, 583. <https://doi.org/10.3390/axioms12060583>

Academic Editor: Simeon Reich

Received: 22 April 2023

Revised: 8 June 2023

Accepted: 9 June 2023

Published: 12 June 2023



Copyright: © 2023 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) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Fault detection and identification play a critical role in various fields, contributing to the overall safety, reliability, and efficiency of systems and processes. It is crucial across different domains, such as manufacturing, chemical processing, power generation, oil and gas distribution, transportation and automotive systems, energy and utilities, aerospace, defense, smart infrastructures, environmental monitoring, and medical devices and healthcare.

In modern industrial systems, fault detection and identification is a key requirement. Because control systems are becoming more complex and data dimensionality is growing, a fault can result in catastrophic harm. Indeed, in the worst circumstances, system failure may result in human injuries, as well as damage with significant economic impact, concerning production items, devices, and equipment. Control engineers are therefore focused on researching effective fault detection and identification techniques. Improving speed and efficiency, optimizing the management of systems and their maintenance, increasing the reliability and health of systems, maximizing production, and minimizing repair time, are additional reasons to address fault detection and identification.

Ensuring the system's continuously safe or, at least, acceptable operation requires appropriate fault detection and isolation mechanisms, which are able to distinguish between non-faulty and faulty system behavior and find the location and nature of the fault in case some abnormal condition is signaled. Often, controller reconfiguration is required in response to the fault [1,2].

Several methods for detecting faults have been proposed. Finding or creating methods exhibiting high accuracy and speed for dealing with a system's faults is the most crucial issue. Indeed, after a fault occurs, there is not much time to identify it, and thus, immediate actions are required to minimize the damage. In light of this, if the method chosen for fault detection and identification is not reliable, then incorrect actions may result, and the fault may spread to additional layers of the system. Even though the fault's primary cause may be unknown, it is crucial that it can be determined through diagnosis. The sort of fault that occurred must also be identified after the issue has been located. To prevent the fault from spreading to higher levels, the system's component where the problem occurred can be temporarily removed by identifying the type of fault [3].

A fault is defined by the International Federation of Automatic Control (IFAC) Technical Committee as an unapproved deviation of at least one characteristic or system parameter from the conditions that are considered acceptable/normal/standard. Such a problem can affect a single process, set of sensors, or set of actuators [1,2].

In general, three categories of faults have been taken into account in earlier studies [1]:

- (1) Process or component fault: Process faults arise when a system's components behave negatively, affecting the dynamics of the system.
- (2) Actuator fault: An actuator fault is a discrepancy between the actuator's input command and its actual output.
- (3) Sensor fault: Sensor faults result in discrepancies between the measured and real values of the system's variables [1,4].

This article looks at approaches that are currently available for diagnosing and identifying faults in control systems, combining them to provide a new technique for accurate diagnosis and fault-type identification. In the devised approach, kernel principal component analysis (KPCA) is first used to minimize the dimensions of the data collected from the system. This streamlines computations and expedites the fault-finding procedure. Then, the presence of a fault in the system is detected using statistical indices, which are assigned threshold values in such a way that whenever the required statistical indices surpass some upper limits, a warning is sent, denoting that a fault has occurred. The K-means algorithm is adopted for data clustering as a step to determine the type of fault. The data are separated into three groups, and clustering is carried out for each category in accordance with the variation range of the variables. The type of fault is then determined using a long short-term memory (LSTM) artificial neural network, meaning a recurrent neural network with extremely high accuracy. The labeled data in the K-means algorithm is used to train the LSTM. The proposed technique is tested for detecting malfunctions on a continuous stirred tank reactor (CSTR) system when considering up to 10 sensor faults and other system abnormal conditions. The effectiveness of the method is compared with the principal component analysis (PCA) approach for fault detection, and with the support vector machine (SVM), K-nearest neighbors (KNN), and decision trees algorithms, for determining the type of fault. The results show the superior performance of the proposed methodology in both fault detection and fault identification.

The paper is structured as follows. Section 2 reviews prior studies on fault detection and identification. Section 3 introduces the proposed approach. Section 4 presents the CSTR system used as a test bed. Section 5 reports several simulation results to illustrate the effectiveness of the new method. Finally, Section 6 presents the main conclusions.

2. Literature Review

There are two primary categories of fault detection and identification approaches. These are statistical methods and machine learning techniques. A statistical method for fault detection and isolation utilizing inverse dynamic models for robot arms and partial least squares (PLS) was provided in [5]. The PLS is a linear process control technique used to locate and monitor industrial processes. The PCA employing a serial model structure, often known as serial PCA (SPCA), as a new linear-nonlinear statistical method for nonlinear process monitoring was reported in [6]. A machine learning approach based

on the SVM for fault detection was employed in [7]. By categorizing the data linked to the fault and the data connected to the system's typical operation period, the SVM could detect faults. This approach was thought to be a simple way to identify sensor failures. Using convolutional neural networks (CNN), a highly precise modular multilevel converter (MMC) circuit monitoring system for early failure detection and identification was proposed in [8]. Reference [9] described a technique using Bayesian networks to identify sensor and process faults, as well as faults involving several sensors or processes. A new hybrid system based on the Hilbert–Huang (HH) transform and the adaptive neuro-fuzzy inference system (ANFIS) with optimal parameters was proposed in [10]. As a detection index for the KNN-based fault detection approach, which could separate several sensor faults, a new separation index using KNN distance decomposition was provided in [11]. Reference [12] presented the development of two independent reduced kernel partial least squares (IRKPLS) regression models for fault detection in large-scale nonlinear and uncertain systems. The use of a weighted kernel independent component analysis (WKICA) based Gaussian mixed model (GMM) for monitoring and fault identification in nonlinear and non-Gaussian processes was proposed in [13]. The probabilities of KICA were estimated using GMM for the first time in the WKICA approach. A model for fault identification using a variational Bayesian Gaussian mixture model with canonical correlation analysis (VBGMM-CCA) was provided in [14]. In Reference [15], a method for fault detection and identification was suggested, which combined artificial neural networks (ANN) and the wavelet transform (WT) multi-loop analysis methodology. A novel method for fault identification was provided in [16], which was based on KPCA and SVM.

Motivated by the above discussion, the present paper proposes a novel method that integrates KPCA, K-means, and LSTM neural networks for detecting and identifying faults in multi-process systems. The method introduces several unique aspects that differentiate it from existing approaches and offers improved fault detection and identification performance.

The main contributions of this paper are:

1. The KPCA is performed to decrease the dimension of the original data set while detecting the existence of potential faults. Thus, in subsequent fault identification steps, computational burden and transmission energy consumption is reduced, which is very important in wireless sensor networks.
2. In the reduced data space, the K-means is used for clustering data into different groups and detecting faults using statistics. By using clustering, faults in different processes can be detected, which is an advantage typically overlooked in traditional approaches.
3. An LSTM network is trained in order to identify faults by reconstruction. The LSTM excels at capturing temporal dependencies in sequential data, identifying fault patterns that unfold over time. This temporal modeling capability enhances fault identification and enables the detection of complex and dynamic fault scenarios.
4. Simulations demonstrate the effectiveness of the method in the detection and identification of faults. Thus, based on measured data, the method can identify crashed sensors and actuators and components' misbehavior.

By combining KPCA for fault detection, K-means for data preparation, and LSTM for fault identification, the proposed approach offers a unique and comprehensive solution to fault detection and identification. It surpasses the limitations of individual methods by integrating nonlinear feature extraction, clustering-based data preparation, and temporal modeling. This integration leads to improved fault detection and identification performance, enabling the system to handle complex fault dynamics, adapt to different fault types, and achieve accurate fault identification. To the best of the authors' knowledge, the work proposed herein is the first to effectively combine those three algorithms together for the actual detection and identification of faults.

3. The Proposed Method

The new hybrid technique for detecting and identifying faults is illustrated in Figure 1. The KPCA is initially used for fault diagnosis, which may trigger subsequent fault identi-

fication. Thus, once a fault is detected, the data are tagged using K-means clustering, as a preparation for the next stage of fault identification, based on their embedded inter-distances. Finally, an LSTM neural network determines the type of fault. Thus, the method consists of three fundamental steps:

- The KPCA for fault detection.
- The K-means clustering for processing faulty data.
- The LSTM neural network for determining the fault type.

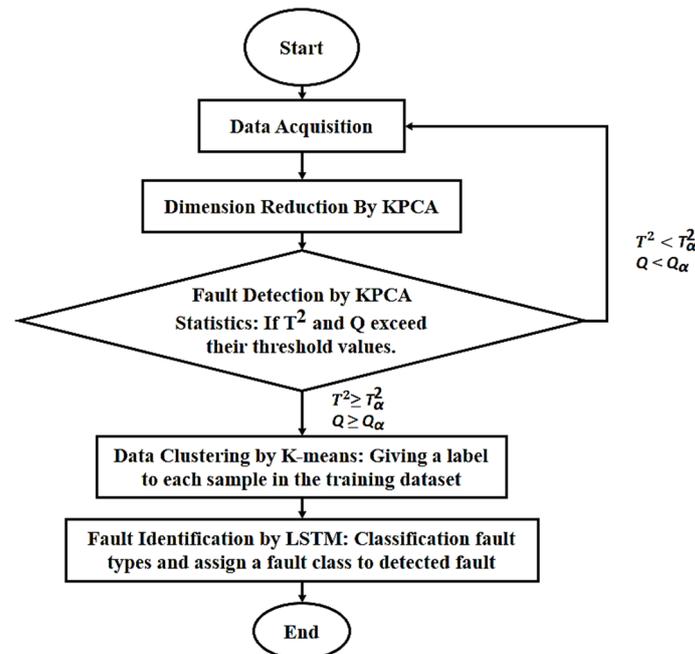


Figure 1. Overview of the proposed method.

These steps are described in the next subsections.

3.1. Fault Detection by KPCA

This step results in data dimensionality reduction in addition to reliable and fast fault detection in nonlinear data. The KPCA is a generalization of the PCA method and one of the most cutting-edge approaches for leveraging kernel functions to monitor nonlinear systems [17,18]. The rationale of KPCA is to extract the nonlinear principal components from the PCA decomposition in the feature space after first mapping the original input data to a high-dimensional defined space using a nonlinear function. In other words, let us consider that the data consists of N observations, $X_k = [u_k^T \ y_k^T]^T, k = 1, \dots, N$, where u and y are the input and output of the process, respectively [18]. The data X_k are normalized, yielding \hat{X}_k , so that their mean is zero and their variance is equal to one. In the PCA, features are extracted only in a linear space. Therefore, we must first use the nonlinear mapping $\Phi(\cdot)$ to represent the data from the original nonlinear input space onto a linear feature one, F , assuming that $\sum_{k=1}^N \Phi(\hat{X}_k) = 0$ is used [18].

The PCA seeks to solve the eigenvalue problem in the covariance matrix in the linear space F as given by [18]:

$$C^F = \frac{1}{N} \sum_{k=1}^N \Phi(\hat{X}_k) \Phi(\hat{X}_k)^T, \quad C^F w = \lambda w \tag{1}$$

where C^F is the sample covariance in the F space, w denotes the eigenvector, and λ is the eigenvalue. To avoid the explicit use of $\Phi(\cdot)$, the kernel functions K are defined in the form [18]:

$$K(X_i, X_j) \triangleq K_{ij} = \langle \Phi(X_i), \Phi(X_j) \rangle \tag{2}$$

$$\langle \Phi(\hat{X}_k), C^F w \rangle = \lambda \langle \Phi(\hat{X}_k), w \rangle, \tag{3}$$

where $w = \langle v, \Phi(\hat{X}_k) \rangle$ and:

$$\hat{K} v = N \lambda v \tag{4}$$

where v is the eigenvector and \hat{K} is defined as [18]:

$$\hat{K} = K - 1_N K - K 1_N + 1_N K 1_N, \tag{5}$$

where $1_N \in R^{N \times N}$ and $(1_N)_{ij} = \frac{1}{N}$. Then, using the matrix \hat{K} obtained from Equation (4), the SVD decomposition is performed [18]:

$$\frac{\hat{K}}{N} = S \Lambda S^T \tag{6}$$

where S contains N eigenvectors of v , and Λ contains N eigenvalues of λ . Next, only r main components that contain 99% of the variances remain and are placed in the matrix $S_r \in R^{N \times r}$ such that [18]:

$$T \triangleq [t_k] = S_r^T \hat{K} \in R^{r \times N} \tag{7}$$

Each test data X_k^{test} is normalized and converted into $X_k^{\hat{test}}$, and then it is mapped to the F space according to [18]:

$$K_k^{test} = K(X_k^{\hat{test}}, \hat{X}_j) \tag{8}$$

where \hat{X}_j is the training data from $j = 1, \dots, N$ and K_k^{test} is normalized according to:

$$X_k^{\hat{test}} = K_k^{test} - 1_N^{test} K - K_k^{test} 1_N + 1_N^{test} K 1_N \tag{9}$$

and, finally:

$$t_k^{test} = S_r^T (K_k^{\hat{test}})^T \tag{10}$$

To identify the fault by KPCA, two statistical indices T^2 and Q are generally used [18]. The values of these two parameters are obtained by:

$$T^2 = (t^{test})^T \Lambda_r^{-1} t^{test}, \tag{11}$$

$$Q = \left\| S^T (K_k^{\hat{test}})^T \right\|^2 - (t^{test})^T t^{test}.$$

A threshold value is defined for each index. If T^2 and Q exceed their threshold values; then it means that there is a fault in the system.

The threshold values are defined as [18]:

$$T_\alpha^2 = \frac{r(m-1)}{m-r} F_{r,m-r,\alpha}, \tag{12}$$

$$Q_\alpha = \theta_1 \left[\frac{C_\alpha h_0 \sqrt{2\theta_2}}{\theta_1} + 1 + \frac{\theta_2 h_0 (h_0 - 1)}{\theta_1^2} \right]^{\frac{1}{h_0}},$$

where r is the number of remaining principal components, $F_{r,m-r,\alpha}$ is the value of the F distribution corresponding to a significance level α , with degrees of freedom r and $m - r$ for numerator and denominator, respectively, $\theta_i = \sum_{j=r+1}^n \lambda_j^i$, ($i = 1, 2, 3$), $h_0 = 1 - \frac{2\theta_1 \theta_3}{3\theta_2^2}$ and $C_\alpha = 100(1 - \alpha)$ [17]. In this paper, the two statistical indices T^2 and Q are calculated, and then, by determining T_α^2 and Q_α , the occurrence of a fault is detected [18].

3.2. Faulty Data Labeling Using the K-Means Technique for Clustering

The faulty data must be labeled in a consistent manner after the fault has been found in order to enter and train the neural network. The data are labeled using the K-means clustering algorithm, a subset of machine learning. By giving a label to each sample in the training dataset, the approach simplifies the data for the neural network and improves network learning. Following data labeling, the LSTM network performs data classification among various types of faults by learning about the different types. In order to obtain insight into the structure of the data, one of the most popular exploratory data analysis approaches is clustering, which is in charge of identifying subgroups in the data so that that points can be grouped together. While data points in various clusters are highly varied, data points within a subgroup (cluster) are very similar. The unsupervised learning algorithm K-means divides unlabeled datasets into K groups, specifying in advance how many clusters will be produced. This approach enables the grouping of data into many categories and can be used to indicate the grouping of a set of unlabeled data. Each cluster has a center because it is a center-based algorithm. This algorithm's major objective is to reduce the overall distance between data points and the clusters that correspond to those locations. This algorithm separates the input unlabeled dataset into K clusters and then iterates through the clusters until it finds the best ones. Essentially, the K-means clustering method does two things:

1. Utilizes an iterative technique to choose the greatest value for K center points.
2. Chooses the closest K center for each data point. A cluster is formed by the data points that are near the K 's center.

As a result, each cluster is distinct from the others while sharing some characteristics with the data [19]. K-means clustering is utilized in this paper to prepare the data before it is fed into the neural network.

3.3. Using an LSTM Neural Network to Identify the Type of Fault

When the KPCA detects a fault, its source must be quickly determined using a highly accurate procedure. This paper introduces a novel approach to fault detection using an LSTM neural network. This network has very high accuracy in fault identification because of its recursive structure. Additionally, compared to other machine learning techniques, the inclusion of long-term memory makes this network a powerful tool for fault identification. The K-means approach is used to train the LSTM network with clustered training data, which represent various kinds of faults. The network then assigns a category to any fault whenever new data arrives.

The LSTM is an improvement on the recurrent neural network, which is often applied to time series, such as speech recognition or natural language processing. Gradient vanishing is a problem in long time series, in which the weights of neural networks do not update well [20]. The LSTM can eliminate this problem by adjusting the amount of information within a cell through the forget gate, input gate, and output gate. A forget gate specifies how much information should be forgotten from previous cells. Input gates determine how much information the current input value will receive, whereas output gates determine how much information the cell should transmit. The useful information can be stored for a long time using these three gates, while the useless information can be forgotten [21].

In order to develop good LSTM models, the appropriate parameters must be found. The LSTM model for fault identification requires parameters such as hidden state dimension, optimizer type, regularization degree, and dropout rate. Training the LSTM models requires the first two parameters while avoiding overfitting requires the last two.

The dimensionality of hidden states refers to the size of the shared weight matrix, which should be determined by considering the training data size. The optimizer finds the global optimum of the loss function. Among the candidates for the proper optimizer are RMSprop, AdaGrad, and Adam. It is necessary to adjust the learning rate during the training process and check the trend of error decrease to prevent falling into a local optimum [22].

In the meantime, big neural network models often suffer from overfitting. Regularization, which gives penalties corresponding to the big weights, is helpful in preventing the issue. Regularization must be adjusted to the appropriate degree. The dropout method prevents overfitting by ignoring some weights randomly depending on the training dropout rate. In addition to improving the representation of training data, the method can also prevent overfitting [22].

In this paper, the LSTM network incorporates a hidden state with a dimension of 100. This hidden state allows the network to capture and remember relevant information from previous time steps, enabling it to learn long-term dependencies in the data. The Adam optimizer is selected to optimize the network's performance during training, combining the benefits of adaptive learning rates and momentum-based updates. Additionally, a lambda value of 0.1 is set for kernel regularization, which helps to prevent overfitting by adding a penalty term to the loss function. Moreover, a dropout rate of 0.3 is implemented within the model. Dropout randomly sets a fraction of the LSTM units to zero during training, reducing the network's reliance on specific units and improving its ability to generalize to unseen data. A minimum batch size of 32 is chosen, specifying the number of samples processed before the weights are updated. These configurations collectively contribute to the LSTM network's ability to effectively capture complex temporal patterns and generalize well to new data.

There are also advanced models of LSTMs, which are briefly mentioned. The bidirectional LSTM (bi-LSTM) refers to a neural network that stores sequence information in both backward (future to past) and forward (past to present) directions. Thus, a bidirectional LSTM is characterized by input flowing in both directions. Standard LSTMs can only deal with input flow in one direction, either backward or forward. By contrast, bi-directional input can preserve past and future information by handling flowing in both directions [23,24]. Another advanced LSTM method is the dual-LSTM, which consists of two parallel LSTM networks. In the dual channel LSTM model, the loss is minimized with the objective of obtaining the optimum weights and biases [25].

In this paper, in order to determine the efficacy and accuracy of the proposed method for identifying faults in a CSTR system, conventional methods of fault identification provided by other researchers are used for comparison. Typical machine learning techniques, such as SVM, KNN, and decision trees, are considered.

3.4. SVM-Based Fault Type Identification

The SVM is a supervised machine learning technique that can be applied to problems involving classification or regression. Each data point is represented as a point in an n -dimensional space (n being the number of features), where each feature corresponds to a particular coordinate. After that, classification is carried out by locating a hyperplane that categorizes the data into various groups. Both linear and nonlinear issues can be resolved with SVM [26]. The SVM approach is used in this paper to identify the categories of faults while comparing it with the proposed approach.

3.5. KNN-Based Fault Type Identification

The KNN is a form of supervised learning algorithm that measures the distance between the test data and all training points in an effort to predict the right class of test data. The KNN method chooses K points that are relatively close to the test data. The test data are then assigned to the category with the highest probability out of the many training data categories that have been chosen after calculating their probabilities [11]. The KNN is used in this paper to identify faults and for comparison purposes.

3.6. Decision Trees Based Fault Type Identification

The supervised learning algorithm family includes the decision tree algorithm. There are roots, branches, nodes, and leaves in a decision tree. A batch of test data for prediction decision trees is started at the tree's base. A feature whose value the test data is compared

with is present at the root. The branch associated with that value is followed, and this branch finishes at a node based on the comparison. Every node has a feature, and the next branch is chosen by comparing the test data with that feature. The test data eventually reach a leaf that defines the test data’s category [27]. The decision tree is used in this study to identify the type of fault and compare its results with those of the proposed approach.

4. Description of the CSTR System

A CSTR is a type of reaction vessel where the products of the reaction simultaneously exit the vessel as reagents and reactants, and frequently solvents flow into the reactor. The tank reactor is regarded as a useful tool for continuous chemical processing. The composition of the material inside the reactor, which depends on residence time and reaction rate, should ideally match the composition of the exit. Several CSTRs may be joined together to form a parallel or linear series mode when the reaction is extremely slow or when there are two immiscible or viscous liquids that need high stirring speed [28].

A basic general model of a tank with a fixed cover, which continuously performs the first-order heat reaction, is shown in Figure 2 [29].

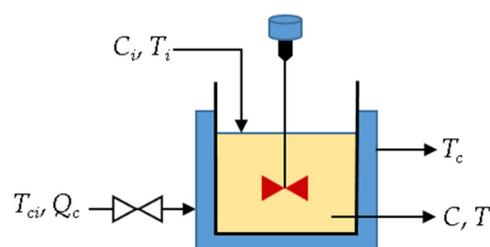


Figure 2. Overview of the CSTR system [29].

The CSTR model when undergoing exothermic first-order reaction is given as [29]:

$$\frac{dC}{dt} = \frac{Q}{V}(C_i - C) - akC + v_1, \tag{13}$$

$$\frac{dT}{dt} = \frac{Q}{V}(T_i - T) - a \frac{(\Delta H_r)kC}{\rho C_p} - b \frac{UA}{\rho C_p V}(T - T_c) + v_2 \tag{14}$$

$$\frac{dT_c}{dt} = \frac{Q_c}{V_c}(T_{ci} - T_c) + b \frac{UA}{\rho_c C_{pc} V_c}(T - T_c) + v_3, \tag{15}$$

with inputs $u = [C_i \ T_i \ T_{ci} \ Q_c]$ and outputs $y = [C \ T \ T_c]$. Moreover, v_i stands for the process noise, and $k = k_0 \exp(\frac{-E}{RT})$ is an Arrhenius rate constant. The parameters’ values in Equations (13)–(15), defined as in reference [29], are inlet flow rate $Q = 100.0 \text{ L}\cdot\text{min}^{-1}$, tank volume $V = 150.0 \text{ L}$, jacket volume $V_c = 10.0 \text{ L}$, heat of reaction $\Delta H_r = -0.2 \times 10^5 \text{ cal}\cdot\text{mol}^{-1}$, heat transfer coefficient $UA = 7.0 \times 10^5 \text{ cal}\cdot\text{min}^{-1}\cdot\text{K}^{-1}$, pre-exponential factor to $k \ k_0 = 7.2 \times 10^{10} \text{ min}^{-1}$, activation energy $E/R = 10^4 \text{ K}$, fluid density $\rho, \rho_c = 1000.0 \text{ g}\cdot\text{L}^{-1}$, and fluid heat capacity $C_p, C_{pc} = 1.0 \text{ cal}\cdot\text{g}^{-1}\cdot\text{K}^{-1}$.

There are seven input and output variables in the system. Each variable is measured by a sensor properly located. In general, there are three types of variables: C , T , and Q , which are related to product concentration, reactor temperature, and inlet flow rate, respectively. For each sensor, there is a type of fault that can be considered. Due to the chemical nature of the system, it may also be subject to catalyst decay or fouling. As a result, component faults can also occur. Moreover, it is possible that input variables are noisy.

In this paper, we consider a total of 10 variables, including 7 input and output, as well as 3 noisy input variables. There is a wide range of data in this system, from very small to very large, depending on the variables. Considering this issue, besides reducing dimensions, the data need to be mapped. In order to improve training, 770 datasets from

system faulty and regular operation data are used. Based on simulation experiments in MATLAB software and the lack of outliers, this number was found suitable for use.

A total of 10 categories of faults may exist in this system, where 7 are sensor faults corresponding to each input (4) and output (3) values, 1 is related to the degradation of the catalyst, and 1 is due to the buildup of heat transfer. The combination of these two also constitutes an additional fault. These last 3 faults are related to process faulty conditions and, thus, are considered the most critical ones. When there is no fault, the values of the parameters a and b in the CSTR system model (13)–(15) are equal to 1.00. It is feasible to replicate the deterioration of the catalyst and the deposition of heat transfer by reducing their values to zero. Table 1 summarizes the fault scenarios, as considered in reference [29]. The subscript 0 stands for nominal values and t is measured in minutes.

Table 1. Scenarios of CSTR primary faults [29].

Number	Designation	Description	Progression Rate γ	Type
1	Catalyst decay	$a = a_0 \exp(-\gamma t)$	0.0005	Multiplicative
2	Fouling	$b = b_0 \exp(-\gamma t)$	0.001	Multiplicative
3	Simultaneous	faults 1 and 2	-	Multiplicative
4	Sensor	$C_i = C_{i,0} + \gamma t$	0.001	Additive
5	Sensor	$T_i = T_{i,0} + \gamma t$	0.05	Additive
6	Sensor	$T_{Ci} = T_{Ci,0} + \gamma t$	0.05	Additive
7	Sensor	$C = C_0 + \gamma t$	0.001	Additive
8	Sensor	$T = T_0 + \gamma t$	0.05	Additive
9	Sensor	$T_C = T_{C,0} + \gamma t$	0.05	Additive
10	Sensor	$Q_C = Q_{C,0} + \gamma t$	-0.1	Additive

5. Simulation Results

The KPCA is used to find faults, as previously described. The method uses both faulty and normal system data. For this reason, samples from the CSTR were created, each having both normal and faulty data. The samples were prepared using the MATLAB software for simulating the CSTR system. For each type of fault, a data set of 300 samples was generated, which was sufficient for learning and characterizing the faults. To gather data for each fault, samples of normal operation were collected first in the simulation, followed by data of faulty operation, until the total number of samples reached 300. For example, in a dataset related to a particular type of fault, the normal data may be displayed from sample 1 to sample 120, while the faulty data appear from sample 121 onwards. Arranging the data in this way is arbitrary. The T^2 and Q indicators are used for fault detection. The simulation's outcomes demonstrate how the first faulty sample was correctly identified. The simulation results for the second type of fault, which are present in samples 121 to 300, are shown in Figure 3.

As seen in Figure 3, sample 121 exhibits the occurrence of the fault. That is, in this kind of fault, the fault is found at the first occurrence it appears, demonstrating the method's high degree of accuracy. Both the statistical indices T^2 and Q successfully identify this issue. Indeed, each of the 10 fault types is in a different range, as shown in Table 2, and is identified by the two indices. The results with the T^2 slightly differ from those with the Q index in terms of how well they identify the faults.

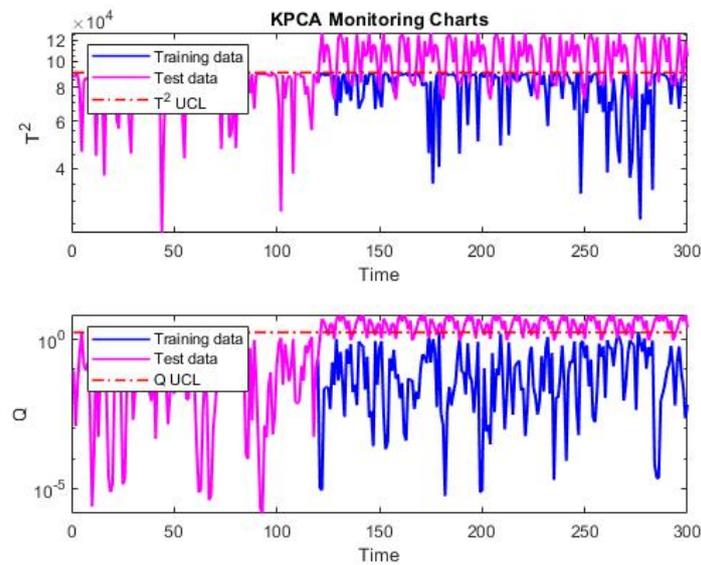


Figure 3. Simulation results of fault detection by KPCA in the CSTR system.

Table 2. Fault sample detected by T^2 and Q for all fault types.

Fault	Faulty Samples	Fault Detection Sample by T^2	Fault Detection Sample by Q
1	151–210	151	151
2	121–300	122	121
3	71–140	73	71
4	131–300	132	131
5	71–250	75	72
6	61–280	62	61
7	121–270	123	121
8	111–300	111	111
9	151–300	152	151
10	151–300	153	151

For comparison, the PCA approach was utilized for fault detection. Figure 4 depicts the results of the second kind of fault with PCA. We verify that the PCA approach does not effectively detect faults in nonlinear data.

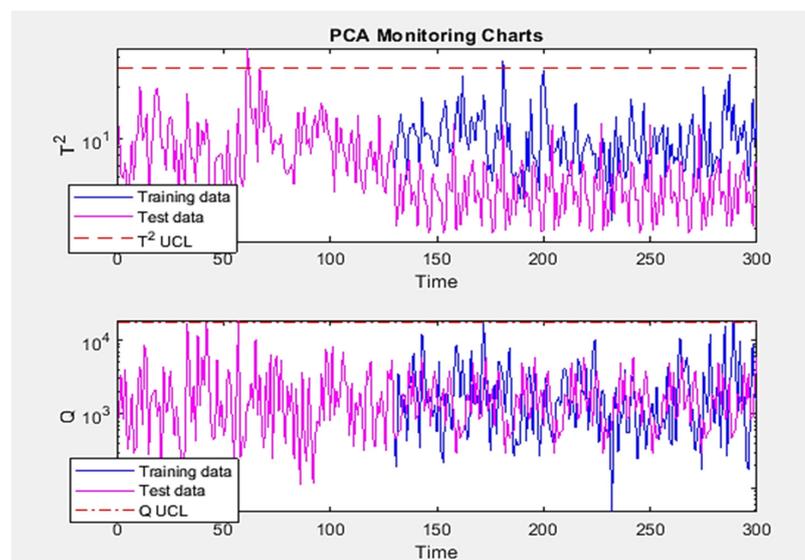


Figure 4. Simulation results of fault detection by PCA in the CSTR system.

For fault identification, it is necessary to cluster or label the data in order to incorporate the values of the model variables for usage in the neural network. The initial values of these variables spread out over wide intervals and are not suited for use in neural networks. As a result, the K-means method is applied to create 64 initial clusters. In other words, each sample, which consists of 10 variables, is assigned a label from 1 to 64 and placed in one of these 64 clusters. The values of the samples are transformed into a label column using this clustering. The accuracy cannot be increased enough by utilizing these labels, so the variables are first divided into three groups, and separate clustering is carried out for each category. Three columns of labels are eventually produced for each sample, greatly enhancing the accuracy. The proposed approach was evaluated based on the common quality metrics of accuracy, recall, and precision, which are explained in Equations (16)–(18):

$$\text{Accuracy} = \frac{tp + tn}{tp + tn + fp + fn'} \tag{16}$$

$$\text{Recall} = \frac{tp}{tp + fn'} \tag{17}$$

$$\text{Precision} = \frac{tp}{tp + fp'} \tag{18}$$

where tp , tn , fp , and fn stand for true positives, true negatives, false positives, and false negatives, respectively. A confusion matrix can be used to obtain these indices directly. Results obtained for various fault identification modes are shown in Table 3.

The method proposed in this study, which entails clustering data for each category of variables and then identifying the fault via the LSTM network, improves fault identification accuracy to nearly 99%, as shown in Table 3. Additionally, in the simulations, the third mode of the LSTM network, in which the variables are divided into three categories and clustered, operates at a significantly faster rate than the other two modes.

Table 3. The accuracy, recall, and precision of different fault identification modes.

Various Implementations of the LSTM Network	Fault Type	Precision %	Recall %	Accuracy %
No clustering	1	80.6	79.1	81.5
	2	77.3	83.5	
	3	81.4	76.9	
	4	75.3	80.1	
	5	83.2	77.6	
	6	81.1	82.6	
	7	79.9	80.8	
	8	83.2	79.8	
	9	78.5	79.3	
	10	82.03	80.6	
Clustering all data at once	1	86.6	88.1	89
	2	90.5	91.3	
	3	89.3	87.9	
	4	90.2	89.1	
	5	87.4	88.9	
	6	84.8	87.3	
	7	91.6	93.2	
	8	90.6	90.3	
	9	88.7	89.5	
	10	89.9	90.5	
Data divided into three categories, with individual clustering for each category	1	100	100	99.09
	2	98.6	99.5	
	3	99.01	99.3	
	4	100	100	
	5	99.3	99.06	
	6	98.9	99.07	
	7	99.5	98.8	
	8	99.1	98.06	
	9	99.00	99.4	
	10	99.8	99.2	

Due to its memory and recursion characteristics, the LSTM network appears to be an appropriate method for fault diagnosis. The sorts of faults have been determined by using this network and labels that are connected to 10 different types of faults that already exist. A total of 770 training examples, including 10 different fault types and a normal system operation, are used to train this network (11 categories in total). Once the network has been trained, test data from each of the 11 categories are fed into it, and the network then uses these data to identify which category each sample belongs. This network’s accuracy was estimated to be 99.09%. As a result, this method’s results for fault identification are highly accurate, and the method may be used to find many kinds of faults in actual systems.

Table 4 compares the accuracy, recall, and precision of fault identification using the LSTM, SVM, KNN, and decision tree approaches.

Table 4. Comparing the accuracy, recall, and precision of different machine learning methods for fault identification.

Various Machine Learning Methods	Fault Type	Precision %	Recall %	Accuracy %
SVM	1	21.2	22.0	23
	2	22.3	25.8	
	3	26.5	28.7	
	4	20.4	21.3	
	5	24.1	20.1	
	6	25.3	26.7	
	7	22.2	23.2	
	8	24.2	24.6	
	9	21.1	23.2	
	10	22.8	21.3	
KNN	1	27.1	28.2	27.5
	2	28.3	27.3	
	3	26.5	26.2	
	4	29.9	25.1	
	5	28.7	24.6	
	6	27.3	29.1	
	7	27.8	23.7	
	8	27.1	28.2	
	9	29.0	27.1	
	10	26.5	24.5	
Decision Tree	1	41.3	46.2	45.2
	2	42.2	48.6	
	3	39.8	41.9	
	4	48.9	44.3	
	5	45.5	46.1	
	6	44.3	40.2	
	7	49.3	41.8	
	8	51.2	48.1	
	9	55.1	49.2	
	10	43.9	40.8	
LSTM	1	100	100	99.09
	2	98.6	99.5	
	3	99.01	99.3	
	4	100	100	
	5	99.3	99.06	
	6	98.9	99.07	
	7	99.5	98.8	
	8	99.1	98.06	
	9	99.00	99.4	
	10	99.8	99.2	

We verify that the classification performance of different machine learning methods for different sorts of faults varies greatly. The SVM approach, with a 23% accuracy rate, the KNN, with a 27.5% accuracy rate, and the decision tree, with a 45.2% accuracy rate, are worse than the new method based on the LSTM network, with a 99.09% accuracy rate. Indeed, the LSTM demonstrates very good performance in identifying the CSTR system faults. As a result, the fault identification model by the LSTM network can be

used in practice, as it enables precise identification of the type of fault that has occurred in the system. Industries place a high value on precision because this helps them to avoid expensive repairs and damages.

6. Conclusions

In this paper, a novel hybrid approach was used to address fault detection and identification in an industrial system by employing KPCA, K-means, and LSTM neural networks. The method was tested in a CSTR system. First, a dataset was created using both non-faulty and faulty data from the CSTR. Next, a fault detection stage was carried out using the KPCA method, and faults could be found in the datasets. In the following phase, the data were processed and labeled using the K-means approach. Finally, an LSTM neural network was used to identify fault types. The network was taught using the information on the names of 10 different types of system faults and the normal system condition. Then, the LSTM network carried out fault identification using testing data.

Apart from being effective and practical, the proposed approach can be applied to faults that occur across a variety of processes. The method can be used as an easily implementable generic fault detection procedure. Dimension reduction contributes to energy savings in data transmission, clustering allows fault detection in different processes, and ANN-based fault identification permits the isolation of malfunctioning sensors, actuators, and components. The advantages of this comprehensive method make it useful for industrial processes or other systems that produce large amounts of data and perform a variety of processes.

The following conclusions can be drawn from the overall findings using the suggested methodology:

- In the sample where the fault appears, the KPCA fault detection method finds the existing fault very well and with excellent accuracy.
- The KPCA approach minimizes the data dimensions, sometimes even to half of the real dimensions, which decreases the number of calculations and speeds up computer processing.
- Unlike the PCA approach, the KPCA method finds faults in all data and performs well for nonlinear data.
- When faults are classified using K-means clustering, the identification is substantially more accurate.
- Due to its recursion, the LSTM network produces relatively significant results when compared to other machine learning techniques.
- The LSTM network correctly identifies faults, and its accuracy is very high, reaching 99.09%.
- Naturally, the proposed approach can be applied to detect and identify faults in other types of equipment.

It is important to note that real-world data are affected by uncertainty, which can be aleatoric or epistemic. The first occurs due to the inherent randomness and variability that accompanies the data generation process, such as measurement errors and noise. The second originates from the incomplete knowledge or understanding of the underlying system or model parameters [30]. Neglecting these types of uncertainty in the context of fault detection and identification can limit the reliability and generalizability of the proposed method. Therefore, it is crucial to highlight that addressing and quantifying aleatoric and epistemic uncertainties in real-world data is an avenue for further research, which the author will pursue to improve the robustness and accuracy of fault detection and identification techniques.

Author Contributions: Conceptualization, N.J.; methodology, N.J. and A.M.L.; software, N.J.; validation, N.J. and A.M.L.; investigation, N.J. and A.M.L.; writing—original draft preparation, N.J.; writing—review and editing, A.M.L.; visualization, A.M.L.; funding acquisition, A.M.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

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

References

1. Hwang, I.; Kim, S.; Kim, Y.; Seah, C.E. A survey of fault detection, isolation, and reconfiguration methods. *IEEE Trans. Control Syst. Technol.* **2010**, *18*, 636–653. [[CrossRef](#)]
2. Thirumarimurugan, M.; Bagyalakshmi, N.; Paarkavi, P. Comparison of Fault Detection and Isolation Methods: A Review. In Proceedings of the IEEE 10th International Conference on Intelligent Systems and Control (ISCO), Coimbatore, India, 7–8 January 2016.
3. Khalili, M.; Zhang, X.; Cao, Y.; Polycarpou, M.M.; Parisini, T. Distributed adaptive fault-tolerant control of nonlinear uncertain second-order multi-agent systems. In Proceedings of the IEEE Annual Conference on Decision and Control (CDC), Osaka, Japan, 15–18 December 2015.
4. Kersten, J.; Rauh, A.; Aschemann, H. Analyzing Uncertain Dynamical Systems after State-Space Transformations into Cooperative Form: Verification of Control and Fault Diagnosis. *Axioms* **2021**, *10*, 88. [[CrossRef](#)]
5. Muradore, R.; Fiorini, P. A PLS-Based Statistical Approach for Fault Detection and Isolation of Robotic Manipulators. *IEEE Trans. Ind. Electron.* **2012**, *59*, 3167–3175. [[CrossRef](#)]
6. Deng, X.; Tian, X.; Chen, S.J.; Harris, C. Nonlinear Process Fault Diagnosis Based on Serial Principal Component Analysis. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *29*, 560–572. [[CrossRef](#)] [[PubMed](#)]
7. Zidi, S.; Moulahi, T.; Alaya, B. Fault detection in Wireless Sensor Networks through SVM classifier. *IEEE Sens. J.* **2017**, *6*, 340–347. [[CrossRef](#)]
8. Kiranyaz, S.; Gastli, A.; Ben-brahim, L. Real-Time Fault Detection and Identification for MMC using 1-D Convolutional Neural Networks. *IEEE Trans. Ind. Electron.* **2019**, *66*, 8760–8771. [[CrossRef](#)]
9. Krishnamoorthy, G.; Ashok, P.; Tesar, D. Simultaneous Sensor and Process Fault Detection and Isolation in Multiple-Input—Multiple-Output Systems. *IEEE Syst. J.* **2015**, *9*, 335–349. [[CrossRef](#)]
10. Rohani, R.; Koochaki, A. A Hybrid Method Based on Optimized Neuro-Fuzzy System and Effective Features for Fault Location in VSC-HVDC Systems. *IEEE Access* **2020**, *8*, 70861–70869. [[CrossRef](#)]
11. Zhou, Z.; Wen, C.; Yang, C. Fault Isolation Based on K-Nearest Neighbor Rule for Industrial Processes. *IEEE Trans. Ind. Electron.* **2016**, *63*, 2578–2586. [[CrossRef](#)]
12. Fezai, R.; Nounou, M.; Messaoud, H. Reliable Fault Detection and Diagnosis of Large-Scale Nonlinear Uncertain Systems Using Interval Reduced Kernel. *IEEE Access* **2020**, *8*, 78343–78353. [[CrossRef](#)]
13. Cai, L.; Tian, X.; Chen, S. Monitoring Nonlinear and Non-Gaussian Processes Using Gaussian Mixture Model-Based Weighted Kernel Independent Component Analysis. *IEEE Trans. Neural Netw. Learn. Syst.* **2015**, *28*, 122–135. [[CrossRef](#)]
14. Jiang, Q.; Yan, X. Multimode Process Monitoring Using Variational Bayesian Inference and Canonical Correlation Analysis. *IEEE Trans. Autom. Sci. Eng.* **2019**, *16*, 1814–1824. [[CrossRef](#)]
15. Li, W.; Monti, A.S.; Ponci, F. Fault Detection and Classification in Medium Voltage DC Shipboard Power Systems With Wavelets and Artificial Neural Networks. *IEEE Trans. Instrum. Meas.* **2014**, *63*, 2651–2665. [[CrossRef](#)]
16. Ni, J.; Zhang, C.; Yang, S. An Adaptive Approach Based on KPCA and SVM for Real-Time Fault Diagnosis of HVCBs. *IEEE Trans. Power Deliv.* **2011**, *26*, 1960–1971. [[CrossRef](#)]
17. Pilario, K.E.S.; Cao, Y.; Shafiee, M. Mixed kernel canonical variate dissimilarity analysis for incipient fault monitoring in nonlinear dynamic processes. *Comput. Chem. Eng.* **2019**, *123*, 143–154. [[CrossRef](#)]
18. Lee, J.M.; Yoo, C.K.; Choi, S.W.; Vanrolleghem, P.A.; Lee, I.B. Nonlinear process monitoring using kernel principal component analysis. *Chem. Eng. Sci.* **2004**, *59*, 223–234. [[CrossRef](#)]
19. Likas, A.; Vlassis, N.; Verbeek, J.J. The global K-means clustering algorithm. *Pattern Recognit.* **2003**, *36*, 451–461. [[CrossRef](#)]
20. Shang, Y.; Zhou, B.; Wang, Y.; Li, A.; Chen, K.; Song, Y.; Lin, C. Popularity Prediction of Online Contents via Cascade Graph and Temporal Information. *Axioms* **2021**, *10*, 159. [[CrossRef](#)]
21. Qiao, M.; Yan, S.; Tang, X.; Xu, C. Deep Convolutional and LSTM Recurrent Neural Networks for Rolling Bearing Fault Diagnosis under Strong Noises and Variable Loads. *IEEE Access* **2020**, *8*, 66257–66269. [[CrossRef](#)]
22. Park, D.; Kim, S.; An, Y.; Jung, J. LiReD: A Light-Weight Real-Time Fault Detection Neural Networks. *Sensors* **2018**, *18*, 2110. [[CrossRef](#)]
23. Elsheikh, A.; Yacout, S.; Ouali, M. Bidirectional Handshaking LSTM for Remaining Useful Life Prediction. *Neurocomputing* **2018**, *323*, 148–156. [[CrossRef](#)]
24. Stoean, C.; Zivkovic, M.; Bozovic, A.; Bacanin, N. Metaheuristic-Based Hyperparameter Tuning for Recurrent Deep Learning: Application to the Prediction of Solar Energy Generation. *Axioms* **2023**, *12*, 266. [[CrossRef](#)]
25. Shi, Z.; Chehade, A. A Dual-LSTM Framework Combining Change Point Detection and Remaining Useful Life Prediction. *Reliab. Eng. Syst. Saf.* **2020**, *205*, 107257. [[CrossRef](#)]
26. Jeong, K.; Choi, S.B.; Choi, H. Sensor Fault Detection and Isolation Using a Support Vector Machine for Vehicle Suspension Systems. *IEEE Trans. Veh. Technol.* **2020**, *69*, 3852–3863. [[CrossRef](#)]

27. Lee, C.; Alena, R.L.; Robinson, P. Migrating fault trees to decision trees for real time fault detection on international space station. In Proceedings of the IEEE Aerospace Conference, Big Sky, MT, USA, 5–12 March 2005.
28. Yang, X.; Wei, Q. Adaptive Critic Designs for Optimal Event-Driven Control of a CSTR System. *IEEE Trans. Ind. Inform.* **2021**, *17*, 484–493. [[CrossRef](#)]
29. Pilario, K.E.S.; Cao, Y. Canonical variate dissimilarity analysis for process incipient fault detection. *IEEE Trans. Ind. Inform.* **2018**, *14*, 5308–5315. [[CrossRef](#)]
30. Zhuang, L.; Xu, A.; Wang, W.L. A prognostic driven predictive maintenance framework based on Bayesian deep learning. *Reliab. Eng. Syst. Saf.* **2023**, *234*, 109181. [[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.