

Research Article

Enhanced Anomaly-Based Fault Detection System in Electrical Power Grids

Wisam Elmasry  and Mohammed Wadi 

Electrical & Electronics Engineering Department, Istanbul Sabahattain Zaim University, Istanbul, Turkey

Correspondence should be addressed to Wisam Elmasry; wisam.elmasry@izu.edu.tr

Received 23 October 2021; Accepted 23 November 2021; Published 14 February 2022

Academic Editor: Muhammad Mansoor Alam

Copyright © 2022 Wisam Elmasry and Mohammed Wadi. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Early and accurate fault detection in electrical power grids is a very essential research area because of its positive influence on network stability and customer satisfaction. Although many electrical fault detection techniques have been introduced during the past decade, the existence of an effective and robust fault detection system is still rare in real-world applications. Moreover, one of the main challenges that delays the progress in this direction is the severe lack of reliable data for system validation. Therefore, this paper proposes a novel anomaly-based electrical fault detection system which is consistent with the concept of faults in the electrical power grids. It benefits from two phases prior to training phase, namely, data preprocessing and pretraining. While the data preprocessing phase executes all elementary operations on the raw data, the pretraining phase selects the optimal hyperparameters of the model using a particle swarm optimization (PSO)-based algorithm. Furthermore, the one-class support vector machines (OC-SVMs) and the principal component analysis (PCA) anomaly-based detection models are exploited to validate the proposed system on the VSB dataset which is a modern and realistic electrical fault detection dataset. Finally, the results are thoroughly discussed using several quantitative and statistical analyses. The experimental results confirm the effectiveness of the proposed system in improving the detection of electrical faults.

1. Introduction

Nowadays, there is a rapid growth in the electrical power grids in terms of size and complexity [1]. This growth includes all sectors of electrical power industry starting from generation to transmission and distribution [2]. One of the conventional problems encountered in the electrical power system is the sudden occurrence of electrical faults across transmission or distribution lines [3]. An electrical fault is deemed to be an abnormal change in current and voltage values, that is, higher values of current and voltage than those commonly expected to be under normal operating conditions. This deviation of voltage and current from nominal states is caused by human errors, environmental conditions, and equipment failures [4]. Furthermore, when an electrical fault occurs, it imposes excessively high current to flow across the network that may cause damage to devices and equipment [5]. Therefore, an early and accurate fault

detection is pivotal to prevent equipment damage, service interruption, and loss of human and animal lives [6].

Although electrical fault detection systems based on binary classification have been extensively researched during the last decade [7], it was reported that there is a research gap in this domain including the automation and validation of the system [8]. Hence, there is a dire need for an intelligent system that acts efficiently in the real-world power systems.

The anomaly detection in machine learning refers to a special class of detection methods that seeks to identify anomalous samples or events in a dataset [9]. Basically, anomalies (also named outliers) are extremely different from the expected pattern of a dataset, and they are quantitatively scarce compared to the majority (normal) of samples [10]. Likewise, electrical faults rarely occur in real-world power systems (less than 5%) while the rest of the signals are normal [11]. Therefore, employing anomaly detection will fit the electrical fault detection problem instead of using the

conventional binary classification where there is a need of enough amount of faulty signals in the dataset to prevent the binary classifier from biasing to the “normal” class [12]. The anomaly-based detection models are being trained on the normal samples exclusively in such a way that these models can discover the normal behaviour of data. Then, they can detect any unseen data which deviate from the preserved behaviour [13, 14].

The aims and contributions of this research are threefold, as follows.

- (i) An anomaly-based detection system is suggested to reveal electrical faults as they occur in the electrical power grids.
- (ii) To enhance detection of electrical faults, two preliminary phases are introduced just before the training phase.
- (iii) The VSB dataset is utilized to validate two anomaly-based detection models leveraged from the proposed technique.

The rest of this paper is organized as follows. A list of related works in the domain of electrical fault detection is introduced in Section 2. Section 3 describes the main characteristics of the VSB dataset. In Section 4, the proposed anomaly-based detection system is explained in detail with a brief description of each of the used models. Then, Section 5 presents which evaluation metrics are used along with their formulas, and the experimental results are also given. The obtained results are discussed and the proposed system is validated using various analytical and statistical aspects in Section 6. In Section 7, the conclusion of the study is drawn.

2. Literature Review

In the open literature, dozens of studies had been published in the area of electrical fault detection and classification. The artificial neural network (ANN) classifier has been utilized in many previous studies for fault detection. For instance, Jamil et al. tested an ANN model for three-phase power line fault detection and classification [15]. They used simulated data in MATLAB and obtained good results. Similarly, several ANN models with different structures were introduced to detect electrical faults on simulated dataset [16]. In [17], an ANN model was exploited to classify and detect electrical faults in simulated six-phase transmission line. Atul and Navita simulated a double-circuit transmission line using MATLAB and applied an ANN classifier on their simulated data for the purpose of fault detection and classification [18]. The fault detection and fault location in extra high voltage (EHV) environments were investigated using an ANN model and simulated dataset [19]. In [20], electrical faults in simulated transmission line were detected and classified using an ANN model. A simulation of the Nigerian power system using MATLAB was introduced in [21, 22]. Then, they used the simulated data for fault detection, fault classification, and fault location using an ANN model. Different ANN architectures with the backpropagation (BP) technique were proposed for both fault detection and classification [23, 24].

Three variations of the ANN model, which are the adaptive network fuzzy inference system (ANFIS), probabilistic neural network (PNN), and generalized regression neural network (GRNN), were utilized for three different tasks, namely, fault detection, fault classification, and fault location [25]. The proposed models were trained and tested using simulated dataset in Simulink. Ekici et al. extended the former study by using the PNN model to classify faults while they used the resilient propagation (RPROP) technique to identify the location of faults [26].

In [27], the prediction of fault occurrence and fault location using the concurrent fuzzy logic (CFL) was employed on data of different cases of simulated transmission lines. A novel technique for detecting fault locations in simulated transmission line was introduced in [28]. The frequency transformations, such as the wavelet transform, were also used for fault detection. Koley et al. suggested a hybrid technique for fault classification, fault location, and fault detection tasks [29]. The suggested technique in the former study exploited the wavelet transform along with the modular ANN model to accomplish these tasks in simulated six-phase transmission lines. In [30], Wani et al. investigated the effectiveness of using the wavelet transform and different ANN architectures on simulated data to detect and classify faults as well as to identify their locations.

Another widely used machine learning method for fault detection is the support vector machine (SVM). For example, three SVM models with different kernel functions were used to detect faults using simulated data [31]. It was reported in the former study that the SVM model based on the Gaussian radial basis kernel function (RBF) was superior to other SVM models. Singh et al. performed fault detection and fault classification using a SVM classifier [32]. They also simulated a 3-phase transmission line in MATLAB framework and exploited the simulated data to validate their SVM model. In [33], a novel method using the SVM model was developed in order to detect faults and their type and location in simulated transmission lines.

In a similar study [34], a new seasonal and trend decomposition using loess (STL) method was proposed, and a SVM model with the RBF kernel was utilized to recognize partial discharge (PD) activities. They trained and tested their model on the VSB dataset [35]. They obtained a recall value of 88% of actual PD signals. A unique anomaly-based fault detection technique is proposed and investigated by using the OC-SVM and PCA anomaly-based models [36]. The two models are validated on the VSB dataset and gained a good performance with accuracy of 80%.

To put all together, most of aforementioned studies in the area of electrical fault detection suffer from the following two shortcomings. (i) They mainly depended on the binary classification-based methods to detect faults in transmission lines, which is inappropriate in the case of electrical faults since the electrical faults are rare in reality [36]. (ii) They exploited simulated datasets to validate their proposed techniques, which cannot accurately represent the actual pattern of electrical faults in real-world power systems [3, 6–8]. Therefore, proposing an enhanced anomaly-based electrical fault detection system that is based on real-time data is still very desirable.

3. VSB Dataset

The VSB (Technical University of Ostrava) dataset is a modern dataset which was published online in Kaggle Competition website in 2018 [35]. In addition to that, it is a realistic fault detection dataset because it was created by the ENET Center at Technical University of Ostrava using a new device for capturing electrical signals passed through real power lines [37]. Regarding structure of the VSB dataset, it has 8712 samples, and each sample is merely an electrical signal that has 800,000 voltage measurements stored as integer values. These signals are captured from a real 3-phase electrical power grid that operates at 50 Hz, and all signals are recorded over a single complete grid cycle (20 milliseconds).

Furthermore, there is a feature, named “Class,” in the VSB dataset that determines the type of each signal, i.e., “normal” and “faulty” classes are labeled as “0” and “1,” respectively. On the other hand, the majority of samples in the VSB dataset belong to normal signals (8187 samples), while the rest (525 samples) are faulty signals. This severe defect between the number of normal and faulty samples in the VSB dataset may lead to poor classification results because the classifiers will bias to the majority class (“normal”). Hence, employing anomaly-based detection models is inevitable with such an imbalanced dataset.

4. Methodology and Models

The proposed system seeks to recognize anomalous patterns in the electrical power line’s voltage signals. Since anomaly-based detection models cannot deal with an electrical signal in its raw form, the input signal has to undergo a preprocessing phase where voltage measurements of the signal are filtered from noise and decomposed into chunks. Afterwards, a feature extraction process is executed to characterize the pattern of voltage measurements, and then these features are put into a data record and normalized. These data records of the input signals are used along with a PSO-based algorithm in the pretraining phase to determine the optimal hyperparameter vector of the selected anomaly-based detection model which fits the used model with underlying fault detection task. Thereafter, the optimized anomaly-based detection model is trained on data records of the normal signals which help the used model to build precisely a profile for the normal signals. Finally, the trained anomaly-based fault detection model will be ready to detect any faulty signals from the normal signals. Figure 1 shows the mechanism of the proposed anomaly-based electrical fault detection system. In the next sections, the methodology of executing our empirical experiments which accomplishes the aforementioned proposed system and a brief description of anomaly-based detection models are explained in detail.

4.1. Methodology. Our methodology is developed to be evident and uncomplicated. It comprises four successive phases, namely, data preprocessing, pretraining, training, and testing. These phases are elaborated in the following sections. Figure 2 depicts the diagram of our methodology.

4.1.1. Data Preprocessing Phase. As the name of this phase may imply, it performs all elementary operations on the samples of the VSB dataset. It is very vital because it prepares data for modeling and analyzing by the used anomaly-based fault detection models. Five successive operations are applied on the VSB dataset in this phase: signal denoising, signal decomposition, feature extraction, data normalization, and dataset splitting, as follows.

(1) Signal Denoising. Generally, machine learning specialists or scientists concern with applying machine learning methods or algorithms on the dataset instead of collecting samples or observations. Thus, they consider the collected data as a ready-for-use dataset and no further work is required. Unfortunately, this is not correct most of the time because usually the collected data are dirty, that is, containing a noise. There are several causes of noise existence in the collected data such as failure of the measurement devices, unexpected event, or casual environmental conditions. Indeed, there is no way to prevent capturing noise during data collecting process. In front of this fact, there is no option rather than accepting the existence of noise in the collected data. On the contrary, using dirty data to train models has a serious downside regarding the quality of data modeling and analyzing. This can be explained by the fact that all the descriptive statistics of the collected data such as the mean and standard deviation are sensitive to noise which can cause tests to either miss significant findings or distort real results. Therefore, the most effective solution in this case is filtering data from the noise [38].

To have better results in data cleaning, the concept of noise inside data should be realized firstly. As mentioned in Section 1, noise is another face of outliers in data. The noise in data is a set of rare samples that fall far away from the majority of data. There is no precise way to identify noise in general. But with the help of the concept of noise explained above, some statistical methods can be utilized to find out noise candidates. One of the well-known noise filtering methods is the interquartile range (IQR). The advantages of using the IQR method are not only because it does not depend on a specific distribution of data but also because it is relatively robust to the presence of noise compared to the other quantitative methods [39]. In this paper, each signal in the VSB dataset is filtered from noise separately, as follows. Firstly, a copy of all voltage measurements of particular signal is saved and ordered in an ascending order. Afterwards, the first quartile $Q1$ (25% percentile) and the third quartile $Q3$ (75% percentile) of the signal measurements are calculated. Then, the value of IQR (the middle 50% of the signal) is computed according to the following equation:

$$IQR = Q3 - Q1. \quad (1)$$

After that, the value of IQR is multiplied by k value which is an adjustment factor. The aim of using such a factor is to determine the strength of outliers. In statistics, there are two widely used values of k , which are 1.5 and 3 [39]. While the value of 1.5 is used to identify weak (minor) outliers, the value of 3 is used to determine strong (major) outliers in

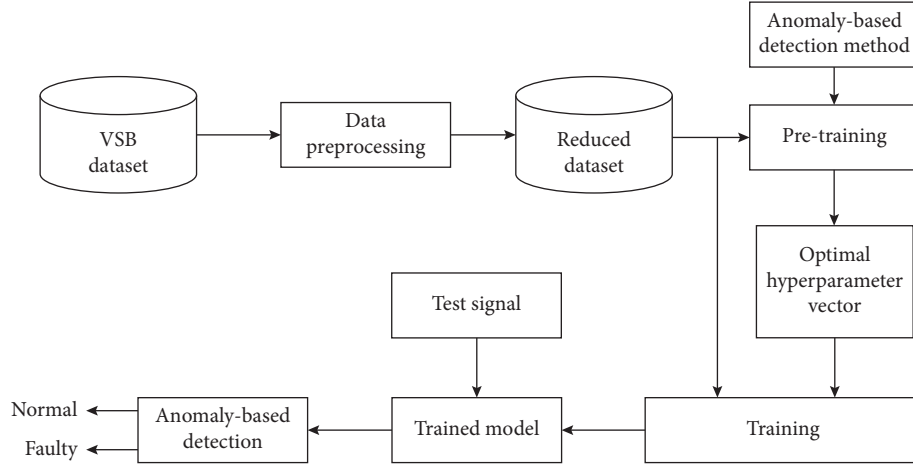


FIGURE 1: Mechanism of the proposed anomaly-based fault detection system.

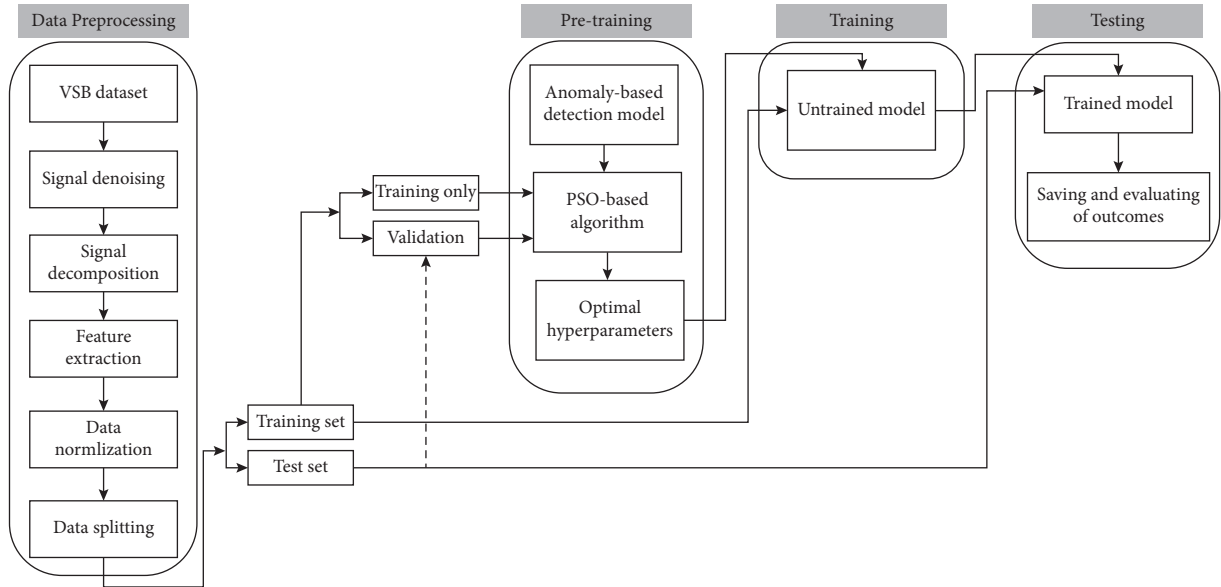


FIGURE 2: Diagram of experiments' methodology.

data. In the electrical fault detection problem, two types of outliers are familiar: fault measurements which slightly deviate from the normal values and noise measurements which extremely differ from the normal values. As a result of that, the fault measurements have to be kept in order to accomplish the fault detection task, whereas the noise measurements have to be removed. Accordingly, in this paper, k value is set to be 3.

$$\text{Threshold} = 3 * \text{IQR}. \quad (2)$$

Thereafter, the threshold value is exploited to determine the lower and upper fences of the noise measurements using (3 and 4), respectively.

$$\text{Lower_Fence} = Q1 - \text{threshold}, \quad (3)$$

$$\text{Upper_Fence} = Q3 + \text{threshold}. \quad (4)$$

Finally, any voltage measurements of particular signal less than the lower fence value or greater than the upper fence value will be removed from the original signal data in the VSB dataset. The filtering process will step into the next signal in the VSB dataset and repeat the same procedure until all signals in the VSB dataset are filtered successfully.

(2) *Signal Decomposition.* After signal denoising process is finished, the remaining voltage measurements of the i^{th} signal in the VSB dataset are equal to $(800,000 - l)$, where l is the number of voltage measurements in the i^{th} signal that are identified as a noise and removed. However, detecting faults in the remaining voltage measurements of each signal is still a difficult task because the few faulty measurements are located within a wide range of normal measurements of the signal. To overcome this problem, the signal decomposition process is indispensable [40].

Signal decomposition is the process of partitioning the remaining voltage measurements of each signal into smaller chunks that are easier to detect faults within them. This will yield to shorten the range of voltage measurements in each chunk significantly and foster better understanding of faults within the range of their related normal measurements [41]. Obviously, if there are more chunks, the performance of the

model will be raised [41]. However, in order to explore the performance differences in this paper, each signal in the VSB dataset is decomposed into 1, 2, 4, and 8 chunks in separate experiments. Let M denote the number of chunks; then, the signal decomposition process breaks up the remaining voltage measurements of each signal in the VSB dataset into M chunks, as follows:

$$\text{chunk_size} = \text{ROUND}\left(\frac{(\text{remaining_measurements_of_signal}_i)}{M}\right), \quad (5)$$

where chunk_size is the size of the chunk and ROUND is a function that rounds a number to an integer value.

$$\text{Chunk}_j^i = X_{[(j-1) * \text{chunk_size}]+1}^i, X_{[(j-1) * \text{chunk_size}]+2}^i, \dots, X_{[(j-1) * \text{chunk_size}]+\text{chunk_size}}^i, j = 1, 2, \dots, M, \quad (6)$$

where X_d^i is the d^{th} voltage measurement of the i^{th} signal and Chunk_j^i is the j^{th} chunk of the i^{th} signal.

(3) *Feature Extraction.* Although the number of voltage measurements for each signal is reduced after performing signal denoising and signal decomposition, the number of voltage measurements in each chunk of the signal is estimated at tens or hundreds of thousands. This due to the fact that originally each signal in the VSB dataset consists of a numerous number of voltage measurements (800,000). Basically, each of the remaining voltage measurements of the signal will act as an input to the used models with a high-dimensional input space. Such a high-dimensional space of inputs is one of the emerging challenges in machine learning domain because high dimensionality is impracticable for the most of machine learning models and definitely it will cause a model's failure with poor performance [12].

To overcome this problem, a feature extraction process is performed to reduce dimensions of the feature space. Thus, 19 features from the existing voltage measurements are extracted for each chunk of the signal separately. After extracting features from all chunks of the signal, all extracted features are combined together along with the "Class" label of the signal in order to form a data record for that signal. The feature extraction process stops when all signals are processed and all resulting data records are put into a new dataset (the reduced dataset). Obviously, the number of features in the reduced dataset differs according to the specified number of chunks, that is, it is equal to $(19 * M) + 1$, where M is the number of chunks.

Indeed, the extracted features from each chunk of the signal are widespread statistics which can give us an informative picture about distribution and behaviour of the voltage measurements. All the extracted features are numeric values, described as follows.

- (i) Mean is the average of a set of numbers and can be obtained using the following equation:

$$\text{Mean}_j = \frac{\sum_{i=1}^{\text{chunk_size}} X_i^j}{\text{chunk_size}}, \quad (7)$$

where Mean_j is the mean of the j^{th} chunk of the signal and X_i^j is the i^{th} voltage measurement of the j^{th} chunk of the signal.

- (ii) Standard deviation is a statistical measure that gives information about dispersion of a discrete set of numbers. As the value of standard deviation increases, the variation among these numbers increases too, and vice versa.

$$\text{Standard_deviation} = \sqrt{\frac{\sum_{i=1}^{\text{chunk_size}} (X_i^j - \bar{X}_j)^2}{\text{chunk_size}}}, \quad (8)$$

where $\text{Standard_deviation}_j$ is the standard deviation of the j^{th} chunk of the signal and \bar{X}_j is the mean of the j^{th} chunk of the signal.

- (iii) Maximum value of the voltage measurements existed in particular chunk of the signal.
- (iv) Minimum value of the voltage measurements existed in particular chunk of the signal.
- (v) Percentile is a statistical measure that allows the chunk to be analyzed in terms of percentage [42]. For instance, the n^{th} percentile is a number where $n\%$ of the voltage measurements fall below that number. In this paper, the 1%, 25%, 50%, 75%, and 99% percentiles of each chunk of the signal are computed. Mathematically, the percentile value can be obtained by selecting the element of rank z after sorting the voltage measurements of particular chunk in an ascending order.

$$z = \lceil \frac{P}{100} \times \text{chunk_size} \rceil, \quad (9)$$

where P is the value of percentage.

- (vi) Relative percentile is the amount of deviation of a specific data from the mean. In this study, the 0%, 1%, 25%, 50%, 75%, 99%, and 100% relative percentiles of each chunk of the signal are calculated using the following equation:

$$P\%_Relative_Percentile_j = P\%_Percentile_j - \text{Mean}_j, \quad (10)$$

where $P\%_Relative_Percentile_j$ is the $P\%$ relative percentile of the j^{th} chunk of the signal and $P\%_Percentile_j$ is the $P\%$ percentile of the j^{th} chunk of the signal.

- (vii) Lower and upper bounds are the lowest and highest bands of the voltage measurements of particular chunk of the signal, as calculated using the following equations.

$$\begin{aligned} \text{Lower_Bound}_j &= \text{Mean}_j - \text{Standard_deviation}_j, \\ \text{Upper_Bound}_j &= \text{Mean}_j + \text{Standard_deviation}_j, \end{aligned} \quad (11)$$

where Lower_Bound_j and Upper_Bound_j are the lower and upper bands of the j^{th} chunk of the signal, respectively.

- (viii) Height is the distance measured from the minimum to the maximum of the voltage measurements of particular chunk of the signal.

$$\text{Height}_j = \text{Maximum}_j - \text{Minimum}_j, \quad (12)$$

where Height_j , Maximum_j , and Minimum_j are the height, maximum, and minimum of the j^{th} chunk of the signal, respectively.

(4) *Data Normalization.* For each data record in the reduced dataset, features (except the ‘‘Class’’ feature) are normalized into [0,1] using the min-max transformation, as follows.

$$x_i = \frac{x_i - \text{Min}}{\text{Max} - \text{Min}}, \quad (13)$$

where x_i is the numeric feature of the i^{th} data record in the reduced dataset and Min and Max are the minimum and maximum values for each numeric feature, respectively.

(5) *Dataset Splitting.* The reduced dataset obtained from the previous step is split according to the concept of anomaly detection, that is, the training set contains only normal samples, whereas the test set contains both normal and faulty samples. Hence, 7100 (81.5%) of normal samples are randomly selected without replacement and inserted in the training set. The rest of normal samples (1087) and all faulty samples (525) are put into the test set with a total of 1612

samples (18.5%). Table 1 summarizes the main characteristics of the reduced dataset after the data preprocessing phase is finished. Figure 3 shows the flowchart of data preprocessing and its operations.

4.1.2. Pretraining Phase. The pretraining phase is designed with the aim of improving the performance of the anomaly-based detection model by selecting its optimal hyperparameters which fit the electrical fault detection task. There are many swarm intelligence-based meta-heuristics that can be utilized for hyperparameter selection, but the PSO-based algorithm which is proposed by Elmasry et al. [13] has attracted much attention due to its simplicity, stability, and generality [14]. Figure 4 depicts the diagram of the PSO-based algorithm and its functionality.

The basic idea behind the PSO-based algorithm is that it selects the hyperparameter vector of particular model that maximizes the accuracy of that model on the given dataset. Accordingly, the first step in the PSO-based algorithm is to adjust it with the optimal operating parameters. Table 2 shows the values of main operating parameters of the PSO-based algorithm. The selected values in Table 2 are obtained after executing a grid search for each PSO parameter in its recommended domain. In addition to that, the domains in Table 2 are recommended in many theoretical and empirical previous studies [12, 13]. Afterwards, the user determines a list of the model’s hyperparameters and their recommended domains.

Then, in the second step, a copy of the training set is split into two independent sets: training only and validation. The hold-out sampling without replacement method is utilized to select randomly 6850 normal samples in the training-only data. The same sampling method is used to select randomly 250 normal samples as well as 125 faulty samples in the validation set (375 samples). Thereafter, for each iteration, the PSO-based algorithm tries many possible combinations of the model’s hyperparameters within their specified ranges. Once the model is tuned by a set of hyperparameters, the training-only data will be used for training and the validation sets for testing. Then, the accuracy value of the model will be computed and stored.

Finally, when the stopping criteria are satisfied, the third step of the PSO-based algorithm outputs the optimal hyperparameter vector which maximized the accuracy value of the given anomaly-based model over all iterations. Table 3 presents the hyperparameters of the used models, their ranges, and the optimal values after finishing the pretraining phase.

4.1.3. Training and Testing Phases. The training phase is started when the optimized anomaly-based model is constructed using the optimal hyperparameters and trained on the full training set. Subsequently, the testing phase is put forward by testing the trained model on the test set. Finally, the obtained outcomes are stored for further processing later.

TABLE 1: Main characteristics of the reduced dataset.

Characteristics	Value
Year	2018
Samples	8712
Classes	2
Data type	Numeric
Number of features	20 for 1 chunk 39 for 2 chunks 77 for 4 chunks 153 for 8 chunks
Training set distribution	Normal = 7100 Faulty = 0 Total = 7100
Test set distribution	Normal = 1087 Faulty = 525 Total = 1612

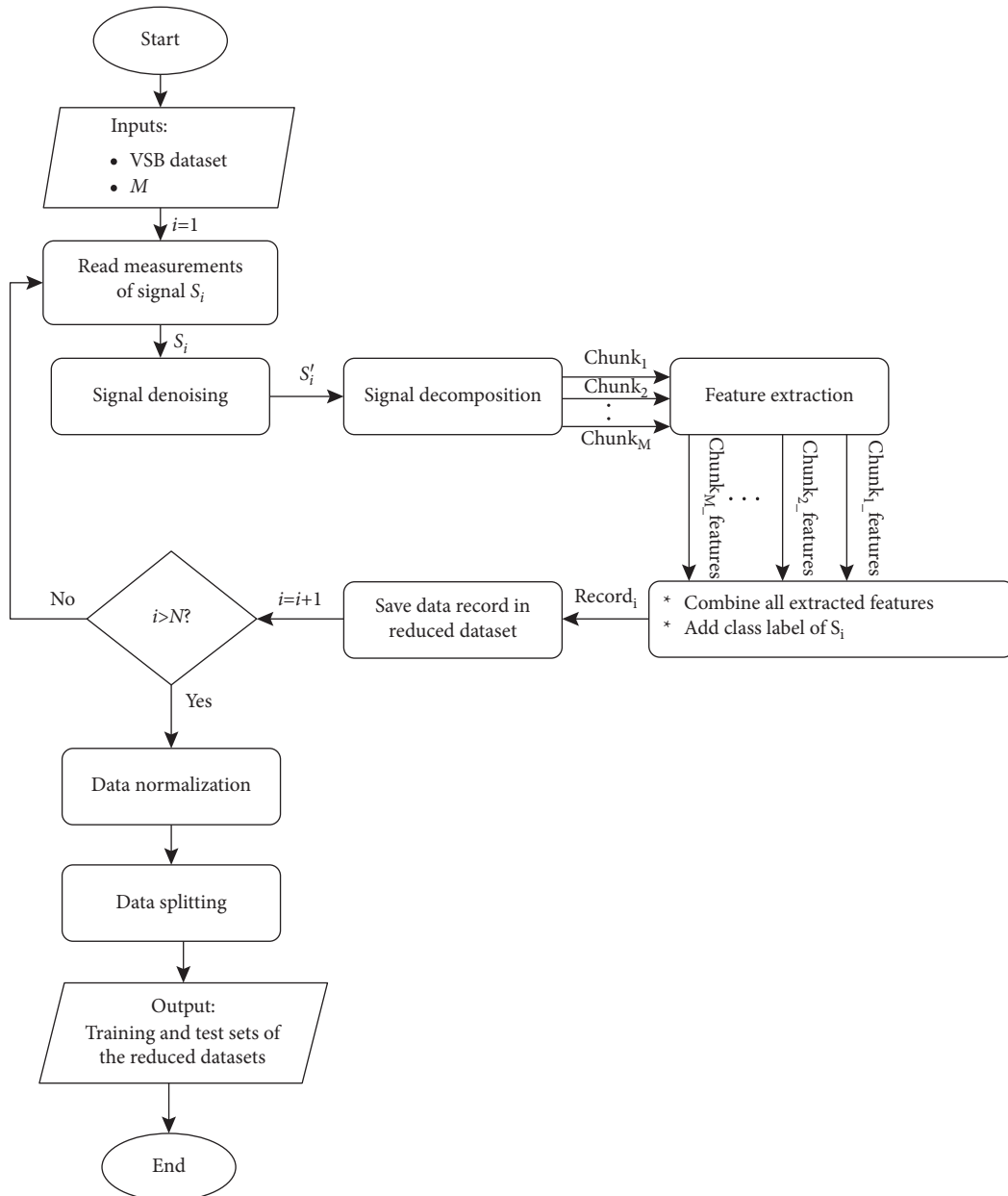


FIGURE 3: Flowchart of the data preprocessing phase (M : number of chunks, i : current iteration, S_i : the i^{th} signal in the VSB dataset, S_i' : the filtered signal of S_i , record_i : the i^{th} record in the reduced dataset, and N : the number of signals in the VSB dataset).

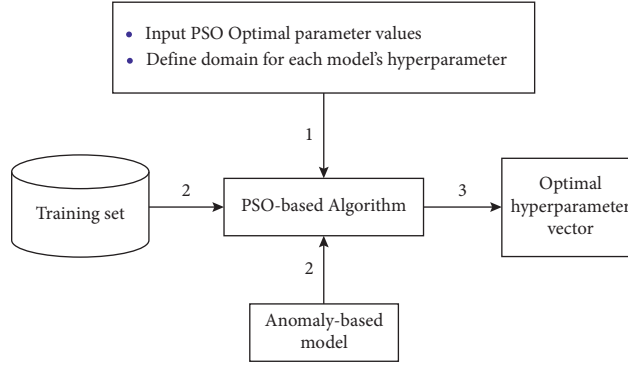


FIGURE 4: Diagram of the PSO-based algorithm for hyperparameter selection.

TABLE 2: PSO parameters and their domains and selected values [12, 13].

PSO parameter	Domain	Selected value
Swarm size	[5,40]	40
Velocity (min)	[0,1]	0
Velocity (max)	[0,1]	1
Coefficients of acceleration	[1,6]	1.43
Constant of inertia weight	[0.39,0.99]	0.69
Number of iterations (max)	[20,100]	50
Stopping factor	[0.01,0.001]	0.001

TABLE 3: The resulting optimal hyperparameters of each anomaly-based detection model.

Model	Parameter	Range	Optimal value
OC-SVM	Nu	[0.001,0.1] Step = 0.01	0.1
	Epsilon	[0.001,0.1] Step = 0.01	0.001
PCA	Rank	[2,10] Step = 2	2
	Oversampling	[2,10] Step = 2	4
	Center	{True, False}	False

4.2. Anomaly-Based Detection Models. The experiments are designed and executed in the Azure Machine Learning (AML) studio [43] using the OC-SVM and PCA anomaly-based detection models. The AML is a free cloud-based platform that can provide users with many useful capabilities. For instance, it is a collaborative tool for designing and analyzing various machine learning experiments with massive computing resources [44, 45]. Sections 4.2.1 and 4.2.2 give a brief description of the used models and their operating parameters.

4.2.1. One-Class Support Vector Machine. The OC-SVM is a special case of the traditional support vector machine where it learns from the training data to identify the majority class among other classes. To accomplish this goal, the OC-SVM model only trains on data belonging to a class that has a vast majority of the dataset samples (“normal” class of the signals). This helps the OC-SVM model to infer properties of

the normal samples, and from these properties, it can determine the boundaries of these samples [46].

Mathematically, the OC-SVM model tries to identify the smallest hypersphere which contains all the normal samples inside. Furthermore, samples located on the boundary of the hypersphere are known as support vectors, and those located outside the hypersphere are considered as anomalies. Accordingly, the problem can be defined as the following constrained optimization form [47]:

$$\min_{r,c,\zeta} r^2 + \frac{1}{\nu N} \sum_{i=1}^N \zeta_i \text{ subject to : } \|\Phi(x_i) - c\|^2 \leq r^2 + \zeta_i \forall i = 1, 2, 3, \dots, N, \quad (14)$$

where r , c , ζ_i , ν , N , x_i , and $\|\Phi(x_i) - c\|^2$ are the hypersphere radius, the hypersphere center, the i^{th} slack value, the nu hyperparameter value, the number of training set samples, the i^{th} sample in the training set, and the distance between

the i^{th} sample and the hypersphere center, respectively. The slack value of a sample means the distance between this sample and the support vectors, and if this value is less than or equal to zero, then the sample will be inside the hypersphere. Otherwise, it will be considered as an outlier. From Karush–Kuhn–Tucker optimal conditions, the center of the hypersphere can be found [48], as follows.

$$c = \sum_{i=1}^N \alpha_i \Phi(x_i), \quad (15)$$

where α_i 's are the solutions of the following constrained optimization problem:

$$\begin{aligned} \max_{\alpha} \sum_{i=1}^N \alpha_i k(x_i, x_i) - \sum_{i,j=1}^N \alpha_i \alpha_j k(x_i, x_j) \text{ subject to : } \sum_{i=1}^N \alpha_i \\ = 1 \text{ and } 0 \leq \alpha_i \leq \frac{1}{\nu N} \forall i = 1, 2, \dots, N, \end{aligned} \quad (16)$$

where k is the kernel function of the OC-SVM model. In this paper, the RBF kernel function is used since it is very common.

There are two hyperparameters which control the performance of the OC-SVM model, namely, nu (ν) and epsilon (ϵ). The nu hyperparameter is a value that determines the upper bound on the fraction of outliers [49]. This upper bound lets the user to trade off between outliers and normal cases. Moreover, the epsilon hyperparameter is deemed to be a stopping factor which affects the number of iterations reached when optimizing the OC-SVM model. Once the value is exceeded, the OC-SVM model stops iterating on a solution [46].

4.2.2. Principle Component Analysis. The PCA was firstly proposed by Karl Pearson in 1901 [50]. It is frequently exploited in machine learning to explore data because it not only describes the inner structure of data but also determines the variance in data. It is deemed to be an orthogonal linear transformation method that converts the data space into a more compact space, i.e., the principal components. This can be handled by analyzing data and looking for correlations among the features to determine the combination of values that best describes differences in outcomes.

The PCA-based anomaly detection model only trains on the normal samples and learns from them which feature set constitutes the “normal” class. In the testing phase, each unseen sample is projected on the eigenvectors as well as a normalized error value is computed in order to identify whether this sample is normal or not. The PCA-based anomaly detection model has three hyperparameters which can adjust its performance, namely, oversampling, rank, and center [51].

Let X denote the training set matrix of size $N \times P$ where the sample mean of each column is a zero empirical mean. Then, the PCA algorithm computes a set of size l of P -dimensional vectors of coefficients $w_{(k)}$ that transform each

sample $x_{(i)}$ to a new principle component scores vector $t_{(i)}$ using the following equation:

$$t_{(i)}^k = x_{(i)} \cdot w_{(k)}, \quad \forall i = 1, 2, \dots, N \text{ and } k = 1, 2, \dots, l, \quad (17)$$

where the first coefficient $w_{(1)}$ can be computed as follows.

$$w_{(1)} = \arg \max \left\{ \frac{w^T X^T X w}{w^T w} \right\}. \quad (18)$$

Finally, the k^{th} coefficient can be calculated by abstracting the first $k-1$ principle components from the matrix X using (19), and then $w_{(k)}$ coefficient can be found using the following equation:

$$\hat{X}_k = X - \sum_{s=1}^{k-1} X w_{(s)} w_{(s)}^T, \quad (19)$$

$$w_{(k)} = \arg \max \left\{ \frac{w^T \hat{X}_k^T \hat{X}_k w}{w^T w} \right\}. \quad (20)$$

5. Experimental Results

The data preprocessing and pretraining phases are carried out using the Python programming language version 3.9.6 [52] associated with the NumPy library [53]. On the other hand, the training and testing phases are hosted in the AML environment. The next two sections present the evaluation metrics and experimental results.

5.1. Evaluation Metrics. The outcome of the testing phase is merely a binary classification process indicating whether a sample is “normal” or “faulty.” Hence, a confusion matrix will be constructed after the testing phase is finished. This confusion matrix has four cells which contain the following measures: true positive (TP), true negative (TN), false positive (FP), and false negative (FN). These measures are required to compute eight commonly used evaluation metrics, as follows.

- (i) Accuracy is the ratio of the number of true classifications to the size of test set.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \quad (21)$$

- (ii) Precision is the ratio of the number of correctly classified faulty samples to all samples labeled as “faulty.”

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \quad (22)$$

- (iii) Recall is the ratio of the number of correctly classified faulty samples to all faulty samples. It is also named hit, true positive rate (TPR), detection rate (DR), or sensitivity.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \quad (23)$$

- (iv) *F1 score* is a balanced metric that includes both the recall (R) and precision (P) values. It is also called *F1 metric*.

$$F1 \text{ score} = \frac{2 \times P \times R}{P + R}. \quad (24)$$

- (v) *False alarm rate (FAR)* is the ratio of the number of normal samples that is wrongly classified as “faulty” to all normal samples. It is also named *false positive rate (FPR)*.

$$FAR = \frac{FP}{FP + TN}. \quad (25)$$

- (vi) *Specificity* is the ratio of the number of correctly classified normal samples to all normal samples. It is also known as *true negative rate (TNR)*.

$$\text{Specificity} = \frac{TN}{TN + FP}. \quad (26)$$

- (vii) *False negative rate (FNR)* is the ratio of the number of faulty samples that is wrongly classified as “normal” to all faulty samples.

$$FNR = \frac{FN}{FN + TP}. \quad (27)$$

- (viii) *Matthews correlation coefficient (MCC)* is a balanced measure that, when calculated, considers all major outcomes of classification. Usually, the MCC metric is useful in the case of using imbalanced datasets [54]. In addition to that, the MCC value can be in $[-1,1]$, where -1 and 1 values indicate weak and perfect classifiers, respectively [55].

$$MCC = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP + FN) \times (TP + FP) \times (TN + FP) \times (TN + FN)}}. \quad (28)$$

5.2. Performance Analysis. The evaluation metrics, mentioned in Section 5.1, will be the key factors for assessing the performance of the used anomaly-based detection models in the electrical fault detection. Obviously, the higher the values of true classification metrics and the lower the values of misclassification metrics, the more effective the model. Table 4 shows the percentage of the evaluation metrics for each anomaly-based detection model in all experiments. Furthermore, the bold and italicized values in Table 4 represent the best results of the used models in the same experiment and among all experiments, respectively, whereas the “Normal” column in Table 4 refers to the results of the used models without using our proposed system, and the “Enhanced” columns refer to the results when using the proposed system.

Clearly, the proposed anomaly-based fault detection system enhanced the performance of all models compared to same models without using the proposed system. This can be noticed when comparing the values of evaluation metrics of

the used models in “Normal” and “Enhanced for 1 chunk” columns in Table 4. For instance, the recall values are increased by 14%, and the FAR values are decreased by 8%. This can be explained by the impact of data preprocessing and pretraining phases that helps anomaly-based detection models to detect faults aggressively. Moreover, when the number of chunks is increased, the values of evaluation metrics for all models are improved significantly. For example, the recall and FAR values of the OC-SVM and PCA models for one chunk are (69.33%, 9.20%) and (65.52%, 9.84%), respectively. But when the number of chunks is increased to 8, they become (88.19%, 2.58%) and (90.10%, 1.47%).

Unfortunately, improving the model’s performance by partitioning the voltage measurements of the signal into more chunks does not come without a penalty, that is, as the number of chunks increases, the complexity of the model’s structure will be increased too. This is because increasing the number of chunks will result in increasing the number of extracted features from these chunks, and accordingly the number of model’s inputs will be increased linearly. Such a feature space with high dimensionality will become less reliable for shallow machine learning models. Hence, using deep learning models in such cases will be more efficient [11, 13].

Regarding performance of the OC-SVM and PCA models, the OC-SVM model is superior to the PCA model when it is applied to a low number of chunks, whereas the PCA model outperformed the OC-SVM model only when the number of chunks is equal or greater than 4. This is due to the ability of the PCA model to find out the smallest feature subset in high dimensional space and to explore the characteristics of “normal” samples with this subset of features. On the other hand, this advantage of the PCA model does not exist in the support vector machines at all. However, all the used models are still fit to the electrical fault detection.

To put all together, the proposed anomaly-based system improved the electrical fault detection, but it encounters an immediate challenge, namely, the increase of feature space which can be resolved by using deep learning methods instead of using shallow learners and harnessing feature selection algorithms to eliminate any irrelevant or redundant features [12]. Due to space limitation, only the accuracy, recall, FAR, and MCC evaluation metrics are presented in Figure 5. Figure 5 presents a visual comparison between the used models when the number of chunks varies. The performance of the used models is enhanced drastically as the number of chunks increases, in such a way that the accuracy, recall, and MCC values for all used models are increased as well as the FAR value is decreased in all cases. Another way to visually interpret the performance of the used models is to draw the critical difference diagram (CDD) [56]. Figure 6 depicts the CDD of the used models for all number of chunks. It is worthy to mention that the notation ($Model_m$) in Figure 6 indicates experiment of the *Model* with m chunks. The critical difference (CD) value, which is drawn above the figure as a bar, equals 7.4244.

TABLE 4: Results of our empirical experiments.

Model	Evaluation metric	Normal	Enhanced			
			Number of chunks			
			1	2	4	8
OC-SVM	Accuracy	75.68	83.81	87.72	90.51	94.42
	Precision	64.88	78.45	82.77	85.77	94.30
	Recall	55.24	69.33	78.67	84.95	88.19
	F1 score	59.67	73.61	80.66	85.36	91.14
	FAR	14.44	9.20	7.91	6.81	2.58
	Specificity	85.56	90.80	92.09	93.19	97.42
	FNR	44.76	30.67	21.33	15.05	11.81
	MCC	42.71	62.24	71.72	78.34	87.18
PCA	Accuracy	73.39	82.13	85.30	92.62	95.78
	Precision	60.21	76.27	79.88	90.28	96.73
	Recall	53.90	65.52	73.33	86.67	90.10
	F1 score	56.88	70.49	76.46	88.44	93.29
	FAR	17.20	9.84	8.92	4.51	1.47
	Specificity	82.80	90.16	91.08	95.49	98.53
	FNR	46.10	34.48	26.67	13.33	9.90
	MCC	37.84	58.13	65.93	83.05	90.34

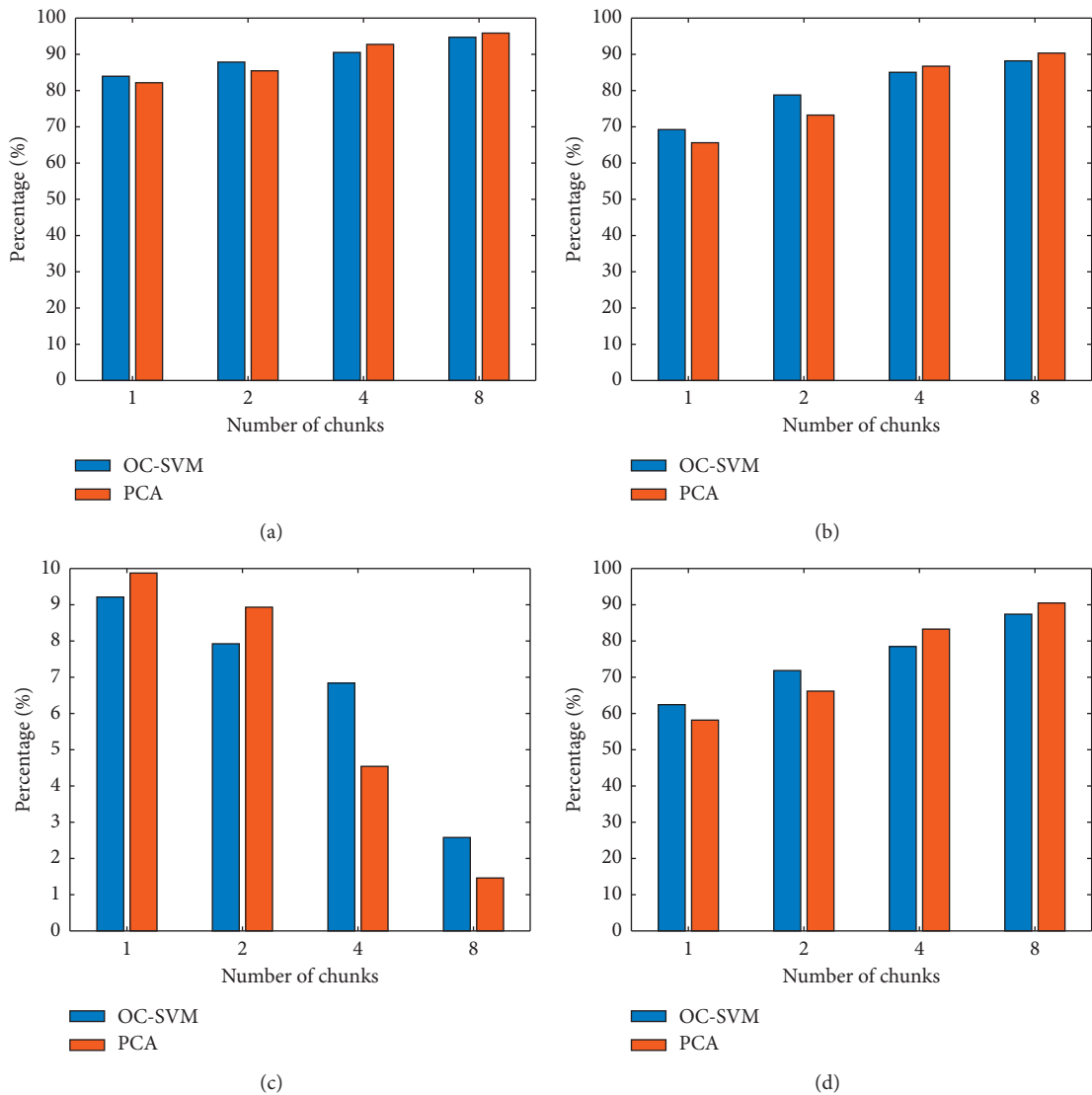


FIGURE 5: Comparison between some evaluation metrics of the used models. (a) Accuracy. (b) Recall. (c) FAR. (d) MCC.

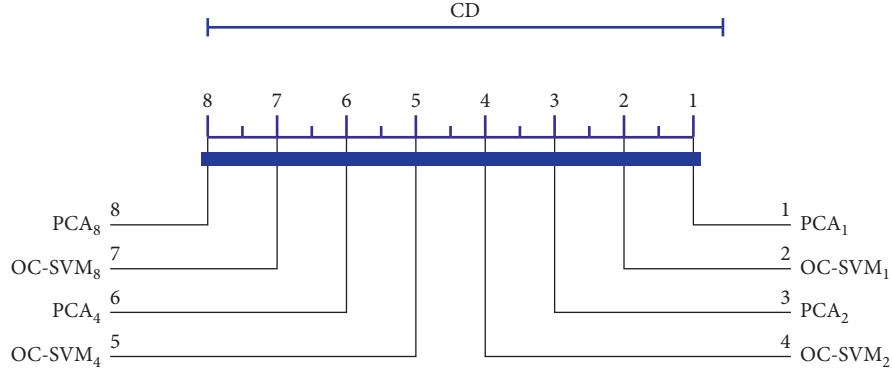


FIGURE 6: Critical difference diagram of the used models for different number of chunks.

Another critical issue in analyzing the performance of an electrical fault detection system is related to the ability of that system to detect faults regardless of type of faults. Indeed, there are mainly two types of faults in the electrical power system. Those are symmetrical and unsymmetrical faults [3]. Firstly, the symmetrical faults (also named as balanced faults) are considered as very serious but infrequent faults in the electrical power grids [6]. Specifically, the symmetrical faults come with two forms in the 3-phase grid, namely, the three lines to ground (L-L-L-G) and three lines (L-L-L) faults [8]. According to many practical studies, these faults are likely to occur with 2 to 5 percent in the entire electrical power system [7]. Although such faults rarely occur, their occurrence usually causes severe damage to both the electrical power system and equipment [7]. However, the entire electrical power system remains balanced even with that damage [7].

On the other hand, the unsymmetrical faults, also called unbalanced faults, are predominant and less dangerous than the symmetrical faults [8]. Three forms of unsymmetrical faults could occur in the 3-phase electrical grid, namely, the line to ground (L-G), line to line (L-L), and double line to ground (L-L-G) faults [7]. It was reported that they are very common in the electrical power systems with a percentage of 65% to 70% for the line to ground faults, 15% to 20% for the double line to ground faults, and 5% to 10% for the line to line faults [3]. Even though they are safer to the electrical power system and equipment than symmetrical faults, they cause unbalancing in the entire system in such a way that they generate unbalanced current to flow in the 3 phases [6].

Regarding the VSB dataset structure based on the fault types, there are 525 faulty samples: 4.5 percent of them are symmetrical faults, whereas 95.5 percent are unsymmetrical faults (line to ground = 69.15%, double line to ground = 18.74%, and line to line = 7.61%). This confirms the reliability of the VSB dataset, and it reflects the occurrence pattern of electrical faults in real-world electrical power grids. Figure 7(a) shows the percentage of detection rate for each of symmetrical and unsymmetrical faults when using the proposed system or not. Obviously, using the proposed system enhanced detection of the symmetrical faults by 43.98% and the unsymmetrical faults by 32.01% compared to those when not using the proposed system. Furthermore,

Figure 7(b) elaborates the impact of number of chunks in detecting the symmetrical and unsymmetrical faults. It can be perceived from Figure 7(b) that increasing the number of chunks helps both the OC-SVM and PCA models to detect more faults whether they are symmetrical or unsymmetrical. Therefore, the proposed system is effective considering detection of all types of electrical fault.

5.3. *ROC Analysis.* Alternatively, the ranking methods can be utilized to trade off between several models which are applied on the same dataset in order to choose the optimal classifier. One of widespread ranking methods is the receiver operating characteristic (ROC) curve. The ROC curve is a plot of the recall metric as a function of the FAR metric of a classifier [57]. The reference line with a model's performance equal to 50% is the diagonal line of the ROC curve. Moreover, the classifier reaches 100% of the performance only if its ROC curve located on the top-left corner. Figure 8 shows the ROC curves of the used models per number of chunks.

The area under the ROC curve (AUC) is a numerical measure that describes the corresponding ROC curve quantitatively [58]. The AUC value of a classifier is in $[0,1]$, where the higher the AUC value, the better the performance of the classifier. Furthermore, the AUC value of a classifier can be computed approximately using equation (29) [59]. Table 5 presents the AUC values of the ROC curves which are plotted in Figure 8. Based on findings, all used models have convenient performance particularly when the number of chunks is higher than 2. Moreover, the impact of the signal decomposition process on enhancing the performance of the used anomaly-based detection models is evident, especially when the number of chunks equals 8 in such a way that the AUC values for all models are higher than 0.9.

$$AUC = \frac{1}{2} \times (\text{recall} + \text{specificity}). \quad (29)$$

6. Discussion

In this section, the obtained results in Section 5 using several statistical and comparative analyses are deeply discussed.

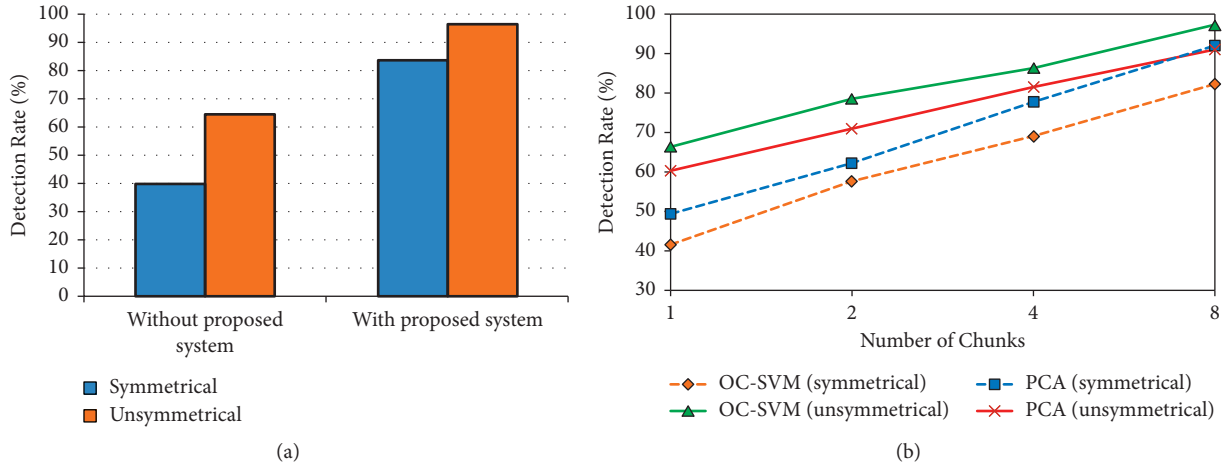


FIGURE 7: Percentage of detection rate for the symmetrical and unsymmetrical faults when (a) using the proposed system or not and (b) using different number of chunks for each model.

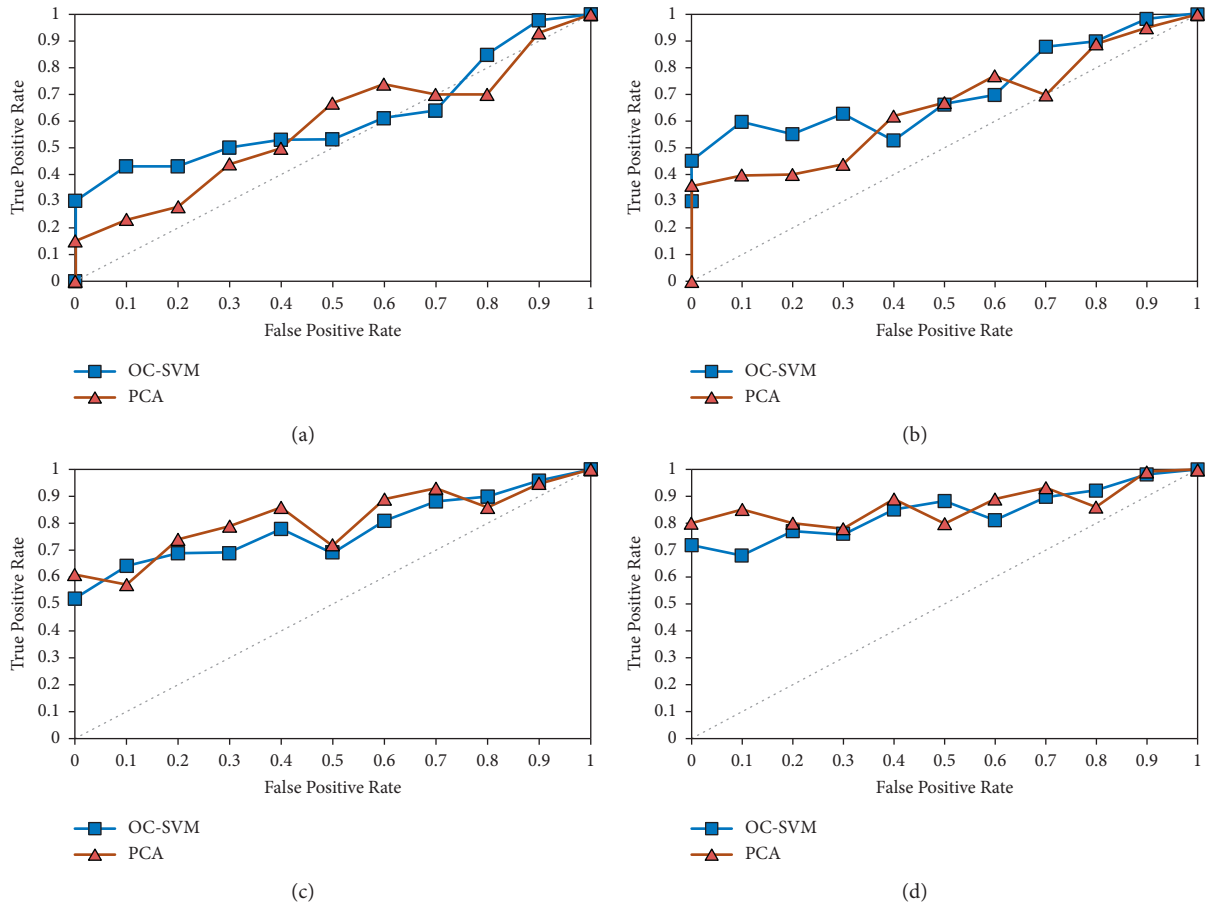


FIGURE 8: ROC curves of the used models for (a) 1 chunk, (b) 2 chunks, (c) 4 chunks, and (d) 8 chunks.

6.1. *Stability and Sensitivity Analyses.* To prove the stability of the proposed system and consistency of the results, some statistical tests can be performed. Due to space limitation, only the Friedman test is applied on the results in Table 4. Friedman test is a renowned statistical test which seeks to discover the distinctions between several repeated

treatments [60]. The Friedman test has many advantages such as its simplicity and generality, and it is a nonparametric test that assumes that your data do not come from a specific distribution. Practically, in Table 4, there are four experiments related to different number of chunks. In addition to that, in each experiment, there are two subjects

TABLE 5: AUC values of the ROC curves.

Number of chunks	Models	AUC
1	OC-SVM	0.8007
	PCA	0.7784
2	OC-SVM	0.8538
	PCA	0.8220
4	OC-SVM	0.8907
	PCA	0.9108
8	OC-SVM	0.9281
	PCA	0.9431

The bold values provide the best results between the used models for each number of chunks.

related to the used models. The Friedman test assumes in its null hypothesis that all these experiments have identical effects. To reject the null hypothesis, two main conditions must be satisfied. The first is that the critical value (FC) is less than the calculated statistic (FS). The second condition is that the significance level value (α) is larger than the calculated probability value (P_value). In this study, the α value is selected to be 0.05 because it is very common. Table 6 shows the results of the Friedman test when it is applied on TP, TN, FN, and FP outcomes. From Table 6, the null hypothesis of the Friedman test is rejected because the two conditions are satisfied in all cases ($FC < FS$ and $\alpha > P_value$). Therefore, the outcomes of the used models are significant and different from each other.

Sensitivity analysis is often applied in classification tasks to specify the relationship between input (independent) and target (dependent) variables under a given set of assumptions [61]. Due to space limitations, only the number of chunks as an input variable is analyzed to show its influence on the recall value of the used models as an output. Indeed, the sensitivity analysis can be done using various approaches, but the one-at-a-time (OAT) analysis is the most common approach. In the first step in the OAT analysis, the base case of the models is defined which in this study is the recall values of the used models with one chunk. Afterwards, the recall values of the used models will be calculated for different number of chunks, leaving all other assumptions unchanged. Finally, the sensitivity statistics will be calculated using the following formula [62].

$$\text{Sensitivity statistic} = \frac{\% \text{ change in the output variable}}{\% \text{ change in the input variable}} \quad (30)$$

The higher the sensitivity statistic is, the more sensitive the recall is to changes in the number of chunks. Table 7 presents the results of the OAT sensitivity analysis when the base case is one chunk. From Table 7, the recall value of the used anomaly detection models is sensitive to the number of chunks and it significantly increases as the number of chunks increases.

6.2. Feature Selection Methods. In this section, the impact of performing feature selection process in the data pre-processing phase is examined. Some recent feature selection methods such as the fitness proportionate selection binary particle swarm optimization and entropy (FPSBPSO-E)

[12, 63, 64], stochastic fractal search-based guided whale optimization algorithm (SFS-guided WOA) [65], hybrid of grey wolf optimization and particle swarm optimization (GWO-PSO) [66], hybrid of grey wolf optimization and genetic algorithm (GWO-GA) [66], biogeography-based optimizer (BBO) [67], firefly algorithm (FA) [68], and satin bowerbird optimizer (SBO) [69] are compared in terms of the mean of AUC and feature reduction rate (FRR) [12] metrics. The FRR metric is the complement of the ratio of selected features to all feature set and can be calculated as follows [12].

$$\text{FRR} = 1 - \frac{\text{number of selected features}}{\text{number of all features}} \quad (31)$$

The experiment is conducted on number of chunks equal to 1, that is, the size of full feature set is 20. Furthermore, the anomaly-based detection models are trained and tested on the feature subsets which are generated by the feature selection methods. Table 8 presents the results of feature selection process when using different methods. It can be perceived that the FPSBPSO-E method not only is better than other feature selection methods in terms of performance but also selected the smallest feature subset.

6.3. Hyperparameter Optimization Methods. The performance of the used PSO-based hyperparameter optimization method is evaluated by comparing it with other popular optimization algorithms such as the original genetic algorithm (GA) [70], grasshopper optimization algorithm (GOA) [71], whale optimization algorithm (WOA) [72], grey wolf optimization (GWO) [73], bat algorithm (BA) [74], and multiverse optimization (MVO) [75] in terms of the mean AUC values of the anomaly-based detection models. Table 9 shows the results of hyperparameter optimization process when using different algorithms. Clearly, the used PSO-based algorithm for hyperparameter selection outperformed other optimization algorithms.

6.4. Anomaly-Based Models vs. Binary Models. This section is dedicated to investigate performance differences between some binary classification models and the used anomaly-based detection models in the electrical fault detection problem. The binary classification models such as the ANN [45], support vector machine (SVM) [45], Naive Bayes (NB) [76], boosted

TABLE 6: Results of the Friedman test.

Outcome	FC	FS	α	P-value
TN	6	8	0.05	0.001 56
FP	6	8	0.05	0.001 56
TP	6	8	0.05	0.001 56
FN	6	8	0.05	0.001 56

TABLE 7: Results of the OAT sensitivity analysis.

Number of chunks	Sensitivity statistic (%)	
	OC-SVM	PCA
1	–	–
2	13.47	11.92
4	22.53	32.28
8	27.20	37.52

TABLE 8: Results of using some feature selection methods in the pretraining phase.

Feature selection method	Number of selected features	FRR (%)	AUC (%)
SFS-guided WOA	14	30	85.56
GWO-PSO	16	20	83.73
GWO-GA	15	25	84.82
BBO	17	15	82.65
FA	18	10	81.97
SBO	19	5	81.22
FPSBPSO-E	13	35	86.11

The bold values provide the best results among all the used feature selection methods.

TABLE 9: Results of using some hyperparameter selection algorithms in the pretraining phase.

Optimization algorithm	AUC (%)
GA	78.39
GOA	76.42
WOA	78.15
GWO	79.05
BA	75.54
MVO	77.70
PSO	80.76

The bold values provide the best results among all the used hyperparameter selection algorithms.

TABLE 10: Comparison between some binary classification and anomaly-based detection models (one chunk).

Detection method	Model	AUC (%)
Binary classification	ANN	52.16
	SVM	60.57
	NB	57.99
	BDT	61.23
	DF	65.55
	DJ	68.06
	QSVM	70.31
Anomaly-based models	OC-SVM	80.07
	PCA	77.84

The bold value provides the best result among all used models.

decision tree (BDT) [77], decision forest (DF) [78], decision jungle (DJ) [79], and quantum support vector machine (QSVM) [80] are utilized without using the proposed fault detection system. Then, their performance is compared to

performance of the OC-SVM and PCA anomaly-based detection models using the proposed system. Table 10 presents the results of binary classification and anomaly-based detection models in terms of the AUC metric. It can be noticed that the

TABLE 11: Comparison with some related works.

Study	Technique	Accuracy	Precision	Recall	F1 score
[34]	STL + SVM (linear kernel)	–	0.72	0.7	0.7
	STL + SVM (polynomial kernel with degree 6)	–	0.67	0.55	0.45
	STL + SVM (RBF kernel)	–	0.75	0.73	0.72
	STL + SVM (sigmoid kernel)	–	0.53	0.52	0.5
[36]	Anomaly detection + SVM (RBF kernel)	0.798 4	0.698 4	0.670 5	0.684 2
	Anomaly detection + PCA	0.792 8	0.667 3	0.725 7	0.695 3
Our study	Anomaly detection + PSO + OC-SVM	0.944 2	0.943 0	0.881 9	0.911 4
	Anomaly detection + PSO + PCA	0.957 8	0.967 3	0.901 0	0.932 9

The bold values provide the best results among all models.

anomaly-based detection models are superior to the binary classification models due to the cooperated mechanisms in the proposed fault detection system.

6.5. Comparison with Related Works. As explained in Section 2, most of the previous studies in electrical fault detection used simulated datasets which cannot reflect the pattern of electrical faults in the real-world electrical power grids. Our proposed anomaly-based fault detection system is compared only with two studies [34, 36], since they used the same real-time dataset (VSB dataset) in their works. Thus, the readers can compare the results easily. Table 11 presents a comparison between our findings and those of related works. Obviously, the proposed system enhanced the performance of all anomaly-based detection models compared to the models in the studies [34, 36]. This confirms the effectiveness of the proposed fault detection system in the electrical fault detection problem.

7. Conclusion

Effective fault detection in electrical power industry will lead to maximizing service quality as well as minimizing economic losses. Therefore, an anomaly-based fault detection system is proposed to cope with drawbacks and limitations of the existing electrical fault detection systems. The proposed system consists of four phases, namely, data preprocessing, pretraining, training, and testing. Most of the work is done in the first two phases. The data preprocessing phase prepares raw data of the signals by performing signal denoising, signal decomposition, feature extraction, data normalization, and dataset splitting. On the other hand, the pretraining phase selects the optimal hyperparameters needed by the anomaly-based fault detection model using a PSO-based algorithm. Moreover, two anomaly-based fault detection models leveraged from the proposed system are validated on the VSB dataset. The experimental results show a significant improvement in electrical fault detection which increased the true classification metrics such as the accuracy and recall metrics by 14% as well as decreased the misclassification metrics such as the FAR metric by 8% compared to same models without using the proposed system. Based on our results, the OC-SVM model is better than the PCA model only in the case of a low number of chunks, and vice versa. However, increasing the number of chunks led to

increase in the feature space that weakened the used models. Thus, using the proposed system with deep learning models for further enhancement in electrical fault detection is recommended as a possible future work.

Abbreviations

AML:	Azure Machine Learning
ANFIS:	Adaptive network fuzzy inference system
ANN:	Artificial neural network
AUC:	Area under the ROC curve
BA:	Bat algorithm
BBO:	Biogeography-based optimizer
BDT:	Boosted decision tree
BP:	Backpropagation
CDD:	Critical difference diagram
CD:	Critical difference
CFL:	Concurrent fuzzy logic
DF:	Decision forest
DJ:	Decision jungle
DR:	Detection rate
EHV:	Extra high voltage
FAR:	False alarm rate
FA:	Firefly algorithm
FNR:	False negative rate
FN:	False negative
FPR:	False positive rate
FPSBPSO-E:	Fitness proportionate selection binary particle swarm optimization and entropy
FP:	False positive
FRR:	Feature reduction rate
GA:	Genetic algorithm
GOA:	Grasshopper optimization algorithm
GRNN:	Generalized regression neural network
GWO:	Grey wolf optimization
IQR:	Interquartile range
MCC:	Matthews correlation coefficient
MVO:	Multiverse optimization
NB:	Naive Bayes
OAT:	One-at-a-time
OC-SVM:	One-class support vector machine
PCA:	Principal component analysis
PD:	Partial discharge
PNN:	Probabilistic neural network
PSO:	Particle swarm optimization
QSVM:	Quantum support vector machine

RBF:	Radial basis kernel function
ROC:	Receiver operating characteristic
RPROP:	Resilient propagation
SBO:	Satin bowerbird optimizer
SFS-guided	Stochastic fractal search-based guided whale
WOA:	Whale optimization algorithm
STL:	Seasonal and trend decomposition using loess
SVM:	Support vector machine
TNR:	True negative rate
TN:	True negative
TPR:	True positive rate
TP:	True positive
VSU:	Technical University of Ostrava
WOA:	Whale optimization algorithm
α :	Friedman test significance level value
ν :	Nu hyperparameter of OC-SVM model
\bar{X} :	Mean of voltage measurements of the signal
ζ :	Slack value
c :	Hypersphere center
FC:	Friedman test critical value
FS:	Friedman test calculated statistic
k :	Adjustment factor
M :	Number of chunks
Max:	Maximum value
Min:	Minimum value
N :	Number of signals in the VSB dataset
P :	Percentage value
P_value :	Friedman test calculated probability value
Q1:	25% percentile
Q3:	75% percentile
r :	Hypersphere radius
S :	A signal in the VSB dataset
S_f :	Filtered signal
X :	Voltage measurement of the signal
x :	Numeric feature
z :	Rank of voltage measurement
ϵ :	Stopping criteria.

Data Availability

The data used to support the findings of this study are available at <https://www.kaggle.com/c/vsb-power-line-fault-detection/data>.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

- [1] M. Wadi and W. Elmasry, "Statistical analysis of wind energy potential using different estimation methods for Weibull parameters: a case study," *Electrical Engineering*, vol. 103, pp. 2573–2594, 2021.
- [2] M. Wadi and W. Elmasry, "Modeling of wind energy potential in marmara region using different statistical distributions and genetic algorithms," in *Proceedings of the 2021 International Conference on Electric Power Engineering – Palestine (ICEPE-P)*, pp. 1–7, Gaza, Palestine, March 2021.
- [3] A. Raza, A. Benrabah, T. Alquthami, and M. Akmal, "A review of fault diagnosing methods in power transmission systems," *Applied Sciences*, vol. 10, no. 4, 1312 pages, 2020.
- [4] M. Wadi and M. Baysal, "Reliability assessment of radial networks via modified RBD analytical technique," *Sigma Journal of Engineering and Natural Sciences*, vol. 35, pp. 717–726, 2017.
- [5] M. Wadi, M. Baysal, and A. Shobole, "Comparison between open-ring and closed-ring grids reliability," in *Proceedings of the 2017 4th International Conference on Electrical and Electronic Engineering (ICEEE) IEEE*, pp. 290–294, Ankara, Turkey, August 2017.
- [6] A. Prasad, J. Belwin Edward, and K. Ravi, "A review on fault classification methodologies in power transmission systems: Part-I," *Journal of electrical systems and information technology*, vol. 5, no. 1, pp. 48–60, 2018.
- [7] A. Prasad, J. Belwin Edward, and K. Ravi, "A review on fault classification methodologies in power transmission systems: Part-II," *Journal of Electrical Systems and Information Technology*, vol. 5, no. 1, pp. 61–67, 2018.
- [8] K. Chen, C. Huang, and J. He, "Fault detection, classification and location for transmission lines and distribution systems: a review on the methods," *High Voltage*, vol. 1, no. 1, pp. 25–33, 2016.
- [9] V. Chandola, A. Banerjee, and V. Kumar, "Outlier detection: a survey," *ACM Computing Surveys*, vol. 1415 pages, 2007.
- [10] V. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artificial Intelligence Review*, vol. 22, no. 2, pp. 85–126, 2004.
- [11] W. Elmasry, A. Akbulut, and A. H. Zaim, "Empirical study on multiclass classification-based network intrusion detection," *Computational Intelligence*, vol. 35, no. 4, pp. 919–954, 2019.
- [12] W. Elmasry, A. Akbulut, and A. H. Zaim, "Evolving deep learning architectures for network intrusion detection using a double PSO metaheuristic," *Computer Networks*, vol. 168, Article ID 107042, 2020.
- [13] W. Elmasry, A. Akbulut, and A. H. Zaim, "Deep Learning Approaches for Predictive Masquerade Detection," *Security and Communication Networks*, vol. 2018, Article ID 9327215, 24 pages, 2018.
- [14] W. Elmasry, A. Akbulut, and A. H. Zaim, "A design of an integrated cloud-based intrusion detection system with third party cloud service," *Open Computer Science*, vol. 11, no. 1, pp. 365–379, 2021.
- [15] M. Jamil, S. K. Sharma, and R. Singh, "Fault detection and classification in electrical power transmission system using artificial neural network," *SpringerPlus*, vol. 4, no. 1, pp. 1–13, 2015.
- [16] R. Singh, "Fault detection of electric power transmission line by using neural network," *International Journal of Emerging Technology and Advanced Engineering*, vol. 2, no. 12, pp. 530–538, 2012.
- [17] E. Koley, A. Jain, A. Thoke, A. Jain, and S. Ghosh, "Detection and classification of faults on six phase transmission line using ANN," in *Proceedings of the 2011 2nd International Conference on Computer and Communication Technology (ICCCCT-2011) IEEE*, pp. 100–103, Allahabad, India, September 2011.
- [18] A. Jain, A. S. Thoke, and R. N. Patel, "Fault detection and fault classification of double circuit transmission line using artificial neural network," *International Journal of Electrical Systems Science and Engineering*, vol. 1, no. 4, pp. 750–755, 2008.

- [19] T. Bouthiba, "Artificial neural network-based fault location IN EHV transmission lines," *International Journal of Applied Mathematics and Computer Science*, vol. 14, no. 1, pp. 69–78, 2004.
- [20] M. Sanaye-Pasand and H. Khorashadi-Zadeh, "Transmission line fault detection & phase selection using ANN," in *Proceedings of the 2003 International Conference on Power Systems Transients*, pp. 1–6, New Orleans, LA, USA, August 2003.
- [21] U. Uzubi, A. Ekwue, and E. Ejiogu, "Artificial neural network technique for transmission line protection on Nigerian power system," in *Proceedings of the IEEE Power Engineering Society Conference and Exposition in Africa, PowerAfrica IEEE*, pp. 52–58, Accra, Ghana, 2017.
- [22] P. O. Mbamaluikem, A. A. Awelewa, and I. A. Samuel, "An artificial neural network-based intelligent fault classification system for the 33-kV Nigeria transmission line," *International Journal of Applied Engineering Research*, vol. 13, no. 2, pp. 1274–1285, 2018.
- [23] E. B. M. Tayeb, "Faults detection in power systems using artificial neural network," *American Journal of Engineering Research*, vol. 2, no. 6, pp. 69–75, 2013.
- [24] E. B. M. Tayeb and O. A. A. Rhim, "Transmission line faults detection, classification and location using artificial neural network," in *2011 International Conference & Utility Exhibition on Power and Energy Systems: Issues and Prospects for Asia (ICUE): IEEE*, pp. 1–5, Pattaya, Thailand, September 2011.
- [25] M. Othman, M. Mahfouf, and D. Linkens, "Transmission lines fault detection, classification and location using an intelligent power system stabiliser," *IEEE in Proceedings of the 2004 IEEE International Conference on Electric Utility Deregulation, Restructuring and Power Technologies*, vol. 1, pp. 360–365, Hong Kong, China, April 2004.
- [26] S. Ekici, S. Yildirim, and M. Poyraz, "A Neural Network Based Approach for Transmission Line Faults," in *5th International Conference on Electrical and Electronics Engineering, ELECO'07*, Turkey, January 2007.
- [27] P. S. P. Eboule, J. H. C. Pretorius, N. Mbuli, and C. Leke, "Fault detection and Location in power transmission line using concurrent neuro fuzzy technique," in *Proceedings of the 2018 IEEE Electrical Power and Energy Conference (EPEC) IEEE*, pp. 1–6, ON, Canada, October 2018.
- [28] R. Kumar, "Three phase transmission lines fault detection, classification and location," *International Journal of Science and Research (IJSR) vol. ISSN (Online): 2319-7064*, pp. 1–4, 2013.
- [29] E. Koley, K. Verma, and S. Ghosh, "An improved fault detection classification and location scheme based on wavelet transform and artificial neural network for six phase transmission line using single end data only," *SpringerPlus*, vol. 4, no. 1, pp. 1–22, 2015.
- [30] N. S. Wani, R. Singh, and M. U. Nemade, "Detection classification and localization of faults of transmission lines using wavelet transform and neural network," *International Journal of Applied Engineering Research*, vol. 13, no. 1, pp. 98–106, 2018.
- [31] M. Singh, B. Panigrahi, and R. Maheshwari, "Transmission line fault detection and classification," in *Proceedings of the 2011 International Conference on Emerging Trends in Electrical and Computer Technology IEEE*, pp. 15–22, Nagercoil, India, March 2011.
- [32] M. R. Singh, T. Chopra, R. Singh, and T. Chopra, "Fault classification in electric power transmission lines using support vector machine," *International Journal of Innovative Science and Research Technology*, vol. 1, no. 12, pp. 388–400, 2015.
- [33] S. S. Gururajapathy, H. Mokhlis, and H. A. B. Illias, "Classification and regression analysis using support vector machine for classifying and locating faults in a distribution system," *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 26, no. 6, pp. 3044–3056, 2018.
- [34] M. Dong, Z. Sun, and C. Wang, "A pattern recognition method for partial discharge detection on insulated overhead conductors," in *proceedings of the 2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE) IEEE*, pp. 1–4, AB, Canada, May 2019.
- [35] *VSB Power Line Fault Detection*, Kaggle, SF, USA, 2018, <https://www.kaggle.com/c/vsb-power-line-fault-detection/data>.
- [36] M. Wadi and W. Elmasry, "An anomaly-based technique for fault detection in power system networks," in *Proceedings of the 2021 International Conference on Electric Power Engineering – Palestine (ICEPE- P) IEEE*, pp. 1–6, Gaza, Palestine, March 2021.
- [37] ENET Centre, VSB, Czech Republic, 2020, <https://cenet.vsb.cz/en/>.
- [38] K. Naveed, M. T. Akhtar, M. F. Siddiqui, and N. ur Rehman, "A statistical approach to signal denoising based on data-driven multiscale representation," *Digital Signal Processing*, vol. 108, Article ID 102896, 2021.
- [39] D. García-Gil, J. Luengo, S. García, and F. Herrera, "Enabling smart data: noise filtering in big data classification," *Information Sciences*, vol. 479, pp. 135–152, 2019.
- [40] X. Yue, "Data Decomposition for Analytics of Engineering Systems: Literature Review, Methodology Formulation, and Future Trends," in *Proceedings of the 14th International Manufacturing Science and Engineering Conference*, vol. 58745, p. V001T02A011, 2019 American Society of Mechanical Engineers (ASME), PA, USA, June, 2019.
- [41] M. Yasir and B.-H. Koh, "Data decomposition techniques with multi-scale permutation entropy calculations for bearing fault diagnosis," *Sensors*, vol. 18, no. 4, 1278 pages, 2018.
- [42] R. J. Hyndman and Y. Fan, "Sample quantiles in statistical packages," *The American Statistician*, vol. 50, no. 4, pp. 361–365, 1996.
- [43] *Azure Machine Learning Studio*, Microsoft, NM, USA, 2020, <https://studio.azureml.net/>.
- [44] R. Barga, V. Fontama, and W. H. Tok, "Introducing microsoft azure machine learning," in *Predictive Analytics with Microsoft Azure Machine Learning*, pp. 21–43, Springer, 2015.
- [45] W. Elmasry, A. Akbulut, and A. H. Zaim, "Comparative evaluation of different classification techniques for masquerade attack detection," *International Journal of Information and Computer Security*, vol. 13, no. 2, pp. 187–209, 2020.
- [46] *One-Class Support Vector Machine*, Microsoft Docs, NM, USA, 2019, <https://docs.microsoft.com/en-us/azure/machine-learning/studio-module-reference/one-class-support-vector-machine>.
- [47] Z. Noumir, P. Honeine, and C. Richard, "On simple one-class classification methods," in *proceedings of the 2012 IEEE International Symposium on Information Theory Proceedings IEEE*, pp. 2022–2026, MA, USA, July 2012.
- [48] S. S. Khan and M. G. Madden, *A Survey of Recent Trends in One Class Classification, A Survey of Recent Trends in One Class Classification. In Proceedings of the Irish Conference on Artificial Intelligence and Cognitive Science Springer*, pp. 188–197, Dublin, Ireland, August 2009.

- [49] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Computation*, vol. 13, no. 7, pp. 1443–1471, 2001.
- [50] K. Pearson, "LIII. On lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, no. 11, pp. 559–572, 1901.
- [51] *PCA-based Anomaly Detection*, Microsoft Docs, WA, USA, 2019, <https://docs.microsoft.com/en-us/azure/machine-learning/studio-module-reference/pca-based-anomaly-detection>.
- [52] *Python*, 2019, <https://www.python.org>.
- [53] *NumPy*, 2019, <https://www.numpy.org>.
- [54] S. Boughorbel, F. Jarray, and M. El-Anbari, "Optimal classifier for imbalanced data using Matthews Correlation Coefficient metric," *PLoS One*, vol. 12, no. 6, Article ID e0177678, 2017.
- [55] B. W. Matthews, "Comparison of the predicted and observed secondary structure of T4 phage lysozyme," *Biochimica et Biophysica Acta (BBA) - Protein Structure*, vol. 405, no. 2, pp. 442–451, 1975.
- [56] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.
- [57] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006.
- [58] A. P. Bradley, "The use of the area under the ROC curve in the evaluation of machine learning algorithms," *Pattern Recognition*, vol. 30, no. 7, pp. 1145–1159, 1997.
- [59] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information Processing & Management*, vol. 45, no. 4, pp. 427–437, 2009.
- [60] M. Friedman, "The use of ranks to avoid the assumption of normality implicit in the analysis of variance," *Journal of the American Statistical Association*, vol. 32, no. 200, pp. 675–701, 1937.
- [61] S. Sahmoud, W. Elmasry, and S. Abudalfa, "Enhancement the security of AES against modern attacks by using variable key block cipher," *International Arab Journal of e-Technology*, vol. 3, no. 1, pp. 17–26, 2013.
- [62] M. Cem Kasapbaşı and W. Elmasry, "New LSB-based colour image steganography method to enhance the efficiency in payload capacity, security and integrity check," *Sādhanā*, vol. 43, no. 68, pp. 1–14, 2018.
- [63] L. Cervante, B. Xue, M. Zhang, and L. Shang, "Binary particle swarm optimisation for feature selection: a filter based approach," in *proceedings of the 2012 IEEE Congress on Evolutionary Computation IEEE*, pp. 1–8, QLD, Australia, June 2012.
- [64] Z. Zhou, X. Liu, P. Li, and L. Shang, "Feature selection method with proportionate fitness based binary particle swarm optimization," in *Proceedings of the Asia-Pacific Conference on Simulated Evolution and Learning*, pp. 582–592, Springer, Dunedin, New Zealand, December 2014.
- [65] E.-S. M. El-Kenawy, A. Ibrahim, S. Mirjalili, M. M. Eid, and S. E. Hussein, "Novel feature selection and voting classifier algorithms for COVID-19 classification in CT images," *IEEE Access*, vol. 8, pp. 179317–179335, 2020.
- [66] F. A. Şenel, F. Gökçe, A. S. Yüksel, and T. Yiğit, "A novel hybrid PSO-GWO algorithm for optimization problems," *Engineering with Computers*, vol. 35, no. 4, pp. 1359–1373, 2019.
- [67] D. Simon, "Biogeography-based optimization," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 6, pp. 702–713, 2008.
- [68] I. Fister, X. S. Yang, I. Fister, and J. Brest, "Memetic Firefly Algorithm for Combinatorial Optimization," arXiv preprint arXiv:1204.5165, 2012.
- [69] S. H. Samareh Moosavi and V. Khatibi Bardsiri, "Satin bowerbird optimizer: a new optimization algorithm to optimize ANFIS for software development effort estimation," *Engineering Applications of Artificial Intelligence*, vol. 60, pp. 1–15, 2017.
- [70] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*, MIT press, MA, USA, 1992.
- [71] S. Saremi, S. Mirjalili, and A. Lewis, "Grasshopper optimization algorithm: theory and application," *Advances in Engineering Software*, vol. 105, pp. 30–47, 2017.
- [72] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, 2016.
- [73] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.
- [74] I. Karakonstantis and A. Vlachos, "Bat algorithm applied to continuous constrained optimization problems," *Journal of Information and Optimization Sciences*, vol. 42, no. 1, pp. 57–75, 2021.
- [75] S. Mirjalili, S. M. Mirjalili, and A. Hatamlou, "Multi-verse optimizer: a nature-inspired algorithm for global optimization," *Neural Computing & Applications*, vol. 27, no. 2, pp. 495–513, 2016.
- [76] H. Zeineddine, U. Braendle, and A. Farah, "Enhancing prediction of student success: automated machine learning approach," *Computers & Electrical Engineering*, vol. 89, p. 106903, 2021.
- [77] *Two-Class Boosted Decision Tree*, Microsoft Docs, WA, USA, 2019, <https://docs.microsoft.com/en-us/azure/machine-learning/studio-module-reference/two-class-boosted-decision-tree>.
- [78] *Two-Class Decision Forest*, Microsoft Docs, WA, USA, 2019, <https://docs.microsoft.com/en-us/azure/machine-learning/studio-module-reference/two-class-decision-forest>.
- [79] *Two-Class Decision Jungle*, Microsoft Docs, WA, USA, 2019, <https://docs.microsoft.com/en-us/azure/machine-learning/studio-module-reference/two-class-decision-jungle>.
- [80] V. Havlíček, A. D. Córcoles, K. Temme et al., "Supervised learning with quantum-enhanced feature spaces," *Nature*, vol. 567, no. 7747, pp. 209–212, 2019.