information

MDPI

Article

An Intrusion Detection System Based on a Simplified Residual Network

Yuelei Xiao [1,2,*] and Xing Xiao [1]

[1] Institute of IOT and IT-based Industrialization, Xi'an University of Post and Telecommunications, Xi'an 710061, China; king_xiao139@163.com
[2] Shaanxi Provincial Information Engineering Research Institute, Xi'an 710075, China
* Correspondence: xiaoyuelei@xupt.edu.cn; Tel.: +86-29-8730-3602

Received: 21 October 2019; Accepted: 14 November 2019; Published: 18 November 2019

Abstract: Residual networks (ResNets) are prone to over-fitting for low-dimensional and small-scale datasets. And the existing intrusion detection systems (IDSs) fail to provide better performance, especially for remote-to-local (R2L) and user-to-root (U2R) attacks. To overcome these problems, a simplified residual network (S-ResNet) is proposed in this paper, which consists of several cascaded, simplified residual blocks. Compared with the original residual block, the simplified residual block deletes a weight layer and two batch normalization (BN) layers, adds a pooling layer, and replaces the rectified linear unit (ReLU) function with the parametric rectified linear unit (PReLU) function. Based on the S-ResNet, a novel IDS was proposed in this paper, which includes a data preprocessing module, a random oversampling module, a S-Resnet layer, a full connection layer and a Softmax layer. The experimental results on the NSL-KDD dataset show that the IDS based on the S-ResNet has a higher accuracy, recall and F1-score than the equal scale ResNet-based IDS, especially for R2L and U2R attacks. And the former has faster convergence velocity than the latter. It proves that the S-ResNet reduces the complexity of the network and effectively prevents over-fitting; thus, it is more suitable for low-dimensional and small-scale datasets than ResNet. Furthermore, the experimental results on the NSL-KDD datasets also show that the IDS based on the S-ResNet achieves better performance in terms of accuracy and recall compared to the existing IDSs, especially for R2L and U2R attacks.

Keywords: intrusion detection system; simplified residual network; simplified residual block; random oversampling; full connection; over-fitting

1. Introduction

With the rapid development of computer networks, network security issues have become increasingly prominent, resulting in huge economic losses. At present, the most widely used network security systems are firewalls, intrusion detection systems (IDSs) and intrusion prevention systems (IPSs). Among them, the IDS is a proactive security protection technology. It collects information from computer network system and analyses it to find out whether there are any violations of security policies and signs of being attacked in the network system. As an effective complement to the firewall, IDS is usually installed behind the firewall to detect intrusions and illegal activities of attack users. IPS is a kind of network security system which is different from a firewall and IDS. It can implement active defense and real-time blocking to intrusion activities and attacks, which greatly improves the security of the network system.

The traditional IDSs mainly check attacks based on feature detection. This method has many shortcomings, such as lack of accuracy of its feature library, lack of ability to automatically update features, etc. With the rapid development of network technology, it is very difficult to automatically

extract intrusive and normal behavior by manual analysis alone. Therefore, there is a trend to apply intelligent data analysis technology to IDS. In recent years, with the development of machine learning and deep learning, they are also gradually used in IDS [1], including various IDSs based on feature selection [2–14], IDSs based on machine learning [15–26], IDSs based on deep learning [27–32] and IDSs based on hybrid model [33–35]. However, these IDSs fail to provide better performance, especially for remote-to-local (R2L) and user-to-root (U2R) attacks.

Recently, residual networks (ResNets) have received quite a bit of attention at IT conventions [36], and they are being considered for helping with the training of deep networks. They can help to preserve good results by using residual blocks in deep learning networks. However, a ResNet is prone to over-fitting for low-dimensional and small-scale datasets, such as datasets for IDS. To solve these problems, we put forward a simplified residual network (S-ResNet), which consists of several cascaded, simplified residual blocks. Then, we present a novel IDS based on the S-ResNet. The experimental results indicate that our proposed IDS based on the S-ResNet has a higher accuracy, recall and F1-score, and faster convergence velocity than the equal-scale, ResNet-based IDS. This means that the S-ResNet reduces the complexity of the network and effectively prevents over-fitting; thus, it is more suitable for low-dimensional and small-scale datasets than ResNet. And the IDS based on the S-ResNet has higher accuracy and recall than the other IDSs, especially for R2L and U2R attacks. Hence, our proposed IDS provides better performance than the existing IDSs.

The remainder of this article is organized as follows. Section 2 provides an overview of the existing IDSs based on machine learning and deep learning. In Section 3, we provide the rationale of our proposed S-ResNet. Section 4 gives a novel IDS based on the S-ResNet. In Section 5, we demonstrate the experimental details and results, and discuss them. Finally, we draw some conclusions in Section 6.

2. Related Works

Pertaining to IDSs based on feature selection, Ambusaidi et al. [2] proposed two feature selection algorithms, i.e., flexible mutual information based feature selection (FMIFS) and flexible linear correlation coefficient based feature selection (FLCFS), which were compared with a mutual information based feature selection (MIFS) algorithm, and 18, 22 and 23 features were selected respectively. Finally, the classification was performed by combining it with a support vector machine (SVM) algorithm. Ghazy et al. [3] and Aljawarneh et al. [4] proposed some feature selection methods based on the correlation feature selection (CFS), information entropy and wrapper. Kang et al. [5] proposed a feature selection algorithm based on local search and meta-heuristic. Firstly, K-means clustering algorithm was applied to the training set, and the accuracy obtained was used as a cost function to select the optimal feature subset. Secondly, 25 features were screened out by the proposed algorithm. Finally, the multi-layer perception (MLP) model was applied to the optimal subset. Salo et al. [6] proposed a hybrid dimension reduction method based on information gain and principal component analysis (PCA), which combines SVM, instance-based k-nearest neighbors (IBK) and multi-layer perceptron (MLP) algorithms. They reported an accuracy of 98.24% on the NSL-KDD dataset [7]. Beulah et al. [8] proposed an improved hybrid feature selection (IHFS) method, which combines four feature selection methods; i.e., CFS, gain ratio (GR), one rule (OneR) and symmetrical uncertainty (SU). Finally, six attributes (i.e., service, flag, src_bytes, dst_bytes, logged_in and srv_serror_rate) were selected as the optimal subset, and a Bayesian network was combined with logistic regression, nearest neighbor, NBTree and SVM classifiers. They reported an accuracy of 79.6% for the NSL-KDD dataset. Bostani et al. [9] proposed a feature selection method based on binary gravitational search algorithm (BGSA) and mutual information (MI). Then, five features (i.e., service, flag, src_bytes, dst_bytes and serror_rate) were selected, and an SVM classifier was used. They reported an accuracy of 88.362% for the NSL-KDD dataset. Acharya et al. [10] proposed a natural heuristic optimization algorithm; i.e., intelligent water drops (IWD). Then, nine features were selected and an SVM classifier was used. They reported an accuracy of 99.0915% on the KDD CUP'99 dataset [11]. Akashdeep et al. [12] combined information gain with correlation to sort the features, and removed redundant information. Then,

25 features were selected, and classified by an artificial neural network (ANN) model. Akyol et al. [13] proposed a feature selection method of discernibility function based feature selection (DFBFS), and then, the MLP and C4.5 algorithms were applied. They reported an accuracy of 98.03% for the KDD CUP'99 dataset. Bhattacharya et al. [14] proposed a layered wrapper feature selection approach. Finally, 16 features were selected, and naive Bayesian, SVM, k-nearest neighbor (kNN) and AdaBoost methods were applied. The accuracy of the naive Bayesian was 83.14% for the NSL-KDD dataset.

In respect to IDSs based on machine learning, Panda et al. [15] constructed a new hybrid intelligent system by combining a naive Bayesian with a decision tree and a rule-based classifier based on non-nested generalized samples, and extended repeated incremental pruning. Ahmad et al. [16] compared the three algorithms of SVM, random forest and extreme learning, and the experimental results show that the accuracy of extreme learning for intrusion detection is better than the other two methods. Aburomman et al. [17] introduced the integration methods of bagging, boosting, AdaBoost, stacking and other basic classifiers. Preecha et al. [18] applied PCA to reduce dimensionality, and then used a simplified fuzzy adaptive resonance theory map (SFAM) to classify, improving the detection rate of R2L. Alabdallah et al. [19] combined with layered sampling, a loss function and a weighted support vector machine (WSVM), achieving an accuracy of 98.31% for the NSL-KDD dataset. And the detection rates of U2R and R2L were improved. Li et al. [20] combined the binary aggregation module with the kNN algorithm, achieving an anomaly detection rate of 91.35% for the NSL-KDD dataset. And the detection rates of U2R and R2L are higher than those of other methods. Demir et al. [21] improved the stacking model by using the logical regression, decision tree and naive Bayesian methods as base classifiers, and designed 13 groups of experiments. They reported an accuracy of 92.55% on the KDD CUP'99 dataset. The naive Bayesian method as a combiner can detect U2R and R2L attacks very well. Kamarudin et al. [22] proposed a Logitboost ensemble algorithm, and selected 10 features by using hybrid feature selection (HFS), and the random forest method was used as a combination of weak classifiers. They reported an accuracy of 99.1% on the NSL-KDD dataset. Tian et al. [23] proposed a robust and sparse ramp loss function to the original one-class SVM (Ramp-OCSVM) method. The non-differentiable non-convex optimization problem of the obtained model was solved by using a concave–convex process. Kabir et al. [24] proposed an optimum allocation-based least square support vector machine (OA-LS-SVM) method. By using the least square support vector machine (LS-SVM) method, different attack categories and normal categories were grouped into subsets, and then the proposed model was applied to each subset. Ahmim et al. [25] proposed an IDS based on the combination of the probability predictions of a tree of classifiers—a two-layer model. The first layer is a classification tree, and the second layer is a classifier, which combines the probability prediction of the tree. They reported an accuracy of 89.75% on the NSL-KDD dataset. Aburomman et al. [26] proposed a weighted one-against-rest SVM (WOAR-SVM) method based on SVM, and applied a differential evolution (DE) optimization algorithm to model selection. They reported an accuracy of 80.65% for the NSL-KDD dataset.

In the aspect of IDSs based on deep learning, Yan et al. [27] proposed a local adaptive gated recurrent unit (LA-GRU) model, which processes unbalanced data using local adaptive synthetic minority oversampling technology (LA-SMOTE), and then classifies based on the gated recurrent unit (GRU) networks. Idhammad et al. [28] proposed a sequential inertial semi-supervised machine learning method based on network entropy estimation, collaborative clustering, information gain ratio and extra-trees algorithm. They reported an accuracy of 98.23% on the NSL-KDD dataset. Mohammadi et al. [29] used a deep auto-encoder for feature coding, and then applied a linear memory classifier to the NSL-KDD dataset, achieving a detection rate of 98.11%. Imamverdiyev et al. [30] proposed an improved Gaussian–Bernoulli type restricted Boltzmann machine (RBM) method, and compared with the Bernoulli–Bernoulli RBM, Gaussian–Bernoulli RBM and deep belief network (DBN) methods. They reported an accuracy of 73.23% for the NSL-KDD dataset. Ma et al. [31] proposed a spectral clustering and deep neural network (SCDNN) model. In this model, a spectral clustering algorithm is applied, and then a subset of clustering is put into the deep neural network (DNN) method.

They reported an accuracy of 92.1% on the NSL-KDD dataset. The model was compared with the back propagation (BP) network, SVM, random forest and Bayes tree models. Shamshirband et al. [32] proposed a cooperative fuzzy Q-learning (Co-FQL) method, which was compared with the fuzzy logic controller, Q-learning and fuzzy Q-learning methods. They reported an accuracy 89.68% on the NSL-KDD dataset.

Regarding IDSs based on a hybrid model, Al-Qatf et al. [33] proposed a self-taught learning intrusion detection system (STL-IDS). In the STL-IDS, new features are constructed by sparse self-coding, and then classified by the J48, naive Bayes, random forest and SVM methods. The experimental results show that the new features accelerate the training of SVM. They reported an accuracy of 99.396% for the NSL-KDD dataset. Hussain et al. [34] combined the advantages of SVM and BP network. In the first stage, SVM is used to classify normal and abnormal. And in the second stage, a BP network is used to identify attack categories in abnormal. Li et al. [35] proposed a model combining a Gini index and gradient boosting decision tree (GBDT) with particle swarm optimization (PSO). The optimal feature subset is selected by the Gini index, and the network attack is detected by a gradient lifting decision tree algorithm. The parameters of GBDT are optimized by the PSO algorithm. They reported an accuracy of 86.10% on the NSL-KDD dataset.

In summary, the above IDSs are mainly innovations in the aspects of feature selection algorithms and classification prediction algorithms. The classification prediction algorithms used in these IDSs are divided into machine learning models, deep learning models and hybrid models. Currently, the deep learning model has become more attractive and effective than the other models in the IDS field. However, the traditional deep learning networks have the problems of the disappearance of the gradient of the network and the difficulty of training network. To avoid these problems, ResNet was proposed at recent IT conventions, which is more suitable for high-dimensional image data than for low-dimensional and small-scale datasets (e.g., datasets for IDS). In order to keep all the features, we do not use any feature selection algorithm in this paper. But for low-dimensional and small-scale datasets (e.g., datasets for IDS), we propose a S-ResNet based on ResNet, and put forward a novel IDS based on the S-ResNet.

3. Simplified Residual Network

ResNet comes from Microsoft Research [36], an artificial intelligence team of Microsoft. It is the winner of the image classification and object recognition algorithms of the Image Net Large Scale Visual Recognition Competition (ILSVRC) in 2015. It outperforms the third version of GoogLeNet (i.e., Inception v3) [37]. ResNet is a large-scale, convolutional neural network constructed by residual blocks, which is 20 times larger than AlexNet [38] and eight times larger than VGG-16 [39]. Because of this residual effect, the depth of the network can be deeper than that of the traditional networks, which can effectively avoid the disappearance of the gradient of the deep network and the difficulty of training. As the number of layers grows, the performance of ResNet does not deteriorate, but improves to a certain extent. The structure of residual block is shown in Figure 1.

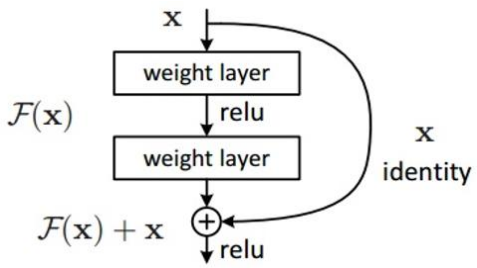

Figure 1. The structure of residual block.

In Figure 1, $\boldsymbol{X}$ denotes the input of a residual block; $F(\boldsymbol{X})$ denotes the output of the residual block before the second activation function. That is to say, $F(\boldsymbol{X}) = \boldsymbol{W}_2\sigma(\boldsymbol{W}_1\boldsymbol{X})$, where $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$ denote the

weights of the first and second layers, σ denotes the rectified linear unit (ReLU) activation function [40] and the output of the residual block is $\sigma(F(\boldsymbol{X}) + \boldsymbol{X})$.

In [41], some variants of the residual block were proposed, as shown in Figure 2.

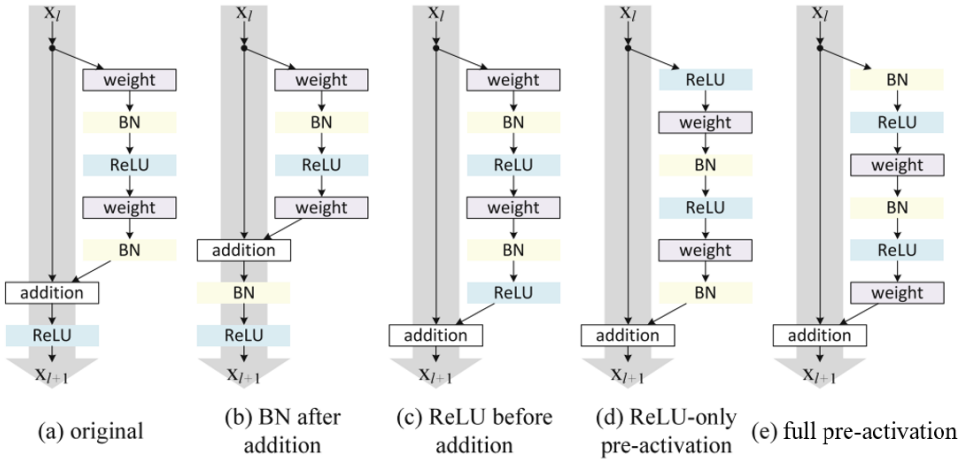

Figure 2. Some variants of the residual block.

In Figure 2, the original residual block is shown, and some variants of the original residual block are shown in Figure 2b–e, which mainly adjusts the order of the components of the original residual block.

ResNet mainly uses residuals to reduce the over-fitting of the model, so that the depth of the network can be greater. And ResNet is more suitable for high-dimensional image data. However, ResNet is prone to over-fitting for low-dimensional and small-scale datasets (e.g., datasets for IDS), resulting in reduced generalization ability of the model.

To solve this problem, this paper proposes a S-ResNet, which is composed of a cascade of simplified residual blocks. Each simplified residual block is mainly composed of a weight layer, a PReLU layer and a pooling layer, as shown in Figure 3.

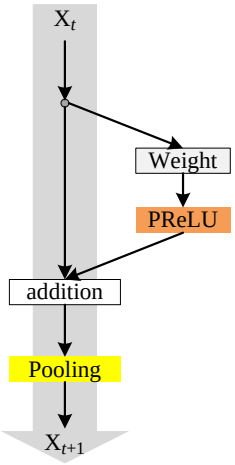

Figure 3. Simplified residual block.

In Figure 3, the input $\boldsymbol{X}_t$ passes through the weight layer; then through the PReLU activation layer; then superimposes with the initial input $\boldsymbol{X}_t$; and finally, passes through the pooling layer as the output of the residual block.

Compared with the original residual block, the simplified residual block deletes one weight layer and two batch normalization (BN) layers; thus, reducing the parameters and complexity of the residual

block, but it retained one weight layer and still uses residual to reduce over-fitting. And the original ReLU activation function is replaced by the parametric rectified linear unit (PReLU) function [40]. Compared with the ReLU function, the PReLU function converges faster and prevents over-fitting. The simplified residual blocks also add a layer of pooling after additions for downsampling, keeping the invariance of the original image and retaining the main features.

From the above description, the simplified residual block can prevent over-fitting and improve the generalization ability of the model, while reducing the parameters and quantity of calculations. As a whole, the simplified residual block is a simplification and optimization of the original residual block for low-dimensional and small-scale datasets. Correspondingly, S-ResNet, which consists of a cascade of simplified residual blocks, can also reduce the complexity of the network, and can effectively prevent over-fitting. Therefore, the S-ResNet can be applied to low-dimensional and small-scale datasets; e.g., datasets for IDS, covering a wider scientific area.

4. Our Proposed IDS

Based on the above S-ResNet, a novel IDS is proposed in this paper, as shown in Figure 4.

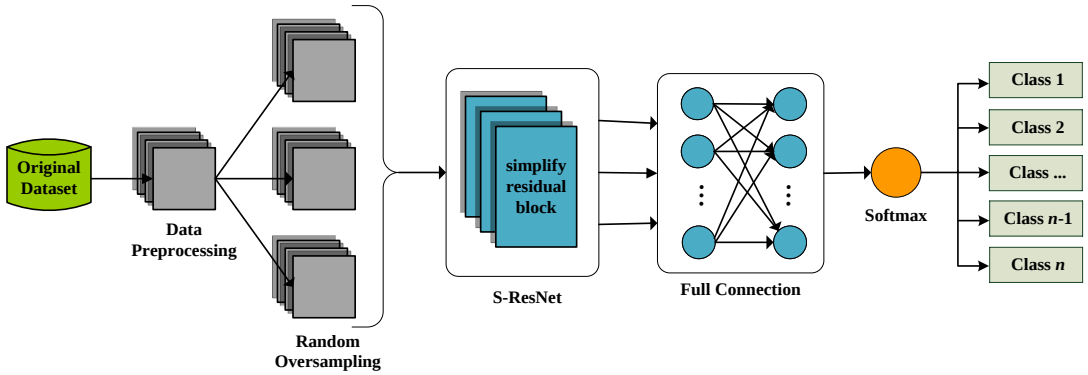

Figure 4. The intrusion detection system (IDS) based on the simplified residual network (S-ResNet).

In this IDS, the original dataset is pre-processed and converted into a series of single channel images (i.e., a series of black and white images). Then, random oversampling is used to balance the number of samples of each category. Finally, the data is run through a S-ResNet layer and a full connection layer, and the prediction probability of the corresponding category is converted through a Softmax layer.

The application scenario of the above IDS is illustrated in Figure 5, including the training phase and the testing phase.

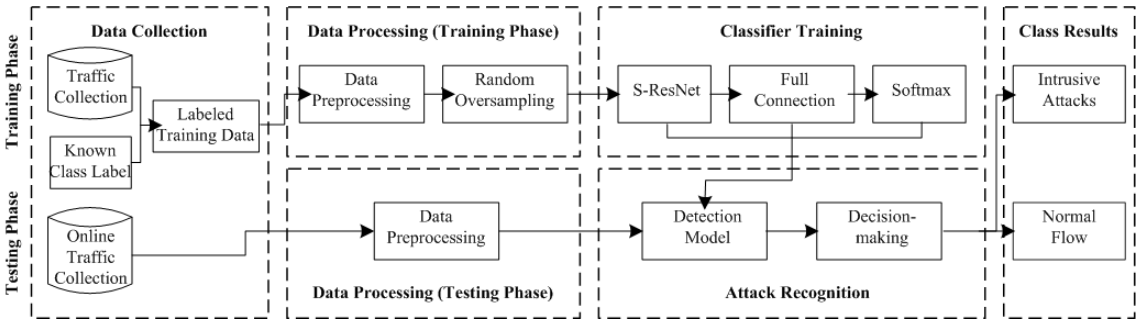

Figure 5. The application scenario of the above IDS.

4.1. Data Preprocessing

The data preprocessing in Figure 4 mainly includes data coding, data standardization and data conversion. One-hot coding, also known as unique hot coding, is used to encode N states with N-bit registers. Each state has its own register bit, and only one is valid at any time. Its function is to convert non-numerical variables into computational numeric forms, and to expand features to generate larger sparse matrices.

Data normalization is to scale the data to a small specific interval, which is used to remove the unit limitations of data and transform it into dimensionless pure values. It is convenient for different units or scales of indicators to compare and weigh, improving the convergence speed of the model and preventing the gradient explosion of the model. The data normalization here adopts Z-score standardization, and its mathematical formula is as follows.

$$z_i^k = \frac{x_i^k - \mu^k}{\sigma^k}, \tag{1}$$

where z_i^k denotes the normalized variable value of attribute k of data sample I; x_i^k denotes the actual variable value of attribute k of data sample i; μ^k denotes the mean value of attribute k of all data samples; and σ^k denotes the standard deviation of attribute k of all data samples.

In this paper, data conversion is mainly through adding the multi-dimensional data of 0 to form n^2-dimensional data. Then, the n^2-dimensional data is converted into a series of $n \times n$ single channel images (i.e., a series of $n \times n$ black and white images). That way, not only is the original information of the data retained, but also the amount of redundant information is not increased.

4.2. Random Oversampling

Resampling technology is a popular method to deal with unbalanced data. It mainly uses a sampling method to improve the imbalance of the sample number of different categories in the dataset. Resampling technology is mainly divided into: oversampling, undersampling and combinations of them. The most commonly used are random undersampling and random oversampling. Random undersampling is to delete most samples from training data randomly, while random oversampling is to copy a few samples from training data randomly.

Random undersampling may lead to loss of information and cannot make full use of the information of the original data, resulting in inadequate training and low accuracy of the model. Other methods of generating new samples, such as synthetic minority oversampling technology (SMOTE) and synthetic minority oversampling technology nominal continuous (SMOTENC) [42], add new information, but if multiple categories are mixed together, the labels of the newly generated samples may not be correct. Consequently, the categories of samples of the testing dataset may be misjudged, reducing the accuracy of the model.

Random oversampling is only repeated sampling, so it can ensure that the samples of the training dataset are correct. Therefore, random oversampling will be used to alleviate the imbalance of the number of samples in the dataset.

4.3. S-ResNet Layer

The S-ResNet layer includes a S-ResNet, which is composed of cascaded simplified residual blocks, as shown in Figure 6.

In Figure 6, the number of simplified residual blocks of the S-ResNet will be set according to the size and dimension of the input dataset. Usually, the smaller the size and dimension of the input dataset, the less the number of simplified residual blocks of the S-ResNet. In the S-ResNet, input datasets are abstracted and simplified layer by layer.

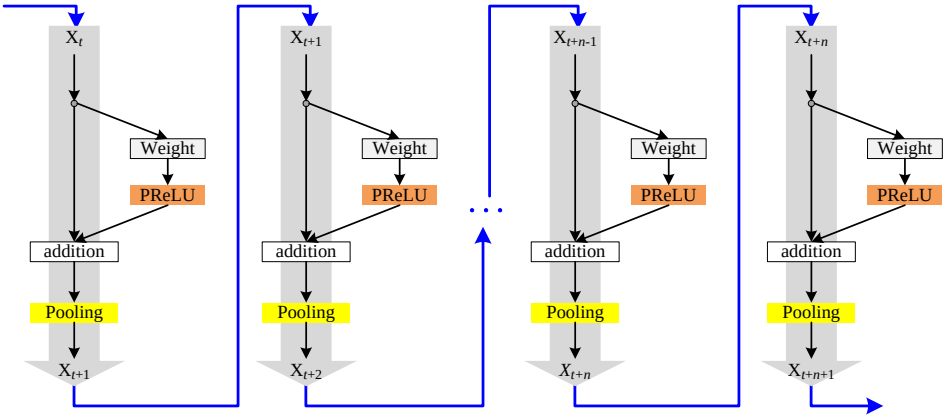

Figure 6. The S-ResNet layer.

4.4. Dense Layer

The dense layer mainly classifies feature vectors and maps the samples from the feature space to the labels. It consists of two parts: The linear part and the nonlinear part. The linear part mainly performs linear transformation and analyzes the input data; the calculation method of this part is linear weighted sum, whose mathematical expression is as follows.

$$\boldsymbol{Z} = \boldsymbol{W} \times \boldsymbol{X} + \boldsymbol{b}, \tag{2}$$

where $\boldsymbol{Z}$ is the output vector of the linear part. It can be expressed as $\boldsymbol{Z} = [\boldsymbol{Z}_1, \boldsymbol{Z}_2, \cdots, \boldsymbol{Z}_m]^T$. $\boldsymbol{X}$ represents the input vector of the linear part, which is expressed as $\boldsymbol{X} = [\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_n]^T$. $\boldsymbol{W}$ is a weight vector matrix of $m \times n$. $\boldsymbol{b}$ is a bias vector, which is expressed as $\boldsymbol{b} = [\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_m]^T$.

The nonlinear part performs nonlinear transformation; namely, corresponding function transformation. This operation has two functions as follows. (1) Data normalization. That is to say, no matter what the previous linear part does, all the values of the nonlinear part will be limited to a certain range. (2) Breaking the linear mapping relation early. In other words, if the dense layer has no non-linear part, it is meaningless to add multiple neural networks in the model.

4.5. Softmax Layer

The Softmax layer is a Softmax activation function. It maps several scalars into a probability distribution, and the value range of its output is from 0 to 1. Its mathematical formula is as follows.

$$\sigma(\boldsymbol{z})_j = \frac{e^{\boldsymbol{z}_j}}{\sum_{k=1}^{K} e^{\boldsymbol{z}_k}}, \tag{3}$$

where $\boldsymbol{z}$ denotes a K-dimensional vector. $\boldsymbol{z}_j$ and $\boldsymbol{z}_k$ denote the elements of the K-dimensional vector. $\sigma(\boldsymbol{z})$ denotes the K-dimensional vector after mapping. j and k denote the subscripts of K-dimensional vector, and $j, k = 1, 2, \ldots, K$.

5. Experiments and Results Analysis

5.1. Experimental Environment and Dataset

The experimental environment of this paper is a Sugon computer A6320r: the processor was a 16× AMD Opteron (tm) Processor 6320, the memory was 65,949 MB, the operating system was Ubuntu 16.04.5 LTS, the programming language was Python 3.6 and the depth learning framework was Keras.

One of the common datasets for intrusion detection experiment is the NSL-KDD dataset [7], which is an improvement version of the KDD Cup'99 dataset [11]. It solves the inherent problems in the KDD Cup'99 dataset. Although the NSL-KDD dataset has the defects of old data, redundant information and

unbalanced numbers of categories [1], the data can be improved after data preprocessing and random oversampling. In addition, these kinds of unbalanced numbers of attack categories exist in actual IDSs. Hence, we chose the NSL-KDD dataset for experimentation in this paper. There are 42 attributes in the dataset, including one category attribute, three non-digital attributes and 38 digital attributes. The training set of the NSL-KDD dataset has 125,973 samples, while the testing set has 22,544 samples.

There are dozens of attack types in the NSL-KDD dataset. However, experts believe that most of the new attacks are variants of known attacks, so these attacks are divided into four categories [7]: (1) Denial of service (DoS): exhausting the resources of the attacked object by savage means; thus, making it unable to provide normal services; paralysis. (2) R2L: unauthorized access to remote computers. (3) U2R: unauthorized access to local superuser privileges. (4) Probe: monitoring and other detection behavior. The types of attacks contained in each category are shown in Table 1.

Table 1. The types of attacks contained in each category.

Attack Category	Attack Types
DoS	apache2, back, land, mailbomb, Neptune, pod, processtable, smurf, teardrop, and udpstorm.
Probe	Ipsweep, portsweep, mscan, nmap, saint, and satan.
U2R	buffer_overflow, loadmodule, httptunnel, perl, ps, sqlattack, rootkit, and xterm.
R2L	ftp_write, guess_passwd, phf, imap, multihop, named, sendmail, snmpgetattack, snmpguess, spy, warezclient, warezmaster, worm, xlock, and xsnoop.

Because the testing set of the NSL-KDD dataset has quite different probability distribution from its training set, and contains many types of attacks that do not appear in the training set, we selected the training set of the NSL-KDD dataset as the experimental dataset. In this paper, the experimental dataset is divided by proportion, where 70% of samples are used for training model, and 30% of samples are used for testing model. And the attack types of the samples in the experimental dataset are converted into five categories: Normal, DoS, Probe, R2L and U2R. The number of samples of each category in the experimental dataset is shown in Figure 7.

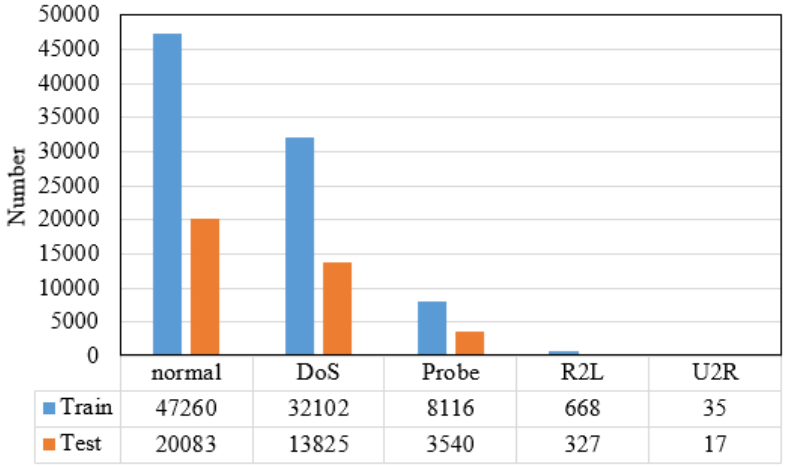

Figure 7. The number of samples of each category in the experimental dataset.

It can be seen from Figure 7 that the proportions of samples of each category in the experimental dataset are 53.6%, 36.4%, 9.2%, 0.76% and 0.04%. Thus, the experimental dataset is a dataset with seriously unbalanced samples in each category.

5.2. Experimental Performance Evaluation

For classification problems, the confusion matrix, including four basic metrics, is generally applied, as shown in Table 2.

Table 2. The confusion matrix, including four basic metrics.

	Predicted Positive Class	**Predicted Negative Class**
Actual positive class	True Positive (TP)	False Negative (FN)
Actual negative class	False Positive (FP)	Ture Negative (TN)

In Table 2, for a sample, TP indicates that the predicted class of the sample is true when the actual class of the sample is true. TN indicates that the predicted class of the sample is false when the actual class of the sample is false. FP indicates that the predicted class of the sample is true when the actual class of the sample is false. FN indicates that the predicted class of the sample is false when the actual class of the sample is true.

To further evaluate the performance of the classification models, several widely applied metrics, i.e., accuracy, precision, recall and F1-score, are generally used. They are calculated based on the four basic metrics of the confusion matrix shown in Table 2. And the expressions of them are as follows.

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN}, \tag{4}$$

$$precision = \frac{TP}{TP + FP}, \tag{5}$$

$$recall = \frac{TP}{TP + FN}, \tag{6}$$

$$F1 - score = \frac{2(precison \times recall)}{precision + recall} = \frac{2TP}{2TP + FP + FN}. \tag{7}$$

Accuracy is the proportion of the number of samples correctly predicted to the total number of samples. Precision is the proportion of the number of actual positive samples that are correctly predicted to the number of all samples predicted to be positive. Recall is the proportion of the number of actual positive samples that are correctly predicted to the number of all actual positive samples. F1-score is the harmonic mean of precision and recall. In other words, it can be interpreted as a weighted average of the precision and recall.

Accuracy is the most intuitive performance of an IDS. It directly reflects the superiority of the IDS. However, if we use accuracy alone to evaluate the performance of the IDS, it may lead to serious skew of classification, resulting in high accuracy of most classes, but low accuracy of a few classes. Precision and recall are two important performance factors of the IDS, but there may be conflicts between them. Besides, F1-score is usually more useful than accuracy, especially for an imbalanced class distribution. Therefore, this paper mainly uses accuracy, recall and F1-score to evaluate the performance of our proposed IDS.

5.3. Experimental Results and Analysis

For the above experimental dataset, we first carried out one-hot coding, transforming non numerical variables into computable numerical forms, and expanding the original 41 dimensions to 122 dimensions, and then carrying out Z-score standardization. If the feature filtering method in [43] was used directly here, then the attribute with the least correlation would be removed from the above 122-dimensional dataset, and the 121-dimensional dataset will be converted into a series of single channel images of 11×11. However, it will reduce the amount of original information and cause information loss. In order to avoid this problem, we added a 22-dimensional dataset of 0 to form a 144-dimensional dataset, and then converted the 144-dimensional dataset into a series of 12×12 single

channel images. That way, the original information was retained and the redundant information was not increased.

After the above data preprocessing, we used random oversampling to make that the proportions of samples of normal, DOS, Probe, R2L and U2R in the dataset were 1:1:1:1:1:1, and then used it to train our proposed IDS. According to the size and dimensions of the above converted dataset, the S-ResNet used in the experiment is shown in Figure 8.

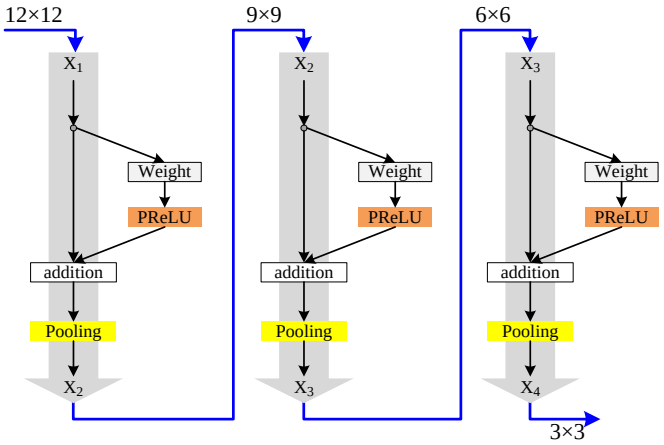

Figure 8. The S-ResNet used in the experiment.

In Figure 8, a series of 12 × 12 single channel images that denote the above converted dataset were inputted into the S-ResNet. And then the optimal architecture of the S-ResNet was chosen experimentally. At the same time, in order to compare with ResNet, we used the original residual block to replace the simplified residual block in Figure 8, and added a pooling layer to form an equal scale ResNet to replace the S-ResNet of the IDS shown in Figure 4, and then carry out a similar experiment.

Through experiments, the confusion matrixes are shown in Figures 9 and 10.

According to Figures 9 and 10, the accuracy, recall and F1-score of the IDS based on the S-ResNet, and the accuracy, recall and F1-score of the equal scale ResNet-based IDS, are shown in Table 3.

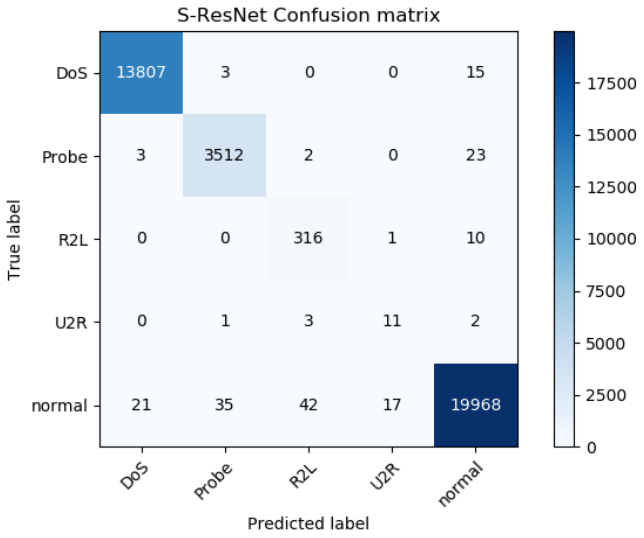

Figure 9. The confusion matrix of the IDS based on the S-ResNet.

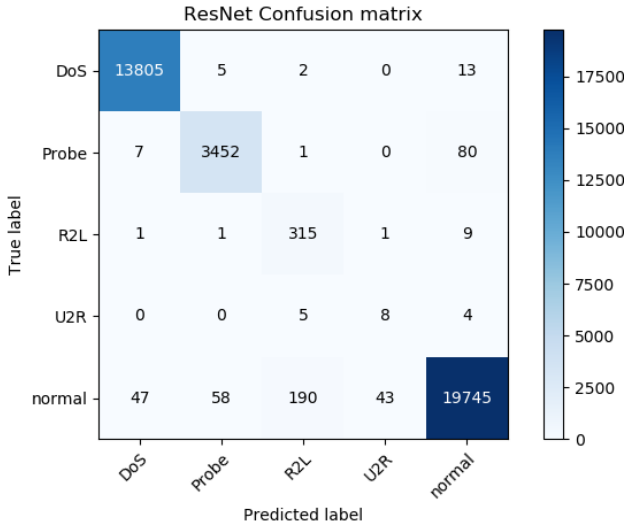

Figure 10. The confusion matrix of the equal scale ResNet-based IDS.

Table 3. The comparison results for the two IDSs.

IDS.	Accuracy (%)	Recall	F1-Score
The IDS based on the S-ResNet	99.529	0.99529	0.99541
The equal scale ResNet-based IDS	98.765	0.98764	0.98857

Additionally, the recall and F1-score of the IDS based on the S-ResNet for each category, and the recall and F1-score of the equal scale ResNet-based IDS for each category, are shown in Figures 11 and 12.

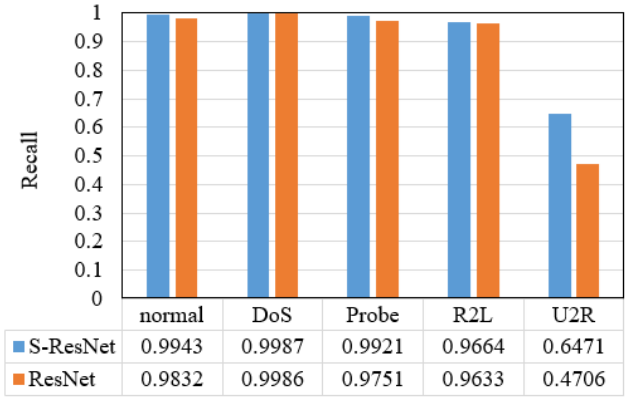

Figure 11. The recall results of the two IDSs for each category.

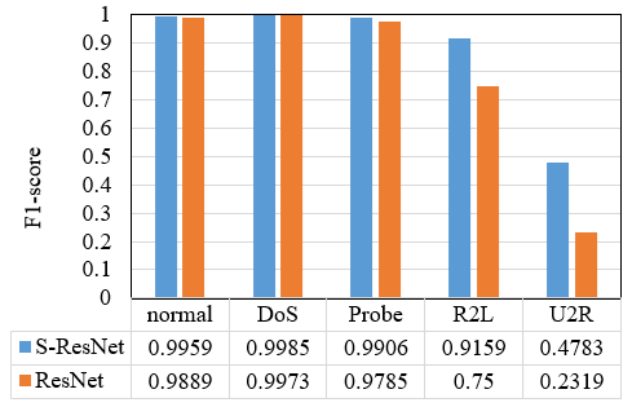

Figure 12. The F1-score results of the two IDSs for each category.

From Table 3, we can see that the IDS based on the S-ResNet has higher accuracy, recall and F1-score than the equal scale ResNet-based IDS. And according to Figures 11 and 12, the recall and F1-score of the IDS based on the S-ResNet for each category are higher than those of the equal scale ResNet-based IDS for each category, especially for R2L and U2R attacks.

Figures 13 and 14 show how the accuracy of the two IDSs changes when training and testing models. Figures 15 and 16 show how the loss of the two IDSs changes when training and testing models.

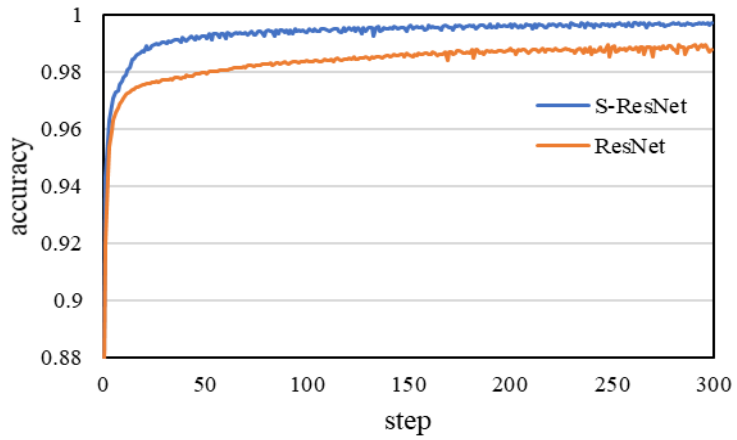

Figure 13. The change of the accuracy of the two IDSs when training models.

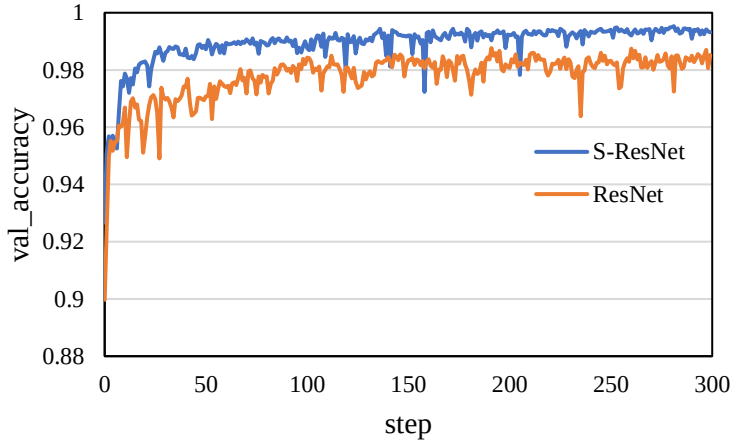

Figure 14. The change of the accuracy of the two IDSs when testing models.

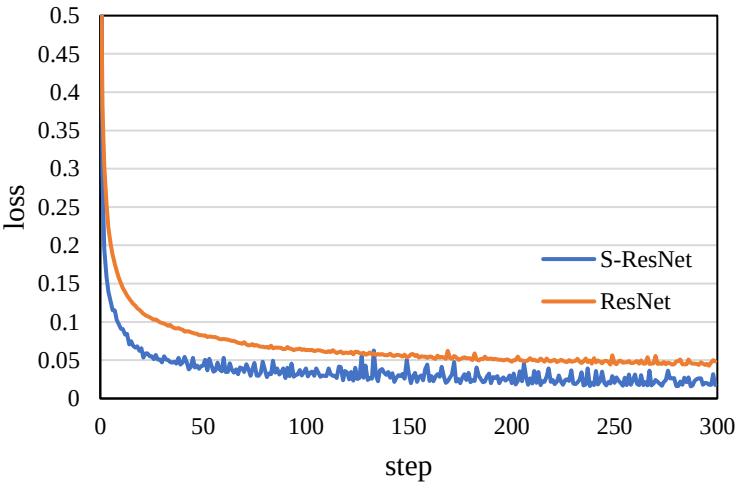

Figure 15. The change of the loss of the two IDSs when training models.

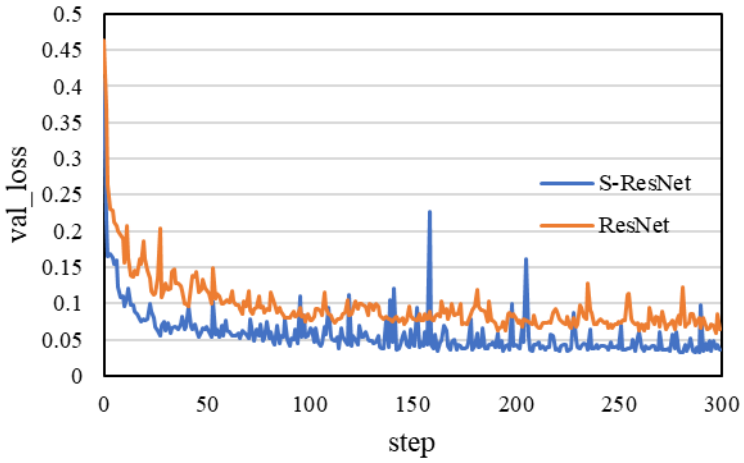

Figure 16. The change of the loss of the two IDSs when testing models.

Compared with the equal scale ResNet-based IDS, the IDS based on the S-ResNet achieves a higher accuracy value at a faster speed according to Figures 13 and 14, and achieves a lower loss value at a faster speed according to Figures 15 and 16.

According to the comparative analysis of the above experimental results, whether accuracy, recall, F1-score or convergence speed, the IDS based on the S-ResNet is better than the equal scale, ResNet-based IDS. It proves that the S-ResNet reduces the complexity of the network and effectively prevents over-fitting; thus, it is more suitable for low-dimensional and small-scale datasets than ResNet.

In addition, Table 4 shows the comparison accuracy of the IDS based on the S-ResNet and the other IDSs on the NSL-KDD dataset. Table 5 shows the comparison recall of the IDS based on the S-ResNet and the other IDSs for each category on the NSL-KDD dataset.

Table 4. The comparison accuracy on the NSL-KDD dataset.

IDSs	Accuracy (%)
GINI-GBDTPSO [35]	86.10
CNN [43]	79.48
LSTM [44]	92.00
DMNB [45]	96.50
DBN-SVM [46]	92.84
TUIDS [47]	96.55
RNN-IDS [48]	81.29
Our Proposed IDS	99.529

Table 5. The comparison recall for each category on the NSL-KDD dataset.

IDSs	Recall				
	Normal	DoS	Probe	R2L	U2R
OS-ELM [49]	0.9907	0.9914	0.9035	0.7810	0.5675
MDPCA-DBN [50]	0.8694	0.6875	0.6320	0.3493	0.6000
Hierarchical SOM [51]	0.9840	0.9690	0.6760	0.7300	0.1570
MOGFIDS [52]	0.9836	0.9720	0.8859	0.1578	0.1101
TVCPSO-SVM [53]	0.9913	0.9884	0.8929	0.6784	0.4038
Our Proposed IDS	0.9943	0.9987	0.9921	0.9664	0.6471

From Tables 4 and 5, our proposed IDS (i.e., the IDS based on the S-ResNet) has higher accuracy than the other IDSs on the NSL-KDD dataset, and higher recall than the other IDSs for each category on the NSL-KDD dataset, especially for R2L and U2R attacks.

6. Conclusions

Based the original residual block, we deleted a weight layer and two BN layers, added a pooling layer, and replaced the ReLU function with the PReLU function to form a simplified residual block, whose purpose is to reduce the parameters and amount of calculations for a model and effectively prevent over-fitting and improve the generalization ability of the model. And then, we proposed a S-ResNet, which is a cascade of simplified residual blocks. On basis of the S-ResNet, we proposed a novel IDS, which includes a data preprocessing module, a random oversampling module, a S-Resnet layer, a full connection layer and a Softmax layer. In this IDS, the original dataset is pre-processed and converted into a series of single channel images (i.e., a series of black and white images). Then, random oversampling is used to balance the number of samples of each category. Finally, through the S-ResNet layer and the full connection layer, and the prediction probability of the corresponding category is converted through the Softmax layer.

After the experiments of our proposed IDS and the equal scale ResNet-based IDS with the NSL-KDD dataset, the experimental results show that our proposed IDS has higher accuracy, recall and F1-score than the equal scale ResNet-based IDS, especially for R2L and U2R attacks. And the former has faster convergence velocity than the latter. It proves that the S-ResNet reduces the complexity of the network and effectively prevents over-fitting, solving the over-fitting problem of ResNet for low-dimensional and small-scale datasets, and providing better performance than ResNet. Therefore, the S-ResNet is more suitable for low-dimensional and small-scale datasets than ResNet. Additionally, the experimental results also show that our proposed IDS has higher accuracy than the other IDSs, and higher recall than the other IDSs for each attack category, especially for R2L and U2R attacks. Hence, our proposed IDS provides better performance than the existing IDSs. In this study, the NSL-KDD dataset was used for experimental verification of our proposed IDS. We will further validate our proposed IDS based on the other IDS datasets, and study the variants of the original residual block, and form new residual network models based on these variants; and propose new IDSs based on these residual network models, achieving better performance in terms of accuracy, precision, recall, F1-score, etc., especially for R2L and U2R attacks.

Author Contributions: Methodology, Y.X.; software, X.X.

Funding: This work was supported by the National Natural Science Foundation of China (61741216, 61402367), the Shaanxi Science and Technology Co-ordination and Innovation Project (2016KTTSGY01-03), the Special Scientific Research Project of Education Department of Shaanxi Province (17JK0704) and the New Star Team Project of Xi'an University of Posts and Telecommunications.

Acknowledgments: The authors would like to thank the anonymous reviewers for their contribution to this paper.

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

References

1. Xin, Y.; Kong, L.S.; Liu, Z.; Chen, Y.L. Machine learning and deep learning methods for cybersecurity. *IEEE Access* **2018**, *6*, 35365–35381. [CrossRef]
2. Ambusaidi, M.A.; He, X.J.; Nanda, P.; Tan, Z.Y. Building an intrusion detection system using a filter-based feature selection algorithm. *IEEE Trans. Comput.* **2016**, *65*, 2986–2998. [CrossRef]
3. Ghazy, R.A.; El-Rabaie, E.M.; Dessouky, M.I.; El-Fishawy, N.A.; Abd El-Samie, F.E. Efficient techniques for attack detection using different features selection algorithms and classifiers. *Wirel. Pers. Commun.* **2018**, *100*, 1689–1706. [CrossRef]
4. Aljawarneh, S.; Aldwairi, M.; Yassein, M.B. Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model. *J. Comput. Sci.* **2018**, *25*, 152–160. [CrossRef]
5. Kang, S.H.; Kim, K.J. A feature selection approach to find optimal feature subsets for the network intrusion detection system. *Cluster Comput.* **2016**, *19*, 325–333. [CrossRef]
6. Salo, F.; Nassif, A.B.; Essex, A. Dimensionality reduction with IG-PCA and ensemble classifier for network intrusion detection. *Comput. Netw.* **2019**, *148*, 164–175. [CrossRef]

7. Tavallaee, M.; Bagheri, E.; Lu, W.; Ghorbani, A.A. A detailed analysis of the KDD CUP 99 data set. In Proceedings of the 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, Ottawa, ON, Canada, 8–10 July 2009; pp. 1–6.
8. Beulah, J.R.; Punithavathani, D.S. A hybrid feature selection method for improved detection of wired/wireless network intrusions. *Wirel. Pers. Commun.* **2018**, *98*, 1853–1869. [CrossRef]
9. Bostani, H.; Sheikhan, M. Hybrid of binary gravitational search algorithm and mutual information for feature selection in intrusion detection systems. *Soft Comput.* **2017**, *21*, 2307–2324. [CrossRef]
10. Acharya, N.; Singh, S. An IWD-based feature selection method for intrusion detection system. *Soft Comput.* **2017**, *22*, 4407–4416. [CrossRef]
11. KDD Cup99. Available online: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html (accessed on 17 October 2019).
12. Akashdeep, S.; Manzoor, I.; Kumar, N. A feature reduced intrusion detection system using ANN classifier. *Expert Syst. Appl.* **2017**, *88*, 249–257. [CrossRef]
13. Akyol, A.; Hacibeyoglu, M.; Karlik, B. Design of multilevel hybrid classifier with variant feature sets for intrusion detection system. *IEICE Trans. Inf. Syst.* **2016**, *ED99*, 1810–1821. [CrossRef]
14. Bhattacharya, S.; Selvakumar, S. LAWRA: A layered wrapper feature selection approach for network attack detection. *Secur. Commun. Netw.* **2015**, *8*, 3459–3468. [CrossRef]
15. Panda, M.; Abraham, A.; Patra, M.R. Hybrid intelligent systems for detecting network intrusions. *Secur. Commun. Netw.* **2015**, *8*, 2741–2749. [CrossRef]
16. Ahmad, I.; Basheri, M.; Iqbal, M.J.; Rahim, A. Performance comparison of support vector machine, random forest, and extreme learning machine for intrusion detection. *IEEE Access* **2018**, *6*, 33789–33795. [CrossRef]
17. Aburomman, A.A.; Reaz, M.B. A survey of intrusion detection systems based on ensemble and hybrid classifiers. *Comput. Secur.* **2017**, *65*, 135–152. [CrossRef]
18. Lilakiatsakun, W.; Somwang, P. Anomaly traffic detection based on PCA and SFAM. *Int. Arab J. Inf. Technol.* **2013**, *12*, 253–260.
19. Alabdallah, A.; Awad, M. Using weighted support vector machine to address the imbalanced classes problem of intrusion detection system. *KSII Trans. Internet Inf. Syst.* **2018**, *12*, 5143–5158.
20. Li, L.J.; Yu, Y.; Bai, S.S.; Hou, Y.; Chen, X.Y. An effective two-step intrusion detection approach based on binary classification and kNN. *IEEE Access* **2018**, *6*, 12060–12073. [CrossRef]
21. Demir, N.; Dalkilic, G. Modified stacking ensemble approach to detect network intrusion. *Turk. J. Electr. Eng. Comput. Sci.* **2018**, *26*, 418–433. [CrossRef]
22. Kamarudin, M.H.; Maple, C.; Watson, T.; Safa, N.S. A LogitBoost-based algorithm for detecting known and unknown web attacks. *IEEE Access* **2017**, *5*, 26190–26200. [CrossRef]
23. Tian, Y.J.; Mirzabagheri, M.; Bamakan, S.M.H.; Wang, H.D.; Qu, Q. Ramp loss one-class support vector machine: A robust and effective approach to anomaly detection problems. *Neurocomputing* **2018**, *310*, 223–235. [CrossRef]
24. Kabir, E.; Hu, J.K.; Wang, H.; Zhuo, G.P. A novel statistical technique for intrusion detection systems. *Future Gener. Comput. Syst.* **2018**, *79*, 303–318. [CrossRef]
25. Ahmim, A.; Derdour, M.; Ferrag, M.A. An intrusion detection system based on combining probability predictions of a tree of classifiers. *Int. J. Commun. Syst.* **2018**, *31*, 1–14. [CrossRef]
26. Aburomman, A.A.; Reaz, M.B. A novel weighted support vector machines multiclass classifier based on differential evolution for intrusion detection systems. *Inf. Sci.* **2017**, *414*, 225–246. [CrossRef]
27. Yan, B.H.; Han, G.D. LA-GRU: Building combined intrusion detection model based on imbalanced learning and gated recurrent unit neural network. *Secur. Commun. Netw.* **2018**, *1*, 1–13. [CrossRef]
28. Idhammad, M.; Afdel, K.; Belouch, M. Semi-supervised machine learning approach for DDoS detection. *Appl. Intell.* **2018**, *48*, 3193–3208. [CrossRef]
29. Mohammadi, S.; Namadchian, A. A new deep learning approach for anomaly base IDS using memetic classifier. *Int. J. Comput. Commun.* **2017**, *12*, 677–688. [CrossRef]
30. Imamverdiyev, Y.; Abdullayeva, F. Deep learning method for denial of service attack detection based on restricted boltzmann machine. *Big Data-US* **2018**, *6*, 159–169. [CrossRef]
31. Ma, T.; Wang, F.; Cheng, J.; Yu, Y.; Chen, X. A hybrid spectral clustering and deep neural network ensemble algorithm for intrusion detection in sensor networks. *Sensors (Basel)* **2016**, *16*, 1701. [CrossRef]
32. Shamshirband, S.; Daghighi, B.; Anuar, N.B.; Kiah, M.L.M.; Patel, A.; Abraham, A. Co-FQL: Anomaly detection using cooperative fuzzy Q-learning in network. *J. Intell. Fuzzy Syst.* **2015**, *28*, 1345–1357.

33. Al-Qatf, M.; Yu, L.S.; Al-Habib, M.; Al-Sabahi, K. Deep learning approach combining sparse autoencoder with SVM for network intrusion detection. *IEEE Access* **2018**, *6*, 52843–52856. [CrossRef]
34. Hussain, J.; Lalmuanawma, S.; Chhakchhuak, L. A two-stage hybrid classification technique for network intrusion detection system. *Int. J. Comput. Int. Syst.* **2016**, *9*, 863–875. [CrossRef]
35. Li, L.J.; Yu, Y.; Bai, S.S.; Cheng, J.J.; Chen, X.Y. Towards effective network intrusion detection: A hybrid model integrating Gini index and GBDT with PSO. *J. Sens.* **2018**, *6*, 1–9. [CrossRef]
36. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), PIEAS, Islamabad, Pakistan, 26–27 August 2016; pp. 770–778.
37. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the Inception Architecture for Computer Vision. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826.
38. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.* **2014**, *115*, 1–37. [CrossRef]
39. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *Comput. Sci.* **2014**, *9*, 1–14.
40. He, K.; Zhang, X.; Ren, S.; Jian, S. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1–11.
41. He, K.; Zhang, X.; Ren, S.; Sun, J. Identity mappings in deep residual networks. In Proceedings of the 2014 European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 11–14 October 2015; pp. 630–645.
42. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic Minority Over-sampling Technique. *J. Artif. Intell. Res.* **2011**, *16*, 321–357. [CrossRef]
43. Wu, K.H.; Chen, Z.G.; Li, W. A novel intrusion detection model for a massive network using convolutional neural networks. *IEEE Access* **2018**, *6*, 50850–50859. [CrossRef]
44. Le, T.T.H.; Kim, Y.; Kim, H. Network intrusion detection based on novel feature selection model and various recurrent neural networks. *Appl. Sci.-Basel* **2019**, *9*, 1392. [CrossRef]
45. Panda, M.; Abraham, A.; Patra, M.R. Discriminative multinomial naive Bayes for network intrusion detection. In Proceedings of the 2010 Sixth International Conference on Information Assurance and Security, Atlanta, GA, USA, 23–25 August 2010; pp. 5–10.
46. Salama, M.A.; Eid, H.F.; Ramadan, R.A.; Darwish, A.; Hassanien, A.E. Hybrid intelligent intrusion detection scheme. *Soft Comput. Ind. Appl.* **2011**, *96*, 293–303.
47. Gogoi, P.; Bhuyan, M.H.; Bhattacharyya, D.; Kalita, J.K. Packet and flow based network intrusion dataset. *Contemp. Comput.* **2012**, *306*, 322–334.
48. Yin, C.; Zhu, Y.; Fei, J.; He, X. A deep learning approach for intrusion detection using recurrent neural networks. *IEEE Access* **2017**, *5*, 21954–21961. [CrossRef]
49. Singh, R.; Kumar, H.; Singla, R.K. An intrusion detection system using network traffic profiling and online sequential extreme learning machine. *Expert Syst. Appl.* **2015**, *42*, 8609–8624. [CrossRef]
50. Yang, Y.Q.; Zheng, K.F.; Wu, C.H.; Niu, X.X.; Yang, Y.X. Building an effective intrusion detection system using the modified density peak clustering algorithm and deep belief networks. *Appl. Sci.-Basel* **2019**, *9*, 238. [CrossRef]
51. Kayacik, H.G.; Zincir-Heywood, A.N.; Heywood, M.I. Ahierarchical SOM-based intrusion detection system. *Eng. Appl. Artif. Intell.* **2007**, *20*, 439–451. [CrossRef]
52. Tsang, C.H.; Kwong, S.; Wang, H. Genetic-fuzzy rule mining approach and evaluation of feature selection techniques for anomaly intrusion detection. *Pattern Recognit.* **2007**, *40*, 2373–2391. [CrossRef]
53. Bamakan, S.M.H.; Wang, H.; Yingjie, T.; Shi, Y. An effective intrusion detection framework based on MCLP/SVM optimized by timevarying chaos particle swarm optimization. *Neurocomputing* **2016**, *199*, 90–102. [CrossRef]