mathematics

MDPI

Article

Deep Machine Learning Model-Based Cyber-Attacks Detection in Smart Power Systems

Abdulaziz Almalaq [1,*], Saleh Albadran [1] and Mohamed A. Mohamed [2,*]

[1] Department of Electrical Engineering, Engineering College, University of Ha'il, Ha'il 55476, Saudi Arabia; s.abadran@uoh.edu.sa
[2] Electrical Engineering Department, Faculty of Engineering, Minia University, Minia 61519, Egypt
* Correspondence: a.almalaq@uoh.edu.sa (A.A.); dr.mohamed.abdelaziz@mu.edu.eg (M.A.M.)

Abstract: Modern intelligent energy grids enable energy supply and consumption to be efficiently managed while simultaneously avoiding a variety of security risks. System disturbances can be caused by both naturally occurring and human-made events. Operators should be aware of the different kinds and causes of disturbances in the energy systems to make informed decisions and respond accordingly. This study addresses this problem by proposing an attack detection model on the basis of deep learning for energy systems, which could be trained utilizing data and logs gathered through phasor measurement units (PMUs). Property or specification making is used to create features, and data are sent to various machine learning methods, of which random forest has been selected as the basic classifier of AdaBoost. Open-source simulated energy system data are used to test the model containing 37 energy system event case studies. In the end, the suggested model has been compared with other layouts according to various assessment metrics. The simulation outcomes showed that this model achieves a detection rate of 93.6% and an accuracy rate of 93.91%, which is greater compared to the existing methods.

Keywords: cyber-attack detection; deep machine learning; smart power grid; data processing

MSC: 94-10

Citation: Almalaq, A.; Albadran, S.; Mohamed, M.A. Deep Machine Learning Model-Based Cyber-Attacks Detection in Smart Power Systems. *Mathematics* **2022**, *10*, 2574. https://doi.org/10.3390/math10152574

Academic Editors: Gurami Tsitsiashvili and Alexander Bochkov

Received: 6 June 2022
Accepted: 22 July 2022
Published: 25 July 2022

1. Introduction

1.1. Necessity of the Research

Cyber-physical systems (CPS) attempt to couple the physical and cyber-worlds, and they are extensively employed by industrial control systems (ICS) to provide users with all the data they need in real-time [1]. Power distribution systems and waste-water treatment plants are among the areas where CPS is being used. Nevertheless, CPS security problems differ from conventional cyber-security problems in that they include integrity, confidentiality, and availability. In addition to transmitting, distributing, monitoring, and controlling electricity, a smart grid (SG) would greatly enhance energy effectiveness and reliability. Such systems may fail and result in temporary damage to infrastructures [2]. Power grids are regarded as essential infrastructure nowadays by many societies, which have developed security measures and policies related to them [3]. Phasor measurement units (PMUs) are adopted in modern electrical systems to improve reliability as they become more complex in their structure and design. Utilizing the gathered information for quick decision making is one of the advantages. There is still the possibility that hacker exploits vulnerabilities to result in branch overloaded tripping, which will lead to cascading failures and, therefore, leads to considerable damage to SG systems [4]. As the operators monitor and manage the energy grid, they must consider possible attacks on the grid. To accomplish this, much energy and grid expertise is required. However, deep machine learning (DML) methods are used because of their capability to recognize patterns and learn, as well as being quickly able to identify potential security boundaries [5].

1.2. Literature Review

Network systems, usually referred to as essential infrastructure systems, have been usually applied to link the systems for monitoring and collecting equipment operations in real-time. The supervisory control and data acquisition (SCADA) system is highly vulnerable to cyber-attacks, and such attacks need to be handled with extreme caution [6]. Sensor's fingerprints and noise processing are used in [7] for detecting hidden cyber-attacks in CPS, and the data set from the actual-world water treatment plants is employed to validate the approach, and the outcomes indicated an accuracy of 98%. In [8], a semantic instruction detection system on the basis of the network was examined for detecting attacks on water plant processes by analyzing network traffic. These findings highlight the need for CPS investigation. Cyber and physical systems are part of the SG. Intrusion detection problems are solved using DML, as seen in recent research [9–11]. The intrusion detection method on the basis of DML is examined in [9]. The data set employed was a SWAT-produced datum from various attacks of 10 various kinds. A quick one-class classification scheme that overcomes the problem of vast sensitivity to out-of-range data is employed in [10], and an actual data set is used to test the suggested algorithm. The data sets employed in this study have also been utilized in numerous other types of research. The authors in [11] examined the method with accuracy rates of around 90% for JRipper + Adaboost and 75% for random forest compared to the whole multiclass data set. The privacy preservation intrusion diagnosing method on the basis of the correlation coefficient and expectation maximization (EM) clustering techniques is presented in [12] to select significant sections of data and recognize intrusive occurrences. There was an 88.9% recall rate in the model compared to the multiclass data sets with 75% of features. Authors in [13] have improved the detection process by dropping the defense target from rejecting attacks to preventing outages to decreasing the necessary number of secured PMUs. In [14], the authors investigated the effect of cyber-attack on the PMU state estimation process using the Cartesian equations and in the case of zero injection buses. In [15], it is tried to develop an allocation method for fault observability using PMU data considering zero injection buses. In [16], the authors have introduced a fault detecting and classifying, and placement approach based on advanced machine learning in radial distribution systems.

1.3. Contributions

A model based on machine learning is presented in this study for detecting system behaviors by analyzing historical data and related log data. Although unsupervised learning is beneficial for detecting zero-day attacks since it requires no training in attack scenarios, it is also vulnerable to false positives [17]. Furthermore, supervised learning can clearly improve the detection's confidence. The experiments are then performed using the supervised machine learning approach. The main contributions in this paper are summarized as follows:

(1) Feature construction engineering is performed, and 16 novel features are constructed via an analysis of the features and possible links of the raw data in the electrical network. It is possible to construct novel features using a combination of attributes that could help more effectively utilize possible types of data instances, which could be used in machine learning models for better application.

(2) A new process for handling abnormal data, such as not the number and infinity amounts in the data sets, is proposed. The suggested approach could significantly enhance accuracy in comparison to conventional processes of processing abnormal data.

(3) A classification model based on machine learning is constructed. The average accuracy of 0.9389, precision of 0.938, recall of 0.936, and F1 score of 0.935 on 15 data sets demonstrate that the suggested model successfully distinguished 37 kinds of behaviors such as power grid fault and single-line-to-ground (SLG) fault replay, relay setting varies, and trip command injection attacks.

Following are the remaining sections of the study. A detailed explanation of the methodology is provided in Section 2. The results of the classification are discussed in Section 3. The conclusion appears in Section 4.

2. Model Structure

Scenarios where disturbances and attacks happen in the electric grid, as well as the meaning of features in the data set, are presented in this part. The suggested model and data processing are detailed here.

2.1. Introduction to Power System Framework Configuration

The suggested data set consisting of measurements associated with normal, fault, and cyber-attack behavior, and so on [18–20]. The electrical network block diagram is shown in Figure 1 [21]. Relay, control panel, snort, and PMU/synchronous are primarily used for recording measurement data. Following are some of the most significant components. Power generators are shown by P1 and P2, and the intelligent electronic device (IED) is relay R1, which could switch breaker1 (BR1) on or off. Transmission lines (TLs) are represented by L1 and L2. The phasor data concentrator is shown by PDC that stores and displays Synchron-phasor data as well as records historical data. The IED incorporates a distance protection mechanism that can trip the breaker if it detects faults. Due to the absence of internal verification approaches for detecting changes, the breaker will be tripped regardless of whether the fault is valid or not. BR1-4 can be tripped by manually sending relevant commands to IEDs. In the event that lines or other components are to be maintained, the manual override will be necessary.

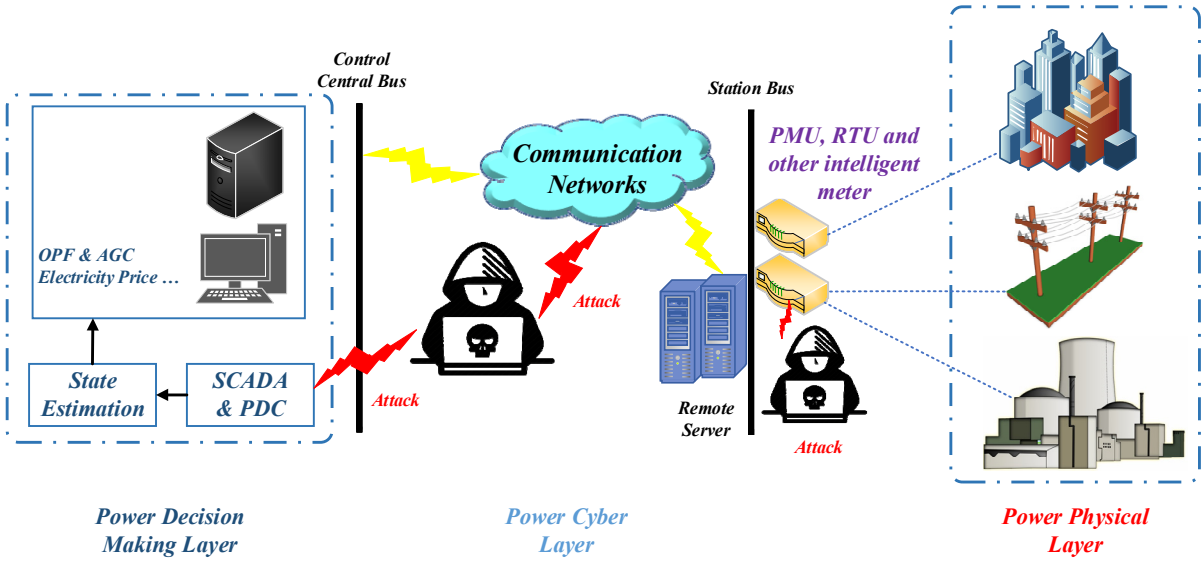


Figure 1. The power system framework configuration.

This experiment applied a data set that contains 128 features recorded using PMUs 1 to 4 and relay snort alarms and logs (Relay and PMU have been combined). A synchronous phasor, or PMU, measures electric waves on a power network using a common time source. A total of 29 features could be measured by every PMU. The data set also contains 12 columns of log data from the control panel and one column of an actual tag. There are three main categories of scenarios in the multiclass classification data set: No Events, Events, Intrusion, and Natural Events. Table 1 summarizes the scenarios, and a brief explanation of each category is provided in the data set.

(a) SLG fault: A fault occurs whenever the current, voltage frequency of the system changes abnormally, and many faults in electrical systems occur in line-to-ground and

line-to-line (LL). The simulated SLG faults are represented as short circuits at diverse points along the TL in the data set.

(b) Line maintenance: This type of attack is caused when one or more relays have been deactivated on a particular line to maintain.

(c) Data injection: More research is being conducted into false data injection state estimation in electrical networks. False data injection attacks are one of the main forms of network attacks, which could affect the power system estimation method. Attackers alter phase angles in order to create false sensor signals. The objective of such attacks is to blind the operators and to avoid raising an alarm, which could lead to economic or physical damage to the electrical systems. Attackers synchronize the phasor measurement with the fault's SLG and next send a relay trip command on the affected lines. A data set modeled the conditions by varying variables, such as current, voltage, and sequence components, which caused faults on various levels ([10 to 19]%, [20 to 79]%, [80 to 90]%) of the TLs.

(d) Remote tripping command injection attack: This occurs when a computer on the communications network uses unexpected relay trip commands to relay at the end of a TL. For achieving attacks, command injection has been applied versus single relays (R [1–4]) or double relays (R3 and R4, R1 and R2).

(e) Relay adjusting variation attack: The relay is configured with a distance protection layout. Attackers change the setting, so the relay responds badly to authentic faults. In the data sets, faults were caused via deactivating the relay functions at diverse parts of TLs with R1 or R2 or R3 or R4 deactivated and fault.

Table 1. Explanation of scenarios.

Case Study No.	**41**	**1–6**	**13, 14**	**7–12**	**15–20**	**21–30, 35–40**
Explanation	Usual operation load variations	SLG faults	Line maintenance	Data injection	Remote tripping command injection	Relay setting vary
Kind	No events	Natural events			Intrusion events	

2.2. Methodology

Despite the fact that the machine learning approach is capable of detecting disturbances and cyber-attacks on electric grids, it can have these drawbacks. Currently, references just discuss how to diagnose attacks in the electrical grids and seldom examine the data relationship. In contrast, when working with multi-classification problems, many algorithms convert them into multi-two-class situations. Nonetheless, the AdaBoost algorithm is able to handle multi-classification situations directly. It utilizes weak classifiers well for cascading and is capable of using various classification algorithms as weak classifiers. In terms of the error rate of misclassification, the AdaBoost algorithm is highly competitive [22]. With an increase in data amount, the fitting ability is affected both by generalization problems and by the increasing difficulty of computing. Machine learning requires a large amount of calculating to find the best solution. Additionally, the accuracy rates on the model presented in [11,12] are about 90% compared to the multiclass data sets, which provides considerable space for development. As a consequence of these findings, this paper constructs a model that can perform superior feature engineering and next can split the data by the diverse PMUs to minimize computation overhead. It should be noted that the PMU allocation in the smart grid is performed in the planning stage and might be implemented according to different purposes. While the high cost might be a limitation, the high number of PMUs is always preferred to cover all areas of the smart grid. It is worth noting that PMU allocation is out of the scope of this work but can be found in other research works widely. In addition, the AdaBoost algorithm for detecting the 37-class fault and cyber-attack case studies in the electric grids is adopted in this paper.

About the feature selection process, it should be noted that this experiment applied a data set that contains 128 features recorded using PMUs 1 to 4 and relay snort alarms and logs (relay and PMU have been combined). Please also note that each PMU can record 29 different features. In this regard, and in order to obtain enriched and integrated informative data, feature construction engineering is performed, and 16 novel features are constructed via an analysis of the features and possible links of the raw data in the electrical network. Technically, it is possible to construct novel features using a combination of attributes that could help more effectively utilize possible types of data instances, which could be used in machine learning models for better application. It is worth noting that we made use of the random forest method to create and classify features. Finally, based on anticipation weighted voting, 37 various case studies were implemented for simulation purposes.

2.3. Diagnosing Attack Behavior Model Structure

A model architecture diagram is shown in Figure 2 to detect faults and cyber-attack in electrical grids. According to Figure 2, the model architecture usually consists of four stages: property making, data dividing, weight voting, and layout training as follows:

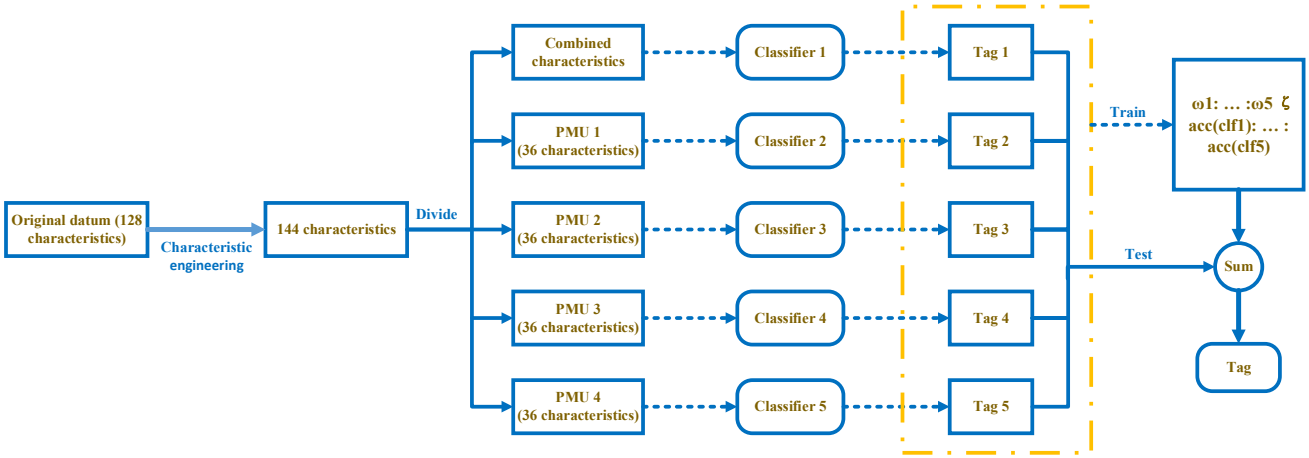


Figure 2. Explanation of layout to detect disturbance and cyber-attack in electrical networks.

Stage.1. Property making. By creating novel features manually from the original data set, it is able to improve the dimension of the data. A novel piece of data is generated by integrating the novel features with several original ones. The upper limit of the model is determined by the features and data, and the algorithm can just approximate the upper limit as closely as feasible. In order to achieve maximum accuracy and improve robustness, feature construction engineering is essential. It is important for feature construction using the original data to obtain more flexible features, and therefore increase data sensitivity and increase the ability to analyze it in the case of sending it to models for classification and training. The target of helpful features is to be simple to understand and maintain. The results of the analysis have led to the construction of 16 novel features. There is also a tendency in machine learning problems to include a large number of features for training instances, and it results in excessive computational overhead and overfitting, leading to poor efficiency. The curse of dimensionality has usually been used to describe this problem. Feature selection and feature extraction have been widely applied to mitigating the problems caused by high dimensionality in learning problems [23].

Stage.2. Datum dividing and training. The test and training sets are divided through 9:1 through the data splitting module. There is too much noise in the classifier if too many features are used [24]; therefore, every original data has been split into four parts according to features from various PMUs. While doing this, a section of the main characteristics is picked and sent to the AdaBoost layout to train alongside the novel features as well. This step is necessary for reducing the effect of errors resulting from bad PMU measurements. In case the feature dimension increases, the classifier's performance decreases. As a result

of this step, several of the original features are combined with novel ones in order to reduce the dimension. The original features are sorted using feature importance, and afterward, a variety of proportions of the features are selected, explained in more detail in Part 3. In addition, several classifier models are developed for personalizing the features following splitting. Various classifiers are set up to make every section of the data display the greatest impact on the classifier, i.e., the training model. Using five classifiers and later obtaining five tags following transferring the information to the layout reduces the effect of the alone classifier generalization error.

Stage.3. Weights for voting. It is the responsibility of the module to assign diverse weights to the tags derived from diverse classifiers and vote on the last classification tag of the data. According to the accuracy ratio of every classifier in the training set, the ratio of various weights has been thus determined. Various tags are generated by the test set following they have passed through the trained classifier, and the weights are determined for the last voting session based on the tags of the relevant classifier. By updating the weights in real-time, the entire system can become more robust and generalizable.

2.4. In-Depth Explanation of the Attack-Diagnosing Layout

2.4.1. Properties Making

During property making, 16 novel features have been extracted from every PMU measurement feature and incorporated into the original data set for preparing for the next step. Raw data is mainly used for extracting novel features based on corresponding computations. Table 2 shows the name, explanation, and extraction process of the extracted feature.

Table 2. Explanation of extracted characteristics.

Feature	**VCA4**	**VCA1**	**SI**
Description	PA7:VH-PA 10:IH	Sin (PA1:VH-P4:PA4:IH-PA7:VH-PA10:IH)	Sin (PA4:IH-PA 10:IH)
Feature	**SV**	**VCM1**	**VCM2**
Description	Sin (PA1:VH-PA 7:VH)	(PM1:V-PM7:V)/ (PM4:1-PM10:I)	(PM2:V-PM8:V)/ (PM5:I-PM11:I)

2.4.2. Data Processing

It is important to process the data prior to sending it to the machine learning model. The normalization of the data is an important part of data processing. The benefit of this method is that it speeds up and improves the accuracy of iterations for finding the best solution for gradient descent. Among the most common techniques of data normalization are z-score standardization and min-max standardization. Basically, min-max standardization works by changing the original data linearly toward an outcome between [0, 1] shown below:

$$X_{scale} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (1)$$

In addition, Z-score standardization has been known as standard deviation standardization, and it has been mostly applied for characterizing deviations from the average. The data analyzed through this technique assure the standard usual distribution, which is that the standard deviation and average are equal to one and zero, respectively. The data processed using the process can satisfy the standard normal distribution, meaning the mean equals 0 and the standard deviation Equation (1). Following is the transformation function, the mean amount of the instant data is shown by μ, and the standard deviation is represented by σ. This study adopts this normalization process.

$$X_{scale} = \frac{x - \mu}{\sigma} \quad (2)$$

A data set may contain the not a number (NAN) and infinity (INF) amount, but it has been usually substituted through the mean amount or zero. For the data set applied here, the novel replacement process is proposed to avoid underflows in the final replacement value and the data being overly discrete. *log_mean* value is used for replacing NAN and INF values present in the data. It can be calculated as follows:

$$log\_mean = \frac{\sum \log|k_i|}{Num(k_i)} \cdot \left(1 - 2 \lessdot \left(\frac{\sum k_i}{Num(k_i)} < 0\right)\right) \quad (3)$$

Here, the number of digits in a column is shown by $Num(k_i)$ and the indicator function is represented by $\lessdot(x)$, which can be described in the following way:

$$\lessdot(x) = \begin{cases} 1 \; if \; x \; is \; true \\ 0 \; otherwise \end{cases} \quad (4)$$

Comparative experiments are conducted on various treatment approaches in this study. Section 3 shows the outcomes that show that the suggested process succeeds.

2.4.3. Establish Classifier Layouts

During the process of making the classifier scheme, the features and characteristics of the SG information are considered, and various DML classification schemes are established for the data obtained from every PMU. Various experiments have shown that random forest is the best for the data gathered through every PMU, and AdaBoost is the ideal layout for combined features, including a section of the main characteristics as well as properties derived from the property making. With AdaBoost, several basic classifiers are combined into a robust classifier. The experiment proposes a new model in which random forest has been applied as the basic classifier of AdaBoost, followed by weighted voting on the anticipation outcomes (AWV).

Stage. (1) Set the training data's weights of observation $= (\omega_1, \ldots \omega_2, \ldots \omega_n)$ $\omega_i = 1/n$.
Stage. (2) For $t = 1$:T

(I) Select random forest classifier $RFC^{(t)}$ as the base classifier of Adaboost.

(II) Calculate classification error $\varepsilon^{(t)} = \sum_{i=1}^{n} \omega_i^{(t)} \lessdot (y_i \neq RFC^{(t)}(X_i) / \sum_{i=1}^{n} \omega_i{}^{(t)}$

Here, X_i shows the ith input feature vector, the actual tag of the ith input property vector is represented by y_i. The predicted outcome is shown through $RFC^{(t)}(X_i)$.

(III) Calculate $\alpha^{(t)} = 0.5 \ln\left(\frac{1-\varepsilon^{(t)}}{\varepsilon^{(t)}}\right)$.

(IV) Update the weights through $\omega_i^{(t+1)} = \omega_i^{(t)} \exp\left(\alpha^{(t)} \lessdot \left(y_i \neq RFC^{(t)}(X_i)\right)\right)$

(V) Renormalize so that $\sum_{i=1}^{n} \omega_i = 1$.

Stage. (3) Output $C(x) = argmax_y \sum_{t=1}^{T} \alpha^{(t)} \lessdot (RFC^{(t)}(X) = y$

Here, $argmax_x(f(x))$ function is meaned return the amount of x which maximizes $f(x)$. Here, for 37-class classification problem, so $\in (1, 2, \ldots, 37)$, and $\sum_{t=1}^{T} \alpha^{(t)} \lessdot (RFC^{(t)}(X) = y$ is a 37-dimensional vector. When various probabilities are associated with various tags for one feature vector X_i, the last output is determined through the probability with the highest amount.

2.4.4. Voting with Weights

Hard combination and soft combination are two ways of addressing the final multiple tags [25]. The hard combination is training the similar data set section with various DML methods and assigning the similar weight to the achieved last tags for voting. The result is the tag with the highest weight value. Similar to that, the soft combination involves adopting various DML methods for a similar section of the data set. However, the tags are assigned with different weights, and the end result is the tag with the highest weight. To summarize, the main difference between the hard and soft combinations is whether or not

the weights are equal. In a classifier, weights represent the probability value of a tag or its confidence level. The present study sets up various machine learning models for various data blocks to address multi-tag problems so as to make the model perform effectively for the data set. Lastly, different weights are assigned to tags to determine the final results. Algorithm 1 describes these steps.

Algorithm 1: Weight Voting Scheme

Input:	144 characteristics
Output:	Tag
(1)	Divide data by random Num (training set):Num (test set) = 9:1
(2)	Divide 144 characteristics into 4 section $PMU_i\_charectristics$ (i = 1, 2, 3, 4)
(3)	Transfer training set to the various machine learning; layout and take the precision rate acc (clf_i) (i = 1, 2, 3, 4, 5)
(4)	Transfer trail information to the trained layout and produce five tags; $label_i$ (i = 1, 2, 3, 4, 5)
(5)	Initialize weight ω_i (i = 1, 2, 3, 4, 5) and $\omega_1 : \ldots : \omega_5 \approx acc$ (AdaBoost) : acc(RFC_1) : . . . : acc(RFC_4)
(6)	Merge tags with weights $[[abel_1, w_1], \ldots, [abel_5, w_5]]$
(7)	Constitute a tag set (tag), and compute the weight set W regarding the tag in the set
(8)	Chose the tag with the largest weight in the W as the last outcome

3. Experiment and Evaluation

In machine learning, classifications and regressions are the primary learning tasks. It is obvious that the classification problem is addressed in this study. The next experiments are designed to test whether the model structure described in this study is capable of distinguishing fault and disturbance in electrical systems. A comparison is made between the model and various conventional models, such as convolution neural network (CNN), gradient boosting decision tree (GBDT), extreme gradient boosting (XGBoost), decision tree (DT), support vector machine (SVM), and k-nearest neighbor (KNN).

Additionally, the accuracy achieved through transferring information has been compared after the property making is compared.

3.1. Data Set

A multiclass classification data set for ICS cyber-attacks is used in the present study. There are a total of 15 groups in the multiclass data set, each with about 5000 pieces of data. Each group's situation is shown in Table 3. Across all tag kinds, the distribution of data can be fairly uniform. ARFF (Attribute-Relation File Format) is the main file template of the data set. An ARFF file is the ASCII text format, which represents a set of attributes shared by several samples. To ease the process, ARFF files are converted to CSV (Comma Separated Values) template. In CSV files, textual/numeric tabular information is stored in plain text. AUC, F1 score, ROC curve, ROC curve, precision, accuracy, and recall area are primarily used to evaluate classification models in machine learning. There are several terms applied in machine learning that require an explanation. The true positive (TP) is the positive sample that the layout predicts to be positive, the false positive (FP) is the negative sample that the layout predicts to be positive, and the false negative (FN) is the positive sample that the model predicts to be negative, the true negative (TN) is the negative sample that the model predicts to be negative. The suggested layout is evaluated using accuracy, precision, recall, and F1 score. An F1 score is basically the harmonic value of precision and recall, which are calculated according to the following equations:

$$accuracy = (TP + TN)/(TP + FP + FN + TN) \tag{5}$$

$$precision = TP/(TP + FP) \tag{6}$$

$$recall = TP/(TP + FN) \tag{7}$$

$$F1\ score = \frac{2TP}{2TP + FN + FP} = \frac{2 \cdot precision \cdot recall}{precision + recall} \tag{8}$$

Table 3. Multiclass instance data statistics.

Data set	**Data 1**	**Data 2**	**Data 3**	**Data 4**	**Data 5**	**Data 6**	**Data 7**	**Data 8**
Data number	4966	5069	5415	5202	5161	4967	5236	5315
Data set	**Data 9**	**Data 10**	**Data 11**	**Data 12**	**Data 13**	**Data 14**	**Data 15**	**Entire**
Data number	5340	5569	5251	5224	5271	5115	5276	78,377

3.2. Experiment Outcome

3.2.1. Machine Learning Model

In this experiment, KNN, SVM, GBDT, XGBoost, CNN, and others were applied as conventional models.

(A) Based on the distance among feature values, the K-nearest neighbor algorithm has been categorized. Distance is calculated primarily using Euclidean/Manhattan distances formulation.

(B) The SVM [26] layout uses the sample as a spot in the region and applies various mapping functions for mapping the input into the great-dimensional property region for constructing the hyperplane group or hyperplane. According to intuition, the further away the boundary is from the point of data training, the more accurate the classification will be. $\omega^T x + b = 0$ shows the formulation to divide the hyperplane, in which the normal vector is shown by ω determining the hyperplane's direction., and the displacement term is shown by b determining the distance between the hyperplane and the origin. $\gamma = \left(\omega^T x + b\right)/||\omega||$ show the formulation for the interval from each spot x to the hyperplane in the region, γ must be maximized within the conditions, which the hyperplane properly divides the training instances, i.e.:

$$\begin{gathered} \max_{\omega,b} \frac{2}{||\omega||} \\ subject\ to\ y_i\left(\omega^T x + b\right) \geq 1 \end{gathered} \tag{9}$$

Calculating the limitation problem via the Lagrange function is more efficient, and an objective function can be derived from the following formula, in which α_i shows the Lagrange multiplier and $\alpha_i \geq 0$.

$$L(\omega, b, \alpha) = \frac{1}{2}||\omega||^2 + \sum_{i=1}^{m} \alpha_i \left(1 - y_i\left(\omega^T x + b\right)\right) \tag{10}$$

Determine $L(\omega, b, \alpha)'s$ partial derivatives and make them 0:

$$\frac{\partial L(\omega, b, \alpha)}{\partial \omega} = 0, \frac{\partial L(\omega, b, \alpha)}{\partial b} = 0 \tag{11}$$

The dual problem can be as follows:

$$\max_{\alpha} \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j x_i^T x_i \ \ subject\ to\ \sum_{i=1}^{m} \alpha_i y_i = 0,\ \alpha_i \geq 0 \tag{12}$$

(C) The decision tree algorithm starts with a group of instances/cases and then makes a tree information framework, which is applied to novel cases. A group of amounts/symbolic amounts describes every case [27]. Entropy is used in C4.5 and C5.0 for the spanning tree algorithm.

(D) A boosting algorithm has been used to improve the XGBoost [28] classifier algorithm. The model is based on residual lifting. Based on the error function, the objective function is calculated by taking the prime and second derivatives of every data spot. The

loss function is a square loss. Here is its objective function, in which l shows a differential convertible loss function, which shows variation among the prediction $\hat{y}_i$ and the purpose y_i. The second part Ω can penalize the pattern complexity, and T shows the leaves number in the tree. The γ and λ show the tree's complexity, the greater their amount, and the simpler the framework of the tree.

$$L(\phi) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k) \;\; where\, \Omega(f) = \gamma T + \frac{1}{2}\lambda ||\omega||^2 \tag{13}$$

(E) The random forest exhibits excellent efficiency and has been extensively applied [29]. RF utilizes the decision tree as its base classifier and shows an extension of Bagging. RF uses two very significant procedures. The first technique involves introducing random features in the procedure of decision tree making, and the second involves an out-of-bag estimation. The RF method can be described below. The first step is to randomly select a sample from every data, and afterward, to return the sample to the original data. As a root sample for a decision tree, the chosen samples have been applied for training the decision tree. Second, for splitting the nodes of the decision tree, m attributes have been chosen randomly (there are a total of M attributes and ensuring $<< M$). Choose an attribute to be the dividing feature of the node using the strategy, such as information gain. Continue to do this until the decision tree can no longer be divided.

(F) Among the more popular deep learning networks is CNN. There are usually input, output, latent, and max-pooling layers in a CNN model. Several great results have been obtained in numerous areas of computer vision. Here, one-dimension property vectors are used as input, and a one-dimension convolution kernel in convolution layers is adopted. The convolution layer extracts properties from the input, and here the kernel size is three. The process of the CNN model is shown in Figure 3.

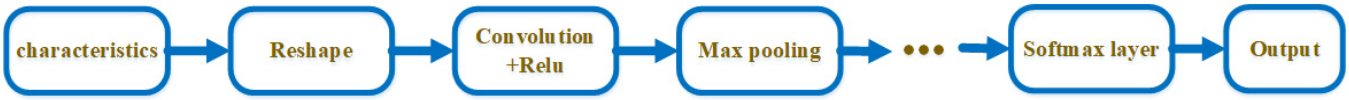


Figure 3. The procedure of CNN layout.

Actually, the main purpose of this research is to show the high and successful role of the deep learning models in reinforcing the smart grid against various cyber-attacks. In this regard, the proposed model would detect and stop cyber-hacking at the installation location rather than focusing on the cyber-attack type. Therefore, the localization procedure would be attained through the diverse detection models located in the smart grid, but the cyber-attack type detection requires more data that can be made later based on the recorded abnormal data.

3.2.2. Outcomes

This study considers 37 varied scenarios for events. In order to determine the need for various models (fault analysis), we performed some comparative experiments according to various PMU kinds. In one group, properties of localization/segmentation are sent to the related DML model in order to train, and in the other one, whole features are sent to various machine learning models. Moreover, it is shown in Table 4 that data can be effectively split according to the PMU resources. Splitting the data can enhance the accuracy of classification models as well as reduce data dimensions and enhance training speed and minimize computing sources. The score of the significant features is shown in Figure 4.

Table 4. Transfer diverse characteristics to the layout for comparison.

Technique	Characteristics	
	Entire	**Split**
Accuracy	0.9344	0.9387

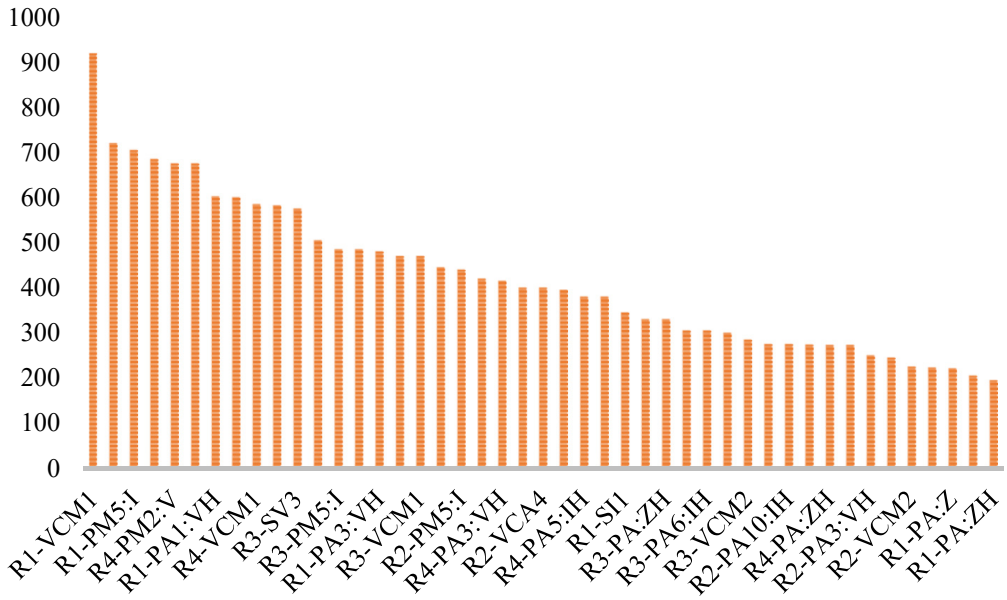


Figure 4. Significance features score.

Several corresponding experiments are conducted on various ways of replacing abnormal values in data. Table 5 shows the outcomes. The replacement method is shown in the left column, and the suggested approach is represented by *log_mean*. Zero shows a process to replace NAN and INF with zero values, and mean shows a process to replace with the mean value. The AWV model is utilized as a trial model, and the accuracy is adopted as the assessment metrics, that is, the right column in Table 5.

Table 5. Diverse methods to procedure Inf and Nan.

Method	**Zero**	**Mean**	***Log-Mean***
Accuracy	0.9361	0.9342	0.9387

Applying the *log_mean* technique for replacing the unusual amount in the data is intuitively the best approach. According to the outcome, the suggested process in order to process abnormal values has proven successful.

Comparison experiments are also conducted to verify feature selection. First, the significance of the original features is determined, and afterward, they are arranged based on significance. A variety of mixtures of features has been selected for training, and Table 6 shows these outcomes.

The approach was verified practically through a comparative test. The test extracts the test group and training group from 15 multiclass data sets in a 9:1 ratio at random, and afterward, these data sets have been combined into 1 training group. The training group has been transferred to the layout to train and learn. Table 7 presents the outcomes of 15 test sets transferred to the model for practically simulating the efficiency of the model applications. It is apparent that the model's accuracy has decreased. It is because data interaction would occur by increasing the number of data resulting in changing the model, and whenever whole data has been combined, there would unavoidably be abnormal

points and noises. Due to the fact that such noises and anomalies have not been separated in training, the model's indexes alter, and the robustness decreases.

Table 6. Assessment of characteristics chosen.

Characteristics	**Only New Characteristics**	**12.5% Main Characteristics and New Characteristics**	**25% Main Characteristics and New Characteristics**	**37.5% Main Characteristics and New Characteristics**	**50% Main Characteristics and New Characteristics**
Mean accuracy	0.7492	0.9390	0.9350	0.9337	0.9334
Characteristics	**62.5% Main Characteristics and New Characteristics**	**75% Main Characteristics and New Characteristics**	**87.5% Main Characteristics and New Characteristics**	**100% Main Characteristics and New Characteristics**	
Mean accuracy	0.9335	0.9331	0.9324	0.9353	

Table 7. Layout accuracy on 15 trail sets in the actual simulation.

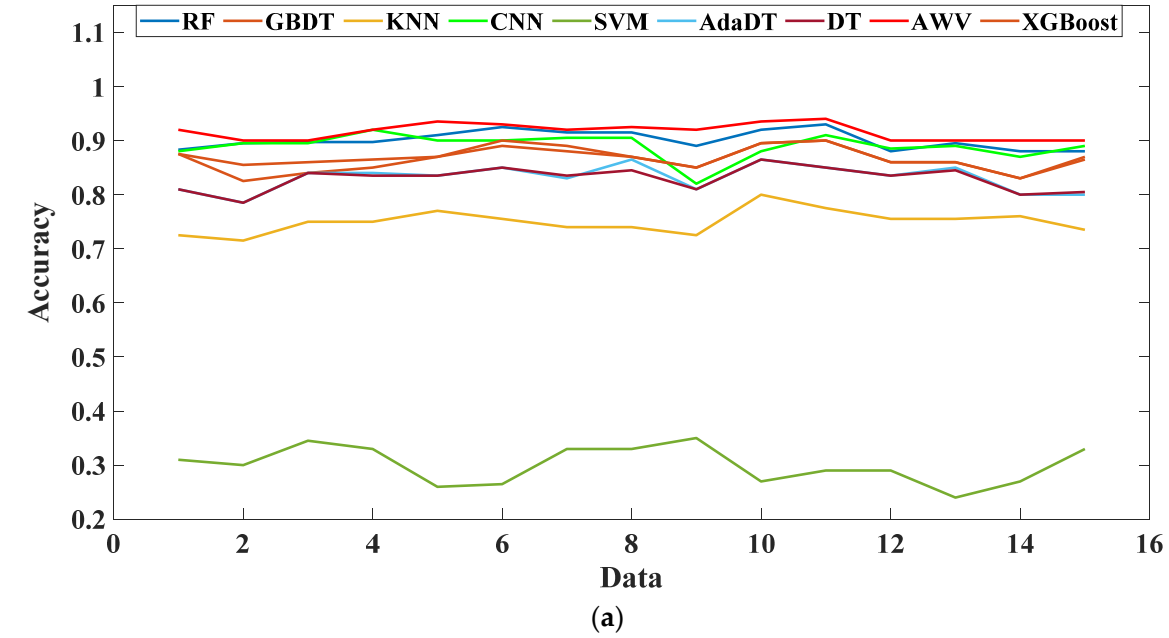
Data set	**Data 1**	**Data 2**	**Data 3**	**Data 4**	**Data 5**	**Data 6**	**Data 7**	**Data 8**
Data number	0.8894	0.8699	0.9097	0.8830	0.9092	0.9096	0.9066	0.9193
Data set	**Data 9**	**Data 10**	**Data 11**	**Data 12**	**Data 13**	**Data 14**	**Data 15**	**Entire**
Data number	0.9083	0.9229	0.9241	0.9007	0.9016	0.8966	0.9130	0.9043

Firstly, the efficacy of the features created from the feature construction engineering in the model is determined by sorting the significance of features. Model interpretability can be determined by determining the significance of features. Weight, gain, cover, and so on are general indicators of feature significance. In the XGBoost method [30], the number of times a property appears in a tree has been shown by weight, the mean gain of the slot using the property has been represented by the gain, and the mean coverage of the slot using the property is shown by the cover. According to Figure 4, weight calculates feature significance. The abscissa indicates the names of the beat 45 properties, and the ordinate indicates the assessment score. The origin features are shown by the gray part. The features derived from feature construction engineering are represented by the red mark. It is evident that each of the 16-making properties is in the best 45.

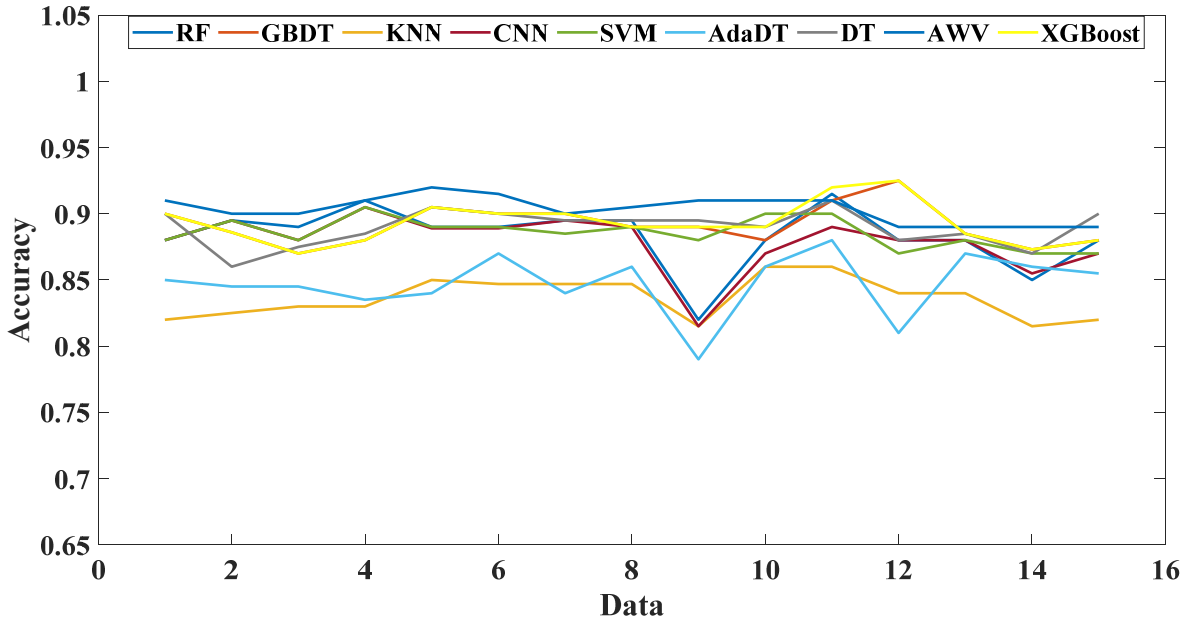
The test trains 15 sets of multiclass classification data sets and tests respectively and uses accuracy as an assessment metric. The accuracy of the trail data sent to the layout before and after optimization based on the main 128 properties is shown in Figure 5. The classification accuracy of the trail group on various layouts with default variables is shown in Figure 5a, and the accuracy of the trail group on the layout applying optimized variables is represented in Figure 5b. For a more intuitive visualization of the variation in accuracy after layouts are optimized, Figure 5a and b are combined, and the mean of the accuracy values for whole sets are adopted, i.e., Figure 5c. Figure 5 shows that the SVM layout with default variables has an accuracy of approximately 0.30, but after optimization, it grows to 0.85, which represents a near 200% advancement. Other models have improved significantly in accuracy after optimization as well. The best accuracy of the suggested AWV model is 0.9217.

Table 3 shows that every data set has about 5000 segments of data; therefore, the CNN layout cannot be used. The semantic relationships among features might also be ignored by several neural networks, such as CNN and long-short-term memory (LSTM) layouts. Thus, in several cases, statistical features according to the manual design could positively affect model accuracy as well. Moreover, the tree-based algorithm outperforms KNN and SVM.

The test set had better performance on the model suggested in this study in comparison to the conventional DML and CNN, as shown in Figure 5.

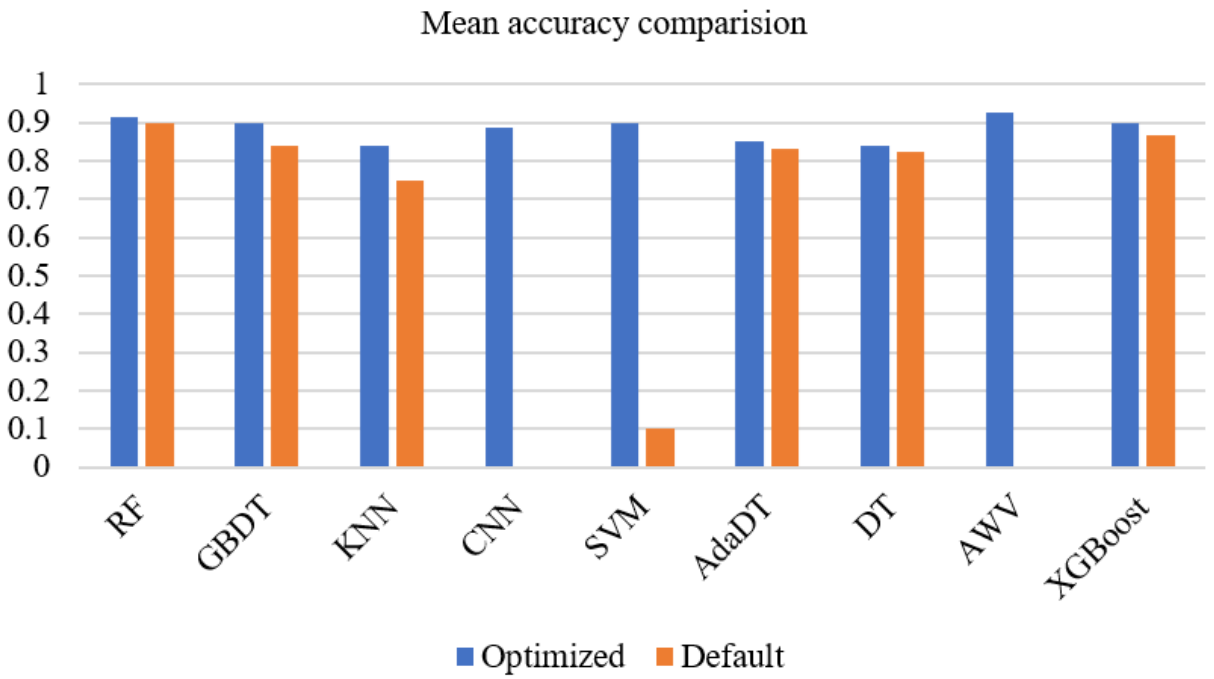


(a)



(b)

Figure 5. *Cont.*



(**c**)

Figure 5. Proficiency comparisons of variables through applying 128 properties (**a**); (**b**) precision over 15 data sets through applying optimum variables; (**c**) mean accuracy comparison.

4. Conclusions

Various SG information as the experimental foundation is used in the present study, and property making for the original data is applied. The layout for identifying faults and cyber-attack in the electrical system is proposed. The present study uses various DML assessment indexes for evaluating the suggested model and conventional DML methods in the experiment. According to the outcomes, the information analyzing process improves the model's accuracy, and the AWV layout detects 37 types of behavior in electrical systems efficiently. As a result, machine learning can be used in the power grid to assist operators in making decisions. In other words, the smart grid operator can always check the health level of the data gathering by the PMUs all around the grid. In the case that any abnormality is detected, the possibility of an intentional cyber-attack exists, and thus, some cautious pre-operation strategies shall be considered to keep the power and demand balance. Moreover, if the data readings from any PMU are unusual, the system operator can decide to estimate the system status without this PMU and rely more on the data coming from the other healthy PMUs.

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

Funding: This research has been funded by the Scientific Research Deanship at the University of Ha'il—Saudi Arabia through project number RG-21079.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Almalaq, A.; Albadran, S.; Alghadhban, A.; Jin, T.; Mohamed, M.A. An Effective Hybrid-Energy Framework for Grid Vulnerability Alleviation under Cyber-Stealthy Intrusions. *Mathematics* **2022**, *10*, 2510. [CrossRef]
2. Reich, J.; Schneider, D.; Sorokos, I.; Papadopoulos, Y.; Kelly, T.; Wei, R.; Armengaud, E.; Kaypmaz, C. Engineering of Runtime Safety Monitors for Cyber-Physical Systems with Digital Dependability Identities. In Proceedings of the International Conference on Computer Safety, Reliability, and Security, Lisbon, Portugal, 15 September 2020; Springer: Cham, Switzerland, 2020; pp. 3–17.
3. Li, Y.; Wang, B.; Wang, H.; Ma, F.; Zhang, J.; Ma, H.; Mohamed, M.A. Importance Assessment of Communication Equipment in Cyber-Physical Coupled Distribution Network Based on Dynamic Node Failure Mechanism. *Front. Energy Res.* **2022**, 654. [CrossRef]
4. Zhang, L.; Cheng, L.; Alsokhiry, F.; Mohamed, M.A. A Novel Stochastic Blockchain-Based Energy Management in Smart Cities Using V2S and V2G. *IEEE Trans. Intell. Transp. Syst.* **2022**, 1–8. [CrossRef]
5. Chen, J.; Alnowibet, K.; Annuk, A.; Mohamed, M.A. An effective distributed approach based machine learning for energy negotiation in networked microgrids. *Energy Strategy Rev.* **2021**, *38*, 100760. [CrossRef]
6. Al-Mhiqani, M.N.; Ahmad, R.; Yassin, W.; Hassan, A.; Abidin, Z.Z.; Ali, N.S.; Abdulkareem, K.H. Cyber-security incidents: A review cases in cyber-physical systems. *Int. J. Adv. Comput. Sci. Appl.* **2018**, *1*, 499–508.
7. Luo, Y.; Cheng, L.; Liang, Y.; Fu, J.; Peng, G. Deepnoise: Learning sensor and process noise to detect data integrity attacks in CPS. *China Commun.* **2021**, *18*, 192–209. [CrossRef]
8. Kaouk, M.; Flaus, J.M.; Potet, M.L.; Groz, R. A review of intrusion detection systems for industrial control systems. In Proceedings of the 2019 6th International Conference on Control, Decision and Information Technologies (CoDIT), Paris, France, 23 April 2019; IEEE: Toulouse, France, 2019; pp. 1699–1704.
9. Dehghani, M.; Kavousi-Fard, A.; Dabbaghjamanesh, M.; Avatefipour, O. Deep learning based method for false data injection attack detection in AC smart islands. *IET Gener. Transm. Distrib.* **2020**, *14*, 5756–5765. [CrossRef]
10. Taormina, R.; Galelli, S.; Tippenhauer, N.O.; Salomons, E.; Ostfeld, A.; Eliades, D.G.; Aghashahi, M.; Sundararajan, R.; Pourahmadi, M.; Banks, M.K.; et al. Battle of the attack detection algorithms: Disclosing cyber attacks on water distribution networks. *J. Water Resour. Plan. Manag.* **2018**, *144*, 04018048. [CrossRef]
11. Chang, Q.; Ma, X.; Chen, M.; Gao, X.; Dehghani, M. A deep learning based secured energy management framework within a smart island. *Sustain. Cities Soc.* **2021**, *70*, 102938. [CrossRef]
12. Keshk, M.; Sitnikova, E.; Moustafa, N.; Hu, J.; Khalil, I. An integrated framework for privacy-preserving based anomaly detection for cyber-physical systems. *IEEE Trans. Sustain. Comput.* **2019**, *6*, 66–79. [CrossRef]
13. Huang, Y.; He, T.; Chaudhuri, N.R.; la Porta, T. Preventing Outages under Coordinated Cyber-Physical Attack with Secured PMUs. *IEEE Trans. Smart Grid* **2022**, *13*, 3160–3173. [CrossRef]
14. Alexopoulos, T.A.; Korres, G.N.; Manousakis, N.M. Complementarity reformulations for false data injection attacks on pmu-only state estimation. *Electr. Power Syst. Res.* **2020**, *189*, 106796. [CrossRef]
15. Alexopoulos, T.A.; Manousakis, N.M.; Korres, G.N. Fault location observability using phasor measurements units via semidefinite programming. *IEEE Access* **2016**, *4*, 5187–5195. [CrossRef]
16. Mamuya, Y.D.; Lee, Y.-D.; Shen, J.-W.; Shafiullah, M.; Kuo, C.-C. Application of Machine Learning for Fault Classification and Location in a Radial Distribution Grid. *Appl. Sci.* **2020**, *10*, 4965. [CrossRef]
17. Chaithanya, P.S.; Priyanga, S.; Pravinraj, S.; Sriram, V.S. SSO-IF: An Outlier Detection Approach for Intrusion Detection in SCADA Systems. In *Inventive Communication and Computational Technologies*; Springer: Singapore, 2020; pp. 921–929.
18. Chen, J.; Mohamed, M.A.; Dampage, U.; Rezaei, M.; Salmen, S.H.; Obaid, S.A.; Annuk, A. A multi-layer security scheme for mitigating smart grid vulnerability against faults and cyber-attacks. *Appl. Sci.* **2021**, *11*, 9972. [CrossRef]
19. Avatefipour, O.; Al-Sumaiti, A.S.; El-Sherbeeny, A.M.; Awwad, E.M.; Elmeligy, M.A.; Mohamed, M.A.; Malik, H. An intelligent secured framework for cyberattack detection in electric vehicles' CAN bus using machine learning. *IEEE Access* **2019**, *7*, 127580–127592. [CrossRef]
20. Wang, B.; Ma, F.; Ge, L.; Ma, H.; Wang, H.; Mohamed, M.A. Icing-EdgeNet: A pruning lightweight edge intelligent method of discriminative driving channel for ice thickness of transmission lines. *IEEE Trans. Instrum. Meas.* **2020**, *70*, 1–12. [CrossRef]
21. Alnowibet, K.; Annuk, A.; Dampage, U.; Mohamed, M.A. Effective energy management via false data detection scheme for the interconnected smart energy hub–microgrid system under stochastic framework. *Sustainability* **2021**, *13*, 11836. [CrossRef]
22. Chen, L.; Liu, Z.; Tong, L.; Jiang, Z.; Wang, S.; Dong, J.; Zhou, H. Underwater object detection using Invert Multi-Class Adaboost with deep learning. In Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, 19 July 2020; IEEE: Toulouse, France, 2020; pp. 1–8.
23. Shafizadeh-Moghadam, H. Fully component selection: An efficient combination of feature selection and principal component analysis to increase model performance. *Expert Syst. Appl.* **2021**, *186*, 115678. [CrossRef]
24. Roshan, K.; Zafar, A. Deep Learning Approaches for Anomaly and Intrusion Detection in Computer Network: A Review. *Cyber Secur. Digit. Forensics* **2022**, *73*, 551–563.

25. Aceto, G.; Ciuonzo, D.; Montieri, A.; Pescape, A. Traffic classification of mobile apps through multi-classification. In Proceedings of the GLOBECOM 2017-2017 IEEE Global Communications Conference, Singapore, 4 December 2017; IEEE: Toulouse, France, 2017; pp. 1–6.
26. Pham, B.T.; Bui, D.T.; Prakash, I.; Nguyen, L.H.; Dholakia, M.B. A comparative study of sequential minimal optimization-based support vector machines, vote feature intervals, and logistic regression in landslide susceptibility assessment using GIS. *Environ. Earth Sci.* **2017**, *76*, 371. [CrossRef]
27. Jena, M.; Dehuri, S. Decision tree for classification and regression: A state-of-the art review. *Informatica* **2020**, *44*, 405–420. [CrossRef]
28. Chen, R.C.; Caraka, R.E.; Arnita, N.E.; Pomalingo, S.; Rachman, A.; Toharudin, T.; Tai, S.K.; Pardamean, B. An end to end of scalable tree boosting system. *Sylwan* **2020**, *165*, 1–11.
29. Lulli, A.; Oneto, L.; Anguita, D. Mining big data with random forests. *Cogn. Comput.* **2019**, *11*, 294–316. [CrossRef]
30. Franklin, J. The elements of statistical learning: Data mining, inference and prediction. *Math. Intell.* **2005**, *27*, 83–85. [CrossRef]