symmetry

MDPI

Article

Application of GA-WELM Model Based on Stratified Cross-Validation in Intrusion Detection

Chen Chen [1], Xiangke Guo [2,*], Wei Zhang [1,*], Yanzhao Zhao [1], Biao Wang [1], Biao Ma [1] and Dan Wei [1]

[1] China Xi'an Satellite Control Center, Xi'an 710043, China
[2] College of Air and Missile Defense, Air Force Engineering University, Xi'an 710051, China
* Correspondence: afeu_guoxiangke@163.com (X.G.); blissnail@163.com (W.Z.)

Abstract: Aiming at the problem of poor detection performance under the environment of imbalanced type distribution, an intrusion detection model of genetic algorithm to optimize weighted extreme learning machine based on stratified cross-validation (SCV-GA-WELM) is proposed. In order to solve the problem of imbalanced data types in cross-validation subsets, SCV is used to ensure that the data distribution in all subsets is consistent, thus avoiding model over-fitting. The traditional fitness function cannot solve the problem of small sample classification well. By designing a weighted fitness function and giving high weight to small sample data, the performance of the model can be effectively improved in the environment of imbalanced type distribution. The experimental results show that this model is superior to other intrusion detection models in recall and McNemar hypothesis test. In addition, the recall of the model for small sample data is higher, reaching 91.5% and 95.1%, respectively. This shows that it can effectively detect intrusions in an environment with imbalanced type distribution. Therefore, the model has practical application value in the field of intrusion detection, and can be used to improve the performance of intrusion detection systems in the actual environment. This method has a wide application prospect, such as network security, industrial control system, and power system.

Keywords: intrusion detection; weighted extreme learning machine; stratified cross-validation; weighted fitness function; imbalanced dataset

check for updates

Citation: Chen, C.; Guo, X.; Zhang, W.; Zhao, Y.; Wang, B.; Ma, B.; Wei, D. Application of GA-WELM Model Based on Stratified Cross-Validation in Intrusion Detection. *Symmetry* **2023**, *15*, 1719. https://doi.org/10.3390/sym15091719

Academic Editor: Aviv Gibali

Received: 15 July 2023
Revised: 28 July 2023
Accepted: 5 September 2023
Published: 7 September 2023

1. Introduction

With the wide application of computer technology and network in various fields, the network security situation is becoming more and more serious. The firewall, an early security measure, cannot meet the current network security requirements. How to find attacks using the network has become the primary goal of preventing network intrusion [1]. The core of intrusion detection system (IDS) is to collect and analyze the data in the network and check whether behaviors in the network are safe. According to the detection results, the corresponding defensive measures are started. As a proactive security defense technology, intrusion detection can effectively guarantee the security of the network [2]. Intrusion detection technology can generally be divided into two different detection types: misuse and anomaly [3,4]. Misuse detection is used to match the data in the host or network according to the known attack type information. If the matching results are consistent, it is defined as attack behavior. This detection method can only detect existing attack types in the sample database, but cannot detect unknown attack types [5]. The anomaly detection method is used to establish normal trajectory characteristics in the database, and regards all behaviors that deviate from the normal trajectory as intrusion [6].

With the success of machine learning technology in the fields of image classification, language translation, and speech recognition, the research of machine learning in intrusion detection has attracted more and more attention from network security researchers. Machine learning algorithms such as support vector machine (SVM) [7], BP neural network [8], extreme learning machine (ELM) [9], decision tree [10], and K-nearest neighbor method [11] have achieved good results in intrusion detection. Traditional machine algorithms are based on balanced data. They pay more attention to the large sample data while ignoring the small sample data during training. As a typical imbalanced dataset, the number of normal behaviors in intrusion detection dataset is much larger than the number of attack behaviors [12]. When using machine learning algorithm for intrusion detection, even if all data are classified as normal behaviors, good accuracy can be achieved. In order to solve the problem of data imbalance, many algorithms for imbalanced datasets have been proposed. Among them, undersampling [13] and oversampling [14] are the most widely used. As another sample rebalancing method, weighted extreme learning machine (WELM) can also solve this problem [15]. In addition, as a training method, stratified cross-validation (SCV) can also improve the detection ability of the model for small sample data [16].

When designing a machine learning model, designers tend to subjectively take a default value for the hyperparameters of the model [17]. The default value may perform well on some datasets, but may not perform well on others. This is because different types of datasets have obvious differences in data distribution and size. Therefore, there will not be a fixed hyperparameter that can be well adapted to all types of datasets [18]. In order to train a suitable model, many researchers often pre-set the value of the hyperparameters before training the machine learning model. The quality of the value will directly affect the performance of the model [19]. The commonly used hyperparameter selection methods include trial and error method [20], expert experience method [21], and meta-heuristic optimization algorithm [22]. The trial and error method takes a finite number of hyperparameter values, uses each value for training, and finally takes the value corresponding to the best training result as the hyperparameter value of the model. The expert experience method estimates a suitable value based on one's own research experience, or directly uses the value scheme in other people's research results. The meta-heuristic optimization algorithm uses an optimization model to iterate the hyperparameter values for a limited number of times in a certain range, and finally gets the optimal solution. Designing a good fitness function is very important for the optimization ability of meta-heuristic optimization algorithm [23].

The specific contributions of this paper are as follows:

(1) A model combining SCV and WELM is proposed. SCV improves the generalization performance of the model from a macro perspective based on data distribution. WELM effectively addresses the issue of data imbalance from a micro perspective based on model structure. The model organically integrates macro and micro solutions.

(2) A novel weighted fitness function is designed. Giving different weights to different data types can artificially control the evolution direction of genetic algorithm (GA). This makes the hyperparameters optimized by GA more suitable for solving data imbalance problems.

(3) The improved GA is used to optimize the weight ω and bias b of WELM.

The other parts of this study are as follows. In Section 2, the related work is discussed. Section 3 describes the methods used in this paper. In Section 4, the simulation experiments and results analysis are carried out. Section 5 gives the research conclusion.

2. Related Work

In the field of intrusion detection, more and more researchers are paying attention to the study of machine learning. It has achieved remarkable results in its theoretical development, key technology research, and application. ELM is a machine learning algorithm proposed by Huang et al. [24], which has received more attention since its proposal. It also shows good performance in intrusion detection. Ali et al. [25] proposed a method to detect software-defined networking (SDN) intrusion. The traffic of each forwarding unit in SDN was supervised by the controller. ELM could be used to detect the traffic of each unit. The accuracy of the proposed method on NSL-KDD and KDDCUP99 benchmark datasets is 95% and 99.2%, respectively. Al-Yaseen et al. [26] used a differential evolution algorithm to select useful features. After feature selection, ELM classifier was applied to evaluate the selected features. This method can reduce redundant information in the dataset, improve the performance of IDS, and reduce its processing time. Lin et al. [27] designed a multi feature extraction ELM (MFE-ELM) algorithm for cloud computing, added a multi feature extraction process to cloud servers, and used the MFE-ELM algorithm deployed on cloud nodes to detect and discover network intrusion on cloud nodes. The proposed algorithm can effectively detect and recognize most network data packets with good model performance. Although these methods perform well on accuracy, due to the fact that ELM is a model oriented towards balanced datasets, they often do not perform well when facing imbalanced datasets.

In order to solve this problem, various processing methods for imbalanced datasets have been proposed. Park et al. [28] used generative adversarial network (GAN) to generate reasonable synthetic data for small sample data, which alleviates the data imbalance in an intrusion detection dataset. Finally, the superiority of the proposed method is proved on a variety of intrusion detection datasets. However, the data synthesized by GAN cannot guarantee the authenticity, that is, whether this data are reasonable in the real network. Therefore, the model trained by using GAN technology may not perform well in the actual network. Yan et al. [29] designed a method to optimize WELM based on an improved grey wolf algorithm. This method uses the improved grey wolf algorithm to optimize the weight and bias of WELM, which avoids the problems of low search speed and local optimization. This achieves high search efficiency. However, WELM increases the detection rate of small sample data at the expense of large sample data. Ma et al. [30] used a cross-validation method to divide the dataset, then used extreme gradient boosting (XGboost) to classify it. The F1 value and average AUC of this method are 0.72 and 0.89, respectively. Ma et al. [31] implemented a mixed sampling technique in combination with the Borderline SMOTE and Gaussian mixture model (GMM), which can effectively alleviate the serious class imbalance problem in intrusion detection datasets. The quantum particle swarm optimization algorithm can optimize and select the optimal number of convolution kernels for each one-dimensional convolution layer, and improve the detection rate of the model for small sample data. The precision rates for the small sample data classes R2L and U2R are improved by 68% and 66%, respectively. However, using the SMOTE sampling technique destroys the data distribution of the original dataset, and the model does not adequately mine the distributional patterns in the data. However, cross-validation only improves the data utilization rate and mines the deep-seated rules in the data, but does not improve the performance of the model in essence.

In order to solve the unscientific problem of ELM hyperparameter selection, the method of using an optimization algorithm for hyperparameter optimization has been widely used. Du et al. [32] proposed a lightweight SVM intrusion detection model based on cloud-fog collaboration. The SVM optimized by particle swarm optimization is used to train the dataset and obtain the optimal SVM intrusion detection classifier. This model is superior to other similar algorithms in detection time, detection rate, and accuracy on the KDD CUP 99 dataset, and can effectively solve the intrusion detection problem in fog environment. Zivkovic et al. [33] used the improved firefly algorithm to adjust and optimize the hyperparameter of XGBoost classifier, which was verified on the NSL-KDD dataset and USNW-NB15 dataset. The experimental results show that the optimized model has improved both classification accuracy and average accuracy. Yamin [34] designed a chaotic metaheuristics with optimal multi-spiking neural network-based intrusion detection (CMOMSNN-ID) model. By using the whale optimization algorithm to optimize the hyperparameters of the model, the performance of the model was optimized and the classification ability was improved. The simulation results show that the performance of the proposed model is superior to other existing models, with a maximum accuracy of 99.20%.

The above optimization algorithms choose accuracy as fitness function. However, in the face of imbalanced datasets, the models with over-optimized hyperparameters will be more biased towards large sample data, resulting in the optimization process not being carried out in the envisaged direction.

This paper proposes a GA-WELM model based on SCV (SCV- GA-WELM), which can effectively detect small sample attacks. Firstly, the training set is divided into multiple SCV datasets to improve the data reuse rate. In the stage of optimizing WELM with GA, a novel fitness function is proposed which improves the detection rate of small sample data.

3. Materials and Methods

3.1. Evaluation Indicators

Compare the real label with the predicted label, and calculate the values of true positives (*TP*), true negatives (*TN*), false positives (*FP*), and false negatives (*FN*). The definitions of accuracy, recall, and precision are as follows:

$$\text{accuracy} = \frac{TP + TN}{TP + FP + FN + TN} \tag{1}$$

$$\text{recall} = \frac{TP}{TP + FN} \tag{2}$$

$$\text{precision} = \frac{TP}{TP + FP} \tag{3}$$

3.2. WELM

In this paper, a WELM classification model is established based on ELM. Suppose N training samples $\{x_i, t_i\}_{i=1}^{N}$ are given, where $x_i = [x_{i1}, x_{i2}, \cdots, x_{in}]^{\mathrm{T}} \in R^n, t_i = [t_{i1}, t_{i2}, \cdots, t_{im}]^{\mathrm{T}} \in R^m$, n is the feature number of the sample, and m is the type number of the sample. A feedforward neural network output model with L hidden layer nodes can be expressed as follows:

$$\sum_{h=1}^{L} \beta_h G(\omega_h, b_h, x) = o_i, i = 1, 2, \cdots, N \tag{4}$$

where β_h is the output weight of the h-th hidden layer neuron, G is the activation function of neurons in the hidden layer, ω_h and b_h are the weight and bias of neurons in the h-th hidden layer, respectively, x is the input sample, o_i is the actual output value of the i-th training sample, and t_i is the expected output of the i-th training sample.

For a training sample with a quantity of N, $\{x_i, t_i\}_{i=1}^{N}, x_i \in R^n$, there are ($\omega_h$, b_h) and β_h, with $\sum\limits_{i=1}^{L} \|o_i - t_i\| = 0$, so that the single-hidden layer feedforward network (SLFN) can approach the training set $\{x_i, t_i\}_{i=1}^{N}$ with zero error, namely,

$$\sum_{h=1}^{L} \beta_h G(\omega_h, b_h, x_i) = t_i, i = 1, 2, \cdots, N \tag{5}$$

Equation (5) can be further simplified as:

$$\boldsymbol{H}\beta = \boldsymbol{T} \tag{6}$$

where $\boldsymbol{H}$ is the hidden layer output matrix, β is the output weight matrix of hidden layer, and $\boldsymbol{T}$ is the expected output matrix corresponding to the training sample.

In the training process of ELM, the hidden layer weight ω_h and hidden layer bias b_h are randomly generated when initializing the network hyperparameters, and they remain unchanged throughout the training and testing process. Since the input training samples, the input weight, and bias of hidden layer and the expected output are all known, the whole training process is to find out the hidden layer output weight matrix β in the ELM model, thus obtaining a complete classification model.

It can be solved by Moore–Penrose generalized inverse matrix $\boldsymbol{H}^{+}$ of hidden layer output matrix $\boldsymbol{H}$.

The Moore–Penrose generalized inverse matrix $\boldsymbol{H}^{+}$ of the hidden layer output matrix $\boldsymbol{H}$ can be solved to obtain

$$\hat{\beta} = \boldsymbol{H}^{+}\boldsymbol{T} \tag{7}$$

In the equation, there are many ways to calculate $\boldsymbol{H}^{+}$. In ELM, the orthogonal projection method (KKT) is usually used to solve $\boldsymbol{H}^{+}$. When $\boldsymbol{H}^{T}\boldsymbol{H}$ is a nonsingular matrix, $\boldsymbol{H}^{+} = (\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H})^{-1}\boldsymbol{H}^{\mathrm{T}}$; when $\boldsymbol{H}\boldsymbol{H}^{\mathrm{T}}$ is a nonsingular matrix, $\boldsymbol{H}^{+} = \boldsymbol{H}^{\mathrm{T}}(\boldsymbol{H}\boldsymbol{H}^{\mathrm{T}})^{-1}$.

In order to solve Equation (7), a small enough regular term $\frac{1}{C}$ is added to the diagonal of $\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H}$ or $\boldsymbol{H}\boldsymbol{H}^{\mathrm{T}}$, which makes the classification model have better stability and generalization performance. The output weight of the hidden layer can be expressed as

$$\hat{\beta} = \begin{cases} \boldsymbol{H}^{\mathrm{T}}\left(\frac{1}{C} + \boldsymbol{H}\boldsymbol{H}^{\mathrm{T}}\right)^{-1}\boldsymbol{T}, N < L \\ \left(\frac{1}{C} + \boldsymbol{H}^{\mathrm{T}}\boldsymbol{H}\right)^{-1}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{T}, N \geq L \end{cases} \tag{8}$$

The output function of ELM can be expressed as

$$f(x) = h(x)\hat{\beta} = \begin{cases} h(x)\boldsymbol{H}^{\mathrm{T}}\left(\frac{1}{C} + \boldsymbol{H}\boldsymbol{H}^{\mathrm{T}}\right)^{-1}\boldsymbol{T}, N < L \\ h(x)\left(\frac{1}{C} + \boldsymbol{H}^{\mathrm{T}}\boldsymbol{H}\right)^{-1}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{T}, N \geq L \end{cases} \tag{9}$$

In the classification problem, not all the classified sample data are evenly distributed. In order to solve the classification problem of imbalanced samples, Zong et al. [35] proposed WELM on the basis of ELM. Give each sample a weight according to the weighting scheme.

Weighting scheme $\boldsymbol{W}_1$:

$$\boldsymbol{W}_1 = \frac{1}{\mathrm{Count}(t_i)} \tag{10}$$

where Count(t_i) is the number of samples with the class of t_i in the training samples.

Weighting scheme $\boldsymbol{W}_2$: Push the ratio of small sample data to large sample data towards 0.618:1 (golden ratio). This scheme actually sacrifices the classification accuracy of large sample data in exchange for the classification accuracy of small sample data.

$$\boldsymbol{W}_2 = \begin{cases} \frac{0.618}{\mathrm{Count}(t_i)}, t_i \text{ belongs to large sample data} \\ \frac{1}{\mathrm{Count}(t_i)}, t_i \text{ belongs to small sample data} \end{cases} \tag{11}$$

The output weight of the WELM hidden layer can be expressed as

$$\hat{\beta} = \boldsymbol{H}^{+}\boldsymbol{T}\begin{cases} \boldsymbol{H}^{\mathrm{T}}\left(\frac{1}{C} + \boldsymbol{W}\boldsymbol{H}\boldsymbol{H}^{\mathrm{T}}\right)^{-1}\boldsymbol{W}\boldsymbol{T}, N < L \\ \left(\frac{1}{C} + \boldsymbol{H}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{H}\right)^{-1}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{T}, N \geq L \end{cases} \tag{12}$$

where the weighting matrix is a diagonal matrix of $N \times N$. N main diagonal elements correspond to N samples, and different weights are given to different sample classes, in which the same class has the same weight. In order to get better detection performance of small sample data, this paper adopts weighting scheme $\boldsymbol{W}_2$.

3.3. SCV

Cross-validation [36] is a statistical analysis method used to verify the performance of classifiers. The basic idea is to group the training data, one part as the training set and the other part as the test set. Firstly, the classifier is trained with the training set, and then the trained model is tested with the test set, which is used as the performance index to evaluate the classifier.

K-fold cross-validation is used to divide the training data into K independent subsets, extract one subset as a test set without repetition, and combine the rest K-1 subset data as a training set. Figure 1 shows the 10-fold cross-validation principle.

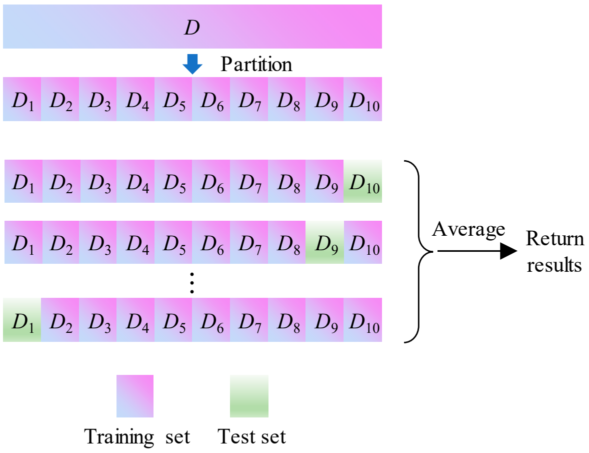

D
Partition
D_1 D_2 D_3 D_4 D_5 D_6 D_7 D_8 D_9 D_{10}
Average
Return results
Training set
Test set

Figure 1. Ten-fold cross-validation principle.

The specific steps of K-fold cross-validation are as follows:

(1) The original training set D is randomly divided into K equal parts (D_1, D_2, . . ., D_K), one of which is taken as the K-fold test set, and the rest K − 1 is taken as the K-fold training set. The K-fold training set is used to train the classifier, and the K-fold test set is classified.

(2) Repeat Step (1) until each fold of data has been tested.

(3) Take the average accuracy of K classifiers obtained from K tests, and the average accuracy obtained is used as the accuracy of K-fold cross-validation.

For most datasets, K-fold cross-validation can avoid over-fitting and under-fitting. However, in the face of the imbalanced dataset such as intrusion detection, the number of normal behaviors is far greater than the number of attack behaviors. K-fold cross-validation randomly assigns different classes of data to each fold. If a small number of attacks are concentrated in a fold, the classification performance of the classifier will be reduced. K-fold SCV is based on K-fold cross-validation to ensure that the proportion of class samples in each fold is roughly the same. SCV is a technique to rearrange data to ensure that each fold can well represent all classes appearing in the data. In this paper, 10-fold SCV is selected.

3.4. Weighted Fitness Function

In this paper, GA is used to optimize hyperparameters in WELM. GA, proposed by Professor Holland in 1975, is a global probabilistic search optimization algorithm based on the theory of natural selection and genetic mutation [37]. When using GA for optimization, it is necessary to encode the candidate solution of the problem, that is, one candidate solution corresponds to one code. The code is usually binary, which is represented by "0" and "1". "1" means that the candidate solution is selected. All candidate solutions are combined together to form chromosomes. In the process of iterative evolution, the fitness function is constructed to calculate the fitness of each individual. The greater the fitness value, the greater the possibility that the individual will be retained.

The setting of fitness function is very important for the optimization ability of GA, and in different optimization problems, the setting of fitness function needs to be based on specific optimization problems. For common classification problems, accuracy is usually used as a fitness function to evaluate the optimization effect.

According to the different proportion of each class of data in the dataset, the dataset can be divided into balanced dataset and imbalanced dataset. Let the dataset consist of three classes of data: A, B, and C, and the number of data is a_T, b_T, and c_T in turn. The model is used to classify datasets, and the correct number of each classification is a, b, and c in turn.

Scenario 1: The dataset is a balanced dataset.

Because the amount of all kinds of data in the balanced dataset is roughly equal, then $a_T = b_T = c_T$. The accuracy of the model can be expressed as:

$$S = \frac{a+b+c}{a_T+b_T+c_T} = \frac{a+b+c}{3a_T} = \frac{1}{3}\left(\frac{a}{a_T}+\frac{b}{b_T}+\frac{c}{c_T}\right) = \frac{1}{3}(R_A+R_B+R_C) \tag{13}$$

In the equation, R_A, R_B and R_C are recall of three classes of data: A, B, and C, respectively.

Scenario 2: The dataset is an imbalanced dataset.

Assuming that Class A represents large sample data, and Class B and Class C represent small sample data, then $a_T \gg b_T$, $a_T \gg c_T$. The accuracy of the model can be expressed as:

$$S = \frac{a+b+c}{a_T+b_T+c_T} = \frac{a+b+c}{a_T} = \frac{a}{a_T}+\frac{b}{a_T}+\frac{c}{a_T} = R_A + m \tag{14}$$

In the equation, $m = \frac{b}{a_T} + \frac{c}{a_T}$; because $b_T \geq b$ and $c_T \geq c$, it can be concluded that $a_T \gg b$ and $a_T \gg c$, so m is a very small number.

From Equations (13) and (14), it can be seen that the changes of R_A, R_B, and R_C have equal influence on S when facing a balanced dataset. This shows that GA will not focus on improving the detection ability of a certain kind of data in the optimization process. In the face of imbalanced datasets, the change of R_A directly determines the size of S. When the model classifies unknown data, it tends to predict Class B and Class C as Class A, which leads to a serious decline in recall of these two classes of data. This shows that there are serious defects in using accuracy as the fitness function of GA when facing imbalanced datasets.

Intrusion detection dataset is a typical imbalanced dataset. In order to overcome the defects of GA in optimizing intrusion detection model, we design a novel weighted fitness function based on the evaluation index of recall. Let a certain intrusion detection dataset have E classes of data in total. The recall of a certain class of data is R_j (j represents the data type), and its corresponding weight is h_j. Therefore, the weighted fitness function F can be expressed as:

$$F = \sum_{j=1}^{E} R_j h_j \tag{15}$$

In Equation (15), by setting a higher weight for small sample data and a lower weight for large sample data, the GA is guided to evolve more appropriate hyperparameters. The optimized model can detect both small sample data and large sample data with excellent performance. Compared with Equation (14), the contribution of small sample data to fitness values is reflected, and GA will inevitably tilt towards small sample data during the optimization process. Due to the scarcity of small sample data, GA needs to consume more computing resources to improve the corresponding recall. GA instinctively searches for more resource-saving evolutionary direction during the process of optimization. In order to make GA violate its "instinct", we must artificially control the evolution direction of GA by adjusting the weight h. Therefore, designing a reasonable weight h is crucial for the evolution of GA. The definition of weight h is as follows:

$$h_i = \begin{cases} \left\lfloor \left(E \times 2^{\frac{\mathrm{num(min)}}{\mathrm{num}(j)}} \right) \right\rfloor & , j \text{ is small sample data} \\ 1 & , j \text{ is large sample data} \end{cases} \tag{16}$$

where num(min) is the minimum number of types among the E classes of data. num(j) is the number of Class j data. $\lfloor \cdot \rfloor$ represents a downward rounding operation.

The weight of large sample data is uniformly assigned to 1, which is a lower weight value. The purpose is to passivate the sensitivity of GA to large sample data. For small sample data, two factors should be considered, one is the number of data types E, and the other is the number of Class j data num(j). E and h are proportional because the larger the

E, the smaller the proportion of small sample data, and the easier it is to be "sacrificed" in the optimization process of GA. Therefore, the larger the value of E, the larger h should be set. num(j) is inversely proportional to h. The smaller the value of num(j), the easier it is for GA to ignore Class j data. Therefore, in order to improve the detection rate of Class j data, it is necessary to increase the value of h. Using an index based on 2 to appropriately reduce num (j) does not change the correlation between num (j) and h, but compresses the scale of the variable. GA can more sensitively perceive the subtle changes of fitness function and find a more suitable direction for the subsequent evolution.

3.5. *SCV-GA-WELM*

The flow chart of SCV-GA-WELM is shown in Figure 2, and the N-S flow chart of SCV-GA-WELM is shown in Figure A1. The specific workflow is as follows:

(1) Input the training set into the model;
(2) According to the requirements of SCV, the training set is divided into K-fold subsets Z_i, $i = 1, 2, \ldots,$ K;
(3) Use GA to update the weight ω and bias b of WELM;
(4) Each fold subset Z_i trains one WELM, and the fitness function values of K WELM are taken as the arithmetic average $\overline{F}$;
(5) Compare the best fitness value F_{best} with $\overline{F}$. If $F_{best} > \overline{F}$, set $F_{best} = \overline{F}$, and record the best ω and b corresponding to F_{best};
(6) If the iteration number n is not greater than the generations of GA, repeat Step (3);
(7) Substitute the best ω and b corresponding to F_{best} into WELM, and conduct experiments on the test set.

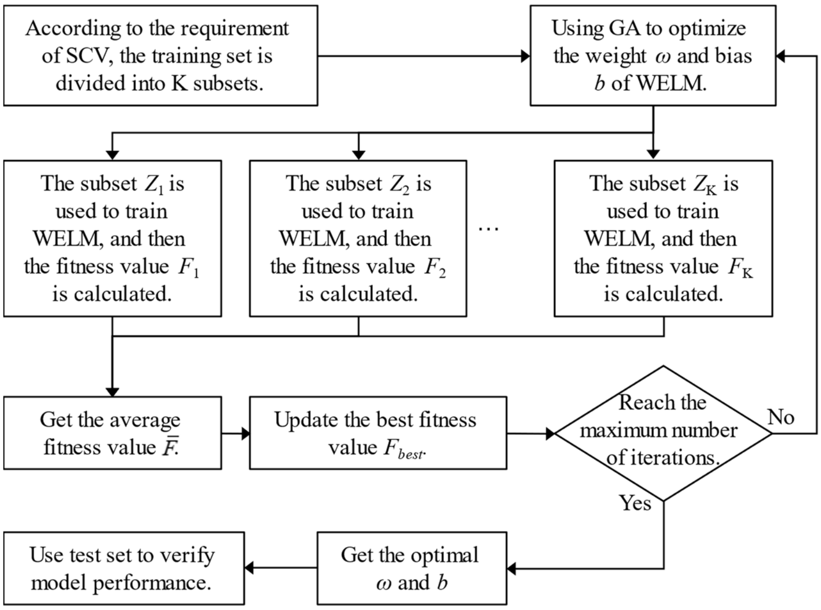

According to the requirement of SCV, the training set is divided into K subsets.
Using GA to optimize the weight ω and bias b of WELM.
The subset Z_1 is used to train WELM, and then the fitness value F_1 is calculated.
The subset Z_2 is used to train WELM, and then the fitness value F_2 is calculated.
…
The subset Z_K is used to train WELM, and then the fitness value F_K is calculated.
Get the average fitness value $\overline{F}$.
Update the best fitness value F_{best}.
Reach the maximum number of iterations.
No
Yes
Use test set to verify model performance.
Get the optimal ω and b

Figure 2. Flow chart of the model in this paper.

4. Results and Analysis

4.1. *Experimental Preparation*

The operating system of the selected server is CentOS 7, the CPU processor is 64 × AMD 7452@2.35 GHz, and the memory is 256 G. The simulation software is MATLAB R2022a. In this paper, the NSL-KDD dataset is selected as the experimental dataset, and KDDTrain+ and KDDTest+ in the NSL-KDD data packet are selected as the training set and test set, respectively. The dataset has 42 dimensions of data, with the first 41 dimensions being dataset features and the 42nd dimension being dataset labels [38]. Labels include normal behaviors and 39 types of attacks, among which the 39 types of attacks belong to four types of attack behaviors: DoS, Probe, U2R and R2L, respectively. Mark the five classes of labels as 1–5, respectively. The training set includes 21 types of attacks, while 18 types of attacks that are not in the training set appear in the test set. These attack types, which only appear in the test set, can be used to evaluate the detection ability of the intrusion detection model in this paper to unknown attacks.

4.2. Comparison of Experimental Settings and Results

Five models, ELM, WELM, GA-WELM, SCV-GA-WELM(I), and SCV-GA-WELM(II), were selected for comparative experiments. Random generation of hyperparameters is used for ELM and WELM. Choose Equation (1) as the fitness function of GA-WELM and SCV-GA-WELM(I). Choose Equation (15) as the fitness function of SCV-GA-WELM(II), and the weight h_j of five classes of data is (1,1,1,10,5). The experimental results are presented in the form of confusion matrix, which is shown in Figure 3. In Figure 3, the diagonal of the confusion matrix is the correct number of classification. Below the confusion matrix is precision, and on the right is recall. The comparison of accuracy, recall, and precision of different models is shown in Tables 1–3, respectively.

Table 1. Accuracy comparison of different models.

Model	Accuracy
ELM	76.11%
WELM	77.94%
GA-WELM	95.79%
SCV-GA-WELM(I)	**96.41%**
SCV-GA-WELM(II)	96.36%

Note: Bold is the maximum value.

Table 2. Recall comparison of different models.

Model	Recall				
	1	2	3	4	5
ELM	97.5%	81.8%	62.7%	4.0%	2.2%
WELM	93.8%	73.7%	81.3%	25.5%	34.4%
GA-WELM	98.0%	97.1%	96.2%	80.0%	85.4%
SCV-GA-WELM(I)	**98.1%**	**97.3%**	**96.5%**	85.0%	89.0%
SCV-GA-WELM(II)	97.0%	96.1%	96.3%	**91.5%**	**95.1%**

Note: Bold is the maximum value.

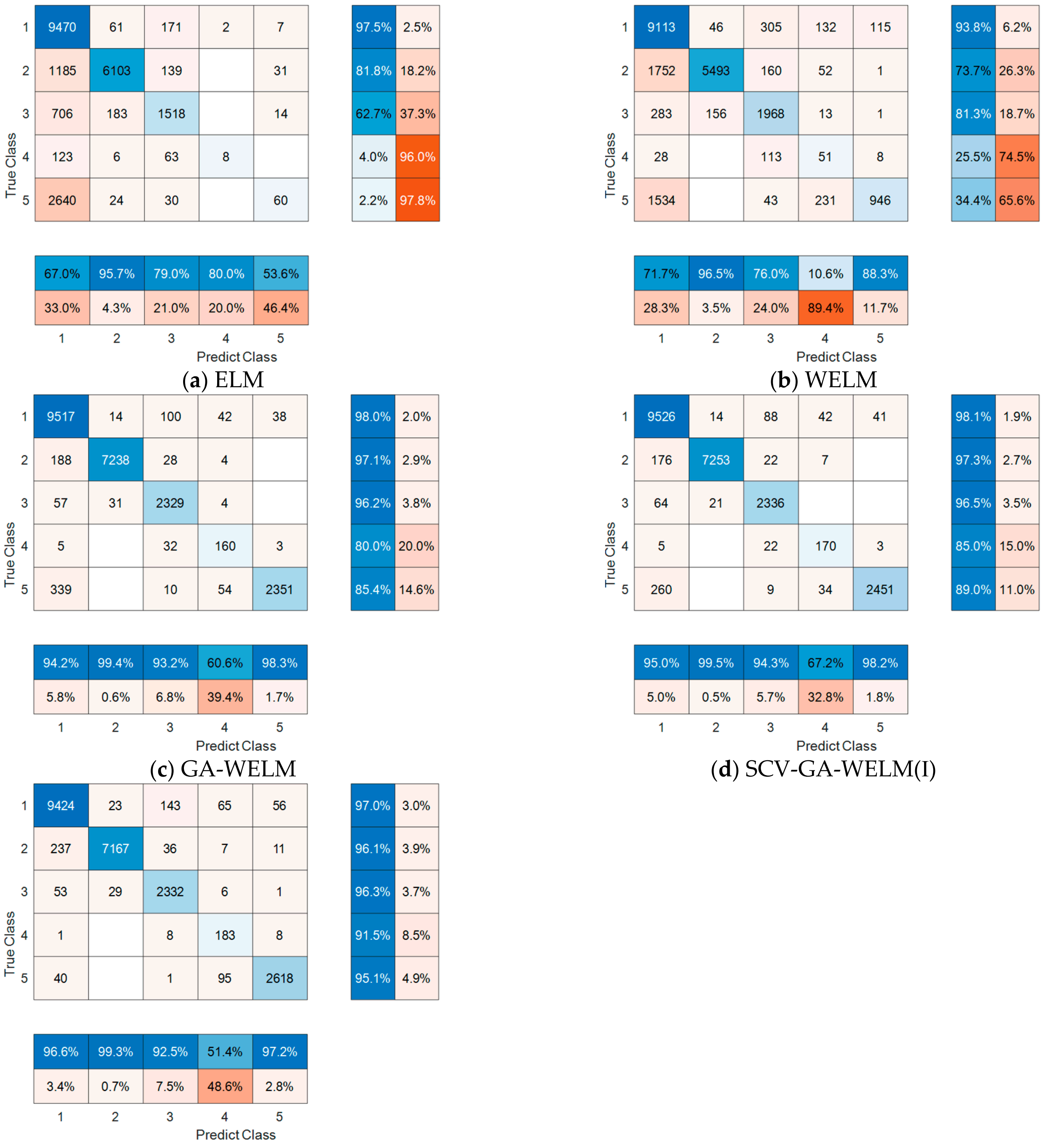
Table 3. Precision comparison of different models.

Model	Precision				
	1	2	3	4	5
ELM	67.0%	95.7%	79.0%	**80.0%**	53.6%
WELM	71.7%	96.5%	76.0%	10.6%	88.3%
GA-WELM	94.2%	99.4%	93.2%	60.6%	**98.3%**
SCV-GA-WELM(I)	95.0%	**99.5%**	**94.3%**	67.2%	98.2%
SCV-GA-WELM(II)	**96.6%**	99.3%	92.5%	51.4%	97.2%

Note: Bold is the maximum value.

4.3. Results Analysis

Comparing ELM with WELM, it can be seen from Figure 3a,b, Tables 2 and 3 that the recall of WELM in Class 4 and Class 5 data has increased by 21.5% and 32.2%, respectively, and the correct number of classification has increased by 43 and 886. This is because Class 4 and Class 5 data are small sample data, and WELM will give them more attention in the process of classification, so the corresponding recall and the correct number of classification have been significantly improved. However, this promotion is based on the premise on sacrificing the recall for large sample data, resulting in a decrease in recall for Class 1 and Class 2 data.



True Class	1	2	3	4	5		
1	9470	61	171	2	7	97.5%	2.5%
2	1185	6103	139		31	81.8%	18.2%
3	706	183	1518		14	62.7%	37.3%
4	123	6	63	8		4.0%	96.0%
5	2640	24	30		60	2.2%	97.8%

Predict Class	1	2	3	4	5
	67.0%	95.7%	79.0%	80.0%	53.6%
	33.0%	4.3%	21.0%	20.0%	46.4%

(**a**) ELM

True Class	1	2	3	4	5		
1	9113	46	305	132	115	93.8%	6.2%
2	1752	5493	160	52	1	73.7%	26.3%
3	283	156	1968	13	1	81.3%	18.7%
4	28		113	51	8	25.5%	74.5%
5	1534		43	231	946	34.4%	65.6%

Predict Class	1	2	3	4	5
	71.7%	96.5%	76.0%	10.6%	88.3%
	28.3%	3.5%	24.0%	89.4%	11.7%

(**b**) WELM

True Class	1	2	3	4	5		
1	9517	14	100	42	38	98.0%	2.0%
2	188	7238	28	4		97.1%	2.9%
3	57	31	2329	4		96.2%	3.8%
4	5		32	160	3	80.0%	20.0%
5	339		10	54	2351	85.4%	14.6%

Predict Class	1	2	3	4	5
	94.2%	99.4%	93.2%	60.6%	98.3%
	5.8%	0.6%	6.8%	39.4%	1.7%

(**c**) GA-WELM

True Class	1	2	3	4	5		
1	9526	14	88	42	41	98.1%	1.9%
2	176	7253	22	7		97.3%	2.7%
3	64	21	2336			96.5%	3.5%
4	5		22	170	3	85.0%	15.0%
5	260		9	34	2451	89.0%	11.0%

Predict Class	1	2	3	4	5
	95.0%	99.5%	94.3%	67.2%	98.2%
	5.0%	0.5%	5.7%	32.8%	1.8%

(**d**) SCV-GA-WELM(I)

True Class	1	2	3	4	5		
1	9424	23	143	65	56	97.0%	3.0%
2	237	7167	36	7	11	96.1%	3.9%
3	53	29	2332	6	1	96.3%	3.7%
4	1		8	183	8	91.5%	8.5%
5	40		1	95	2618	95.1%	4.9%

Predict Class	1	2	3	4	5
	96.6%	99.3%	92.5%	51.4%	97.2%
	3.4%	0.7%	7.5%	48.6%	2.8%

(**e**) SCV-GA-WELM(II)

Figure 3. Models' performance comparison. The darker the color, the higher the value; Conversely, the lighter the color, the lower the value.

As large sample data, the recall of Class 3 data has obviously increased, because it is a large sample compared with Class 4 and Class 5 data, but it is small sample data compared with Class 1 and Class 2 data. As can be seen from Table 1, the accuracy of WELM and ELM is basically the same, which cannot reflect the performance improvement of WELM on small sample data, and the evaluation index is distorted. Therefore, it is not appropriate to use accuracy as the main evaluation index and fitness function for imbalanced dataset.

The precision of WELM on Class 4 data decreased obviously, which deviated from the fact that the correct number of classifications increased by 43, so we could not choose precision as the main evaluation index and fitness function.

Comparing WELM and GA-WELM, it can be seen from Figure 3c, Tables 1–3 that GA-WELM is superior to WELM in recall of all data classes, and the precision and accuracy of GA-WELM have also improved, indicating that GA-WELM has higher detection ability than WELM. The reason is that GA optimizes the hyperparameters of WELM. The WELM model optimized by GA can effectively detect small sample data and also has good detection ability for large sample data.

Compared with GA-WELM and SCV-GA-WELM (I), SCV-GA-WELM (I) achieved the best results in recall and accuracy, and achieved the best results in the precision of the first four classes of data. The precision of the Class 5 data is basically equal to GA-WELM, indicating that SCV-GA-WELM (I) has better classification performance. The recall of SCV-GA-WELM(I) is improved by 0.1%,0.2% and 0.3%, respectively, on large sample data, and by 5% and 3.6%, respectively, on small sample data. Compared to large sample data, the improvement in model performance is more significant on small sample data. This is because the use of SCV can improve the detection ability of GA-WELM, especially for small sample data.

Compared with SCV-GA-WELM(I) and SCV-GA-WELM(II), SCV-GA-WELM(II) has a higher recall on the latter two classes of data. SCV-GA-WELM(I) has a higher recall on the first three classes of data. It is also higher on accuracy. This is because SCV-GA-WELM(II) using weighted fitness function is more sensitive to small sample data, so it has achieved good results in small sample data detection. The fitness function used in SCV-GA-WELM(I) is accuracy, so it focuses more on the performance of large sample data and has achieved good results in detecting large sample data. The fitness function used in SCV-GA-WELM(I) is accuracy, so it pays more attention to the performance of large sample data and has achieved good results in large sample data detection. The value of accuracy mainly depends on the performance of the model on large sample data, so the accuracy of SCV-GA-WELM(I) is also the highest. The precision of SCV-GA-WELM(II) for the Class 4 data has dropped sharply, because the weight of the Class 4 data is much higher than that of the other four classes, so many other classes data are misclassified into Class 4 data. SCV-GA-WELM(II) is not the highest precision for Class 5 data, and the same is true. It is proved that the evolutionary direction of GA can be controlled artificially by the weighted fitness function. The accuracy of SCV-GA-WELM(II), and the precision of the Class 1,2,3 and 5 data, have little change compared with SCV-GA-WELM(I). This proves that although SCV-GA-WELM (II) sacrifices large sample data to improve recall of small sample data, it can reasonably balance the relationship between large sample data and small sample data, and does not blindly divide large sample data into small sample data. SCV-GA-WELM(II) performs well on small sample data and also has high detection ability on large sample data.

4.4. McNemar Hypothesis Test Results

To verify the superiority of SCV-GA-WELM, the McNemar hypothesis test was performed using the SCV-GA-WELM as a benchmark model. The McNemar hypothesis test confirmed the statistical significance of differences between the two methods. The McNemar hypothesis test is a nonparametric test on a 2 × 2 confusion matrix, and it is based on the standardized normal test statistic, which can be expressed as follows:

$$z = \frac{f_{12} - f_{21}}{\sqrt{f_{12} + f_{21}}} \tag{17}$$

where f_{ij} denotes the number of occurrences of elements (i, j) in the confusion matrix; the square of z obeys the chi-square distribution of one degree of freedom.

The test equation can be expressed by:

$$\chi^2 = \frac{(f_{12} - f_{21})^2}{f_{12} + f_{21}} \tag{18}$$

Select four existing models [39–42] as comparison models. In the test, a $p < 0.05$ was considered to indicate statistical significance. The McNemar hypothesis test results of the comparison model and the benchmark model SCV-GA-WELM are shown in Table 4. As shown in Table 4, the p values between the SCV-GA-WELM and the other models were less than 0.05, indicating that the McNemar hypothesis was true. Thus, the detection capability of the SCV-GA-WELM was statistically superior to those of the comparison models.

Table 4. The McNemar hypothesis test results.

Model	**SCV-GA-WELM** ***p*-Value**
HDLNIDS [39]	2.74×10^{-4}
HC-DTTSVM [40]	1.36×10^{-7}
MCLDM [41]	6.20×10^{-3}
two-stage LSTM and DNN [42]	6.21×10^{-7}

5. Conclusions

Aiming at the problem of data imbalance in intrusion detection, this paper proposes a SCV-GA-WELM model, which adopts a brand-new weighted fitness function to guide the evolution direction of GA and improve its optimization ability. At the same time, SCV is used to divide a fairer sample dataset, reduce over-fitting, and improve data utilization. The experimental results show that the SCV-GA-WELM model has excellent performance on recall and McNemar hypothesis tests, especially on small sample data. The selection of hyperparameters in GA has a significant impact on the optimization results. We will conduct detailed discussions and research on how to select appropriate hyperparameters in subsequent work. At the same time, we will consider using conditional generation antagonistic network (CGAN) to construct countermeasures samples with small sample data. On the one hand, it can solve the problem of uneven data distribution, and on the other hand, it can improve the security of the detection model itself.

Author Contributions: Conceptualization, C.C. and X.G.; methodology, C.C.; software, W.Z.; validation, W.Z.; formal analysis, C.C.; resources, Y.Z. and D.W.; writing—original draft preparation, B.W. and B.M.; writing—review and editing, C.C.; supervision, C.C.; funding acquisition, X.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

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.

Appendix A. The N-S Flow Chart of SCV-GA-WELM

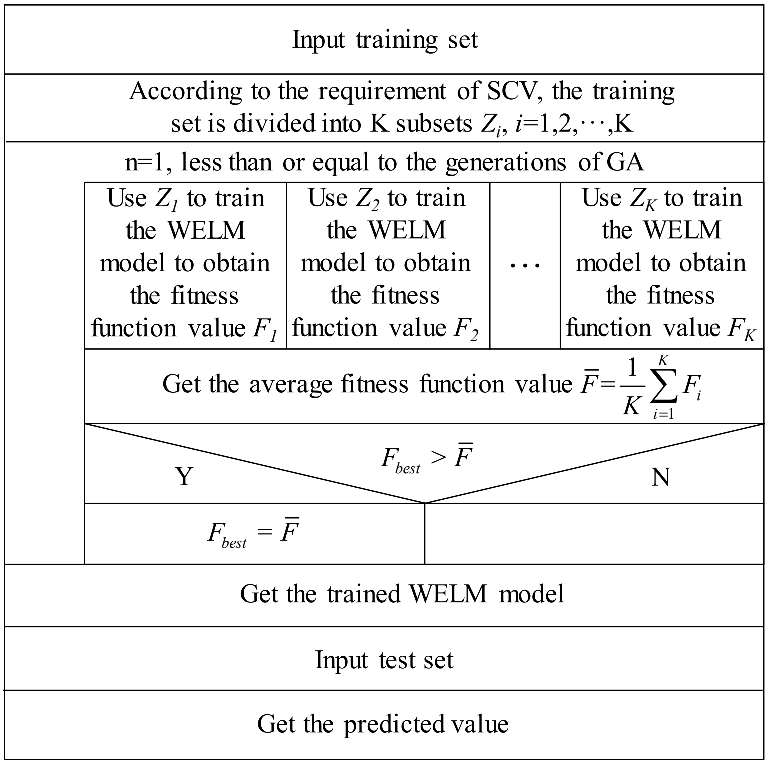

Input training set
According to the requirement of SCV, the training set is divided into K subsets Z_i, i=1,2,⋯,K
n=1, less than or equal to the generations of GA
Use Z_1 to train the WELM model to obtain the fitness function value F_1
Use Z_2 to train the WELM model to obtain the fitness function value F_2
⋯
Use Z_K to train the WELM model to obtain the fitness function value F_K
Get the average fitness function value $\overline{F}=\frac{1}{K}\sum_{i=1}^{K}F_i$
$F_{best} > \overline{F}$
Y
N
$F_{best} = \overline{F}$
Get the trained WELM model
Input test set
Get the predicted value

Figure A1. N-S flow chart of the model in this paper.

References

1. Zhou, G.; Miao, F.; Tang, Z.; Zhou, Y.; Luo, Q. Kohonen neural network and symbiotic-organism search algorithm for intrusion detection of network viruses. *Front. Comput. Neurosci.* **2023**, *17*, 1079483. [CrossRef] [PubMed]
2. Zaib, R.; Zhou, K.-Q. Zero-Day Vulnerabilities: Unveiling the Threat Landscape in Network Security. *Mesopotamian J. CyberSecurity* **2022**, *2022*, 57–64. [CrossRef]
3. Alajanbi, M.; Ismail, M.A.; Hasan, R.A.; Sulaiman, J. Intrusion Detection: A Review. *Mesopotamian J. CyberSecurity* **2021**, *2021*, 1–4.
4. Nassreddine, G.; Younis, J.; Falahi, T. Detecting Data Outliers with Machine Learning. *Al-Salam J. Eng. Technol.* **2023**, *2*, 152–164.
5. Zipperle, M.; Gottwalt, F.; Chang, E.; Dillon, T. Provenance-based Intrusion Detection Systems: A Survey. *ACM Comput. Surv.* **2022**, *55*, 135. [CrossRef]
6. Debicha, I.; Bauwens, R.; Debatty, T.; Dricot, J.-M.; Kenaza, T.; Mees, W. TAD: Transfer learning-based multi-adversarial detection of evasion attacks against network intrusion detection systems. *Future Gener. Comput. Syst.* **2023**, *138*, 185–197. [CrossRef]
7. Anyanwu, G.O.; Nwakanma, C.I.; Lee, J.-M.; Kim, D.-S. RBF-SVM kernel-based model for detecting DDoS attacks in SDN integrated vehicular network. *Ad Hoc Netw.* **2023**, *140*, 9318. [CrossRef]
8. Sheikhi, S.; Kostakos, P. A Novel Anomaly-Based Intrusion Detection Model Using PSOGWO-Optimized BP Neural Network and GA-Based Feature Selection. *Sensors* **2022**, *22*, 9318. [CrossRef]
9. Alzaqebah, A.; Aljarah, I.; Al-Kadi, O. A hierarchical intrusion detection system based on extreme learning machine and nature-inspired optimization. *Comput. Secur.* **2023**, *124*, 102957. [CrossRef]
10. Louk, M.H.L.; Tama, B.A. Dual-IDS: A bagging-based gradient boosting decision tree model for network anomaly intrusion detection system. *Expert Syst. Appl.* **2023**, *213*, 119030. [CrossRef]
11. Chen, C.; Song, Y.; Yue, S.; Xu, X.; Zhou, L.; Lv, Q.; Yang, L. FCNN-SE: An Intrusion Detection Model Based on a Fusion CNN and Stacked Ensemble. *Appl. Sci.* **2022**, *12*, 8601. [CrossRef]
12. Li, X.; Kong, K.; Shen, H.; Wei, Z.; Liao, X. Intrusion detection method based on imbalanced learning classification. *J. Exp. Theor. Artif. Intell.* **2022**, 1–21. [CrossRef]
13. Pimsarn, C.; Boongoen, T.; Iam-On, N.; Naik, N.; Yang, L. Strengthening intrusion detection system for adversarial attacks: Improved handling of imbalance classi-fication problem. *Complex Intell. Syst.* **2022**, *8*, 4863–4880. [CrossRef]
14. Ding, H.; Chen, L.; Dong, L.; Fu, Z.; Cui, X. Imbalanced data classification: A KNN and generative adversarial networks-based hybrid approach for intrusion detection. *Future Gener. Comput. Syst.* **2022**, *131*, 240–254. [CrossRef]
15. Tummalapalli, S.; Kumar, L.; Neti, L.B.M.; Krishna, A. Detection of web service anti-patterns using weighted extreme learning machine. *Comput. Stand. Interfaces* **2022**, *82*, 103621. [CrossRef]

16. Dahiya, M.; Nitin, N.; Dahiya, D. Intelligent Cyber Security Framework Based on SC-AJSO Feature Selection and HT-RLSTM Attack Detection. *Appl. Sci.* **2022**, *12*, 6314. [CrossRef]
17. Chen, C.; Song, Y.; Yue, S.; Xu, X.; Zhou, L.; Lv, Q.; Yang, L. A Network intrusion detection method based on PSOGWO-SVM. *J. Air Force Eng. Univ.* **2022**, *23*, 97–105.
18. Kalita, D.J.; Singh, V.P.; Kumar, V. A novel adaptive optimization framework for SVM hyper-parameters tuning in non-stationary environment: A case study on intrusion detection system. *Expert Syst. Appl.* **2023**, *213*, 119189. [CrossRef]
19. Bin Sarhan, B.; Altwaijry, N. Insider Threat Detection Using Machine Learning Approach. *Appl. Sci.* **2022**, *13*, 259. [CrossRef]
20. Jia, H.; Liu, J.; Zhang, M.; He, X.; Sun, W. Network intrusion detection based on IE-DBN model. *Comput. Commun.* **2021**, *178*, 131–140. [CrossRef]
21. Wang, C.; Sun, Y.; Lv, S.; Wang, C.; Liu, H.; Wang, B. Intrusion Detection System Based on One-Class Support Vector Machine and Gaussian Mixture Model. *Electronics* **2023**, *12*, 930. [CrossRef]
22. Vanitha, S.; Balasubramanie, P. Improved Ant Colony Optimization and Machine Learning Based Ensemble Intrusion Detection Model. *Intell. Autom. Soft Comput.* **2022**, *36*, 849–864. [CrossRef]
23. Edwin Singh, C.; Celestin Vigila, S.M. WOA-DNN for Intelligent Intrusion Detection and Classification in MANET Services. *Intell. Autom. Soft Comput.* **2023**, *35*, 1737–1751. [CrossRef]
24. Huang, G.-B.; Zhu, Q.-Y.; Siew, C.-K. Extreme learning machine: Theory and applications. *Neurocomputing* **2006**, *70*, 489–501. [CrossRef]
25. Ali, H.; Elzeki, O.M.; Elmougy, S. Smart Attacks Learning Machine Advisor System for Protecting Smart Cities from Smart Threats. *Appl. Sci.* **2022**, *12*, 6473. [CrossRef]
26. Al-Yaseen, W.L.; Idrees, A.K.; Almasoudy, F.H. Wrapper feature selection method based differential evolution and extreme learning machine for intrusion detection system. *Pattern Recognit.* **2022**, *132*, 108912. [CrossRef]
27. Lin, H.; Xue, Q.; Feng, J.; Bai, D. Internet of things intrusion detection model and algorithm based on cloud computing and multi-feature ex-traction extreme learning machine. *Digit. Commun. Netw.* **2023**, *9*, 111–124. [CrossRef]
28. Park, C.; Lee, J.; Kim, Y.; Park, J.-G.; Kim, H.; Hong, D. An Enhanced AI-Based Network Intrusion Detection System Using Generative Adversarial Networks. *IEEE Internet Things J.* **2022**, *10*, 2330–2345. [CrossRef]
29. Yan, Y.; Qian, Y.; Ma, H.; Hu, C. Research on imbalanced data fault diagnosis of on-load tap changers based on IGWO-WELM. *Math. Biosci. Eng.* **2023**, *20*, 4877–4895. [CrossRef]
30. Ma, T.; Wu, L.; Zhu, S.; Zhu, H. Multiclassification Prediction of Clay Sensitivity Using Extreme Gradient Boosting Based on Imbalanced Dataset. *Appl. Sci.* **2022**, *12*, 1143. [CrossRef]
31. Ma, W.; Gou, C.; Hou, Y. Research on Adaptive 1DCNN Network Intrusion Detection Technology Based on BSGM Mixed Sampling. *Sensors* **2023**, *23*, 6206. [CrossRef] [PubMed]
32. Du, R.; Li, Y.; Liang, X.; Tian, J. Support Vector Machine Intrusion Detection Scheme Based on Cloud-Fog Collaboration. *Mob. Netw. Appl.* **2022**, *27*, 431–440. [CrossRef]
33. Zivkovic, M.; Tair, M.; Venkatachalam, K.; Bacanin, N.; Hubálovský, Š.; Trojovský, P. Novel hybrid firefly algorithm: An application to enhance XGBoost tuning for intrusion detection classification. *PeerJ Comput. Sci.* **2022**, *8*, e956. [CrossRef]
34. Yamin, M.; Bajaba, S.; AlKubaisy, Z.M. Chaotic Metaheuristics with Multi-Spiking Neural Network Based Cloud Intrusion Detection. *Comput. Mater. Contin.* **2022**, *74*, 6101–6118. [CrossRef]
35. Zong, W.; Huang, G.-B.; Chen, Y. Weighted extreme learning machine for imbalance learning. *Neurocomputing* **2012**, *101*, 229–242. [CrossRef]
36. Szeghalmy, S.; Fazekas, A. A Comparative Study of the Use of Stratified Cross-Validation and Distribution-Balanced Stratified Cross-Validation in Imbalanced Learning. *Sensors* **2023**, *23*, 2333. [CrossRef]
37. Liu, X.; Du, Y. Towards Effective Feature Selection for IoT Botnet Attack Detection Using a Genetic Algorithm. *Electronics* **2023**, *12*, 1260. [CrossRef]
38. Song, J.; Hiroki, T.; Yasuo, O. *Description of Kyoto University Benchmark Data*; Kyoto University: Kyoto, Japan, 2006; Available online: http://www.takakura.com/Kyoto_data/BenchmarkData-Description-v5.pdf (accessed on 15 July 2023).
39. Qazi, E.U.H.; Faheem, M.H.; Zia, T. HDLNIDS: Hybrid Deep-Learning-Based Network Intrusion Detection System. *Appl. Sci.* **2023**, *13*, 4921. [CrossRef]
40. Zou, L.; Luo, X.; Zhang, Y.; Yang, X.; Wang, X. HC-DTTSVM: A Network Intrusion Detection Method Based on Decision Tree Twin Support Vector Machine and Hierarchical Clustering. *IEEE Access* **2023**, *11*, 21404–21416. [CrossRef]
41. Luo, J.; Zhang, Y.; Wu, Y.; Xu, Y.; Guo, X.; Shang, B. A Multi-Channel Contrastive Learning Network Based Intrusion Detection Method. *Electronics* **2023**, *12*, 949. [CrossRef]
42. Han, J.; Wooguil, P. High Performance Network Intrusion Detection System Using Two-Stage LSTM and Incremental Created Hybrid Features. *Electronics* **2023**, *12*, 956. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.