energies

MDPI

Article

Research on Network Intrusion Detection Based on Incremental Extreme Learning Machine and Adaptive Principal Component Analysis

Jianlei Gao, Senchun Chai *, Baihai Zhang and Yuanqing Xia

School of Automation, Beijing Institute of Technology, Beijing 100081, China; jianleixinye@163.com (J.G.); smczhang@bit.edu.cn (B.Z.); xia_yuanqing@163.net (Y.X.)

* Correspondence: chaisc97@163.com; Tel.:+86-1391-145-7765

Received: 24 February 2019; Accepted: 25 March 2019; Published: 29 March 2019

Abstract: Recently, network attacks launched by malicious attackers have seriously affected modern life and enterprise production, and these network attack samples have the characteristic of type imbalance, which undoubtedly increases the difficulty of intrusion detection. In response to this problem, it would naturally be very meaningful to design an intrusion detection system (IDS) to effectively and quickly identify and detect malicious behaviors. In our work, we have proposed a method for an IDS-combined incremental extreme learning machine (I-ELM) with an adaptive principal component (A-PCA). In this method, the relevant features of network traffic are adaptively selected, where the best detection accuracy can then be obtained by I-ELM. We have used the NSL-KDD standard dataset and UNSW-NB15 standard dataset to evaluate the performance of our proposed method. Through analysis of the experimental results, we can see that our proposed method has better computation capacity, stronger generalization ability, and higher accuracy.

Keywords: network intrusion detection (IDS); incremented extreme learning machine (I-ELM); adaptive-principal component analysis (A-PCA); NSL-KDD; UNSW-NB15

1. Introduction

With the development of modern cyber-technologies, Internet technology has developed rapidly and we have entered an era of interconnection with everything. The emergence of Internet technology has brought us into a new world of interconnection, which now makes networking a very important and indispensable part of our modern life, providing us with convenience and promoting the current progress of society.

However, this technology also brings us lots of security problems caused by malicious network intrusions. According to the report of Kaspersky Laboratory in the second quarter of 2018, more than 962, 947, 023 malicious intrusions have been launched in 187 countries, which is significantly higher than the number of previous quarters. Moreover, the cyber attacks against mobile devices have also shown an unexpected trend of explosion (as is shown in Figure 1) due to the popularity of mobile networking, which aggravates the severity of the situation. The most well-known network attack is probably the "WannaCry virus" ransomware incident. It affected many computers and application systems, as well as telecommunication systems, transportation systems, energy production management systems, and industrial control systems in 2017.

In addition, more and more malicious attackers have begun to pay their attention to critical infrastructures, such as the smart city, power transmission, and so on. Once these facilities are subjected to cyber attacks, there will be tremendous trouble in the future. Therefore, protecting devices and systems against malicious attacks has become an important and urgent task, since this intrusion can result in great risks.

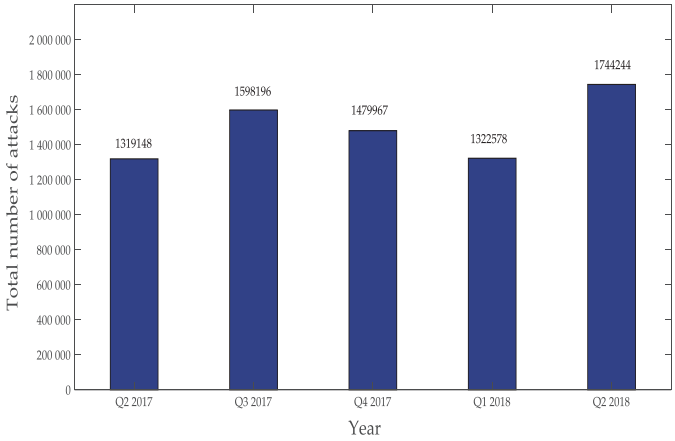

Figure 1. Global mobile threat situation.

IDS, proposed by Anderson [1], is a method/way to protect application systems from malicious attacks, which is considered as the second defending line. It collects, analyses, and distinguishes valid information, including the packet size, packet characteristics, attacker behaviour models, and access rules. Today, IDS has become a hot research topic and a thorny issue [2] due to the increasing amount of data generated by the Internet. Therefore, it makes sense to design an effective IDS, and it is an active security defending strategy that is used widely in the information field.

There are many studies on IDS, and it can be divided into two parts: the signature-based IDS, and the anomaly-based IDS. In order to design and improve the performance of IDS, scholars have proposed many methods, including statistics [3], data mining [4], the artificial immune system [5], clustering-based method [6], decision tree (DT), [7] and so on. What captured our attention was the methods based on machine learning algorithms, such as the artificial neural network. Although these methods are effective at detecting malicious behavior, they are also unable to cope with many other problems.

However, the updated hacker technology and powerful attack abilities can generate a massive amount of data with so many characteristics, such as a huge number of samples, many new attack types, and imbalanced data distribution. Those problems are prevalent in the current cyber world, which undoubtedly reduces the performance of IDS. As is known to us, the traditional IDS cannot perform well while grappling with these issues, such as the HT-assisted DoS attacks (sinkhole and blackhole attacks) in embedded systems [8]. Besides, the requirement of fast detection is also an urgent thing. Therefore, discovering how an IDS can be designed to satisfy this need is still a huge challenge.

In order to help improve the detection accuracy and solve these problems, a method is proposed by us in this paper. It combines the incremental extreme learning machine [9] (I-ELM) with adaptive principal component analysis (A-PCA) as our IDS's detection algorithms. The A-PCA is used to extract effective features automatically according to the parameter constraints, and the I-ELM is responsible for detecting malicious attacks, as is shown in Figure 2.

The major contributions of our paper are summarized as follows: (1) A new method based on I-ELM is proposed in IDS, which is an adjustable network structure ELM and can minimize training error to solve the over-fitting problem, so as to enhance the anomaly detection accuracy. (2) Our method has better performance to identify new types of cyber-attacks. (3) This method provides more computation capacity than SVM, BP, and CNN, which is suitable for dealing with massive data. (4) The proposed method proposed by us has shown different characteristics that are very suitable for IDS, and not only provides better performance but also reduces the consuming time.

The rest of this paper is organized as follows: in Section 2, we present some related research about IDS to explain the reasons for applying the algorithm. A method named the incremented extreme learning machine with adaptive principal component analysis" is selected by us to detect abnormal

network action in Section 3. Then, Section 4 introduces the NSL-KDD dataset and NSW-NB15 dataset to test our method of IDS, which is followed by the experimental results and discussion. Finally, Section 5 concludes the whole paper. The last part describes future work.

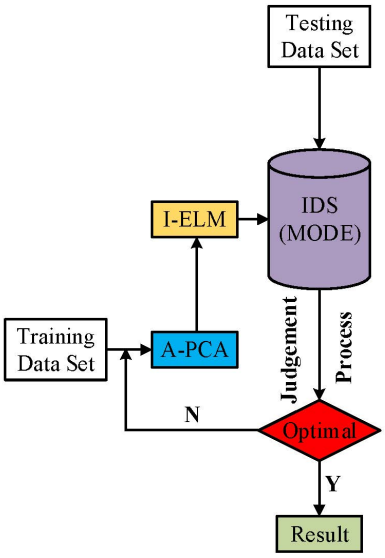

Figure 2. The network detection process of IDS.

2. Related Work

The IDS was firstly proposed by Anderson, and was designed to be a security device used to detect malicious actions on communication networks. The critical part of IDS is the classification method. Many people have been devoted to designing an IDS to protect application systems from intrusions. The traditional IDS was aimed at distinguishing abnormal actions of network environments, which is usually based on statistics technology. However, traditional IDS cannot deal with the above-mentioned problems in Section 1.

Anomaly detection is actually a two-class problem; that is to say, it can separate normal data from abnormal data by its functions. Due to the research and development of artificial intelligence, lots of machine learning algorithms have been widely applied in IDS. Usually, the model of IDS is trained by the training dataset with known attack types. Therefore, many surveillance models are designed to distinguish anomalies in the working environment, such as a collision-free surveillance model in the Internet of things [10], data mining in a smart grid [11], and a UAV surveillance framework in the smart city [12]. However, the most typical and common surveillance learning algorithms in IDS models are the deep belief network [13], the artificial neural networks (ANN) [14], the support vector machine (SVM) [15], the extreme learning machine [16], the Convolutional Neural Network (CNN) [17], and so on, which have also made IDS achieve great progress in anomaly detection. The above-mentioned methods can be used alone or combined with other algorithms.

Although they have all achieved good performance in IDS, they still have many inevitable problems. If we want to obtain a perfect classification efficiency when we use these methods, we must get a great quantity of high-quality datasets to train them, which is usually difficult to meet due to the network dataset with massive samples and imbalanced distribution. Besides, they all have their shortcomings. The BP neural network algorithm has some obvious weaknesses, such as the slow convergence speed, long training time, and the local optimum instead of global optimality. SVM is suitable for processing a small dataset, but is unsuitable for a huge dataset with lots of samples. The CNN is required to adjust its parameters. Moreover, they will lead to a high false alarm rate and take more time to train their networks. Faced with the cyber-dataset with new types of attacks,

huge number of samples, uneven and imbalanced data distributions, and highly complex structures, they tend to be more incompetent.

However, we found that there was another neural network algorithm named the incremental extreme learning machine (I-ELM), which is an improvement of ELM [9,18]. It possesses many advantages, such as stronger learning ability, faster convergence and training, the convenience of approximating a nonlinear function, higher detection efficiency, and especially its training speed, which is thousands of times faster than other methods [18,19]. It is obvious that these capabilities are well-suited for dealing with such massive samples of datasets. I-ELM is an adjustable network structure ELM, and can minimize the training error by adding nodes to solve the overfitting and under-fitting [9,20,21], which overcomes some disadvantages that it needs to optimize all parameters, and lacks appropriate activation functions. Usually, a one-hot encoding method is adopted to keep the original information of the dataset and avoid distribution interference of samples, which is also able to increase the dimension degree and decrease the cost time in the training network, as well as over-fitting. An adaptive principal component analysis (A-PCA) is helpful for reducing the feature dimension and retaining as much information as possible. Therefore, we figured that a combination of these would be ideal for our security framework design. The NSL-KDD dataset, with a massive amount of data and imbalanced sample distributions, can be applied to test our method.

However, the dataset of NSL-KDD used by us was derived from KDD-CUP99 datasets. It also has many shortcomings of the KDD-CUP99 dataset [22–25], such as: (1) the lack of new attack types, (2) a lack of real cyber-world property, (3) a failure to solve the constant changes in attacks and network architectures, (4) little new network service features, and (5) unclear emergence of the selection of network traffic. It is important that the values of the connection's content features are always set to be greater than zero, which will affect the capabilities of machine learning algorithms without a doubt. The low detection accuracy of "R2L" and "U2R" could be a result of this reason. Although this dataset deletes a great deal of redundant samples in KDD-CUP99, it does not have enough of a proper reason. All of these serious flaws may lead to a lack of practicability and validity in our model. As a result, in order to further verify detection effectiveness and make it as least time-consuming as possible, the latest dataset, named UNSW-NB15, was also chosen for application in our paper [25–27].

3. Principles of the Method

In this section, we demonstrate a method named incremental ELM (I-ELM), which is an improvement from ELM, and adaptive principal component analysis (A-PCA), which is a combination of the adaptive control idea and PCA.

3.1. I-ELM

I-ELM is a kind of incremental adjustable structure ELM, which is different from the feedback neural network algorithm, such as the BP-neural network. Due to the lack of feedback calculation, ELM obviously has stronger learning capabilities, and faster convergence and training. A typical I-ELM structure is shown in Figure 3.

The parameters α_i and b_i of the hidden nodes of I-ELM are usually independent of each other. Firstly, we assumed that we had a dataset $N = \{(x_i, o_i)|x_i \in R^n, o_i \in R^m, i = 1, \cdots, N\}$, and this network construction has n hidden nodes. Thus, the output function of the network is:

$$o_n(\mathbf{x}) = \sum_{i=1}^{n} \beta_i g_i(\mathbf{x}), \mathbf{x} \in \mathbf{R}^d, \beta_i \in \mathbf{R} \tag{1}$$

where α_i ,β_i, and g_i are the training parameters of I-ELM. Among them, α_i and β_i denote the input weights and the connection linking the output weights of the ith hidden node, respectively. b_i is the threshold value of the ith hidden node, and $g_i(\mathbf{x})$ denotes the ith hidden node's output. If another node is added, this output becomes $g_i(\mathbf{x}) = g(\boldsymbol{\alpha_i} \cdot \mathbf{x} + b_i)$.

In the I-ELM construction, it firstly always sets a new hidden node. Then, we can change the network structure through randomly adding a node one-by-one to the network. $\mathbf{e}_n \equiv o - o_n$ represents the network error function of the current I-ELM with n hidden nodes, and $o \in L^2(\mathbf{x})$ is the output function we want to achieve. Thus, the following I-ELM iteration function can be obtained:

$$o_n(\mathbf{x}) = o_{n-1}(\mathbf{x}) + \beta_n g(\boldsymbol{\alpha}_n \cdot \mathbf{x} + b_n) \tag{2}$$

Huang et al. [9,19,20] also proved that if $H_n^r \ \beta_n = (\langle \mathbf{e_{n-1}}, H_n^r \rangle / \|H_n^r\|^2)$, and $H_n^r(\mathbf{x}) = H(\alpha_n \cdot \mathbf{x} + b_n)$ are any type of function sequence, it could get a probability of 1 when $\lim_{n\to\infty} \|o - (o_{n-1} + \beta_n H_n^r)\| = 0$:

$$o_j = \sum_{i=1}^{n} \beta_i g_i(\mathbf{x}_j) = \sum_{i=1}^{n} \beta_i g(\boldsymbol{\alpha_i} \cdot \mathbf{x}_j + b_i), j = 1, \cdots, N \tag{3}$$

where $\boldsymbol{\alpha_i}$, β_i and b_i are the training parameters of I-ELM. Among them, $\boldsymbol{\alpha_i}$ is the input weight, β_i is the output weight, and b_i represents the bias of the the hidden node, respectively. $g(\boldsymbol{\alpha_i} \cdot \mathbf{x}_j + b_i)$ is the output activation function, and o_j is the output for the hidden node, j. Usually, due to the need to calculate the integral in reverse, I-ELM adopts the activation function $sig(sigmoid)$ and $tan(tanh)$. The function sig can map the input variables from $(-\infty, +\infty)$ to $(0, 1)$, which meet the probabilistic requirements $(0, 1)$. This excellent property makes ELM always use it as an activation function. The purpose of choosing the appropriate activation function and training process using this machine learning is to find some appropriate parameters that can match the training dataset without error.

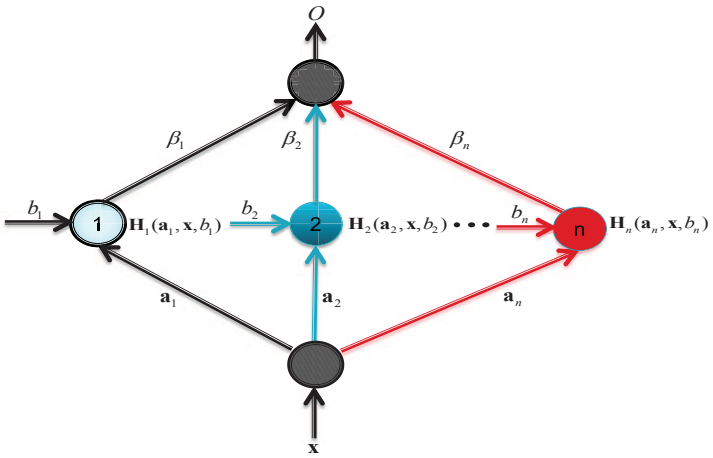

Figure 3. The I-ELM structure.

3.2. A-PCA

Usually, the same kind of data in datasets have a higher concentration and different types of data, which means the variance in the former data is smaller. Principal component analysis (PCA) is based on this principle, and it constructs a set of orthogonal bases to project the high-dimensional data to a hyperplane to convert the high-dimensional data to relative low-dimensional data. At the same time, making the variance of the reduced dataset as large as possible is helpful for retaining most of the original information.

PCA is a data-driven approach, which is applied to data compression and image processing [28], artificial intelligence [29], fault diagnosis [30], decision analysis, and so on. The purpose of using PCA is to reduce the dimensionality of the dataset and keep the original variation by preserving the most important information as much as possible.

The following is the principle of PCA. We have a dataset $N = \{(x_i, t_i) | x_i \in R^n, t_i \in R^m, i = 1, \cdots, N\}$ with N samples, as mentioned above. It can be assumed that every sample has a feature

set $X_i = \{x_{1i}, x_{2i}, \cdots, x_{mi}\}$, of which m is the maximum sample feature. Thus, the whole dataset $X = [X_1, X_2, \cdots, X_n]$ can show that:

$$X = \begin{bmatrix} x_{11} & x_{12} & x_{13} & \cdots & x_{1n} \\ x_{21} & x_{22} & x_{23} & \cdots & x_{2n} \\ x_{31} & x_{32} & x_{33} & \cdots & x_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & \cdots & x_{mn} \end{bmatrix}$$

Firstly, an average observed value μ and an average deviation of observed value δ are defined here:

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{4}$$

Then, it can be get the δ from the decentralization of all samples:

$$\delta_i = x_i - \mu = x_i - \frac{1}{n}\sum_{i=1}^{n} x_i \tag{5}$$

And we get the covariance matrix of the dataset:

$$\boldsymbol{Cov} = \frac{1}{n}\sum_{i=1}^{n}(\delta_i)(\delta_i)^T = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)(x_i - \mu)^T \tag{6}$$

Thus, we can use the singular value decomposition to get the eigenvalue $\{\lambda_1, \lambda_2, \cdots, \lambda_p\}$ and the eigenvector $\{\mu_1, \mu_2, \cdots, \mu_p\}$ of the covariance matrix $\boldsymbol{Cov}$, which correspond to each other, respectively. Furthermore, $\{\mu_1, \mu_2, \cdots, \mu_p\}$ is the maximum linearly independent eigenvector with $1 \leq p \leq m$. Therefore, it can reconstruct a new sample space by choosing some eigenvectors according to the value of the eigenvalue.

Adaptive principal component analysis (A-PCA) is a method that combines the adaptive control theory with PCA, which selects the features after being decomposed by PCA by comparing the given performance indicators we set by automatically adjusting the step size α of r to compress the dataset according to the value of Acc and dimension after using PCA.

This can be defined as $f(r, \eta_{Acc})$, with the following details.

$$f(r, \eta_{Acc}) = \begin{cases} max(\eta_{Acc}) \\ \frac{\sum_{i=1}^{p}}{\sum_{i=1}^{m}} \geq r \\ p = PCA(X) \\ s.t. 0 < r \leq 1 \\ s.t. 1 \leq p \leq m \end{cases} \tag{7}$$

where r is the ratio of A-PCA, and η_{Acc} is the accuracy of I-ELM.

Finally, the work of I-ELM and A-PCA can be described in Figure 4.

3.3. Evaluation Criteria

In this subsection, we want to select some standard performance criteria to evaluate the IDS' performance. There are many quotas to evaluate it. For this paper, we chose the most commonly used indicators, including detection accuracy (η_{Acc}), detection rate (η_{DR}), detection of the false alarm rate (η_{FAR}), and the training and testing time of IDS (T). They are shown in Table 1, and a confusion

matrix shown in Table 2. We will show some concepts before we give the specific definitions of the evaluation criteria.

Table 1. Evaluation criteria calculation about different detection algorithms.

Name	Detection Accuracy η_{Acc}	Detection Rate η_{DR}	Detection of False Alarm Rate η_{FAR}	Time
Formula	$\frac{n_{TP}+n_{TN}}{n_{TP}+n_{TN}+n_{FP}+n_{FN}}$	$\frac{n_{TP}}{n_{TP}+n_{FP}}$	$\frac{n_{FP}}{n_{FP}+n_{TN}}$	T(s)

Table 2. Confusion matrix.

Label	Predicted Label	
	Positive_Sample	**Negative_Sample**
Positive_Sample	n_{TP}	n_{FN}
Negative_Sample	n_{FP}	n_{TN}

Where, "Positive_Sample" is the normal sample in dataset and "Negative_Sample" is the abnormal sample in dataset.

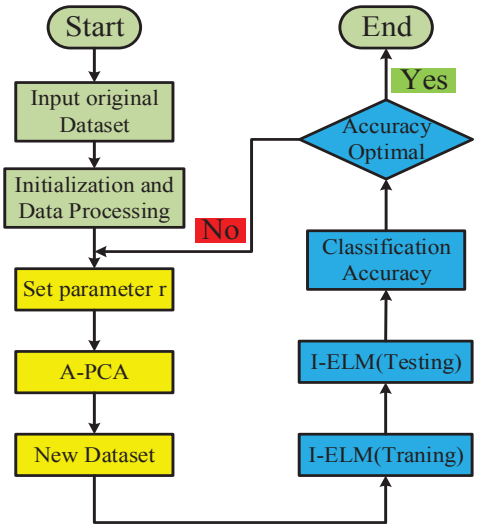

Figure 4. The work-flow chart of IDS.

• **The Number of True Positive (n_{TP})**

This is the sum of the normal sample in the dataset, which is judged by IDS as a normal sample.

• **The Number of False Positive (n_{FP})**

This is the sum of the abnormal sample in the dataset, which is misjudged by IDS as a normal sample.

• **The Number of False Negative (n_{FN})**

This is the sum of the normal sample in the dataset, which is misjudged by IDS as an abnormal sample.

• **The Number of True Negative (n_{TN})**

This is the sum of the abnormal sample in the dataset, which is judged by IDS as an abnormal sample.

• **The Starting Time of IDS (T_s)**

• **The Ending Time of IDS (T_d)**

Therefore, the following can be obtained:

$$\eta_{Acc} = \frac{n_{TP} + n_{TN}}{n_{TP} + n_{TN} + n_{FN} + n_{FP}} \tag{8}$$

$$\eta_{DR} = \frac{n_{TP}}{n_{TP} + n_{FN}} \tag{9}$$

$$\eta_{FAR} = \frac{n_{FP}}{n_{FP} + n_{TN}} \tag{10}$$

$$T = T_d - T_s \tag{11}$$

Therefore, the larger the η_{Acc} is, the higher the detection accuracy will be; the larger the η_{DR} is, the more normal samples can be identified; and the smaller η_{FAR} is, the fewer the samples which can be misidentified, which shows that the model's performance of IDS is better.

4. Experiment

4.1. Experiment Platform

In order to avoid interference from the operating platform, all methods are limited to running on the same platform and using the same programming language as shown in the following Table 3.

Table 3. Experimental platform configuration.

Index	Name	Details
1	OS	Windows 7 Ultimate x64
2	CPU	Intel Core i5 M560 *2 2.67 GHz
3	RAM	8.0 G
4	DISK	320 G
5	Software Platform	Matlab2016b
6	Program Language	M-Language

4.2. Dataset Explanation

4.2.1. NSL-KDD Dataset

The KDD-CUP99 dataset contains more than 5 million training samples and more than 2 million testing samples. Due to the large number of redundant samples in KDD99, a larger recognition and classification error is caused [17]. NSL-KDD [31] is optimized from the KDD-CUP99 dataset, which removes redundant and duplicate records and becomes the most typical dataset in IDS. The NSL-KDD dataset also has 41 features, which contain 9 discrete features and 32 continuous features. They include five types: (1) "DoS" attacks, (2) "probe" attacks, (3) "U2R" attacks, (4) "R2L" attacks, and (5) "normal". In addition, the testing dataset contains a number of different attack patterns from the training dataset, whose details are shown in the Table 4.

Table 4. Specific attack type.

Index	Type	Training Dataset	Testing Dataset
1	DoS	6	10
2	Prob	4	6
3	R2L	8	15
4	U2R	4	8
5	Normal	1	1
6	Total	23	40

The NSL-KDD is divided into four datasets: KDDTrain+, KDDTest, KDDTrain+_20percent (a subdataset of KDDTrain+), and the KDDTest21(a subdataset of the KDDTest). The specific data distribution of four datasets are shown in Table 5.

From Table 5, it can be seen that the category of normal, that is, of no attack, has the most samples, which accounts for more than half of the training dataset. However, the attack categories of "R2L" and "U2R" are less than 1%. The situation of this problem—we call it "imbalance data"—will lead to a great identification bias. As is known to us, although the NSL-KDD dataset is lacking in new practicality, its imbalanced distribution of data can also be applied to test our method.

Table 5. The specific data distribution of four datasets.

Index	Type	KDDTrain+	KDDTest+	KDDTrain+_20Percent	KDDTest-21
1	DoS	45,926	7458	9234	4342
2	Probe	11,655	2421	2289	2402
3	U2R	52	200	11	200
4	R2L	995	2754	209	2754
5	Normal	67,345	9711	13,449	2152
6	Total	125,973	22,544	25,192	11,850

4.2.2. UNSW-NB15 Dataset

The UNSW-NB15 dataset was created by the Australian Center for Cyber Security (ACCS) in 2015. It is a new dataset about IDS in research. The purpose of this dataset is to solve the inherent problems of classical KDD99 and improve the NSL-KDD dataset, which contains some new types of cyber attacks and has modem normal traffic scenarios [26,27]. The UNSW-NB15 dataset has nine different modern attack types with 49 features, which has five more attack types than NSL-KDD.

This dataset consists of 2,540,044 samples, and includes 9 attack types, known as "Fuzzers", "DoS", "Analysis", "Reconnaissance", "Exploit", "Shellcode", "Worm", "Backdoor", and "Generic", whose specific amount is shown in Table 6. For easy use, it was divided into two parts: a training dataset (175,341 samples) and a testing dataset (82,332 samples). Clearly, the UNSW-NB15 dataset also has an imbalance distribution situation. The first category, "Analysis"; second category, "Backdoor"; eighth category, "Shellcode"; and ninth category, "Worms", are also very few, whose sum is less than 2.29% of the total samples.

Table 6. Details of the UNSW_NB15 dataset.

Index	Attack Types	Training Set	Testing Set
1	Analysis	2000	677
2	Backdoor	1746	583
3	DoS	12,264	4089
4	Exploits	33,393	11,132
5	Fuzzers	18,184	6062
6	Generic	40,000	18,871
7	Reconnaissance	10,491	3496
8	Shellcode	1133	378
9	Worms	130	44
10	Normal	56,000	37,000
11	Total	175,341	82,332

4.3. Data Preprocessing

4.3.1. Data Encoding for Symbolic Features

The NSL-KDD dataset contains different features, which are divided into two parts (categorical and continuous). It must convert categorical features to continuous features to ensure the deep learning

can deal with them. In the NSL-KDD dataset, the second feature (protocol type), third feature (network service type of target host), and fourth feature (connection status) are symbolic, which cannot be used directly in machine learning. Other features are discrete. For our paper, we adopted a one-hot code to encode those same types of features and also to take an example of a second feature, as shown in Table 7. After encoding, the second feature (protocol) changed from 1 to 3.

Table 7. Example of a one-hot code for the second feature (protocol feature).

Index	Specific Feature	One-Hot Code
1	TCP	100
2	UDP	010
3	ICMP	001

We used the same encoding method to encode the third feature (service) that includes 70 service types, meaning this feature changed from 1 to 70. In the same way, the fourth feature (flag) changed from 1 to 11. Therefore, the entire features of the dataset became 122 features ($38 + 3 + 70 + 11 = 122$). By using the same encoding method, all 49 features of the UNSW-NB15 dataset became 183 features ($39 + 133 + 11 = 183$).

4.3.2. Data Normalization

Different features contained in the dataset usually have different dimensions or dimensional units, and this would affect the results of data analysis. Due to this fact, it is important to eliminate the dimensional influence. At the same time, it is necessary to keep the values of each sample feature distributed uniformly [32]. There is no doubt that normalization is the best way to resolve these problems. After data normalization, the problem of comparability between the original data's characteristic indicators lies in the unified data scale, so as to facilitate comprehensive comparative evaluation.

For this paper, we adopted the min–max normalization method to normalize data samples, which transforms the original value to make sure that it is mapped between $[0, 1]$. The function is as follows:

$$x' = \frac{(x - min(x))}{max(x) - min(x)} \tag{12}$$

where x' is the new value after normalization, and $min(x)$ and $max(x)$ are the minimum value and maximum value of the sample of the $x-$fitting feature, respectively.

4.4. Experiments of NSL-KDD Dataset

In order to evaluate the performance of our proposed method, rigorous experiments were performed on the NSL-KDD dataset. These experiments also include SVM, the BP neural network, CNN, ELM, and I-ELM. The experimental results show that our method showed better performance.

In the paper, we used the NSL-KDDTest+ as a test dataset to test our proposed method. The details of NSL-KDDTest+ are shown in Table 5, and we selected the best test results and recorded them a hundred times. The parameters of the algorithm are referenced in Section 3. By adjusting the parameters, we ensured that the training accuracy was between 97.80% and 98.50%.

Firstly, five confusion matrices can be obtained from the experimental results, which are shown in Tables 8–13. Each of them shows the detail of detection on five network behaviors (including four abnormal actions and one normal action), respectively.

From these confusion matrices, we can know that the algorithm of ELM has the best performance of detection on a third category (33, total 200), and the algorithm of I-ELM has the best performance of detection on the second category (6301, total 7458), fourth category (963, total 2754), and fifth category (9361, total 9711), and the proposed algorithm has the best performance of detection on the first category (1849, total 2421). However, the performance of our method of the other four terms is

close to the best, which are (6217, total 7458), (22, total 200), (891, total 2754), and (9332, total 9711), respectively, which also shows that the generalization capability is stronger than others in identifying new attack methods.

Table 8. Confusion matrix of SVM.

Label	DoS	Probe	U2R	R2L	Normal
DoS	1295	71	2	0	1053
Probe	50	5704	2	0	1702
U2R	17	0	31	4	148
R2L	1	0	41	380	2332
Normal	265	56	12	17	9361

Table 9. Confusion matrix of BP.

Label	DoS	Probe	U2R	R2L	Normal
DoS	1328	373	175	116	429
Probe	185	5474	693	242	864
U2R	11	15	28	98	48
R2L	1	3	12	819	1919
Normal	152	92	431	148	8888

Table 10. Confusion matrix of CNN.

Label	DoS	Probe	U2R	R2L	Normal
DoS	1215	768	0	1	437
Probe	65	6234	0	5	1154
U2R	2	111	0	4	83
R2L	5	34	0	504	2211
Normal	233	474	0	45	8959

Table 11. Confusion matrix of ELM.

Label	DoS	Probe	U2R	R2L	Normal
DoS	1689	237	11	24	460
Probe	70	6068	1	296	1023
U2R	17	0	33	6	144
R2L	3	11	9	580	2151
Normal	183	87	6	30	9405

Table 12. Confusion matrix of I-ELM.

Label	DoS	Probe	U2R	R2L	Normal
DoS	1062	940	0	0	419
Probe	45	6301	0	0	1112
U2R	0	5	1	16	178
R2L	0	45	0	963	1746
Normal	84	88	0	79	9460

Table 13. Confusion matrix of I-ELM+A-PCA.

Label	DoS	Probe	U2R	R2L	Normal
DoS	1849	258	0	36	278
Probe	12	6217	0	1	1228
U2R	85	0	22	7	86
R2L	1	2	2	891	1858
Normal	234	88	7	50	9332

The proposed algorithm can still achieve good performance in both of the two categories, especially in the third and fourth classes with fewer samples, which indicates that it is a better solution to the problems of dataset imbalance. Even compared with the accuracies of Wu et al. [17], the detection accuracy of this method is the highest.

It can be seen that the detection rates η_{DR} of all algorithms are similar and have little difference between them. However, the method we proposed has the best detection accuracy η_{Acc} and lowest detection of false alarm rates η_{FAR}, which indicates better performance. The time consumed is also the least, besides the method of ELM from Table 14 and Figure 5, which also shows our method has a stronger computation capacity. We can also see the detection details of four abnormal actions, and one normal behaviour from Figure 6.

Table 14. The values of evaluation criteria of different detection algorithms (NSL-KDD).

Method	η_{Acc}	η_{DR}	η_{FAR}	T(s)
SVM	0.7439	0.964	0.4226	20.991619
BP	0.7335	0.9153	0.404	106.2232239
CNN	0.7502	0.9226	0.3803	133.0159
ELM	0.7885	0.9685	0.3478	2.5349877
I-ELM	0.7845	0.9564	0.3456	35.3617911
I-ELM+A-PCA	0.8122	0.961	0.3003	19.97091

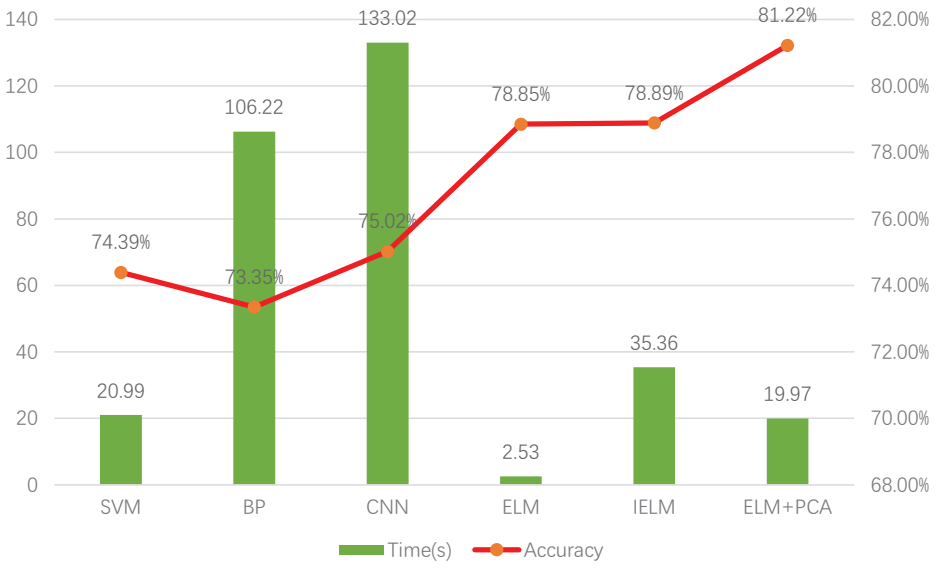

Figure 5. The average time and maximum accuracy of different methods.

Figure 7 shows that we can get different detection accuracies, different dimensions of new feature descriptions, and different average times after different ratios of A-PCA processing when we choose the NSL-KDD dataset. It is clear from Figure 7 that the A-PCA algorithm makes the characteristics of new dataset space less than those of the original dataset. We can see that there are 85 dimensions of new feature space and the $\eta_{Acc} = 79.41\%$ when $r = 0.998$, and that there are 86 dimensions of new feature space and the $\eta_{Acc} = 81.22\%$ when $r = 0.9985$—whose accuracies are better than those of different algorithms in [17]—and the $\eta_{Acc} = 78.88\%$ when we used 122 dimensions of new feature space.

Besides, it is obvious that we can get the $\eta_{Acc} = 78.80\%$ with 17 dimensions of new feature space when $r = 0.92$. These analysis results also show that the algorithm of A-PCA can reduce the dimension of the data without affecting the inherent nature of the dataset. Therefore, our proposed method has better performance than the I-ELM method without A-PCA. At the same time, our proposed method can reduce the time of training and testing.

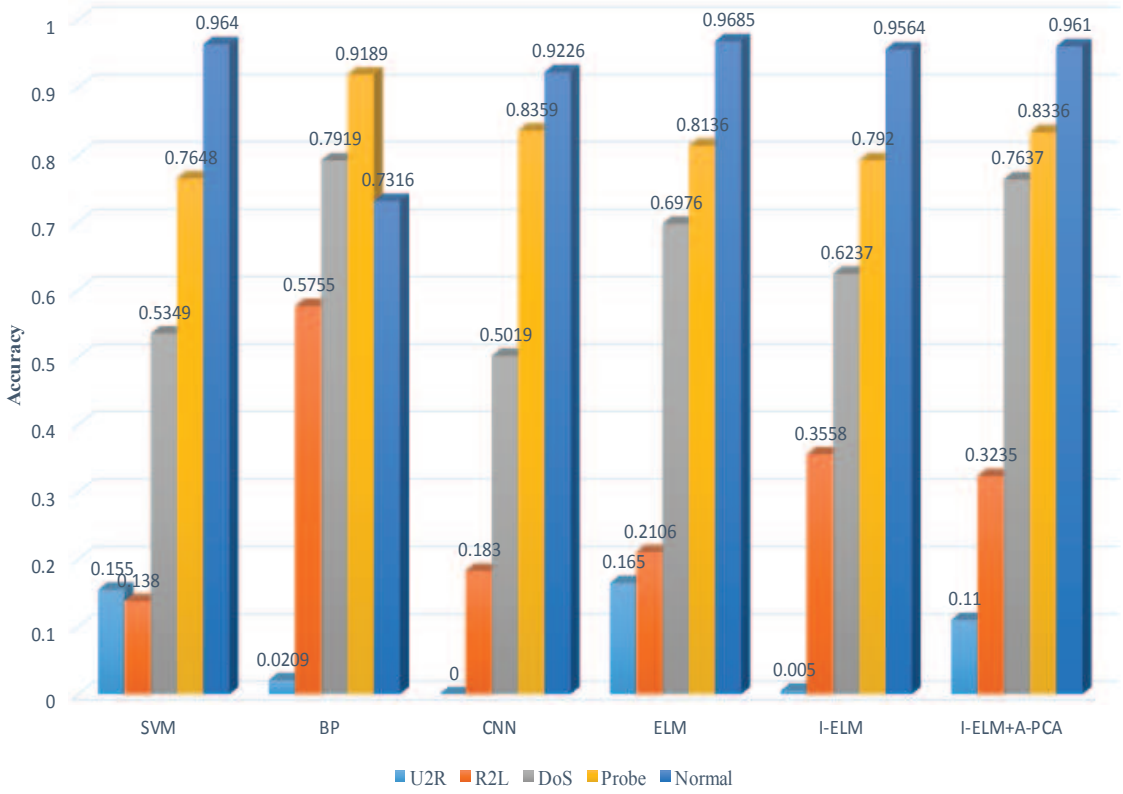

Figure 6. The confusion matrices analysis of different methods (NSL-KDD).

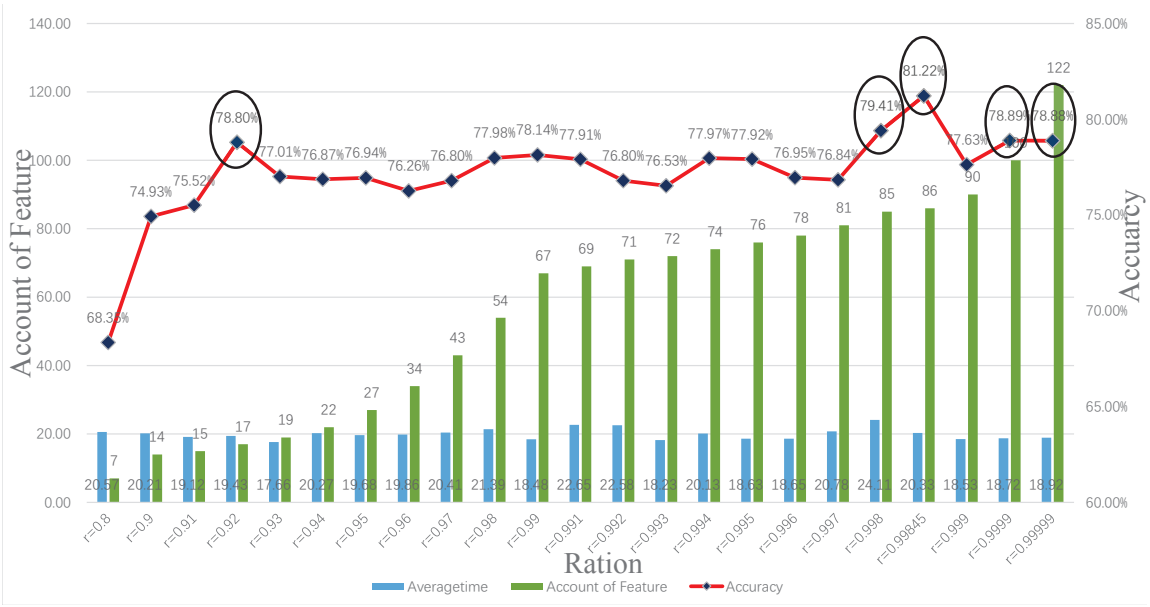

Figure 7. The different ratio of I-ELM+A-PCA (NSL-KDD).

4.5. Experiments of UNSW-NB15 Dataset

There exists a different distribution problem in the training dataset and testing dataset of the NSL-KDD dataset, which could also lead to a great identification bias and efficient disturbing of our IDS model. In order to further verify the performance of our proposed method, some experiments were also performed on the UNSW-NB15 dataset. These algorithms also include SVM, BP, CNN, ELM, and I-ELM.

In this paper, we used the training dataset of the UNSW-NB15 dataset to train our algorithm. Table 15 shows that the detection accuracy η_{Acc}, the detection rates η_{DR}, and the detection of false alarm rates η_{FAR} of all methods are different to each other. However, the method we proposed has the highest detection accuracy η_{Acc} and the highest detection rate η_{DR}. Besides, the costing time is also lower than SVM, BP, and CNN. The detection details of nine attacks and one normal behaviour are displayed in Figure 8.

Table 15. The values of evaluation criteria of different detection algorithms (UNSW-NB15).

Method	η_{Acc}	η_{DR}	η_{FAR}	T(s)
SVM	0.6573	0.6357	0.3231	5266.2600
BP	0.6260	0.6289	0.3764	714.8700
CNN	0.6355	0.6754	0.3972	3055.4718
ELM	0.6784	0.6035	0.2604	4.9271
I-ELM	0.6701	0.6575	0.3196	384.6250
I-ELM+A-PCA	0.7051	0.7736	0.3509	476.1880

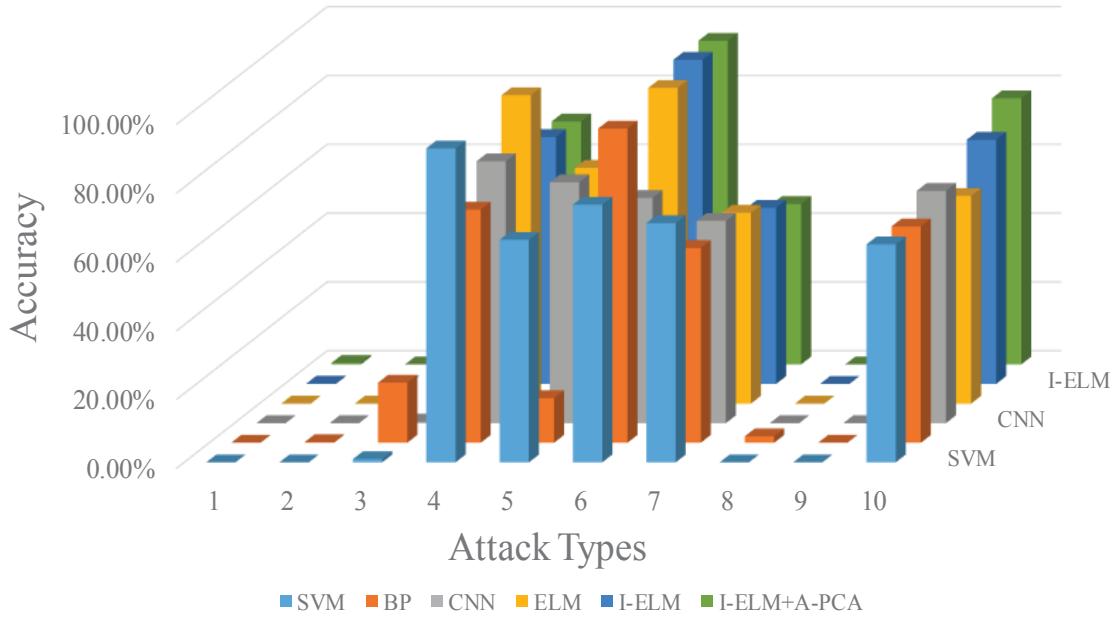

Figure 8. The confusion matrices analysis of different methods (UNSW-MB15).

From Figure 9, we can get different detection accuracies, different dimensions of new feature descriptions, and different average times after different ratios of A-PCA processing when we use the UNSW-NB15 dataset. It is very clear that the A-PCA algorithm makes the characteristics of new dataset space less than those of the original dataset. When we set $r = 0.9995$, the highest detection accuracy score of 70.51% can be achieved, which is greater than the detection accuracy of 65.23% when $r = 0.999$, and the detection accuracy of 69.82% when $r = 0.9999$.

From the above Table 15, and Figures 8 and 9, we know that the proposed method also has better performance by using the UNSW-NB15 dataset, not only in the field of detection accuracy but also the costing time of detection.

Although the UNSW-NB15 dataset has massive samples with many new network types, which brings a great knotty problem in IDS, the proposed method can also achieve good performance. From the Figures 6–9, it can be seen that our method can perform well whether you use the NSL-KDD dataset with an imbalanced data distribution or the UNSW-NB15 dataset.

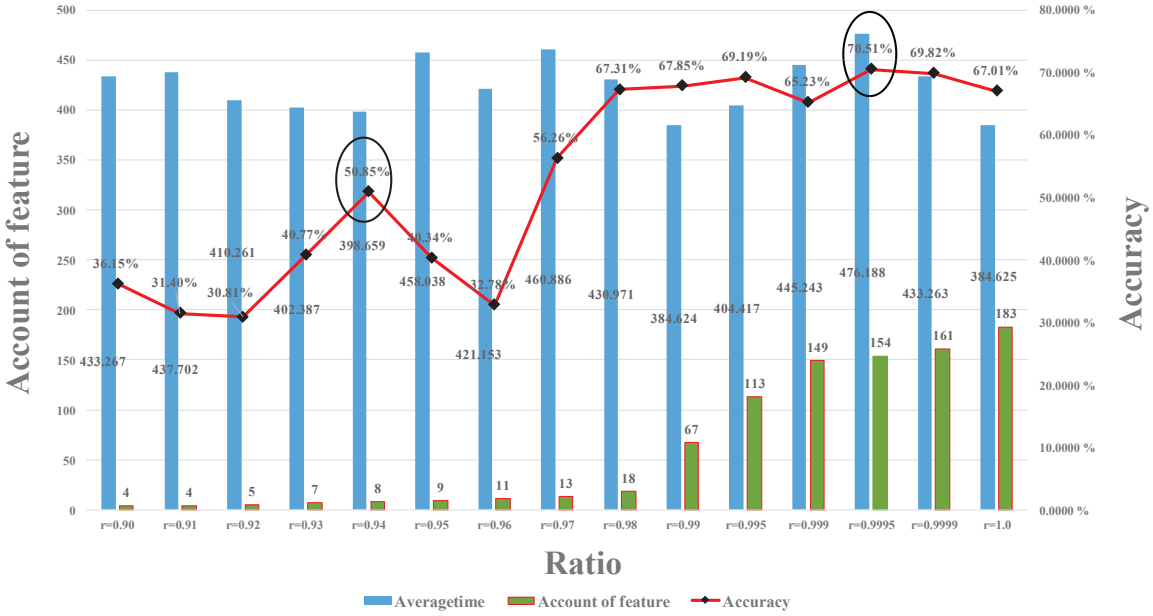

Figure 9. The different ratio of I-ELM+A-PCA (UNSW-NB5).

5. Conclusions

As we all know, the function of IDS is to find the abnormal data quickly and effectively in the dataset. Due to the imbalanced and huge amount of NSL-KDD datasets, it is necessary for the detection to solve these problems. In our research, we proposed a method used in IDS which combines I-ELM and A-PCA to detect anomalies in the network dataset.

We have compared our method with other algorithms of IDS, such as SVM, the BP neural network, CNN, ELM, and I-ELM on the performance of IDS at detecting cyber attacks using the NSL-KDD dataset and UNSW-NB15 dataset, all of which are repeated one hundred times with the same dataset on the same platform. What's more, some indicators were selected to explain the performance of different methods. Finally, the experimental results using datasets processed by A-PCA show that the detection method proposed by us can obtain a stronger capability for detecting new attacks, meaning stronger computing power and highest accuracy, and proving that our method is a better solution to the problems of the dataset with imbalance and massive samples.

6. Future Work

The experimental results show that our method has better performance than other algorithms. This network intrusion detection method can be studied further to improve the detection accuracy and extend our result to industrial control systems.

Author Contributions: J.G. established experiment platform, provided experimental data after conducting numerous experiments and then composed the first draft of this paper; S.C. provided research methods for this paper and also improved it; B.Z. helped revise and finalize the paper; Y.X. revised the paper, and provided the research methods.

Funding: This work has been supported by National Natural Science Foundation of China (No. 61573061).

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

References

1. Ehrenfeld, J.M. WannaCry, Cybersecurity and Health Information Technology: A Time to Act. *J. Med. Syst.* **2017**, *41*, 104. [CrossRef] [PubMed]
2. Yu, Y.; Kang, S.L.; Qiu, H. A new network intrusion detection algorithm: DA-ROS-ELM: INTRUSION DETECTION ALGORITHM DA-ROS-ELM. *IEEJ Trans. Electr. Electron. Eng.* **2018**, *13*. [CrossRef]

3. Amrita, M.A. Fusion of Statistic, Data Mining and Genetic Algorithm for feature selection in Intrusion Detection. *Int. J. Adv. Res. Comput. Eng. Technol.* **2013**, *2*, 1725–1731.
4. Nadiammai, G.V.; Hemalatha, M. Effective approach toward Intrusion Detection System using data mining techniques. *Egypt. Inf. J.* **2014**, *15*, 37–50. [CrossRef]
5. Powers, S.T.; He, J. A hybrid artificial immune system and Self Organising Map for network intrusion detection. *Inf. Sci.* **2012**, *178*, 3024–3042. [CrossRef]
6. Jiang, S.; Song, X.; Wang, H.; Han, J.-J.; Li, Q.-H. A clustering-based method for unsupervised intrusion detections. *Pattern Recognit. Lett.* **2006**, *27*, 802–810. [CrossRef]
7. Vuong, T.P.; Loukas, G.; Gan, D.; Bezemskij, A. Decision Tree-based Detection of Denial of Service and Command Injection attacks on Robotic Vehicles. In Proceedings of the IEEE International Workshop on Information Forensics and Security, Rome, Italy, 16–19 November 2015; pp. 1–6.
8. Zhang, L.; Wang, X.; Jiang, Y.; Yang, M.; Mak, T.; Singh, A. Effectiveness of HT-assisted Sinkhole and Blackhole Denial of Service Attacks Targeting Mesh Networks-on-chip. *J. Syst. Archit.* **2018**, *89*, 84–94. [CrossRef]
9. Huang, G.-B.; Chen, L.; Siew, C.-K. Universal approximation using incremental constructive feedforward networks with random hidden nodes. *IEEE Trans. Neural Netw.* **2006**, *17*, 879–892. [CrossRef] [PubMed]
10. Kim, H.; Benothman, J. A Collision-Free Surveillance System Using Smart UAVs in Multi Domain IoT. *IEEE Commun. Lett.* **2018**, *22*, 2587–2590. [CrossRef]
11. Choi, K.; Chen, X.; Li, S.; Kim, M. Intrusion Detection of NSM Based DoS Attacks Using Data Mining in Smart Grid. *Energies* **2012**, *5*, 4091–4109. [CrossRef]
12. Kim, H.; Mokdad, L.; Ben-Othman, J. Designing UAV Surveillance Frameworks for Smart City and Extensive Ocean with Differential Perspectives. *IEEE Commun. Mag.* **2018**, *56*, 98–104. [CrossRef]
13. Liu, Y.; Zhang, X. Intrusion Detection Based on IDBM, Dependable, Autonomic and Secure Computing. In Proceedings of the Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress, Auckland, New Zealand, 8–12 August 2016; pp. 173–177.
14. Al-Zewairi, M.; Almajali, S.; Awajan, A. Experimental Evaluation of a Multi-Layer Feed-Forward Artificial Neural Network Classifier for Network Intrusion Detection System. In Proceedings of the International Conference on New Trends in Computing Sciences, Amman, Jordan, 11–13 October 2017; pp. 167–172.
15. Hui, L.I.; Guan, X.H.; Xin, Z.; Han, C.Z. Network Intrusion Detection Based on Support Vector Machine. *J. Comput. Res. Dev.* **2003**, *6*, 1–4.
16. Cheng, C.; Tay, W.P.; Huang, G.B. Extreme learning machines for intrusion detection. In Proceedings of the International Joint Conference on Neural Networks, Brisbane, QLD, Australia, 10–15 June 2012; pp. 1–8.
17. Wu, K.; Chen, Z.; Li, W. A Novel Intrusion Detection Model for a Massive Network Using Convolutional Neural Networks. *IEEE Access* **2018**, *6*, 50850–50859. [CrossRef]
18. Huang, G.; Zhu, Q.; Siew, C.K. Extreme learning machine: Theory and applications. *Neurocomputing* **2006**, *70*, 489–501. [CrossRef]
19. Huang, G.B.; Li, M.B.; Chen, L.; Siew, C.K. Incremental extreme learning machine with fully complex hidden nodes. *Neurocomputing* **2008**, *71*, 576–583. [CrossRef]
20. Feng, G.; Huang, G.-B.; Lin, Q.; Gay, R. Error minimized extreme learning machine with growth of hidden nodes and incremental learning. *IEEE Trans. Neural Netw.* **2009**, *20*, 1352–1357. [CrossRef]
21. Miche, Y.; Sorjamaa, A.; Bas, P.; Simula, O.; Jutten, C.; Lendasse, A. OP-ELM: Optimally Pruned Extreme Learning Machine. *IEEE Trans. Neural Netw.* **2010**, *21*, 158–162. [CrossRef]
22. Mchugh, J. Testing Intrusion detection systems: A critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln Laboratory. *Acm Trans. Inf. Syst. Secur.* **2000**, *3*, 262–294. [CrossRef]
23. Hindy, H.; Brosset, D.; Bayne, E.; Seeam, A.; Tachtatzis, C.; Atkinson, R.C.; Bellekens, X.J.A. A taxonomy and survey of intrusion detection system design techniques, network threats and datasets. *arXiv* **2018**, arXiv:1806.03517.
24. Hindy, H.; Hodo, E.; Bayne, E.; Seeam, A.; Atkinson, R.; Bellekens, X. A taxonomy of malicious traffic for intrusion detection systems. *arXiv* **2018**, arXiv:1806.03516.
25. Al Tobi, A.M.; Duncan, I. KDD 1999 generation faults: A review and analysis. *J. Cyber Secur. Technol.* **2018**, *2*, 164–200. [CrossRef]

26. Moustafa, N.; Slay, J. The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set. *Inf. Syst. Secur.* **2016**, *25*, 18–31. [CrossRef]
27. Moustafa, N.; Slay, J. UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In Proceedings of the Military Communications & Information Systems Conference, Canberra, ACT, Australia, 10–12 November 2015.
28. Cocianu, C.; State, L.; Vlamos, P. A new adaptive PCA scheme for noise removal in image processing. In Proceedings of the International Symposium ELMAR, Zadar, Croatia, 10–12 September 2008.
29. Castaño, A. PCA-ELM: A Robust and Pruned Extreme Learning Machine Approach Based on Principal Component Analysis. *Neural Process. Lett.* **2013**, *37*, 377–392. [CrossRef]
30. Hu, Z.; Chen, Z.; Gui, W.; Jiang, B. Adaptive PCA based fault diagnosis scheme in imperial smelting process. *ISA Trans.* **2014**, *53*, 1446–1455. [CrossRef] [PubMed]
31. Tavallaee, M.; Bagheri, E.; Lu, W.; Ghorbani, A.A. A detailed analysis of the KDD CUP 99 data set. In Proceedings of the IEEE International Conference on Computational Intelligence for Security & Defense Applications, Ottawa, ON, Canada, 8–10 July 2009.
32. Dash, T. A study on intrusion detection using neural networks trained with evolutionary algorithms. *Soft Comput.* **2017**, *21*, 2687–2700. [CrossRef]