applied sciences

MDPI

Article

Optimization of the Random Forest Hyperparameters for Power Industrial Control Systems Intrusion Detection Using an Improved Grid Search Algorithm

Ningyuan Zhu [1,2], Chaoyang Zhu [2,*], Liang Zhou [2], Yayun Zhu [2] and Xiaojuan Zhang [2]

[1] School of Electrical and Electronic Engineering, North China Electric Power University, Beijing 102206, China
[2] China Electric Power Research Institute, Beijing 100192, China
* Correspondence: zhucy@epri.sgcc.com.cn

Abstract: The intrusion detection method of power industrial control systems is a crucial aspect of assuring power security. However, traditional intrusion detection methods have two drawbacks: first, they are mainly used for defending information systems and lack the ability to detect attacks against power industrial control systems; and second, although machine learning-based intrusion detection methods perform well with the default hyperparameters, optimizing the hyperparameters can significantly improve its performance. In response to these limitations, a random forest (RF)-based intrusion detection model for power industrial control systems is proposed. Simultaneously, this paper proposes an improved grid search algorithm (IGSA) for optimizing the hyperparameters of the RF intrusion detection model to improve its efficiency and effectiveness. The proposed IGSA boosts the speed of calculation from $O(n^m)$ to $O(n \times m)$. The suggested model is evaluated based on the public power industrial control system dataset after hyperparameter optimization. The experiment results show that our method achieves a superior detection performance with the accuracy of 98% and has more outstanding performance than the same type of work.

Keywords: improved grid search; intrusion detection; hyperparameter importance; random forest; hyperparameter optimization

Citation: Zhu, N.; Zhu, C.; Zhou, L.; Zhu, Y.; Zhang, X. Optimization of the Random Forest Hyperparameters for Power Industrial Control Systems Intrusion Detection Using an Improved Grid Search Algorithm. *Appl. Sci.* **2022**, *12*, 10456. https://doi.org/10.3390/app122010456

Academic Editor: Oscar Duque-Perez

Received: 30 August 2022
Accepted: 26 September 2022
Published: 17 October 2022

1. Introduction

Smart grids will rely on advanced technologies such as big data and the Internet of Things to fulfill the increasing demand of reliable electricity [1]. Smart grids necessitate the use of power industrial control systems and the deployment of digital communication networks (such as Ethernet, cellular services, and satellite signals) for data acquisition and remote control between control centers and a large number of smart grid infrastructures (such as smart substations and photovoltaic power plants) [2]. With the increasing interconnections between power industrial equipment and the Internet, malicious attacks targeting power industrial control systems through the Internet have followed [3].

Power critical information infrastructure is always at the forefront of cyber attacks. Attacks and sabotage actions that result in widespread power outages, severely disrupt social and economic processes, and even jeopardize public safety have become increasingly common in recent years [4]. In 2010, the infamous Stuxnet malware was brought into the Iranian nuclear facility region via a USB stick and leveraged various zero-day vulnerabilities to insert its malicious code into Siemens programmable logic controllers (PLCs), causing centrifuges to wear out at a much higher rate. At the same time, malware falsifies sensor data to conceal operator attacks [5]. In 2015, a malware attack on Ukraine's power infrastructure resulted in a large-scale power outage. The intruder secured legitimate credentials for remote access prior to the cyber attack. During the attack, a denial-of-service attack was launched on the Ukrainian Electric Power Company's website and customer service system, preventing users from reporting incidents and prolonging the outage. KillDisk virus was

used to delete several systems at the end of the cyber attack, delaying recovery efforts [6]. In 2016, Israel disclosed that its electrical authority had been the target of a severe cyber attack. In this attack, the attacker sent phishing emails to the electric power bureau workers that contained ransomware, tricking them into running malicious malware and encrypting the necessary content on their computers, which the electric power bureau staff must pay to unlock. Following the incident, Israeli authorities were obliged to turn down infected computers at power plants to prevent the ransomware from spreading further over the network and causing a larger incident [7]. This is yet another example of a cyberattack on a power grid. This sequence of events raised the alarm for the power system's safety. As a result, enhancing the detection capabilities of assaults on power industrial control systems has become a pressing problem in academics and industry.

Intrusion detection (ID) was first used in traditional information systems to detect attacks that exploit system design faults or vulnerabilities and offer forensic evidence to warn system administrators about network intrusions [8]. Intrusion detection has shown to play a significant role in the world of information systems; thus, it is possible to consider implementing it in the field of electric power industrial control systems. A power industrial control system's objective and structure, on the other hand, are fundamentally different from those of a standard information system. The following are the distinct manifestations: (1) Using real-world equipment. (2) A wide range of tools. (3) A consistent business process. (4) The main objective is to keep the system stable. (5) Numerous proprietary protocols exist. (6) The timing requirements are very strict. Furthermore, the power grid industrial control system's traffic characteristics and significance differ from those of a conventional network environment. The differences are as follows: (1) The data are short and frequent. (2) Network traffic is predictable. (3) The response time is drastically reduced. (4) Data flow is always in the same direction. (5) Because the order of control information is defined, the timing is precise [9]. As a result, typical intrusion detection technology cannot be used directly to a power industrial control system, and associated research is limited. Investigation of intrusion detection technologies for power grid industrial control systems is crucial.

To address the above-mentioned problems, this paper presents an intrusion detection approach for power industrial control systems based on hyperparameter optimization. The main contributions are as follows:

(1) We collect performance data that captures the roc_auc score of the random forest (RF) classifier with hyperparameter settings. We then fit an RF model to this data and use functional ANOVA to decompose the variance of predictions into contributions due to every single hyperparameter. The significance of RF hyperparameters is determined by the magnitude of variance contribution.

(2) To further improve the intrusion detection model's performance, we propose an improved grid search that performs a local grid search for each hyperparameter in turn, based on the importance of the RF hyperparameters. The optimal hyperparameter combination is obtained by substituting the optimal hyperparameter value for the matching base classifier hyperparameter value.

(3) Based on the objective, structure and characteristics of the power industrial control system, we built an RF-based intrusion detection model, taking full advantage of the RF algorithm's advantages of processing high-dimensional data, fast training speed, and good performance. To ensure that our proposed method is effective for power industrial control system intrusion detection and simple to validate, we use publicly available power industrial control system datasets.

The rest of this paper is organized as follows. Section 2 provides the background in intrusion detection. Section 3 discusses our methodology when applying our experiments. In Section 4 we describe our results and compare with similar work. Finally, we draw a conclusion of this article in Section 5.

2. Related Work

The use of RF as an intrusion detection algorithm in traditional network traffic intrusion detection is relatively mature [10–12]. For example, Cao, L. et al. proposed a parallel ensemble learning RF intrusion detection model for the classic NSL-KDD dataset to address the problems of over-fitting and large generalization error in the traditional decision tree method. The proposed model had high accuracy and outperformed the decision tree intrusion detection model [13]. The disadvantage of this article is that it only considered one hyperparameter of the base classifier during hyperparameter optimization, which is obviously insufficient. Gattineni, P. et al. used three intrusion detection methods to lower the false alarm rate: support vector machine (SVM), RF, and extreme learning machine [14]. The results demonstrated that the extreme learning machine approach had the best effect, whereas the RF method had a very low accuracy of only 52.32%. To address the issue of insufficient detection capacity in traditional networks, Wang, Z. et al. proposed an intrusion detection method that combined stacked autoencoders and RF, taking full advantage of the autoencoder feature extraction capabilities as well as the RF classification and detection capabilities [15]. With accuracy, precision, recall, and F-score all of which are 0.998, the model beat commonly used machine learning approaches such as Naive Bayes (NB), SVM, and Decision Tree (DT).

In the field of industrial control system intrusion detection, Anton, S.D.D. et al. used two machine learning-based methods, namely SVM and RF, to detect attacks. Two different datasets were used, one Modbus-based gas pipeline control traffic and one OPC UA-based batch processing traffic [16]. This article compared the performance of the SVM and RF methods on the two datasets, and the results demonstrated that the RF method is more accurate. Zhang, F. et al. analyzed data from a real-time industrial control system test environment and evaluated four data-driven detection models: k-nearest neighbor, bagging technique, DT, and RF, in order to deal with the increasing number of network attacks on industrial control systems [5]. All had a low false negative and false positive rate, according to the results.

In the topic of power industrial control systems, there are just a few related works on intrusion detection. Wang et al. first created a power system attack model and generated a power system network intrusion dataset using a combination of Gaussian clustering model. Then they used the RF method to build an intrusion detection model [17]. The model had a high accuracy and a low false positive rate, according to the dataset test results. The problem is that the assessment indicators are incomplete, as key measures like recall and F1-score are absent. Furthermore, the dataset used are not public, making it hard to evaluate the performance of their intrusion detection method. Morris T. et al. investigated the efficacy of several machine learning methods applied to intrusion detection of power industrial control systems to overcome the problem of high artificial judgment uncertainty induced by more complicated network attacks on power systems [18]. Experiments showed that deploying machine learning methods to enhance existing power system security architectures has practical implications. The problem is that the RF intrusion detection approach employed in this study had a low evaluation index. Binary classification, for example, has an accuracy rate of only 80%, which is insufficient in a power system with exceptionally high stability requirements. Our study and Morris' work both use the same dataset and use the RF method, therefore we compare the performance of our intrusion detection method with Morris' method.

To summarize, machine learning-related technologies for traditional network intrusion detection are relatively mature. Whereas, related research works in the field of industrial control intrusion detection, particularly in the area of power grid industrial control system intrusion detection, are extremely limited, with numerous research gaps. Specifically, they are incapable of detecting attacks on power industrial control systems; therefore, the performance of related intrusion detection technologies should be improved. Furthermore, most studies collect data using their own simulation platforms, which are not open for other

researchers to evaluate. Thus, peer-reviewed intrusion detection methods' performance cannot be compared to their work.

3. Methodology

The flow chart of the proposed intrusion detection method based on hyperparameter optimization is shown in Figure 1.

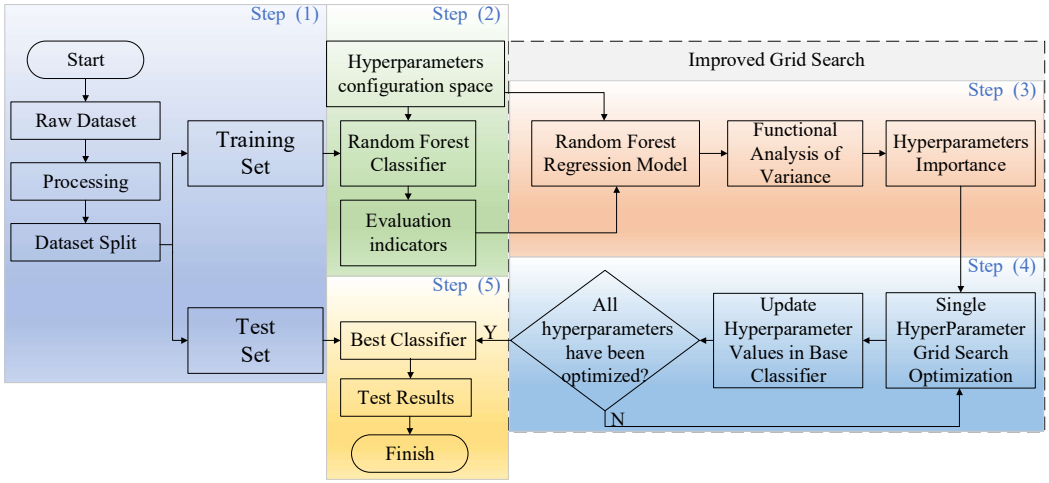

Figure 1. The overview of the proposed method.

The method consists of five steps:

1. Import the power industrial control system's original dataset, which is then preprocessed and divided into training and validation sets using 5-fold cross-validation.
2. Create a hyperparameter configuration space, build RF classifiers with the hyperparameter combinations in the space, train the classifier with the training set, and then classify the test set to obtain the classifiers' performance.
3. Fit a RF regression model with the combination of hyperparameters in the configuration space as a feature and the performance of the corresponding classifier as a label and use functional variance analysis to obtain the importance ranking of hyperparameters.
4. Based on the importance of hyperparameters, apply grid search optimization to each hyperparameter in turn, and the optimal hyperparameter replaces the corresponding hyperparameter value in the base classifier.
5. Obtain the best hyperparameter combination, build the best RF classifier, and validate its classification effect using the test set.

3.1. Random Forest

The RF algorithm is a decision tree-based ensemble learning algorithm [19]. RF utilizes the advantages of the decision tree algorithm's high speed and accuracy when dealing with classification problems by creating several decision tree models. Multiple decision trees have no association, and errors are mutually minimized, resulting in more accurate and robust classification findings. Figure 2 depicts the RF model employed in this study.

The first step in model construction is to choose a sampling method for generating a sub-dataset. Whether to use bootstrap sampling is the first hyperparameter of RF, and the randomness of RF data is reflected here. The second step in model construction is to construct a decision tree, with the number of decision trees being the second hyperparameter. There are four hyperparameters in the decision tree: the maximum depth of the decision tree, the splitting standard, the minimum number of samples for internal node splitting, and the minimum number of samples for leaf nodes. The characteristics of each decision tree are picked at random, and the random forest's seventh hyperparameter is the maximum number of selections. Because seven hyperparameters impact the performance of the RF classifier, this study optimizes the combination of these seven hyperparameters [20].

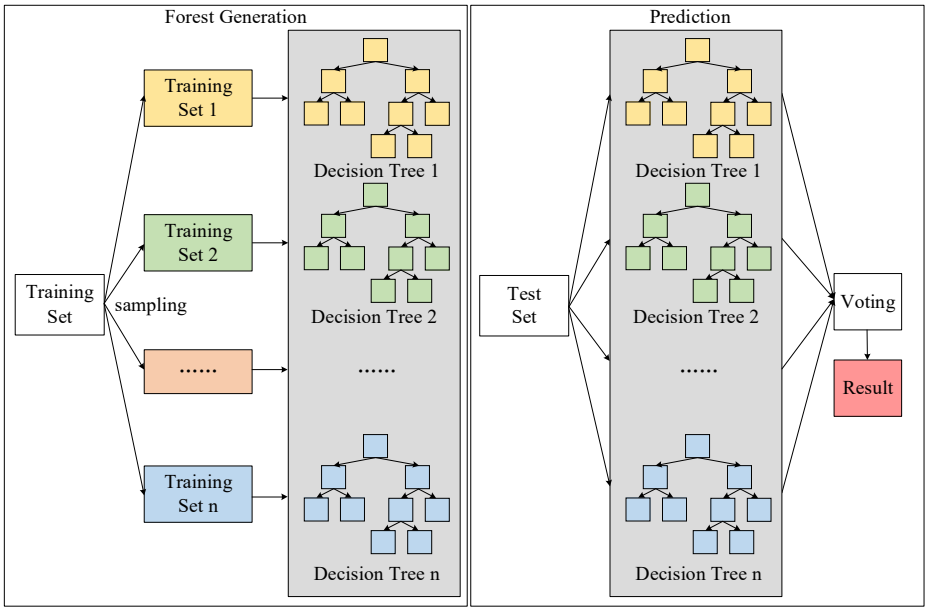

Figure 2. Random forest classifier model.

Following the construction of the RF model, the test set samples are input, and each decision tree in the forest judged separately, with the classification results of the samples output. Finally, the results of all decision trees are combined using the voting mechanism, and the class with the most votes is the class to which the sample belongs [21].

3.2. Functional Analysis of Variance (Functional ANOVA)

The functional ANOVA framework for evaluating the importance of hyperparameters was first proposed by Hutter et al. [22]. Since we will use this technique to obtain the importance of hyperparameters, we now go over it in more depth.

We first introduce notation. Assuming that RF classification has n hyperparameters with domains $\Theta_1, \ldots, \Theta_n$ and configuration space $\Theta = \Theta_1 \times \cdots \times \Theta_n$. Let $N = \{1, \ldots, n\}$ be the set of all hyperparameters of RF. An instantiation of RF is a vector $\theta = \langle \theta_1, \ldots, \theta_n \rangle$ with $\theta_i \in \Theta_i$. A partial example of RF is a vector $\theta_U = \langle \theta_i, \ldots, \theta_j \rangle$ with a subset $U \subseteq N$ of the hyperparameters.

Functional ANOVA decomposes a function $\hat{y} : \Theta_1 \times \cdots \times \Theta_n \to \mathbb{R}$ into additive components that only depend on the subsets of the hyperparameters N:

$$\hat{y}(\theta) = \sum_{U \subseteq N} \hat{f}_U(\theta_U) \tag{1}$$

The components $\hat{f}_U(\theta_U)$ are defined as:

$$\hat{f}_U(\theta_U) = \begin{cases} \hat{f}_\varnothing & \text{if } U = \varnothing \\ \hat{a}_U(\theta_U) - \sum_{W \subsetneq U} \hat{f}_W(\theta_W) & \text{otherwise} \end{cases} \tag{2}$$

where $\hat{a}_U(\theta_U)$ means the average performance of all complete instantiations θ that agree with θ_U in the instantiations of hyperparameters U. The unary functions $\hat{f}_{\{j\}}(\theta_{\{j\}})$, namely main effects, capture the effect of varying hyperparameter j [23].

We construct an RF regression prediction model, the hyperparameter configuration space is utilized as a feature, and the performance (e.g., recall or roc_auc score) of the RF classifier is used as a label. It is equally important in the power system to correctly discriminate between normal and attack samples. The area under curve (AUC) metric, which is defined as the roc_auc score in this paper, is chosen as the hyperparameter assessment criterion because it considers the classifier's capacity to classify both positive and negative data. We then take advantage of functional ANOVA to decompose the

variance of each $\hat{y}$ into contributions due to each subset of hyperparameters, as indicated by the formula:

$$\mathbb{V} = \sum_{U \subset N} \mathbb{V}_U, \text{ with } \mathbb{V}_U = \frac{1}{\|\Theta_U\|} \int \hat{f}_U(\theta_U)^2 d\theta_U \quad (3)$$

where $\frac{1}{\|\Theta_U\|}$ is the probability density of the uniform distribution across Θ_U.

Functional ANOVA thus provides us with the relative variance contributions of each individual hyperparameter.

3.3. Improved Grid Search

This paper proposes an improved grid search algorithm (IGSA) that builds on the traditional grid search algorithm and addresses the problem of the grid search computing time.

The principle of the grid search algorithm is shown in Figure 3. First, the Cartesian product is applied to the value set of each hyperparameter to obtain the hyperparameter configuration space, which contains all possible hyperparameter combinations (the box on the left side of the figure). The grid search algorithm then trains a model for every hyperparameter combination in configuration space. The experiment that yields the best validation set error is then chosen as having found the best hyperparameters, as shown in the box on the right side of the figure [24].

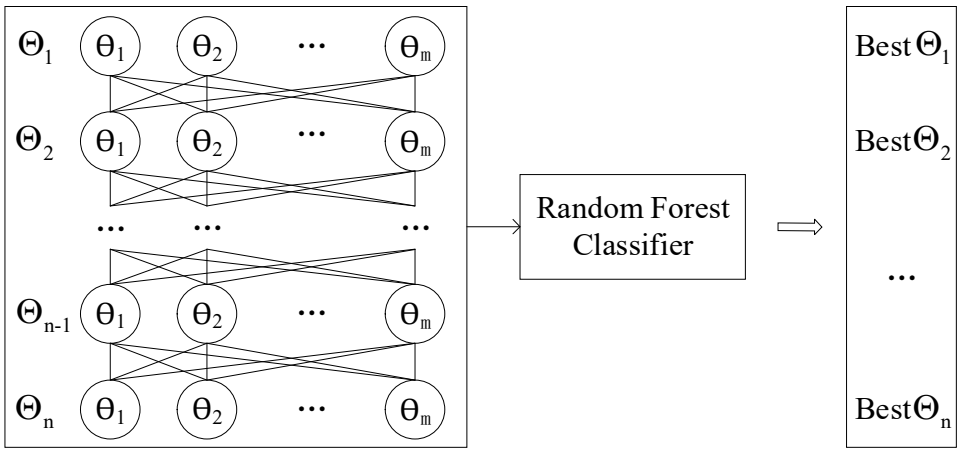

Figure 3. Principle of the grid search algorithm.

The problem is that grid search has a significant disadvantage. Grid search computation time grows exponentially $O(n^m)$ with n hyperparameters, each of which can take m values. This wastes a lot of computational resources, and the processing time is excruciating. To solve this problem, this research provides a IGSA method.

Figure 4 depicts the principle of IGSA. The significance of hyperparameters is first determined via functional ANOVA, and the results are recorded as $\Theta_1, \Theta_2, \ldots, \Theta_n$. Secondly, a single-parameter optimization algorithm is employed to perform grid search optimization on particular hyperparameters in order of importance, then the hyperparameter value of the base classifier is replaced with the best single-hyperparameter value obtained through optimization. The optimal hyperparameter combination is generated by acquiring n ideal hyperparameter values after n times of the grid search. This experiment's base classifier is the RF classifier defaulted in sklearn.

The single-parameter optimization algorithm is shown in Algorithm 1, the input is a single hyperparameter Θ_i with m values. The output is a single optimal hyperparameter.

The performance of the classifier is validated using k-fold cross-validation in Algorithm 1. For k-fold cross-validation, all datasets are divided into k parts, with one serving as the test set without repetition and the remaining k-1 as the training set for training. Then computation of the model's score on the test set scores are then averaged k times to obtain the final score. We use 5-fold cross-validation in this experiment.

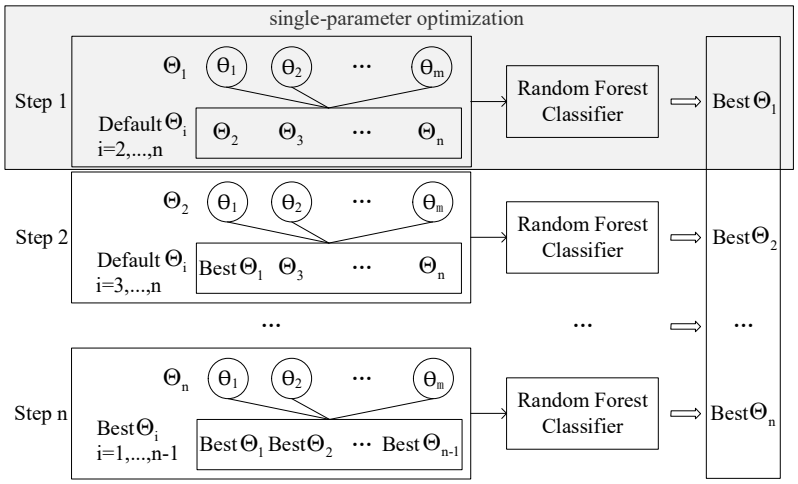

Figure 4. Principle of improved grid search algorithm (IGSA).

Algorithm 1. Improved grid search in parameter space for RF with finer tuning.

input: $\Theta_i, j = [1, 2, \ldots, m]$
best_result = 0
best_para = default value in base RF classifier
for θ_j in Θ_i do:
Train RF with θ_j on Training Set
Evaluate RF classification on ValidationSet
if result > best_result
best_result = result
best_para = θ_j
end if
end for
return best_para

For n hyperparameters with m values, the calculation speed of the optimal hyperparameter search is reduced to $O(n \times m)$ through improvement.

4. Experiments and Result Analysis

The experiments described in this paper were carried out on Google Colab Intel(R) Xeon(R) CPU @ 2.20 GHz with 12 GB RAM, using Python's sklearn library (version 1.0.2) to construct the RF models for this paper.

4.1. The Power Industrial Control System Dataset

In Figure 5 we show the power system framework used in this evaluation [25], a complex mix of supervisory control systems interacting with various smart electronic devices complemented by network monitoring devices such as Snort and Syslog systems. The power system components are as follows: G1, G2 are two generators, R1–R4 are four intelligent electronic devices (IEDs) that can open or close circuit breakers, and BR1-BR4 are four circuit breakers. Each IED automatically controls a circuit breaker: R1 controls BR1, R controls BR2, and so on. When a fault is detected, the IED uses a distance protection scheme that trips the circuit breaker; however, because the device lacks abnormal monitoring capabilities, it is impossible to tell whether the fault is real or fake. The operator can also manually trip circuit breakers BR1-BR4 with the IED. The manual override is used when performing maintenance on the lines or other system components.

The five types of power system status are as follows: (1) malfunctioning short circuit (2) maintaining the line (3) remote trip command injection (attack) (4) changes to the relay settings (attack) (5) injection of data (attack).

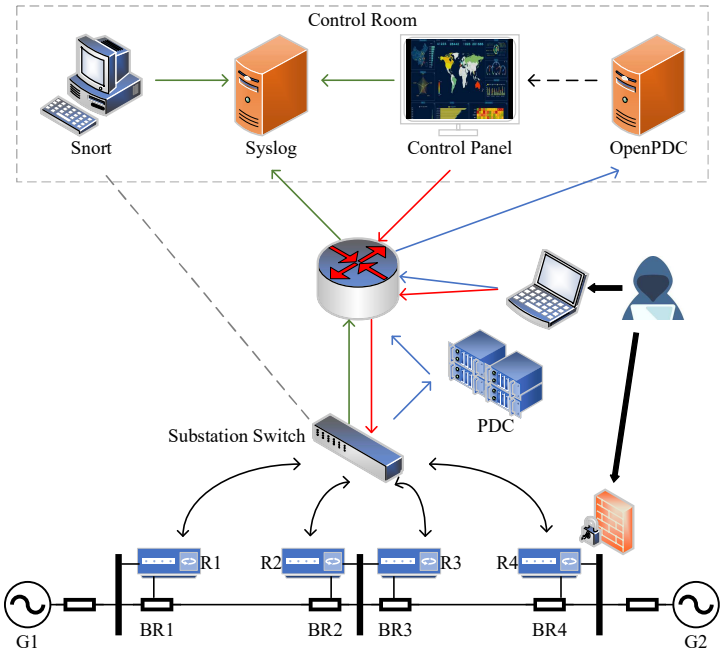

Figure 5. Experiment network diagram.

The dataset has a total of 128 features, including electrical physical quantity features and network traffic features [26]. The phasor measurement unit measures the electrical physical quantity characteristics (PMU). The system employs four PMUs, each measuring 29 electrical characteristics, for a total of 116 electrical characteristics in the dataset. The index of each column is in the form of "R#-Signal Reference" that indicates a type of measurement from a PMU specified by "R#". Each feature's name and description are listed in Table 1. For example, R1-PA1: VH is the phase angle of the phase A voltage as measured by PMU R1. There are 12 features after the PMU measurement data feature, which are the control panel log alerts, Snort alerts, and relay status (relays and PMU integrated) corresponding to the 4 IEDs. The last column is the label, which represents whether a sample is normal or under attack.

Table 1. Dataset features and descriptions.

Feature	Description
PA1:VH-PA3:VH	Phase A-C Voltage Phase Angle
PM1:V-PM3:V	Phase A-C Voltage Phase Magnitude
PA4:IH-PA6:IH	Phase A-C Current Phase Angle
PM4:I-PM6:I	Phase A-C Current Phase Magnitude
PA7:VH-PA9:VH	Pos.-Neg.—Zero Voltage Phase Angle
PM7:V-PM9:V	Pos.-Neg.—Zero Voltage Phase Magnitude
PA10:VH-PA12:VH	Pos.-Neg.—Zero Current Phase Angle
PM10:V-PM12:V	Pos.-Neg.—Zero Current Phase Magnitude
F	Frequency for relays
DF	Appearance Impedance for relays
PA:Z	Appearance Impedance for relays
PA:ZH	Appearance Impedance Angle for relays
S	Status Flag for relays
control_panel_log	control panel remote trip status
relay_log	relay status for relay1–4
snort_log	snort alert status for relay1–4

4.2. Evaluation Metrics

The confusion matrix is required to calculate the evaluation index. The confusion matrix is a table that stores the actual classification results in the dataset as well as the

prediction results of the classification model in machine learning as a matrix. Table 2 depicts the confusion matrix proposed in this paper.

Table 2. Confusion matrix.

		Predicted Value	
		Attack (A)	**Natural (N)**
Actual Value	**Attack (A)**	TA (true attack)	FN (false natural)
	Natural (N)	FA (false attack)	TN (true natural)

In Table 2, TA represents the model's correct detection of attack records, while FA represents the model's misinterpretation of normal system operations as an attack on the system. TN, on the other hand, is a normal record of the model's correct judgment, whereas FN is a misjudgment that the system is under attack during normal system operations. The confusion matrix can be used to assess the proposed model's performance. The goal of this experiment is to reduce the number of RF classifier's FN and FA. We assess our intrusion detection model against this goal using four commonly used effectiveness metrics: precision, recall, accuracy, and F1-score.

The above indicators' calculation formulas can be expressed as:

$$\text{Accuracy} = \frac{\text{TA} + \text{TN}}{\text{TA} + \text{TN} + \text{FA} + \text{FN}} \tag{4}$$

$$\text{Precision} = \frac{\text{TA}}{\text{TA} + \text{FA}} \tag{5}$$

$$\text{Recall} = \frac{\text{TA}}{\text{TA} + \text{FN}} \tag{6}$$

$$\text{F1} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} = \frac{2\text{TA}}{2\text{TA} + \text{FA} + \text{FN}} \tag{7}$$

4.3. Data Preprocessing

The line impedance in the power industrial control system dataset has inf values, which the intrusion detection model cannot identify. As a result, the inf value is set to 8000, a large enough value to replace the inf value.

The features in the power industrial control system dataset have different dimensions and dimensional units. If the order of magnitude difference between the features in the dataset is too large, the information gain provided by different features cannot be compared. A normalization procedure is essential for complete comparative examination, which resolves data comparability. In this work, we have used min-max scaling for data normalization, which maps the value into the range [0, 1] [27]. The min-max normalization transformation function is:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{8}$$

Since the data label is a string, the label needs to be numericized, the attack corresponds to 1, and the normal corresponds to 0.

4.4. Determination of Hyperparameter Importance

The significance of hyperparameters is determined by decomposing the contribution of a single hyperparameter to the variance in the RF regression model using functional ANOVA. The hyperparameter becomes more significant as the contribution increases. The names, ranges, step sizes, and descriptions of the hyperparameters used in the RF regression model are listed in Table 3.

The ranking of the variance contribution of hyperparameters obtained through functional variance analysis is shown in Table 4.

Table 3. Hyperparameter configuration.

Hyperparameter	Range	Step	Description
n_estimators	[10, 510]	50	The number of trees in the forest.
max_depth	[1, 300]	30	The maximum depth of the tree.
min_samples_split	[2, 18]	4	The minimal number of data points required to split an internal node.
min_samples_leaf	[1, 21]	4	The minimal number of data points required in order to create a leaf.
max_features	[8, 128]	10	Number of random features sampled per node.
bootstrap	{true, false}	\	Whether to train in bootstrap samples or on the full train set.
criterion	{gini, entropy}	\	Function to determine the quality of a possible split.

Table 4. Rank of The Hyperparameter Importance.

Rank	Hyperparameter	Importance
1	max_depth	0.7843
2	min_samples_leaf	0.0629
3	min_samples_split	0.0063
4	criterion	0.0044
5	n_estimators	0.0020
6	bootstrap	0.0018
7	max_features	0.0008

4.5. Hyperparameter Optimization

The sklearn RF classifier's tunable hyperparameters include the number of decision trees (n_estimators), the maximum depth of the decision tree (max_depth), the split criteria (criterion), the minimum number of samples for internal node splitting (min_samples_split), the minimum number of samples for leaf nodes (min_samples_leaf), whether bootstrap sampling is used (bootstrap), and the maximum number of features (max_features). In this paper, the hyperparameters are optimized using an improved grid search method.

As shown in Section 4.4, the order of parameter optimization is max_depth, min_samples_leaf, min_samples_split, criterion, n_estimators, bootstrap, and max_features. The roc_auc score is the area under the roc curve, and it is used as the RF classification model's evaluation standard in this experiment. The sklearn RF classifier is used as the experiment's base classifier, and it has been optimized. The default parameters of the base classifier are shown in Table 5.

Table 5. Default Parameters of the Base Classifier.

Hyperparameter	Importance
max_depth	None
min_samples_leaf	1
min_samples_split	2
criterion	gini
n_estimators	100
bootstrap	True
max_features	7

(1) Choose max_depth

The greater the depth, in general, the better the fitting effect, but it increases computational complexity and slows calculation speed, as well as the phenomenon of over-fitting. Figure 6 shows that the roc_auc score is highest when the decision tree depth is 24, reaching 0.98909. The roc_auc score stops increasing as the maximum decision tree depth reaches 30. So max_depth is set to 24.

(2) Choose min_samples_leaf

Figure 7 shows that the optimal min_samples_leaf is the default value of 1 for the base classifier, which has the highest complexity of the classification model. As the

min_samples_leaf increases, the model complexity decreases, and the roc_auc score also decreases, so min_samples_leaf is set to 1.

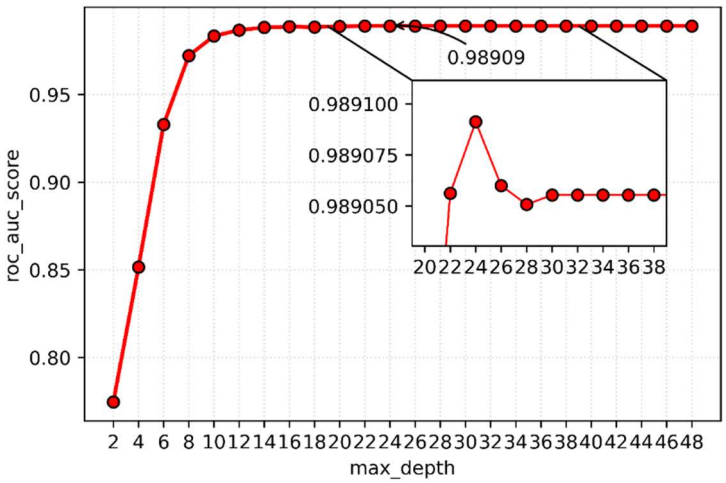

Figure 6. Performance of max_depth.

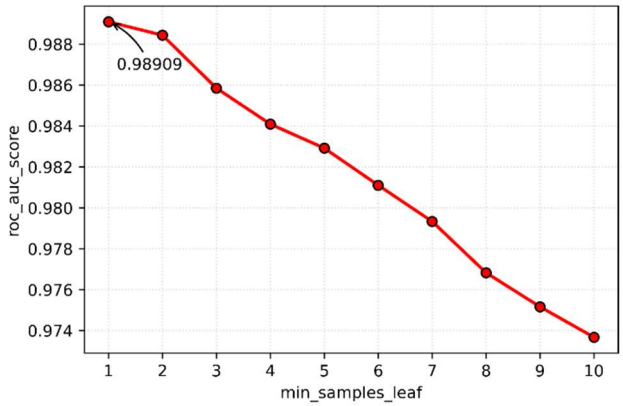

Figure 7. Performance of min_samples_leaf.

(3) Choose min_samples_split

As is illustrated in Figure 8, the best min_samples_split is the base classifier's default value of 2, which has the highest complexity of the model. As min_samples_split increases, the model's complexity and roc_auc score decreases, so min_samples_split is set to 2.

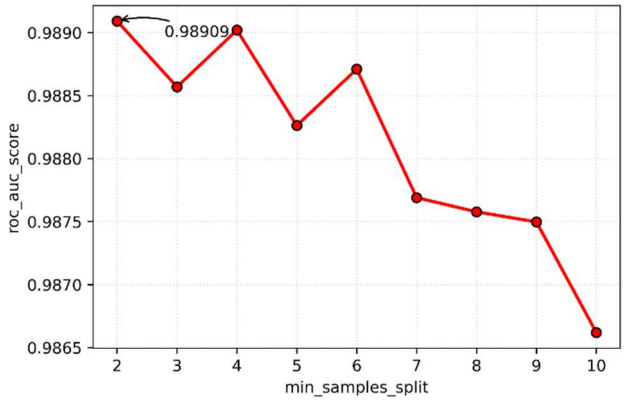

Figure 8. Performance of min_samples_split.

(4) Choose criterion

When the gini coefficient is used as the splitting criterion, the roc_auc score is 0.98909. If the splitting criterion used is entropy, the roc_auc score is 0.98901. When the gini coefficient is used as the splitting criterion, the score is higher, as shown in Figure 9. As a result, the gini coefficient is chosen as the best splitting criterion.

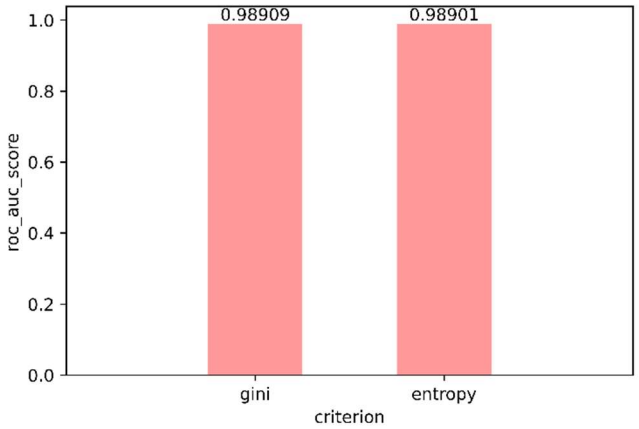

Figure 9. Performance of criterion.

(5) Choose n_estimators

This parameter is primarily used to lower the overall model's variance and improve its fitting ability. The experimental results are shown in Figure 10. The accuracy of the model will improve as the number of decision trees grows. The roc_auc score reaches a maximum of 0.9898 when the number of decision trees is 240, and it no longer increases significantly as the number of decision trees increases, as shown in Figure 10. As a result, a total of 240 decision trees are chosen.

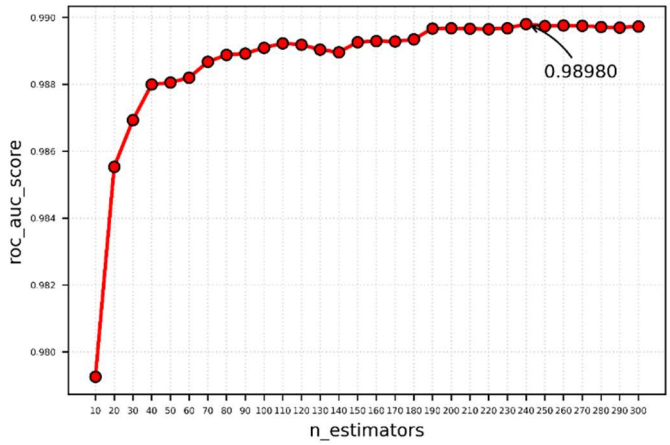

Figure 10. Performance of n_estimators.

(6) Choose bootstrap

The experimental results show that the roc_auc score is the same when sampling with bootstrap and using the entire training set as a sample. As a result, bootstrap sampling uses the base classifier's default value. As with the base classifier, bootstrapping is used for sampling.

(7) Choose max_features

Increasing the number of features in a model can improve performance because the model has more options on each node to consider, but it reduces the diversity of a single tree, which is the unique advantage of random forests, so it's important to pick the right

value. In addition, the model becomes slower as the number of features increases. As shown in Figure 11, the roc_auc score is highest when the maximum number of features is set to 9, reaching 0.99046, and then decreases as the maximum number of features increases. As a result, the maximum number of features is set to 9.

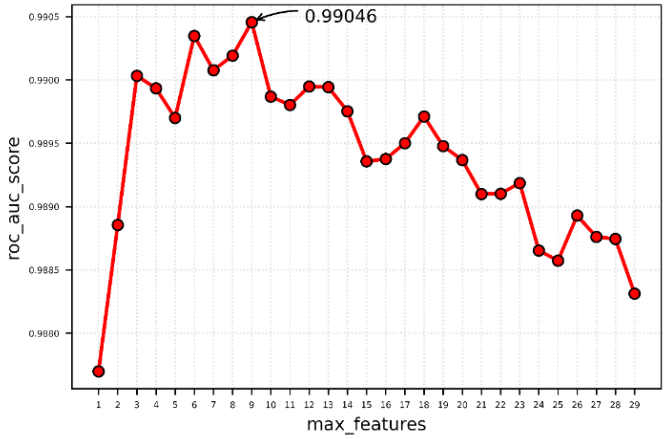

Figure 11. Performance of max_features.

4.6. Experimental Results

Table 6 shows the optimal hyperparameters obtained by IGSA, as well as the improved roc_auc score of a single hyperparameter compared to the base classifier.

Table 6. Optimal hyperparameters and improvement.

Hyperparameter	Value	Improvement
max_depth	24	+0.02129
min_samples_leaf	1	0
min_samples_split	2	0
criterion	gini	0
n_estimators	240	+0.00071
bootstrap	True	0
max_features	9	+0.00066

Experiments aimed to create the best RF classifier model using the hyperparameters in Table 6 and test it on the validation set. On the test set, the roc_auc score is 0.9935, and the roc curve is shown in Figure 12.

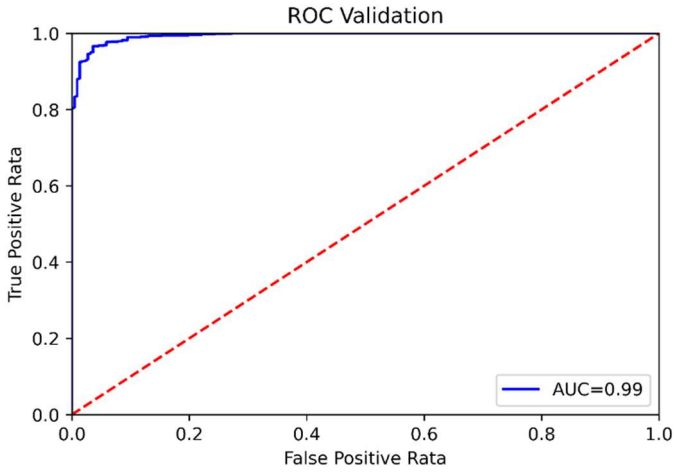

Figure 12. Roc curve of the proposed random forest classifier.

4.7. Comparative Analysis

After training, the roc_auc scores of SVM, NB, and DT on the test set are 0.5, 0.5604, and 0.9249, respectively. On the test set, the proposed method's roc_auc score is 0.9935, which are 0.4935, 0.4331, 0.0686 higher than the scores of the other three classifiers. Because Morris' work lacks roc_auc scores, it is not compared here. The results of the comparison show that the proposed RF classifier outperforms other classifiers in terms of classification performance.

We also compare the evaluation metrics of SVM, NB, DT and our proposed classifier. The comparison results are shown in Table 7. The RF classification model proposed in this paper outperforms SVM, NB, DT in terms of accuracy, precision, recall, and F1-score. The RF classifier outperforms the other three in terms of precision by 38%, 29%, and 3%, respectively. The RF classifier has a recall rate that is 21% higher, 36% higher, and 4% higher than the other three. The F1-score for the other three classes of classifiers are 31%, 34%, and 3% lower than the RF classifier, respectively.

Table 7. Performance comparison of different models.

Metrics	SVM	NB	DT	Work [18]	Work [28]	Proposed Method
Accuracy	0.78	0.63	0.95	0.96	0.96	**0.98**
Precision	0.60	0.69	0.95	0.95	0.96	**0.98**
Recall	0.78	0.63	0.95	0.95	0.96	**0.99**
F1-score	0.68	0.65	0.95	0.96	0.96	**0.99**

The JRipper + Adaboost classification method with the best classification effect in literature [18] is chosen for comparison because the performance of the RF classification method in Morris' work is not ideal. Table 7 demonstrates that the intrusion detection method of power industrial control systems proposed in this paper has 2% higher accuracy, 3% higher precision, 4% higher recall rate, and 3% higher F1-score when compared to the method in literature [18]. Compared to the method in the literature [28], as shown in Table 7, the accuracy is 2% higher, the precision is 2% higher, the recall rate is 3% higher, and the F1-score is 3% higher, the proposed method outperforms the method in [28] comprehensively.

4.8. Efficiency Analysis

In terms of computing time, the operation of Google Colab under the same parameter scale fails to run the final result due to the limited computing power of Google Colab. Small-scale hyperparameter combinations are chosen for comparison in order to perform traditional grid search hyperparameter optimization and improved grid search hyperparameter optimization. Table 8 depicts the configuration space for small-scale hyperparameter combinations.

Table 8. Configuration space for small-scale hyperparameters.

Hyperparameter	Range	Step
n_estimators	[220, 270]	10
max_depth	[20, 32]	2
min_samples_split	[2, 4]	1
min_samples_leaf	[1, 3]	1
max_features	[2, 15]	2
bootstrap	{true, false}	\
criterion	{gini, entropy}	\

In Table 9, computation time and roc_auc score are compared for grid search, genetic algorithm search and IGSA [29]. The roc_auc score of IGSA is close to the global optimum and higher than the genetic algorithm search. More importantly, the hyperparameter

optimization speed is increased by 165 times compared to grid search and 5 times faster than genetic algorithm search, proving the method's effectiveness.

Table 9. Comparison of hyperparameter search methods.

Methods	Time	Roc_auc Score
Grid search	62,104.179 s	0.9906
Genetic algorithm search	1989.95 s	0.9897
Improved grid search	375.524 s	0.9904

5. Conclusions

This paper proposes a RF intrusion detection model for electric power industrial control systems in order to address the problem of insufficient intrusion detection capabilities. A hyperparameter optimization method based on improved grid search is proposed to address the problem that the performance of the RF-based classifier is not optimal. The seven hyperparameters of the random forest model are tweaked in order of importance to generate the best classifier performance. The optimization speed is 165 times faster than grid search. The test results show that the power industrial control system intrusion detection method based on hyperparameter optimization has higher accuracy, precision, recall, F1 score, and, most importantly, roc_auc score than other similar methods. The accuracy reaches 0.98, the precision arrives to 0.98, the recall is up to 0.99 and the F1-score of the proposed method is 0.99. Therefore, the method can be used for power industrial control system intrusion detection.

However, there are still many problems with the current method. For example, more artificial intelligence algorithms, such as deep learning, can be used to detect intrusions in power industrial control systems. Advanced metaheuristics search algorithms for hyperparameter optimization can be studied to improve the performance of the power industrial control intrusion detection model. In addition, future research will focus on making intrusion detection models interpretable.

Author Contributions: Resources, N.Z. and Y.Z.; validation, L.Z. and X.Z.; methodology, N.Z. and C.Z.; writing—original draft preparation, N.Z. and Y.Z.; writing—review and editing, N.Z.; supervision, C.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the science and technology project of Stata Grid Corporation of China "Research on Key Technologies of Network Security Intelligent Hidden Risk Identification and Threat Response for Actual Combat" (Project No. 520940210009).

Data Availability Statement: All data used in this paper can be obtained by contacting the authors of this study.

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

References

1. Pan, S.; Morris, T.; Adhikari, U. Developing a Hybrid Intrusion Detection System Using Data Mining for Power Systems. *IEEE Trans. Smart Grid* **2015**, *6*, 3104–3113. [CrossRef]
2. Sun, C.; Liu, C.; Xie, J. Cyber-Physical System Security of a Power Grid: State-of-the-Art. *Electronics* **2016**, *5*, 40. [CrossRef]
3. Xie, X.; Wang, B.; Wan, T.; Tang, W. Multivariate Abnormal Detection for Industrial Control Systems Using 1D CNN and GRU. *IEEE Access* **2020**, *8*, 88348–88359. [CrossRef]
4. Morris, T.; Gao, W. Industrial Control System Traffic Data Sets for Intrusion Detection Research. In Proceedings of the International Conference on Critical Infrastructure Protection (ICCIP), Heidelberg/Berlin, Germany, 17–19 March 2014; pp. 65–78.
5. Zhang, F.; Kodituwakku, H.A.D.E.; Hines, J.W.; Coble, J. Multilayer Data-Driven Cyber-Attack Detection System for Industrial Control Systems Based on Network, System, and Process Data. *IEEE Trans. Ind. Inform.* **2019**, *15*, 4362–4369. [CrossRef]
6. Liang, G.; Weller, S.R.; Zhao, J.; Luo, F.; Dong, Z.Y. The 2015 Ukraine Blackout: Implications for False Data Injection Attacks. *IEEE Trans. Power Syst.* **2017**, *32*, 3317–3318. [CrossRef]
7. Li, Z.; Tong, W.; Jin, X. Construction of cyber security defense hierarchy and cyber security testing system of Smart Grid: Thinking and enlightenment for network attack events to national power grid of Ukraine and Israel. *Autom. Electr. Power Syst.* **2016**, *40*, 147–151.

8. Liu, H.; Lang, B. Machine learning and deep learning methods forintrusion detection systems: A survey. *Appl. Sci.* **2019**, *9*, 4396. [CrossRef]
9. Lai, Y.; Liu, Z.; Cai, X. Research on intrusion detection of industrial control system. *J. Commun.* **2017**, *38*, 143–156.
10. Denning, D.E. An Intrusion-Detection Model. *IEEE Trans. Soft. Eng.* **1987**, *SE-13*, 222–232. [CrossRef]
11. Vinayakumar, R.; Alazab, M.; Soman, K.P.; Poornachandran, P.; Al-Nemrat, A.; Venkatraman, S. Deep Learning Approach for Intelligent Intrusion Detection System. *IEEE Access* **2019**, *7*, 41525–41550. [CrossRef]
12. Khraisat, A.; Gondal, I.; Vamplew, P.; Kamruzzaman, J. Survey of Intrusion Detection Systems: Techniques, Datasets and Challenges. *Cybersecurity* **2019**, *2*, 1–22. [CrossRef]
13. Cao, Y.; Zhu, G.; Qi, X.; Zou, J. Research on Intrusion Detection Classification Based on Random Forest. *Comput. Sci.* **2021**, *48*, 459–463.
14. Gattineni, P.; Dharan, G.R.S. Intrusion Detection Mechanisms: SVM, random forest, and extreme learning machine (ELM). In Proceedings of the 2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 2–4 September 2021; pp. 273–276.
15. Wang, Z.; Jiang, D.; Huo, L.; Yang, W. An efficient network intrusion detection approach based on deep learning. *Wireless Netw.* **2021**, 1–14. [CrossRef]
16. Anton, S.D.D.; Sinha, S.; Schotten, H.D. Anomaly-based Intrusion Detection in Industrial Data with SVM and Random Forests. In Proceedings of the 2019 International Conference on Software, Telecommunications and Computer Networks (SoftCOM), Split, Croatia, 19–21 September 2019; pp. 1–6.
17. Zhu, G.; Yuan, H.; Zhuang, Y.; Guo, Y.; Zhang, X.; Qiu, S. Research on network intrusion detection method of power system based on random forest algorithm. In Proceedings of the 2021 13th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA), Beihai, China, 16–17 January 2021; pp. 374–379.
18. Hink, R.C.B.; Beaver, J.M.; Buckner, M.A.; Morris, T.; Adhikari, U.; Pan, S. Machine learning for power system disturbance and cyber-attack discrimination. In Proceedings of the 2014 7th International Symposium on Resilient Control Systems (ISRCS), Denver, CO, USA, 19–21 August 2014; pp. 1–8.
19. Liu, C.; Gu, Z.; Wang, J. A hybrid intrusion detection system based on scalable K-means+ random forest and deep learning. *IEEE Access* **2021**, *9*, 75729–75740. [CrossRef]
20. Probst, P.; Wright, M.N.; Boulesteix, A.L. Hyperparameters and tuning strategies for random forest. *Wiley Interdiscip. Rev. Data Mining Knowl. Discov.* **2019**, *9*, e1301. [CrossRef]
21. Resende, P.A.A.; Drummond, A.C. A survey of random forest based methods for intrusion detection systems. *ACM Comput. Surv.* **2018**, *51*, 1–36. [CrossRef]
22. Hutter, F.; Hoos, H.; Leyton-Brown, K. An efficient approach for assessing hyperparameter importance. In Proceedings of the 31st International Conference on Machine Learning, Beijing, China, 21–26 June 2014; pp. 754–762.
23. Van Rijn, J.N.; Hutter, F. Hyperparameter importance across datasets. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, UK, 19–23 August 2018; pp. 2367–2376.
24. Zhang, A.; Lipton, Z.C.; Li, M.; Smola, A.J. Dive into deep learning. *arXiv* **2021**, arXiv:2106.11342.
25. Pan, S.; Morris, T.; Adhikari, U. A specification-based intrusion detection framework for cyber-physical environment in electric power system. *Int. J. Netw. Secur.* **2015**, *17*, 174–188.
26. Tommy Morris—Cyber Attack Datasets. Available online: https://sites.google.com/a/uah.edu/tommy-morris-uah/ics-data-sets (accessed on 20 May 2022).
27. Singh, D.; Singh, B. Investigating the impact of data normalization on classification performance. *Appl. Soft. Comput.* **2019**, *97*, 1568–4946. [CrossRef]
28. Wu, R.; Li, X.; Bin, D. A network attack identification model of smart grid based on XGBoost. *Elect. Meas. Instr.* **2021**, *212*, 1–7.
29. Moubayed, A.; Injadat, M.; Shami, A. Optimized random forest model for botnet detection based on DNS queries. In Proceedings of the 2020 32nd International Conference on Microelectronics (ICM), Aqaba, Joradn, 14–17 December 2020; pp. 1–4.