IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 13 March 2023, accepted 27 April 2023, date of publication 1 May 2023, date of current version 12 July 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3271866

RESEARCH ARTICLE

Network Intrusion Detection Method Based on Hybrid Improved Residual Network Blocks and Bidirectional Gated Recurrent Units

HONGCHEN YU, CHUNYING KANG, YAO XIAO, AND YUTING YANG
School of Data Science and Technology, Heilongjiang University, Harbin 150000, China
Corresponding author: Chunying Kang (kangchunying@hlju.edu.cn)

ABSTRACT With the rapid development of network technology, network intrusion detection plays a vital role in network security. In the era of big data, a large amount of network data is generated in the network all the time. Traditional detection methods do not achieve high accuracy and need to take a long time to detect network data. Therefore, how to improve the efficiency and accuracy of detection has become a hot topic of current research. Since each network traffic data has both spatial and temporal characteristics, this paper proposes a hybrid network classifier consisting of improved residual network blocks and bidirectional gated recurrent units. Before inputting the classification network, the feature dimensionality of the network data is first reduced using an improved autoencoder, and then the processed network data is detected using the constructed hybrid network classifier. In this paper, the proposed research approach is justified using official experimental datasets in the field of network detection (NSL-KDD and UNSW-NB15). The experimental results show that the proposed method in this paper achieves a higher accuracy of 93.40% and 93.26% on the datasets of NSL_KDD and UNSW_NB15, respectively, compared with the known detection methods.

INDEX TERMS Bidirectional gated recurrent units, residual networks block, improved auto encoder, network intrusion detection.

I. INTRODUCTION

With the continuous development of Internet of Things technology, a large number of Internet of Things devices have become more and more intelligent, and have gradually involved in various industries. Such as education, industry, finance, medicine, military, transportation, tourism, etc., which greatly facilitate people's quality of life and improve the efficiency of people's work. At the same time, hacker technology is constantly updated and iterated. Relying solely on firewall technology cannot ensure the security of networks and computer systems. So an intrusion detection system with high accuracy, low false negative rate, and low false positive rate becomes very important [1].

Intrusion detection technology mainly includes signature-based intrusion detection and anomaly-based intrusion detection. Signature-based detection mainly relies on manual experience for pattern matching, which has high detection efficiency, but it cannot detect new attacks and needs to update the pattern library frequently. Anomaly-based detection technology mainly uses machine learning, deep learning, and other methods to find malicious traffic that deviates from normal traffic. Traditional machine learning detection methods have been widely applied to network intrusion detection, such as Bayesian network [2], decision tree [3], etc., and have achieved good detection results. With the great success of deep learning in natural language processing, image recognition, and other fields, some researchers have introduced deep learning technology into the field of intrusion detection, which has greatly improved the detection effect. Reference [4] according to the characteristics of network intrusion behavior with temporal characteristics, the gated recurrent unit memory module is introduced into the recurrent neural network, and an intrusion detection network model based on memory and timing is proposed. This paper only predicts the two classifications of the NSL_KDD

The associate editor coordinating the review of this manuscript and approving it for publication was Giovanni Pau.

dataset. Reference [5] proposed an abnormal traffic detection method based on a long short memory network (LSTM) and improved residual neural network optimization to solve the defects of traditional intrusion detection methods such as poor feature selection and weak generalization ability.

Reference [6] proposed a mobile malware traffic detection method based on value derivative GRU. This method introduces the concept of ' cumulative state change ' to describe the change information of mobile network malicious traffic in different hidden layers of GRU network, and adds pooling operation in the hidden layer, so that the algorithm can capture the key information of mobile malicious traffic and improve the accuracy of intrusion detection. Because the derivative information of each hidden node is calculated and saved, it does not mean that the data with too high dimension and large scale still has a good effect.

Reference [7] introduces the sparrow search algorithm into the particle swarm optimization algorithm to improve the basic particle swarm optimization algorithm, which is easy to fall into local optimum, slow convergence speed, and low accuracy of later optimization. The improved SSAPSO algorithm uses the sparrow's large-scale fast search ability to improve the convergence speed of the particle swarm and improve the performance of the algorithm. However, the model of this paper is based on the KDD99 data set, and the detection effect is good, but the data set is not convincing.

Reference [8] proposed an online learning model combining K-means clustering and GRU neural network for trajectory prediction. This method uses an online learning prediction model based on GRU neural network to learn the trajectory points of each cluster.

Reference [9] proposed a complaint classification model based on a hybrid attention mechanism and GRU neural network (HATT-GRU). The bidirectional GRU neural network is used to capture sequence information of different lengths and learn the correlation between features.

Reference [10] proposed a multivariate time series data region clustering feature extraction model based on CNN-GRU, using CNN to identify the characteristics of each variable, and based on GRU to derive trends over time.

Reference [11] proposed a network intrusion detection method based on CNN_BiLSTM, which extracts network traffic features in parallel by CNN and BiLSTM. However, with the deepening of CNN, network degradation is prone to occur, and only one data set has been tested, which cannot well explain the feasibility of the model.

Through analysis, it can be found that the above network intrusion detection methods all use a machine learning algorithm and a neural network model to classify network intrusion data. But most algorithms just perform simple preprocessing on the data and then use machine learning algorithms for feature selection. The preprocessing is only done by using relevant text analysis and simple normalization of the data, so the optimal features obtained are still redundant. At the same time, only using normalization and other operations to process network traffic data will directly filter out some important features of the original network data samples, which cannot achieve the purpose of optimal data cleaning. Blindly increasing the neural network model will produce phenomena such as training regression, overfitting, gradient disappearance, and generalization ability.

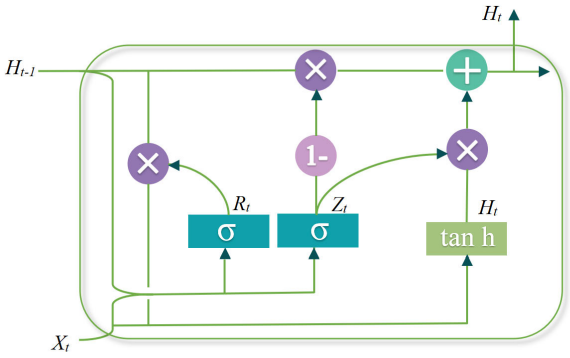

FIGURE 1. Gated Recurrent hidden layer unit.

The main contributions of this work can be briefly highlighted as follows:

- This paper uses a greedy strategy to improve the self-encoder and solves the problem that with the increase of the number of network layers, the conventional self-encoder cannot make good use of shallow network parameters, resulting in unsatisfactory coding results.
- This paper first proposes the concept of a double pooled layer and applies it to bidirectional gated recurrent units to enhance its ability to extract time-series features.
- This paper first proposes the concept of a double-pool layer and applies it to residual network blocks to enhance its ability to extract spatial features.
- This paper proposes a hybrid network intrusion detection method based on improved residual network blocks and improved bidirectional gated recurrent units.

II. MATERIALS AND METHODS

A. BiGRU (BIDIRECTIONAL GATED RECURRENT UNIT)

A gated recurrent unit (GRU) [12] is a commonly used gated recurrent neural network, which is a simplified version of LSTM (Long short-term memory). Compared with LSTM, GRU simplifies the gating unit reduces the network parameters and is less likely to produce overfitting, and GRU achieves better results with the same number of iterations, so GRU can make the network structure simpler while maintaining the LSTM effect.At present, GRU has been widely used. GRU includes an update gate and a reset gate, which determine the retention and discarding of information respectively. The gated Recurrent hidden layer unit is shown in figure 1.

$$Z_t = \sigma(W_z[H_t - 1, X_t]) \tag{1}$$

$$R_t = \sigma(W_R[H_t - 1, X_t]) \tag{2}$$

$$\widetilde{H}_t = tanh(W_H[R_t \bigodot H_t - 1, X_t]) \tag{3}$$

$$H_t = (1 - Z_t) \bigodot H_t - 1 + Z_t \bigodot \widetilde{H}_t \tag{4}$$

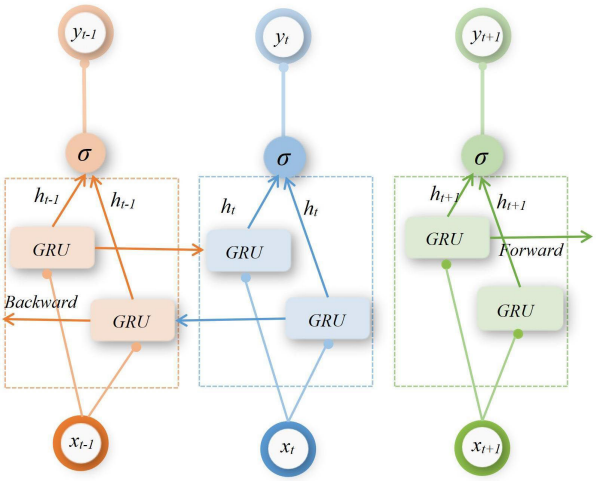

FIGURE 2. The structure of bidirectional gated recurrent units is shown in figure.

Z_t is the update gate at time step t; R_t is the reset gate at time step t; $\widetilde{H}_t$ is the state of the hidden layer unit at time step t; H_t is also used as the input for the next time step; X_t is the input at the current time step t; $H_t - 1$ is the state of the hidden layer unit at the previous moment. Since GRU can only read sequence data from one direction, the influence of subsequent information is not fully considered, so this paper uses BiGRU instead of GRU to process network data. BiGRU [13] is generated by combining forward GRU with reverse GRU for learning from forward and reverse time series data. The hidden layer contains two units with the same input and connected to the same output. Among them, one unit handles the forward time series and the other handles the backward time series, increasing the time series participating in training by better learning features, thus providing higher accuracy for longer time series data, as shown in equation 5. The structure of bidirectional gated recurrent units is shown in figure 2.

$$V_t = [\overrightarrow{H} : \overleftarrow{H}] \tag{5}$$

$\overrightarrow{H}$ is the state of gated recurrent unit for forward; $\overleftarrow{H}$ is the state of gated recurrent unit for backward;

B. IAE(IMPROVED AUTOENCODER)

Autoencoder (AE) [14] consists of two parts: encoder and decoder. It is an unsupervised learning network consisting of input layer, hidden layer and output layer. The sequence X is encoded by the encoder to obtain the hidden layer vector sequence L, and the sequence L is reconstructed by the decoder to obtain the reconstructed signal $\widetilde{X}$ with the same dimension as the input sequence. The pre-training of the preprocessed data by the self-coder can effectively reduce the dimension of high-dimensional data, effectively reduce the detection time and reduce the impact of redundant data features on the performance of the detection method.

$$L_t = f(WX_t + b) \tag{6}$$

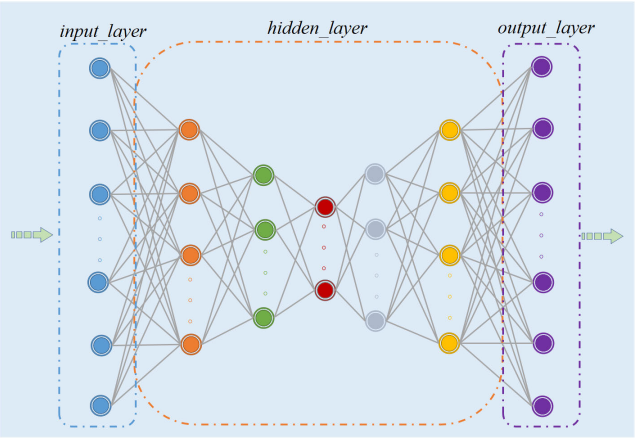

FIGURE 3. Improved Auto Encoder Network structure.

X_t is the input of decoder,W is the weight between input X_t, b is the bias,f is the function of decoding, L_t is the result of decoding.

$$\widetilde{s}_t = g(WL_t + b) \tag{7}$$

g is the function of decoding,$\widetilde{s}_t$ is the result of decoding

$$loss_t = ||X_t - \widetilde{X}_t||^2 \tag{8}$$

$loss_t$ is the loss between the input of encoder and the output of decoder

However, as the number of network layers increases, it is difficult to train low-level network parameters in deep networks. To solve this problem, this paper proposes a greedy hierarchical weight initialization method based on the traditional method, which ensures that the loss of each layer is minimized. This method first trains the first layer of the network to minimize the training loss, and finally uses the output of the first layer as the input of the second layer to train the model to minimize the training loss of the network model, and then uses the output as the input of the third layer to train the model, Minimize the training loss of the network model. The network structure is shown in figure 3.

The network parameters are updated through backpropagation to make the input sequence and the reconstructed sequence as similar as possible to minimize the training loss, and finally, the intermediate result of the hidden layer is used as the input of the anomaly detection network.

C. IMPROVED RESIDUAL NETWORK BLOCK

Convolutional neural networks [15] generally consist of convolutional layers, pooling layers, and fully connected layers. The convolution layer is used to extract the features of the local area. Different convolution kernels are equivalent to different feature extractors. The function of the convergence layer is to select features and reduce the number of features, thereby reducing the number of parameters. However, with the increase in the number of convolutional layers, it is easy to cause problems such as gradient disappearance and gradient explosion, so how determining an effective network depth is

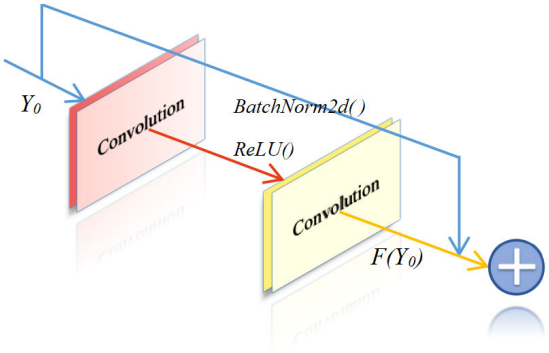

FIGURE 4. Residual convolutional block.

a problem that needs to be overcome. In order to overcome the above problems, this paper uses a residual convolutional neural network to extract local features of the data. It uses skip connections to achieve activation from one layer and suddenly pass it to the next layer, even deeper neural network layers. The structure of residual convolutional block is shown in figure 4.

$$H(Y_0) = F(Y_0) + Y_0 \tag{9}$$

Y_0 is the input of the residual network, $F(Y_0)$ is the output of the residual network, $H(Y_0)$ is the result of skip connections of the residual network block.

Although convolutional neural networks can extract the spatial characteristics of data very well, Convolutional neural networks are prone to the increase of the estimated value variance caused by the limited neighborhood size and the deviation of the estimated mean caused by the parameter error of the convolution layer. Therefore, to overcome the above problems of convolutional networks and enhance the ability of residual network blocks to extract spatial features. This paper makes improvements to residual network blocks. Since average pooling can reduce the increase in variance in estimates caused by neighborhood size constraints and Maximum pooling can reduce the deviation of the estimated mean caused by the parameter error of the convolutional layer. This paper proposed an improved residual network structure based on the above strengths of Average pooling and Maximum pooling. The structure of the improved residual convolutional block is shown in Figure 5.

$$\hat{H}(Y_0) = Max\_pooling(H(Y_0)) + Average\_pooling(H(Y_0)) \tag{10}$$

$\hat{H}(Y_0)$ is the result of double pooling.

III. NETWORK INTRUSION DETECTION MODEL OF THE PROPOSED HYBRID NETWORKS

Since each network traffic data includes spatial features and temporal features. In this paper, we propose a hybrid network-based classifier which consists of improved residual convolutional blocks and bidirectional gated recurrent units.

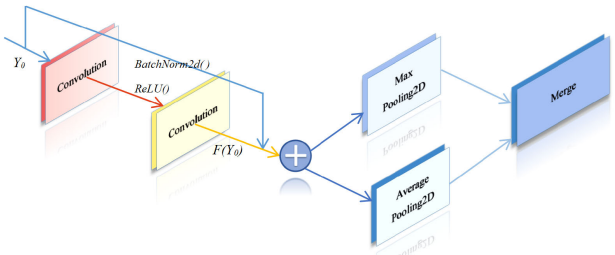

FIGURE 5. The structure of the improved residual convolutional block.

A. THE EXPERIMENTAL PROCESS OF THE PROPOSED METHOD

Currently, traditional network intrusion detection methods cannot detect traffic in the network at all times.It can only detect network malicious attacks at the current moment. Moreover, traditional methods are prone to forgetfulness and network degradation, and cannot make good use of temporal characteristics to detect attacks on network traffic.To better maintain a stable network security environment and improve the accuracy of network intrusion detection. Since each piece of network traffic data have include spatial characteristics and timing characteristics.Therefore, we propose an ensemble hybrid neural network classification method, which extracts the spatial features of data by improved residual convolutional blocks and extracts the time series features of data by using bidirectional gated recurrent units.

This paper aims to be able to extract more timing features and spatial features about the network traffic. Therefore, this paper constructs a two-way gated looping network by bidirectional gated recurrent unit and constructs optimized residual convolutional network by optimized residual convolutional blocks. At the same time, adding regularization can improve the generalization ability of the model and reduce overfitting. Introducing double pooling can reduce the convolution error and improve the expression ability of the detection algorithm. Since the two-dimensional convolution has a wider range than the two-dimensional convolution, in this paper, in order to adapt the data to the two-dimensional convolution, the sequences after dimensionality reduction are processed into gray images. For example, 1*3 convolution only considers the front, middle and back 3 features, while 3*3 convolution considers 9 features at the same time, which can extract more spatial features.

The network intrusion detection method consists of four parts: data processing, time series feature and spatial feature extraction, obtain fusion features, detection and classification. the structure of the network intrusion detection method is shown in Figure 6.

B. EXPERIMENTAL MODEL PARAMETER SETTINGS

The hyper-parameters of the hybrid network intrusion detection model proposed in this paper consist of two parts. One is the hyper-parameters of the iterative IAE feature dimension reduction part, and the other is the hyper-parameters of the hybrid network intrusion detection model.

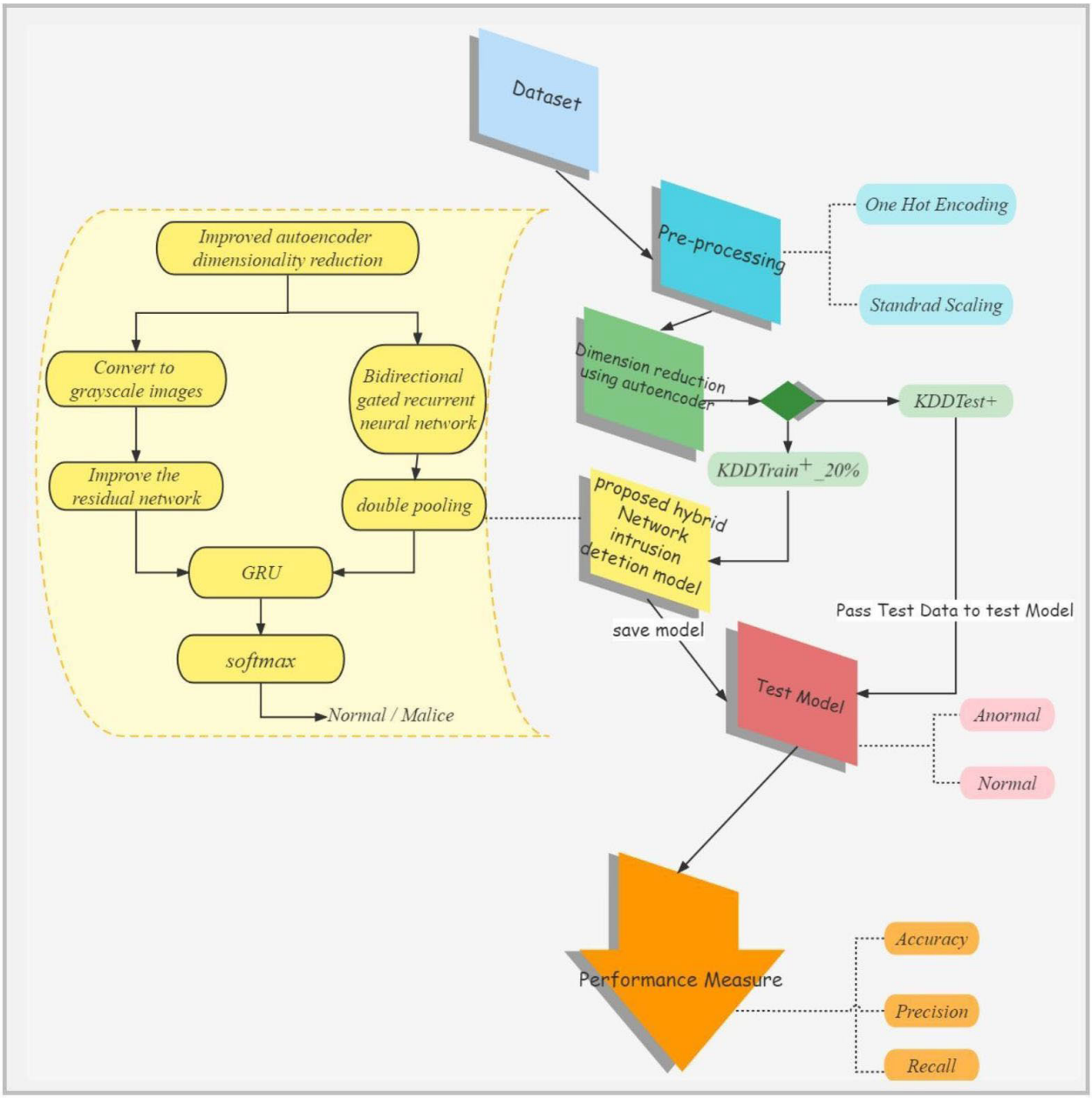

FIGURE 6. The structure of the network intrusion detection method.

After several rounds of hyper-parameter debugging, the detailed experimental hyper-parameters of the model were finally determined, as shown in table 1.

The loss function of AE is square loss function, as shown in equation 11. The loss function of Hybrid Neural network measures the binary cross entropy between the output and the target, as shown in equation 12.

$$loss\_AE = \frac{1}{N}\sum_{i=0}^{N}(X_i - Y_i)^2 \tag{11}$$

$$loss = \frac{1}{N}\sum_{i=0}^{N} W_i[Y_i \log X_i + (1-Y_i)\log(1-Y_i))] \tag{12}$$

IV. EXPERIMENT ANALYSIS

A. EXPERIMENTAL ENVIRONMENT

In this paper, the specific experimental environment is shown in table2.

B. DATASET

The simulation experimental data are selected from the NSL_KDD dataset [16] and the UNSW_NB15 dataset [17]. NSL_KD dataset is a relatively authoritative intrusion detection dataset in the field of network security, which improves some inherent problems of KDDcup99. The training and test sets of the NSL_KDD dataset do not contain redundant records, making detection more accurate. The

TABLE 1. The hyper-parameters of the experiment.

Category	Description	Value
Batch_size1	the size of data each of batch for AE	1024
loss_AE	loss function of AE	nn.MSELoss()
LR_AE	Learning rate of Improved Auto Encoder	0.0001
Epoch_AE	Number of iterations of each of the layer of IAE	2000
Batch	the size of data each of batch for Hybrid Neural network	1024
loss	loss function of Hybrid Neural network	nn.CrossEntropyLoss()
LR	Learning rate of Hybrid Neural network	0.0001
Epoch	Number of iterations of the Hybrid Neural network	80
optimizer	Optimizing the method of Parameter	NAdam

TABLE 2. The configuration of the experimental environment.

Name	Version
GPU	NVIDIA GeForce RTX 3060
CPU	Intel i7-12700H
RAM	32GB
Language	Python3.10
Pytorch	1.12.0

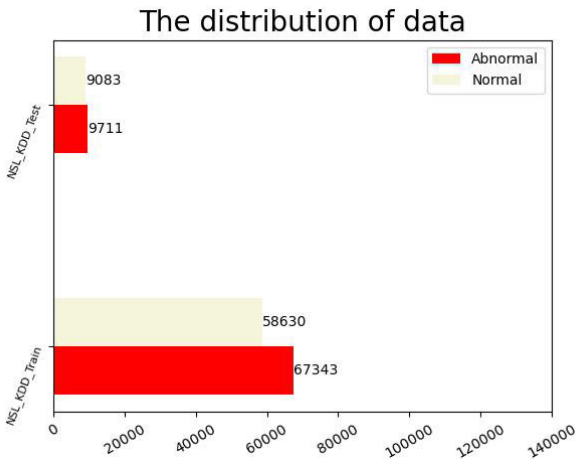

FIGURE 7. Class Distribution of NSL_KDD training and testing set.

dataset contains 41 features. There are 23 attack types in the training set and 38 attack types in the test set. The attack types can be further divided into four categories: Denial of Service(DOS), Remote-to-Login(R2L), User-to-Root(U2R), and PROBE.

1) NSL_KDD

The well-known intrusion detection dataset NSL_KDD has been used to evaluate the methods proposed in this study. We trained with KDDTrain +_20% and tested our proposed model with KDDTest+. The distribution of normal samples and attack samples is shown in Figure 7.

2) UNSW_NB15

The original network packets of the UNSW_NB15 dataset were created by the ACCS Network Scope Lab using the IXI

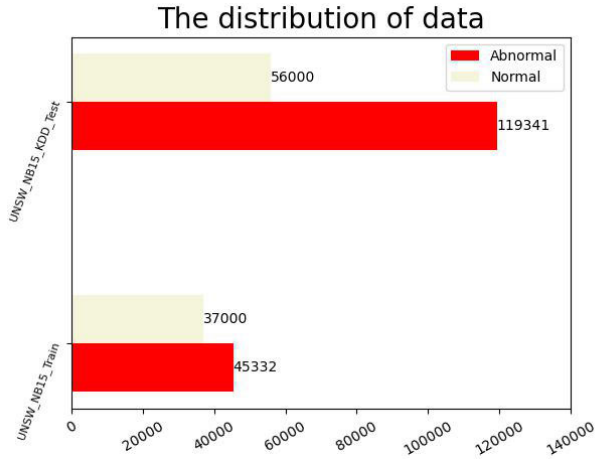

FIGURE 8. Class Distribution of UNSW_NB15 training and testing set.

PrimeSturf tool, and contain both real-world network normal activity and synthetic contemporary attacks. The data set contains nine types of attacks, namely backdoors, penetration analysis, denial of service attacks, exploits, obfuscation testing, generic attacks, stepping stones, shellcode, and worms. This article is divided into two categories, specifically divided into normal traffic: Normal, abnormal traffic: Abnormal. The distribution of normal samples and attack samples is shown in Figure 8.

C. DATASET PREPROCESSING STEP

1) ONE HOT ENCODING

It is found that among the 41-dimensional features of the NSL_KDD dataset, 3-dimensional features are discrete non-numerical features, protocol_type, service, and flag by analyzing of the NSL_KDD dataset and UNSW_NB15 dataset, respectively. Among the 45-dimensional features of the UNSW_NB15 dataset, 3-dimensional features are discrete non-numerical features, proto, service, and state, respectively. Since most deep learning models need to transmit numerical data, they need to be converted into numerical data. This paper applies an independent thermal coding technique to convert discrete non-numerical features into numerical values. This technique counts different values of each feature and assigns a unique index to each value. This article uses the function of pandas(get_dummies( )) to one-hot encode discrete features into numeric variables [18].

2) DATA NORMALIZATION

Data normalization, also known as standardization, helps to reduce training time and converge the model faster. In order to achieve data standardization, there are many techniques, such as minimum and maximum scaling, standard scaling and average scaling. In this paper, the minimum and maximum scaling is used to compress the data between [0,1], such as equation 13.

$$X' = \frac{X - \min(X)}{\max(X) - \min(X)} \tag{13}$$

D. EXPERIMENTAL EVALUATION

TP indicates the number of attack traffic detected by the network intrusion detection model, and the detection result is correct. *FN* indicates the number of attack traffic detected, but its detection If the result is wrong, these traffics are actually normal traffic. *TN* indicates the number of normal traffics detected, and the detection result is correct. *FP* indicates the number of normal traffic detected, and the detection result is wrong, the traffic is actually an attack flow [19]. Accuracy represents the proportion of the overall sample that the classifier correctly classifies the sample, and the higher the value, the better the performance of the network intrusion detection model. Calculated as equation 14.

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \tag{14}$$

Recall represents the proportion of the number of samples predicted as positive samples by the classifier to the total positive samples. The higher the value, the better the performance of the network intrusion detection model. Calculated as equation 15.

$$Recall = \frac{TP}{TP + FN} \tag{15}$$

Precision represents the proportion of positive samples in the positive examples classified by the classifier. The higher the value, the better the false positive performance of the network intrusion detection model. Calculated as equation 16.

$$Precision = \frac{TP}{TP + FP} \tag{16}$$

F1-score is the weighted average of Precision and Recall, which is used to integrate the scores of Precision and Recall. The larger the value, the closer the precision and recall are to 1, and the better the detection performance of the network intrusion detection model for normal traffic. Calculated as equation 17.

$$F1\text{-}score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{17}$$

V. EXPERIMENTAL RESULTS AND ANALYSIS

A. MODEL PERFORMANCE ANALYSIS EXPERIMENT

To verify the feasibility of the network intrusion detection method proposed in this paper, the UNSW_NB15 data set and NSL_KDD data will be used to verify the proposed detection method, and the accuracy, precision, recall, and F1-score of the model will be obtained. And compared with the traditional network intrusion detection model. Experiments were carried out on the UNSW_NB15 and NSL_KDD datasets respectively. it was found that with the increase of training rounds, the accuracy of the UNSW_B15 training set continued to increase, and the loss continued to decrease by analyzing Figure 9 and Figure 10. The model reached convergence and the accuracy and loss gradually stabilized with 80 rounds of training. Finally, the trained model is tested against the test set of UNSW_NB15 using the trained model with an accuracy rate is 93.26%, a precision rate is 91.40%, a recall rate is 99.35%, and an F1-score is 95.21%. As shown in table3 of the classification report.

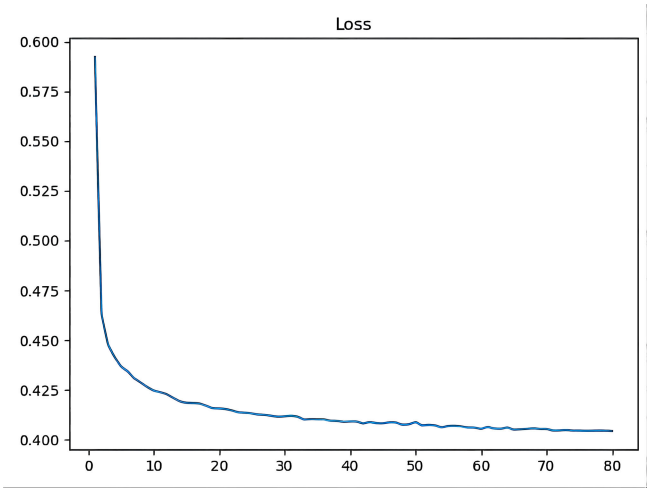

FIGURE 9. UNSW_NB15 scatter plot of the iteration rounds and loss changes of the training set.

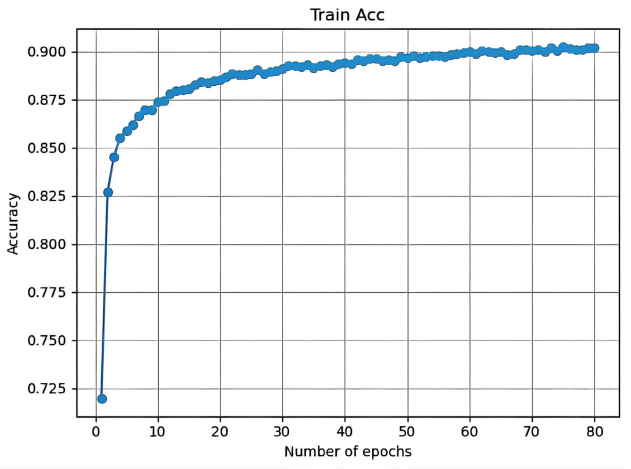

FIGURE 10. UNSW_NB15 scatter plot of the iteration rounds and accuracy changes of the training set.

TABLE 3. UNSW_NB15 test set classification report.

	Precision	Recall	F1-score	Support
Class1	0.98	0.81	0.89	56000
Class2	0.92	0.99	0.95	116032
Accuracy			**0.93**	172032
Macro_avg	0.95	0.90	0.92	172032
Weighted_avg	0.94	0.93	0.93	172032

TABLE 4. NSL_KDD test set classification report.

	Precision	Recall	F1-score	Support
Class1	0.94	0.94	0.94	9837
Class2	0.93	0.93	0.93	8595
Accuracy			**0.93**	18432
Macro_avg	0.93	0.93	0.93	18432
Weighted_avg	0.93	0.93	0.93	18432

It is found that with the increase of training rounds, the accuracy of the NSL_KDD training set continues to rise, and the loss continues to decrease by analyzing Figure 11 and Figure 12. The model reached convergence and the accuracy

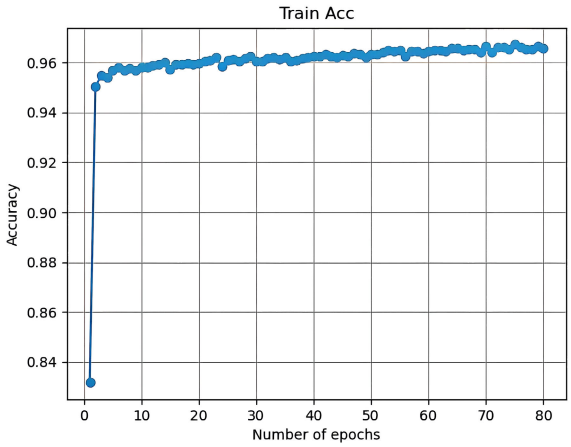

FIGURE 11. NSL_KDD scatter plot of the iteration rounds and accuracy changes of the training set.

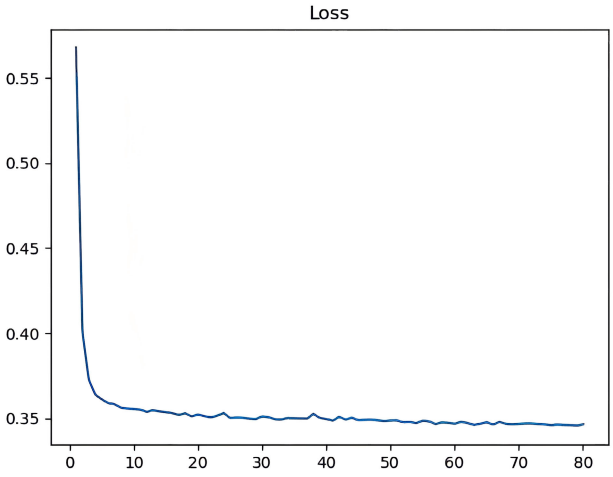

FIGURE 12. NSL_KDD scatter plot of the iteration rounds and loss changes of the training set.

TABLE 5. The experimental results of the proposed method in this paper are compared with traditional classification methods on the UNSW_NB15 test set.

Reference	Method	Accuracy	Precision	Recall
[20]	Naive Bayes	0.79	0.82	0.77
[21]	Decision Tree	0.84	0.85	0.85
[22]	Random forest	0.87	0.86	0.87
The proposed method	**The proposed hybrid Network**	**0.93**	**0.91**	**0.99**

and loss gradually stabilized with 80 rounds of training. Finally, the trained model is tested against the test set of NSL_KDD using the trained model. The accuracy rate is 93.40%, the precision rate is 92.87%, the recall rate is 92.98% and F1 The -score is 92.92%. As shown in table4 of the classification report.

B. COMPARISON EXPERIMENT

1) COMPARISON WITH TRADITIONAL MACHINE LEARNING CLASSIFICATION METHODS

To further verify that the proposed method of network intrusion detection classification in this paper is better than the traditional classification method. A comparison experiment is set up in this paper. The experimental results in table 5 show that on the UNSW_NB15 test set, the proposed method achieves the highest accuracy of 0.93, which is 0.14 higher than that of classification using Bayesian network techniques, 0.09 higher than that of classification using decision trees, and 0.06 higher than that of classification using random forests. In terms of precision rate, the proposed method in this paper is 0.09 higher than the results of classification using Bayesian network technique, 0.06 higher than the results of classification using decision tree, and 0.05 higher than the results of classification using random forest. The proposed method in this paper also achieves the highest recall rate of 0.99 compared to other methods.

TABLE 6. The experimental results of the proposed method in this paper are compared with traditional classification methods on the NSL_KDD test set.

Reference	Method	Accuracy	Precision	Recall
[20]	Decision Tree	0.77	0.66	0.86
[21]	Random forest	0.77	0.65	0.87
[22]	Naive Bayes	0.76	0.65	0.89
The proposed method	**The proposed hybrid Network**	**0.93**	**0.92**	**0.92**

By analyzing the experimental results in table 6, it can be found that the proposed method in this paper achieves the highest accuracy of 0.93 on the NSL_KDD test set, which is 0.17 higher than the results of classification using Bayesian network techniques, 0.16 higher than the results of classification using decision trees, and 0.16 higher than the results of classification using random forests. In terms of precision rate, the proposed method in this paper is 0.27 higher than the results of classification using Bayesian network technique, 0.26 higher than the results of classification using decision tree, and 0.27 higher than the results of classification using random forest. Therefore, the experimental results on UNSW_NB15 and NSL_KDD datasets show that the detection performance of the proposed method is better than several other common machine learning classifier methods.

2) COMPARISON WITH CURRENT ADVANCED CLASSIFICATION METHODS

To further verify the research value of the method proposed in this paper, the experimental results of the method proposed in this paper and the currently known advanced research methods are compared. The experimental results are shown in table 7. By comparing the detection method proposed in this paper with the references [23], [24], [25], and [26], it can be found that the detection method proposed in this paper achieved higher accuracy on the NSL_KDD dataset, 3.39% higher than the accuracy of the detection method proposed in the reference [23], 3.35% higher than the accuracy of the detection method proposed in the reference [24], 19.83% higher than the accuracy of the detection method proposed

TABLE 7. The experimental results of the proposed method in this paper are compared with the currently known advanced research methods.

Reference	Classifier	Dataset	Technique	Results
[23]	DT	NSL_KDD	CFS	Accuracy for NSL_KDD:90.01%
[24]	RepTree	NSL_KDD UNSW_NB15	IG	Accuracy for NSL_KDD:89.85% Accuracy for UNSW_NB15:88.95%
[25]	DT	NSL_KDD UNSW_NB15	CossinMFO	TPR for NSL_KDD:0.891 FPR for NSL_KDD:0.049 Accuracy for NSL_KDD:0.897 F-score for NSL_KDD:0.892 TPR for UNSW_NB15:0.921 FPR for UNSW_NB15:0.030 Accuracy for UNSW_NB15:0.924 F-score for UNSW_NB15:0.942
[26]	Ensemble tree classifier	NSL_KDD	CFS-Bat algorithm	Accuracy for NSL_KDD:73.57% DR for NSL_KDD:73.6% FPR for NSL_KDD:12.92%
[27]	DT	KDD CUP99 UNSW_NB15	GA	DR for KDD CUP99:99.90% DR for UNSW_NB15 :81.24%
[28]	RF	UNSW_NB15	IG	Accuracy for UNSW_NB15:85.78%
The proposed method	The proposed network	NSL_KDD UNSW_NB15	IAE	Accuracy for NSL_KDD: **93.40%** Precession for NSL_KDD:92.87% Recall for NSL_KDD:92.98% F1-score for NSL_KDD:**92.92%** Accuracy for UNSW_NB15: **93.26%** Precession for UNSW_NB15:91.40% Recall for UNSW_NB15:99.35% F1-score for UNSW_NB15:**95.21%**

in the reference [25], 3.70% higher than the accuracy of the detection method proposed in the reference [26], and 19.83% higher than the accuracy of the detection method proposed in the reference [26]. By comparing the experimental results of this paper with the references [24], [25], [27], and [28], it is clear that the detection method proposed in this paper also achieves better detection performance on the UNSW_NB15 dataset. Therefore, the detection method proposed in this paper has a strong research value.

C. ABLATION EXPERIMENT

To further validate the overall performance of the anomalous flow detection method proposed in this paper, an ablation experiment was set up. Under the condition that the experimental conditions such as data pre-processing and training times are guaranteed to be constant, the proposed hybrid network structure is tested for the existence of redundant parts. The experimental results are shown in table8 and table9.

It is found that the detection method proposed in this paper has better detection capability by comparing and analyzing other detection methods, The highest accuracy is achieved on both the UNSW_NB15 dataset and NSL_KDD dataset with 0.93 and 0.93, respectively. the proposed method improves the accuracy of the UNSW_NB15 dataset by 0.04 and of NSL_KDD by 0.13 compared to the BiGRU-only detection method.The proposed method is compared with a hybrid network using a combination of BiGRU and resnet The accuracy of the proposed method is improved by 0.03 on UNSW_NB15 and by 0.05 on NSL_KDD.Therefore, the above analysis reveals that the improved autoencoder, the improved residual convolutional blocks, and the bidirectional gated recurrent units incorporating dual pooling proposed in

TABLE 8. Comparison of UNSW_NB15 ablation experiment.

Method	Accuracy	Precision	Recall
BiGRU	0.89	0.92	0.85
BiGRU_ResNet	0.90	**0.98**	0.96
Proposed method	**0.93**	0.91	**0.99**

TABLE 9. Comparison of NSL_KDD ablation experiment.

Method	Accuracy	Precision	Recall
BiGRU	0.80	0.87	0.78
BiGRU_ResNet	0.88	0.90	0.89
Proposed method	**0.93**	**0.92**	**0.92**

this paper can indeed improve the detection capability of the network intrusion detection model.

VI. CONCLUSION

In the context of the Big Data era, maintaining a stable network environment has become critical. Although researchers have proposed many different methods to deal with cyber threats, none of them have good detection capabilities and either have low detection accuracy or fail to detect all attacks.In addition to this, most of the proposed IDSs use a large number of features with high computational costs.In this work, we summarize previous research work and reduce the number of features in the UNSW_NB15 and NSL_KDD datasets using an improved autoencoder for binary classification.In addition, we propose a new method for detecting network attacks.The method first reduces the number of features of network data by IAE, then extracts spatial features and temporal features in different dimensions using hybrid networks, and finally uses the extracted features to identify whether it is a network attack or not.

The results show that the proposed method is superior to most previous works in terms of accuracy and sensitivity. To study in more detail the ability of hybrid networks to identify network attacks, we used the UNSW_NB15 and NSL_KDD datasets. By analyzing the KDE (kernel density estimation) distribution plots for each feature of the UNSW_NB15 and NSL_KDD datasets, it can be found that the distributions of the training and test sets are inconsistent and the data labels are unbalanced. Since this paper aims to test the accuracy and other metrics of the proposed method with two datasets, the problem of data imbalance is not addressed, but the use of upsampling and downsampling techniques in machine learning can be considered.Currently, deep learning is widely used in intrusion detection, but it faces data privacy issues. We can try to use joint learning to solve the problem of network intrusion detection.

REFERENCES

[1] H. Lin, Q. Xue, J. Feng, and D. Bai, "Internet of Things intrusion detection model and algorithm based on cloud computing and multi-feature extraction extreme learning machine," *Digit. Commun. Netw.*, vol. 9, no. 1, pp. 111–124, Feb. 2023.

[2] L. Xiao, Y. Chen, and C. K. Chang, "Bayesian model averaging of Bayesian network classifiers for intrusion detection," in *Proc. IEEE 38th Int. Comput. Softw. Appl. Conf. Workshops*, Jul. 2014, pp. 128–133.

[3] A. Ahmim, L. Maglaras, M. A. Ferrag, M. Derdour, and H. Janicke, "A novel hierarchical intrusion detection system based on decision tree and rules-based models," in *Proc. 15th Int. Conf. Distrib. Comput. Sensor Syst. (DCOSS)*, May 2019, pp. 228–233.

[4] E. Mushtaq, A. Zameer, M. Umer, and A. A. Abbasi, "A two-stage intrusion detection system with auto-encoder and LSTMs," *Appl. Soft Comput.*, vol. 121, May 2022, Art. no. 108768.

[5] G. Bendiab, S. Shiaeles, A. Alruban, and N. Kolokotronis, "IoT malware network traffic classification using visual representation and deep learning," in *Proc. 6th IEEE Conf. Netw. Softwarization (NetSoft)*, Jun. 2020, pp. 444–449.

[6] H. Zhou, "Mobile malware traffic detection method based on value derivative GRU," *Acta communicati Sinica*, vol. 41, pp. 102–113, 2020.

[7] L. Yang, Z. Li, D. Wang, H. Miao, and Z. Wang, "Software defects prediction based on hybrid particle swarm optimization and sparrow search algorithm," *IEEE Access*, vol. 9, pp. 60865–60879, 2021.

[8] P. Han, W. Wang, Q. Shi, and J. Yue, "A combined online-learning model with K-means clustering and GRU neural networks for trajectory prediction," *Ad Hoc Netw.*, vol. 117, Jun. 2021, Art. no. 102476.

[9] S. Wang, B. Wu, B. Wang, and X. Tong, "Complaint classification using hybrid-attention GRU neural network," in *Proc. Pacific–Asia Conf. Knowl. Discovery Data Mining*. Cham, Switzerland: Springer, 2019, pp. 251–262.

[10] J. Kim and N. Moon, "CNN-GRU-based feature extraction model of multivariate time-series data for regional clustering," in *Advances in Computer Science and Ubiquitous Computing*. Cham, Switzerland: Springer, 2021, pp. 401–405.

[11] J. Sinha and M. Manollas, "Efficient deep CNN-BiLSTM model for network intrusion detection," in *Proc. 3rd Int. Conf. Artif. Intell. Pattern Recognit.*, Jun. 2020, pp. 223–231.

[12] G. Wei, "Research on internet text sentiment classification based on BERT and CNN-BiGRU," in *Proc. 11th Int. Conf. Commun., Circuits Syst. (ICCCAS)*, May 2022, pp. 285–289.

[13] D. She and M. Jia, "A BiGRU method for remaining useful life prediction of machinery," *Measurement*, vol. 167, Jan. 2021, Art. no. 108277.

[14] M. Gokhale, S. K. Mohanty, and A. Ojha, "A stacked autoencoder based gene selection and cancer classification framework," *Biomed. Signal Process. Control*, vol. 78, Sep. 2022, Art. no. 103999.

[15] Y. Li, R. Qian, and K. Li, "Inter-patient arrhythmia classification with improved deep residual convolutional neural network," *Comput. Methods Programs Biomed.*, vol. 214, Feb. 2022, Art. no. 106582.

[16] A. S. Qureshi, A. Khan, N. Shamim, and M. H. Durad, "Intrusion detection using deep sparse auto-encoder and self-taught learning," *Neural Comput. Appl.*, vol. 32, no. 8, pp. 3135–3147, Apr. 2020.

[17] N. Moustafa and J. Slay, "The evaluation of network anomaly detection systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set," *Inf. Secur. J., Global Perspective*, vol. 25, nos. 1–3, pp. 18–31, Apr. 2016.

[18] Z. Lv, H. Ding, L. Wang, and Q. Zou, "A convolutional neural network using dinucleotide one-hot encoder for identifying DNA N6-methyladenine sites in the rice genome," *Neurocomputing*, vol. 422, pp. 214–221, Jan. 2021.

[19] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," *IEEE Access*, vol. 7, pp. 41525–41550, 2019.

[20] N. Sathish and K. Valarmathi, "Detection of intrusion behavior in cloud applications using Pearson's chi-squared distribution and decision tree classifiers," *Pattern Recognit. Lett.*, vol. 162, pp. 15–21, Oct. 2022.

[21] L. Zhu, X. Zhou, and C. Zhang, "Rapid identification of high-quality marine shale gas reservoirs based on the oversampling method and random forest algorithm," *Artif. Intell. Geosci.*, vol. 2, pp. 76–81, Dec. 2021.

[22] S. Manimurugan, "IoT-Fog-cloud model for anomaly detection using improved naïve Bayes and principal component analysis," *J. Ambient Intell. Humanized Comput.*, pp. 1–10, Jan. 2021.

[23] B. Ingre and A. Yadav, "Performance analysis of NSL-KDD dataset using ANN," in *Proc. Int. Conf. Signal Process. Commun. Eng. Syst.*, Jan. 2015, pp. 92–96.

[24] M. Belouch, S. El, and M. Idhammad, "A two-stage classifier approach using RepTree algorithm for network intrusion detection," *Int. J. Adv. Comput. Sci. Appl.*, vol. 8, no. 6, pp. 1–6, 2017.

[25] M. Alazab, R. A. Khurma, A. Awajan, and D. Camacho, "A new intrusion detection system based on Moth–Flame optimizer algorithm," *Exp. Syst. Appl.*, vol. 210, Dec. 2022, Art. no. 118439.

[26] Y. Zhou, G. Cheng, S. Jiang, and M. Dai, "Building an efficient intrusion detection system based on feature selection and ensemble classifier," *Comput. Netw.*, vol. 174, Jun. 2020, Art. no. 107247.

[27] C. Khammassi and S. Krichen, "A GA-LR wrapper approach for feature selection in network intrusion detection," *Comput. Secur.*, vol. 70, pp. 255–277, Sep. 2017.

[28] W. Zong, Y.-W. Chow, and W. Susilo, "A two-stage classifier approach for network intrusion detection," in *Information Security Practice and Experience*. Tokyo, Japan: Springer, Sep. 2018, pp. 329–340.

HONGCHEN YU was born in Zaolin, Bazhou, Bazhong, Sichuan, China, in 1998. He received the B.S. degree in cyberspace security (computer science and engineering) from the Chengdu Neusoft College, in 2021. He is currently pursuing the M.S. degree in cyberspace security with the School of Data Science and Technology, Heilongjiang University.

He published an article in *Sensors*. His research interests include intrusion detection systems, network security, and cyber security defenses. He has won many ACM Arithmetic Competition Awards in China. He received the National Inspirational Scholarship, in 2018, and the National Scholarship in China, in 2019.

CHUNYING KANG was born in Harbin, Heilongjiang, China, in 1976. She received the B.S. degree from the School of Computer Science, Heilongjiang University, and the M.S. degree from the School of Computer Science, Northeastern University.

She is currently a Master's Tutor and a Professor of cyberspace security with the School of Data Science and Technology, Heilongjiang University. She has published many articles and held several invention patents. Her research interests include intrusion detection systems and network defense strategies.

YAO XIAO was born in Zhumadian, Henan, China, in 1997. He received the B.S. degree in physics from Zhengzhou University, in 2020. He is currently pursuing the M.S. degree in cyberspace security with the School of Data Science and Technology, Heilongjiang University. He published many articles. His research interests include network security and cybersecurity defenses.

YUTING YANG was born in Ganzhou, Jiangxi, China, in 2000. She received the B.S. degree in information management and information systems from the Harbin University of Commerce, in 2022. She is currently pursuing the M.S. degree in cyberspace security with the School of Data Science and Technology, Heilongjiang University. Her research interest includes network attack and defense.