IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 12, 2020, accepted October 22, 2020, date of publication October 26, 2020, date of current version November 9, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3034015

A Novel Wireless Network Intrusion Detection Method Based on Adaptive Synthetic Sampling and an Improved Convolutional Neural Network

ZHIQUAN HU[1], LIEJUN WANG[1,2], LEI QI[1], YONGMING LI[1], AND WENZHONG YANG[1]
[1]College of Information Science and Engineering, Xinjiang University, Ürümqi 830046, China
[2]Key Laboratory of Signal Detection and Processing, Xinjiang Uygur Autonomous Region, Xinjiang 830046, China

Corresponding author: Liejun Wang (wljxju@xju.edu.cn)

This work was supported in part by the Xinjiang Uygur Autonomous Region Natural Science Foundation Project under Grant 2020D01C034, in part by the National Natural Science Foundation Project under Grant 61771416 and Grant U1903213, in part by the Xinjiang Uygur Autonomous Region Higher Education Innovation Project under Grant XJEDU2017T002, and in part by the CERNET Innovation Project under Grant NGII20190309.

ABSTRACT The diversity of network attacks poses severe challenges to intrusion detection systems (IDSs). Traditional attack recognition methods usually adopt mining data associations to identify anomalies, which has the disadvantages of a high false alarm rate (FAR), low recognition accuracy (ACC) and poor generalization ability. To ameliorate the comprehensive capabilities of IDS and strengthen network security, we propose a novel intrusion detection method based on the adaptive synthetic sampling (ADASYN) algorithm and an improved convolutional neural network (CNN). First, we use the ADASYN method to balance the sample distribution, which can effectively prevent the model from being sensitive to large samples and ignore small samples. Second, the improved CNN is based on the split convolution module (SPC-CNN), which can increase the diversity of features and eliminate the impact of interchannel information redundancy on model training. Then, an AS-CNN model mixed with ADASYN and SPC-CNN is used for intrusion detection tasks. Finally, the standard NSL-KDD dataset is selected to test AS-CNN. The simulation illustrates that the accuracy is 4.60% and 2.79% higher than that of the traditional CNN and RNN models, and the detection rate (DR) increased by 11.34% and 10.27%, respectively. Additionally, the FAR decreased by 15.58% and 14.57%, respectively, compared with the two models.

INDEX TERMS Intrusion detection, adaptive synthetic sampling, AS-CNN, NSL-KDD.

I. INTRODUCTION

With the rapid popularization of 5G technology, wireless networks are being adopted in more extensive and complex environments. However, due to the open and distributed characteristics of wireless networks, they have become the main attack target [1]; therefore, network security issues have attracted more attention. Network security technologies generally include firewalls, encryption and access authentication technology. However, attack identification mechanisms are undoubtedly the most guaranteed network security measures when these security technologies fail [2]. Network attacks generally include user to root (U2R), denial of service (DoS), remote to local (R2L) and probe attacks. Attack detection means that IDS classifies attacks correctly according to the learned characteristics [3], which can help detect anomalies early and take corresponding preventive measures.

At present, classical machine learning methods are widely used in attack identification tasks, which usually include three steps: data preprocessing, feature selection and classification. Data preprocessing makes classifiers recognize different features better [4], and the most common preprocessing method is feature coding. However, various coding methods make the processed features have different degrees of discretization; generally, the higher the degree of discretization, the better the classification effect of the model. Lin *et al.* [5] compared the influence of different feature coding algorithms on the performance of IDS and proposed a character-level encoding method to mine the association among features. Feature analysis and selection directly affect the detection

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

accuracy and efficiency [6]. Qi *et al.* [7] used the PCA algorithm to select some key features, which significantly improved IDS's efficiency. The optimized linear discriminant analysis (LDA) [8] algorithm and correlation-based characteristic selection (CCS) [9] methods were used to reduce the feature dimension. Wang *et al.* [10] implemented edge density ratio conversion to construct a set of better-quality characteristics. These methods are typical feature reduction and augmentation algorithms. Classifiers are the core of traditional intrusion detection methods. An effective classifier is needed to classify network traffic correctly after data preprocessing. Xiao *et al.* designed an attack identification classifier based on naïve Bayes [11]. In addition, other classifiers include decision trees (DTs) [12], artificial neural networks (ANNs) [13], and support vector machines (SVMs) [14], [15]. Additionally, to synthesize the advantages of classifiers, Peng *et al.* [16] analyzed the detection performance of various classifiers and designed a mixed attack recognition classifier based on decision trees and k-means. Tahir *et al.* [17] presented a hybrid classifier based on SVM and k-means.

The machine learning (ML) algorithms mentioned above depend on complex feature engineering and are difficult to adapt to the growing network environment, while the deep learning (DL) algorithm can autonomously abstract high-level features from basic network traffic without complex feature engineering; therefore, it is widely used in attack recognition tasks. Tan *et al.* [18] obtained a structure-optimal deep belief network (DBN) optimized by the particle swarm optimization (PSO) method, and experiments showed that the accuracy of PSO-DBN on the KDD99 dataset reached 92.44%. Marir *et al.* [19] applied DBN as a feature selector and combined it with an SVM classifier to improve the recognition accuracy of IDS. Yin *et al.* [20] applied a recurrent neural network to an intrusion detection task and achieved a remarkable effect on the NSL-KDD dataset. Compared with the above DL algorithms, CNN reduces the complexity of the model by means of local perception and weight sharing. In addition, the features abstracted by CNN are often superior to traditional feature selection algorithms. Therefore, the accuracy of CNN-based intrusion detection algorithms is generally higher than that of traditional DL algorithms. Generally, the larger the number of training samples, the better the features learned by CNN, and the better-quality features contribute greatly to the classification accuracy of the model. Therefore, CNN-based intrusion detection algorithms are more suitable for today's explosive network environment.

At present, CNN has been successfully applied to various intrusion detection tasks [21], [22], while they ignored the impact of data imbalance and interchannel information redundancy on model training. Wu *et al.* [23] adjusted the weight of loss based on the proportion of samples to solve the problem of data imbalance, where the larger the sample proportion, the greater the weight of the loss function. This loss-based method effectively weakens the impact of sample distribution imbalance on model performance. However, how to strengthen the recognition ability of the model is still the focus of our research. Generally, the more output channels of the convolutional layer, the more diverse the extracted feature maps, and it may create some unnecessary information redundancy. However, these CNN-based intrusion detection algorithms ignore the influence of interchannel feature redundancy on the classification performance of the model.

In summary, there are two problems to be solved in the IDS based on CNN: 1. unbalanced sample distribution; 2. interchannel information redundancy. Therefore, we design a novel attack recognition method based on the ADASYN data augmentation algorithm and an improved CNN. Our main contributions are as follows:

1) We apply the ADASYN data augmentation algorithm to prevent the model from being sensitive to large samples but ignore small samples, which can improve the learning and recognition ability of IDS for small samples.
2) We design a novel CNN model based on channel splitting, which can not only extract multiscale features of data but also effectively solve the problem of interchannel feature redundancy. Finally, different channel features are organically fused through a soft attention operation.
3) An AS-CNN model mixed with the ADASYN algorithm and SPC-CNN is used for intrusion detection tasks. The simulation results illustrate that the comprehensive performance of AS-CNN is superior to that of traditional intrusion recognition models.

The main distribution of this article is as follows. Section 3 introduces the research methods of this article. Then, we provide the rationality analysis and objective evaluation according to the simulation in section 4. Finally, we draw research conclusions and prospects.

II. RELATED WORKS

The development of attack recognition technology has gone through three stages: pattern matching algorithms, machine learning algorithms and deep learning algorithms. The pattern matching algorithm was first applied to intrusion detection tasks based on feature matching. In [24], Wu and Shen analyzed the classical pattern matching algorithms, BM and AC, and proposed the corresponding improved algorithms BMHS and AC-BM; experiments illustrated that the enhanced algorithms greatly optimize the timeliness of IDS. Dagar *et al.* [25] applied RabinKarp and Knuth-MorrisPratt pattern matching algorithms to intrusion detection tasks and compared their execution efficiency. However, these pattern matching algorithms are difficult to adapt to today's network environment due to the diversity of network attacks.

An attack recognition algorithm based on ML has been successfully applied to IDS and achieved excellent performance, which gradually replaced the traditional pattern matching algorithm. SVM is a typical supervised learning model in ML. Thaseen and Kumar [26] presented a novel attack recognition model based on chi-square feature

selection and multi class support vector machine (SVM). The simulation illustrates that removing redundant features significantly improves the calculation accuracy and execution efficiency of the model. Ingre *et al.* [27] established a novel intrusion recognition system by combining the relevant feature screening algorithm with decision trees. Nancy *et al.* [28] designed a dynamic recursive feature selection algorithm. By extending the decision tree algorithm and combining it with convolutional neural networks. They proposed an intelligent fuzzy temporal decision tree algorithm. The new algorithm achieved a high detection rate of unknown attacks on the KDD cup dataset. An improved IDS based on a Bayesian network and feature selection algorithm was proposed in [29]. Although these ML detection algorithms achieved higher recognition accuracy in intrusion detection tasks, they not only need large-scale feature engineering but the model parameters are also difficult to adjust. However, the DL algorithm can autonomously abstract features from basic network traffic without complex feature engineering. Therefore, related research on intrusion detection is gradually focused on the DL method.

An LSTM classifier with a gradient descent optimizer is used in IDS [30], which can effectively mine the association between features from the perspective of time. Su *et al.* [31] combined an attention mechanism and BLSTM (bidirectional long short-term memory) to propose a network anomaly detection model BAT, which extracts coarse-grained features by connecting forward LSTM and backward LSTM. The BAT model uses an attention mechanism to filter the network flow vectors generated by the BLSTM model to obtain the key characteristics of network traffic classification. Wei *et al.* [32] applied particle swarm optimization (PSO) to optimize the structure of DBN, and the improved DBN achieves significant anomaly detection ability.Gao *et al.* [33] designed an effective attack recognition method by combining association rules and improved deep neural networks (DNN), which uses the apriori algorithm to mine the association between discrete features and labels to improve the recognition accuracy. Yin *et al.* [20] presented an effective attack recognition model by using feature enhancement and improved RNN; however, the feature enhancement method also increases the computational complexity of the model. CNN has been successfully applied to intrusion detection tasks because it can extract network traffic characteristics more effectively [34]. Lin *et al.* [5] designed a character-level CL-CNN model. The character-based encoding method makes the features more discretized, which contributes to improving the detection accuracy of IDS. Wu *et al.* [23] designed a CNN model with a simple structure and proved the necessity of converting the original data into a 2D format through experiments. In addition, the combination of this simple CNN and 2D data conversion greatly improves the detection efficiency of the model compared with the RNN model in [20]. Ding and Zhai [35] proposed a convolutional neural network model (MS-CNN) based on multistage features. Multistage features are obtained by connecting the outputs of all convolutional layers to the dense layer with the softmax classifier. By adding supplementary information (such as local information and detailed information lost by higher-level convolutional layers), the expressive ability of the model significantly improves. Yang and Wang [36] extracted diverse features through a cross-layer aggregated CNN model, which greatly improved the expression ability of the model.

Although the above attack recognition algorithms using CNN improve the detection accuracy, they ignore the interchannel information redundancy in the convolution layer. However, we cannot directly discard some channel information because we are not sure which channels are redundant. To reasonably eliminate the problem of interchannel information redundancy, Zhang *et al.* [37] proposed a split-based plug and play convolution (SPC) block, which divides the channel of the convolution layer into two parts, the representative part and the uncertain redundant part, after which hierarchical processing is performed. Inspired by this idea, we propose an SPC-equipped CNN (SPC-CNN) and apply it to attack recognition tasks. The simulation illustrates that the comprehensive performance of SPC-CNN has obvious advantages compared with the traditional CNN model. Before that, the ADASYN data augmentation algorithm is used to prevent the model from being sensitive to large samples and ignores small samples. Then, the AS-CNN model mixed with the ADASYN algorithm and SPC-CNN is used for intrusion detection tasks. The simulation illustrates that the comprehensive performance of the hybrid AS-CNN model is better than that of using SPC-CNN alone.

III. PROPOSED METHODS

A. RESEARCH IDEA

Figure 1 shows the framework of the improved IDS, which mainly consists of the following four parts:

Part 1: Data preprocessing. This part consists of three steps: numerical, normalization and ADASYN data augmentation. This part transforms the original data format, weaken the measurement difference between features and balances the sample distribution, which greatly promote the comprehensive performance of the CNN model.

Part 2: Design network. In this part, to extract multiscale features and reasonably solve the problem of interchannel information redundancy, we propose the SPC-CNN model, which includes the introduction of improved methods and basic data processing processes.

Part 3: Model training and testing. Perform training operations and continuously adjust the network parameters to make the model converge. In addition, a test experiment is carried out after each training cycle, and we decide whether to output the model according to the test results.

Part 4: Evaluation. This section illustrates the feasibility and effectiveness of the novel model. Finally, the advantages and disadvantages of the optimization method are evaluated objectively according to FAR, ACC and DR.

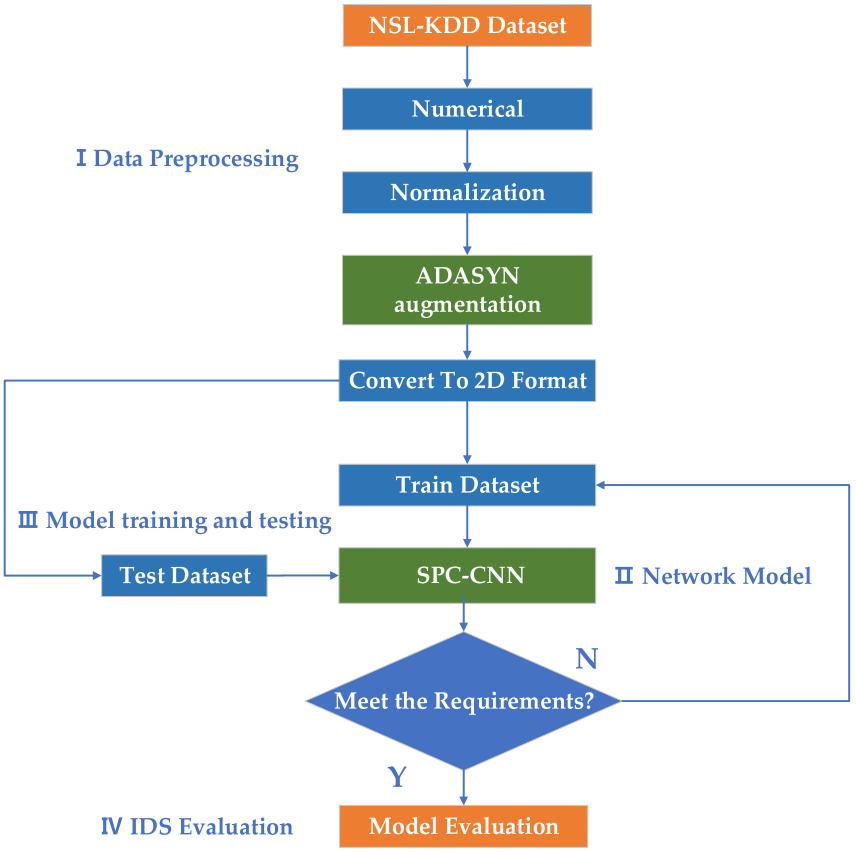

FIGURE 1. The flowchart of AS-CNN model. (AS-CNN is composed of ADASYN and SPC-CNN).

B. DATASET ANALYSIS AND PREPROCESSING

The NSL-KDD dataset has been widely used as a benchmark to evaluate the recognition capability of IDS [38]. The NSL-KDD dataset contains three sub-datasets: KDDTrain, KDDTest-21 and KDDTest+. Every record in the dataset is a connection vector that contains 1 label and 41 characteristics, which contain 38 digital characteristics and 3 symbolic characteristics. Table 1 illustrates the specific distribution of KDDTrain categories. The NSL-KDD dataset contains 5 types of samples (Dos, probe, R2L, U2R, Normal). The 4 abnormal samples can be subdivided into 39 attack types, of which 22 attack types appear in the training dataset and 17 unknown attack types appear in the test dataset. The purpose of this division is mainly to test the generalization ability of the model. The KDDTest-21 and KDDTest+ contain different abnormal samples. Therefore, the KDDTest-21 dataset is often used as a supplementary benchmark for the KDDTest+ dataset to evaluate the generalization ability of the model.

TABLE 1. Category distribution of the KDDTrain dataset.

Category	Number	Ratio
Dos	45927	36.46%
Probe	11656	9.52%
R2L	995	0.79%
U2R	52	0.04%
Normal	67347	53.46%
Total	**125973**	**1.00**

1) NUMERICAL

The original dataset contains three types of symbolic features, including 3 protocol types, 70 service types and 11 connection states. However, the input of SPC-CNN is a standard 2D digital matrix. Therefore, we apply a one-hot encoder to digitize the symbolic features, which helps the model to recognize the features better. For instance, the three symbolic protocol features ICMP, TCP and UDP are encoded as (0, 0, 1), (0, 1, 0), and (1, 0, 0), respectively, as shown in Table 2. Finally, the original 42 features are mapped into 122-dimensional vectors.

2) NORMALIZATION

The value range of 122 features of the NSL-KDD dataset fluctuates greatly, for example, feature Src_Bytes $\in$ [0, 1379963888] while srv_serror_rate $\in$ [0, 1], which will

TABLE 2. One-hot encoding.

Attribute	Encoding	Bit
3 protocol types	(0.1.0)	3
70 types of network services	(0……1……0)	70
11 types of link states	(1…0…0)	11

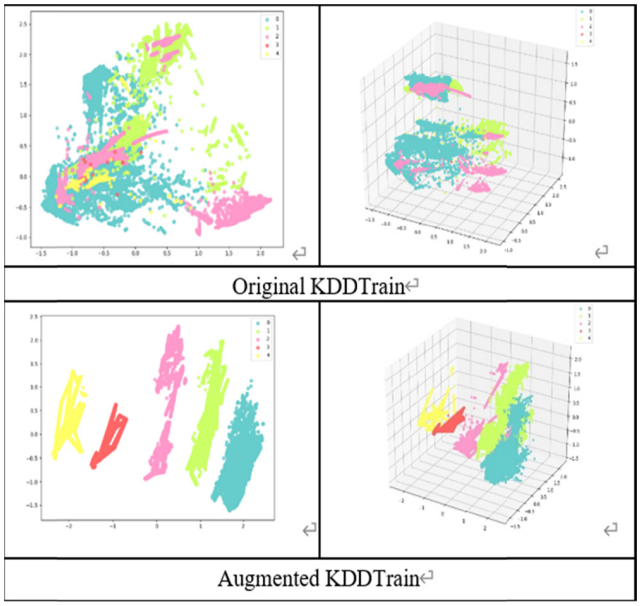

FIGURE 2. Dataset distribution.

seriously affect the reliability of training results. To eliminate the influence of attribute measurement difference on model training, we use the max-min normalization function to normalize the attribute values. The specific formula is as follows:

$$X_n = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (1)$$

where X, X_n represents the original value and normalized value respectively. X_{max}, X_{min} represents the maximum and minimum values of the corresponding feature.

3) DATA AUGMENTATION

Table 1 shows that the sample distribution of the NSL-KDD dataset is extremely unbalanced, in which normal and DoS account for 53.46% and 36.46%, respectively, while R2L and U2R account for less than 1%. Obviously, the sample distribution imbalance will lead to the model being biased towards frequent samples and ignoring the small samples. Therefore, we use the ADASYN algorithm to augment small samples, which will effectively weaken the interference of extremely unbalanced sample distribution on model training. The ADASYN algorithm is described as follows:

Figure 2 shows the distribution of the KDDTrain dataset before and after sampling. Obviously, the augmented data have better comprehensive discrete characteristics.

ADASYN Algorithm

Input: The training set contains m samples $\{(x_1, y_1), (x_2, y_2) \ldots (x_i, y_j)\}$, $i = 1, 2, \ldots, m$. Where x_i represents an n-dimensional eigenvector and $x_i = \{x_{i1}, x_{i2}, x_{i3} \ldots x_{in}\}$, y_j represents the sample label and $y_j \in \{0, 1, 2, 3, 4\}$, $y_j = \{3, 4\}$ and $y_j = \{0, 1, 2\}$ represent a large sample and a small sample, respectively. In addition, we use N_l and N_s to represent a large sample size and a small sample size, respectively, where $N_l + N_s = m$.

Step 1: Calculate the degree of imbalance between samples: $d = N_s/N_l$.
IF $d < d_{th}$ *(where d_{th} represents the imbalance threshold)*

Step 2: Calculate the number of small samples to be synthesized: $Q = \beta(N_l - N_s)$, where the parameter β represents the unbalance degree of the new sample and $\beta \in [0, 1]$.

Step 3: For every small sample x_i in each minority class. We calculate the ratio R_i according to its k-nearest neighbor in n-dimensional space: $R_i = \frac{\gamma_i}{k}$, where $R_i \in (0, 1]$ and γ_i represents the number of k-nearest neighbors of sample x_i belonging to large categories.

Step4: Normalize R according to the formula:

$$R_i' = R_i / \sum_1^{N_s} R_i$$

Step5: Calculate the number of samples to be generated for each x_i according to the formula: $g_i = R_i' * G$, where G represents the total number of samples to be synthesized.

Step 6: Execution loop
For $i < g_i$

(1) Randomly select a small sample x_{zi} from the K neighbors of x_i

(2) Synthesize new samples according to the formula:

$$L_i = x_i + \lambda(x_{zi} - x_i)$$

where λ represents a random number and $\lambda \in [0, 1]$

End

Output: New training dataset

C. REDESIGN OF CONVOLUTIONAL NEURAL NETWORK

The improved CNN is shown in Figure 3. First, the original $k \times k$ matrix is convolved by Conv1 to obtain L feature maps, and the visualization results are shown in Figure 4. It can be seen that there are many similarities between these feature maps; however, these patterns are not exactly the same, which indicates that there is a certain degree of

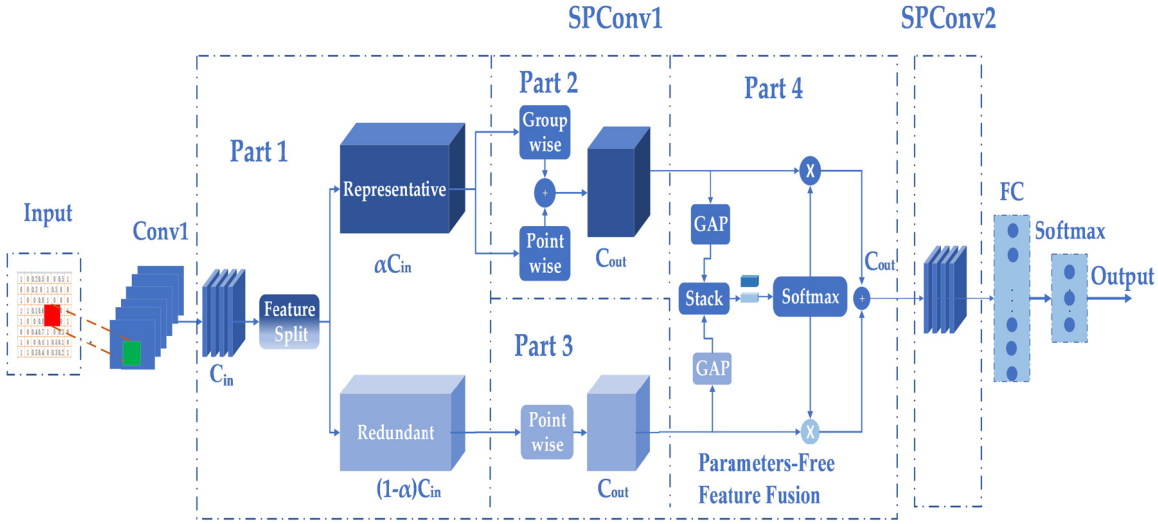

FIGURE 3. The improved convolutional neural network.

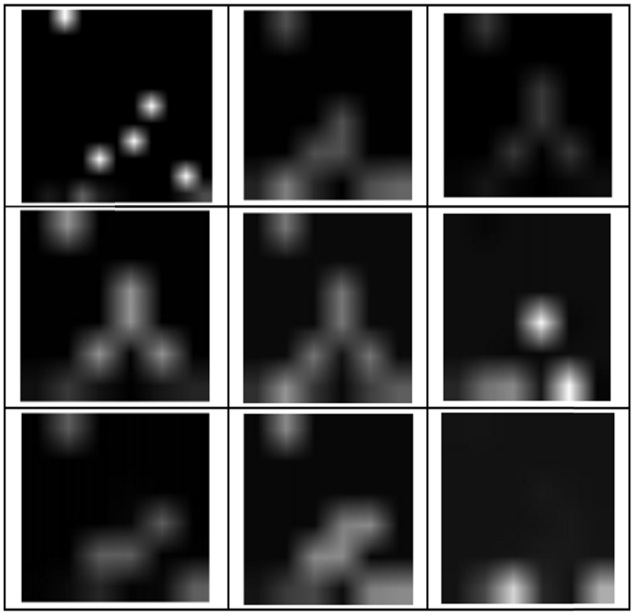

FIGURE 4. Visualization of Conv1 output feature maps.

interchannel information redundancy. However, the traditional attack recognition model based on CNN ignores the redundancy in channels, which directly convolves all feature maps in the next convolutional layer without any feature analysis. Obviously, the redundancy on these channels will reduce the detection performance and execution efficiency of the network. However, these similar features cannot be directly removed because it is difficult to determine whether the similar pattern features contain different details.

To extract multiscale features and reduce the interchannel information redundancy more reasonably, we propose an improved CNN based on the SPConv module. As shown in Figure 3, SPC-CNN consists of seven layers: an input layer, a convolution layer, 2 SPConv modules, a fully connected layer, a softmax layer and an output layer. There are four parts in the SPConv module: a channel splitting block, 2 convolution blocks and a feature fusion block. The basic process of SPC-CNN is as follows.

1) In Part 1, L feature maps are outputted by conv1, which are divided into representative parts and redundant parts by the channel splitting module.

2) Part 2 and Part 3 of the SPConv module adopt different levels of feature extraction for the representative part and redundant part, respectively.

3) In part 4, a soft attention module is used to integrate the features from different channels, which can make the use of the features from different channels more reasonable.

4) We calculate the loss value according to the output of the Softmax layer and optimize the network parameters by error backpropagation to make the model converge.

D. MODEL TRAINING

Convert training samples into standard image format as SPC-CNN specific input. Conv1 outputs L feature maps under the activation of the ReLU function and the transmits them to the SPC module.

1) FEATURE SPLITTING

Assume X ∈ R^(L*h*w) and Y ∈ R^(M*h*w) represent the input and output tensors of the SPConv module, respectively. L input channels are divided into two parts by the SPConv module according to $\boldsymbol{\alpha} : (\mathbf{1} - \boldsymbol{\alpha})$ (representative part and redundant part). The 3×3 convolution kernel is used to extract the intrinsic information for the representative part. Complementarily, the more economical 1×1 convolution operation is used to extract the details from the redundant part. Therefore, the convolution operation of the SPConv

module is shown in formula (2):

$$\begin{bmatrix} o_1 \\ o_2 \\ \vdots \\ o_M \end{bmatrix} = \begin{bmatrix} W_{11} & W_{12} & \cdots & W_{1,\alpha L} \\ W_{21} & W_{22} & \cdots & W_{2,\alpha L} \\ \vdots & \vdots & \ddots & \vdots \\ W_{M1} & W_{M2} & \cdots & W_{M,\alpha L} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{,\alpha L} \end{bmatrix} + \begin{bmatrix} w_{1,\alpha L+1} & w_{1,\alpha L+2} & \cdots & w_{1,L} \\ w_{2,\alpha L+1} & w_{2,\alpha L+2} & \cdots & w_{2,L} \\ \vdots & \vdots & \ddots & \vdots \\ w_{M,\alpha L+1} & w_{M,\alpha L+2} & \cdots & w_{M,L} \end{bmatrix} \begin{bmatrix} x_{\alpha L+1} \\ x_{\alpha L+2} \\ \vdots \\ x_L \end{bmatrix} \tag{2}$$

where $w_{ij}, j \in [\alpha L + 1, L]$ denotes the weight parameter of economical 1×1 pointwise convolution kernels on $(1-\alpha)L$ redundant channels. $W_{ij}, j \in [1, \alpha L]$ denotes the weight parameter of αL representative channels.

2) FURTHER REDUCTION FOR THE REPRESENTATIVE PART

Although the original input channels are divided into representative parts and redundant parts, the representative part may still have feature redundancy. Therefore, groupwise convolution (GWC) is applied to representative channels to further reduce redundancy in channels. The GWC is equivalent to a vanilla convolution composed of sparse block diagonal convolution kernels, where each block corresponds to a part of the channel. However, group convolution will inevitably lose some information because these blocks are independent. To compensate for the loss of information caused by group convolution, pointwise convolution (PWC) is added in part 2 in parallel. Finally, the features extracted by GWC and PWC are summed directly, which means that the combination of GWC and PWC can ensure that the features extracted from the representative channel part are neither redundant nor incomplete.

$$\begin{bmatrix} W^P_{11} & 0 & 0 \\ 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & 0 & W^P_{GG} \end{bmatrix} \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_G \end{bmatrix} + \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1,\alpha L} \\ w_{21} & w_{22} & \cdots & w_{2,\alpha L} \\ \vdots & \vdots & \ddots & \vdots \\ w_{M1} & w_{M2} & \cdots & w_{M,\alpha L} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{,\alpha L} \end{bmatrix} \tag{3}$$

Therefore, the representative part of formula (2) can be expressed as formula (3). where G represents the number of groups in GWC and each group contains g channels. W^P_{vv} represents the weight parameter of the GWC kernel in group v.

3) FEATURE FUSION

SPC-CNN provides a nonparametric feature fusion module to better fuse features from different channels. As shown in Part 4 of Figure 3, GAP (global average pooling) is used to generate channelwise statistics (CS), $CS_1, CS_3 \in R^C$, and then CS is output by compressing the spatial dimension. The c-th parameter of CS is as follows:

$$CS_{kc} = F_{gap}(T_{kc}) = \frac{1}{H * W} \sum_{i=1}^{H} \sum_{j-1}^{W} T_{kc}(i, j), \quad k \in [1, 3] \tag{4}$$

The channel statistics CS_1, CS_3 are stacked together and processed by softmax to output the important vector $\beta, \gamma \in R^c$. Its c-th parameter is as follows:

$$\gamma_c = \frac{e^{S_{1c}}}{e^{S_{3c}} + e^{S_{1c}}}, \quad \beta_c + \gamma_c = 1 \tag{5}$$

The redundant part and the representative part are organically fused according to the important parameters β and γ, and the comprehensive feature O is obtained by cross-channel fusion.

$$O = \gamma T_1 + \beta T_3 \tag{6}$$

In summary, the output of the SPConv1 module is shown in formula (7).

$$\begin{aligned} O &= W'X \\ &\approx \beta \begin{bmatrix} W^P_{11} & 0 & 0 \\ 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & 0 & W^P_{GG} \end{bmatrix} \begin{bmatrix} o_1 \\ o_2 \\ \vdots \\ o_G \end{bmatrix} \\ &\quad + \beta \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1,\alpha L} \\ w_{21} & w_{22} & \cdots & w_{2,\alpha L} \\ \vdots & \vdots & \ddots & \vdots \\ w_{M1} & w_{M2} & \cdots & w_{M,\alpha L} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{,\alpha L} \end{bmatrix} \\ &\quad + \gamma \begin{bmatrix} w_{1,\alpha L+1} & w_{1,\alpha L+2} & \cdots & w_{1,L} \\ w_{2,\alpha L+1} & w_{2,\alpha L+2} & \cdots & w_{2,L} \\ \vdots & \vdots & \ddots & \vdots \\ w_{M,\alpha L+1} & w_{M,\alpha L+2} & \cdots & w_{M,L} \end{bmatrix} \begin{bmatrix} x_{\alpha L+1} \\ x_{\alpha L+2} \\ \vdots \\ x_L \end{bmatrix} \end{aligned} \tag{7}$$

IV. SIMULATION AND ANALYSIS

A. ENVIRONMENT AND PARAMETER SETTING

The detailed configuration information is shown in Table 3. The learning rate of the network is set at 0.1, and dropout is 0.5 to achieve the best detection effect according to multiple experiments. Additionally, the epoch and batch size are set to 200 and 50, respectively. Table 4 shows the hyperparameters of the SPC-CNN model.

TABLE 3. Configuration information.

Item	Configuration
OS	Ubuntu16.04 (X64)
GPU	4* GTX 1080Ti
Python	3.6

TABLE 4. Parameter setting of SPC-CNN.

Layer	Attribute	Size	Strides	Active Function
L_1	$Conv_1$	2*2*8	1	Relu
L_{21}	GWC_1	3*3*16	1	Relu
L_{22}	PWC_1	1*1*16	1	Relu
L_{23}	PWC_2	1*1*16	1	Relu
L_{31}	GWC_2	3*3*32	1	Relu
L_{32}	PWC_3	1*1*32	1	Relu
L_{33}	PWC_4	1*1*32	1	Relu
-	GAP	2*2	1	Relu
L_4	FC	-	-	Dropout
L_5				Softmax

TABLE 5. The relationship between metrics.

Label	Normal	Attack
Normal	TN	FP
Attack	FN	TP

B. EVALUATION CRITERIA

We select three criteria of ACC, FAR and DR to evaluate the performance of IDS [21]. The related parameters: TP (true positive) and TN (true negative) represent the number of attacks and normal samples correctly classified, respectively, while FP (false positive) and FN (false negative) represent the number of misclassifications. A summary of the metrics is as follows.

$$\mathrm{ACC} = \frac{TN + TP}{TN + FN + TP + FP} \tag{8}$$

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

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

where ACC indicates the ratio of the samples correctly identified by IDS to the total. DR represents the ratio of attack samples identified by IDS to total abnormal samples, which indicates the recognition degree of attack samples. FAR reflects an error level of IDS in distinguishing normal samples. Therefore, the reliability of the intrusion detection system pursues higher ACC and DR and lower FAR.

C. SIMULATION

The standard KDDTest+ and KDDTest-21 datasets are selected to test the validity of the SPC-CNN model in this article. Then, the three evaluation criteria, ACC, DR and FAR, are calculated according to the output five-dimensional confusion matrix. The simulation results are shown in Table 6.

Table 6 shows the test results of standard datasets on SPC-CNN and traditional CNN model. Compared with the traditional CNN model in [23], the ACC and DR of the improved model on KDDTest+ are increased by 4.35% and 5.95%, respectively, and FAR is reduced by 5.49%. In addition, the results in Table 6 show that the detection performance of SPC-CNN on the KDDTest-21 dataset also significantly improved compared to traditional CNN. Our SPC-CNN can effectively solve the inter-channels information redundancy and improve the diversity of features, which contributes greatly to strengthening the recognition performance and generalization ability of the model.

TABLE 6. Simulation of SPC-CNN and traditional CNN.

Dataset	Model	ACC(%)	DR(%)	FAR(%)
KDDTest+	CNN	79.48	68.66	27.90
	SPC-CNN	**83.83**	**74.61**	**22.41**
KDDTest-21	CNN	60.71	58.47	71.88
	SPC-CNN	**69.42**	**66.44**	**60.17**

To illustrate the synergistic contribution of the ADASYN data augmentation operation and SPC-CNN model on IDS, the augmented KDDTrain dataset is used to train the SPC-CNN and save an optimal model. Then, the KDDTest+ and KDDTest-21 datasets are applied to test the hybrid AS-CNN model. The simulation results are shown in Table 7. Obviously, the comprehensive performance of hybrid AS-CNN is better than that of using SPC-CNN alone, which is shown in Table 6. The DR of the AS-CNN model on the KDDTest+ and KDDTest-21 datasets increased by 5.39% and 7.21%, respectively, and the FAR decreased by 10.09% and 14.51%, respectively. We can conclude that the ADASYN algorithm balances the sample distribution and discretizes the categories, which greatly improves the classification performance of IDS.

TABLE 7. Simulation of AS-CNN.

Dataset/Criteria	ACC	DR	FAR
KDDTest+	84.08	80.00	12.32
KDDTest-21	72.54	73.65	45.66

Table 8 and Table 9 show the confusion matrix obtained by testing the AS-CNN model with standard KDDTest+ and KDDTest-21 datasets, where the figures on the main diagonal represent the number of correctly identified samples. Then, the evaluation indicators ACC, DR and FAR of the AS-CNN model are calculated according to the confusion matrix. As shown in Figure 5 and Table 10.

Many classic intrusion detection algorithms have been used in intrusion detection tasks. These include Bayesian [11]; NBTree [12]; SVM [14]; CL-CNN [5]; RNN [20]; CNN [23] and MS-CNN [35]. Figure 5 illustrates the ACC of multiple

TABLE 8. KDDTest+ classification confusion matrix of AS-CNN model.

Actual \ Predicted	Normal	Probe	Dos	U2R	R2L
Normal	**8690**	314	209	52	446
Probe	68	**1891**	304	10	148
Dos	552	112	**6192**	32	570
U2R	95	18	0	**34**	53
R2L	506	14	9	77	**2148**

TABLE 9. KDDTest-21 classification confusion matrix of AS-CNN model.

Actual \ Predicted	Normal	Probe	Dos	U2R	R2L
Normal	**1453**	254	145	41	259
Probe	68	**1876**	300	10	148
Dos	552	111	**3085**	32	562
U2R	95	18	0	**34**	53
R2L	506	14	9	77	**2148**

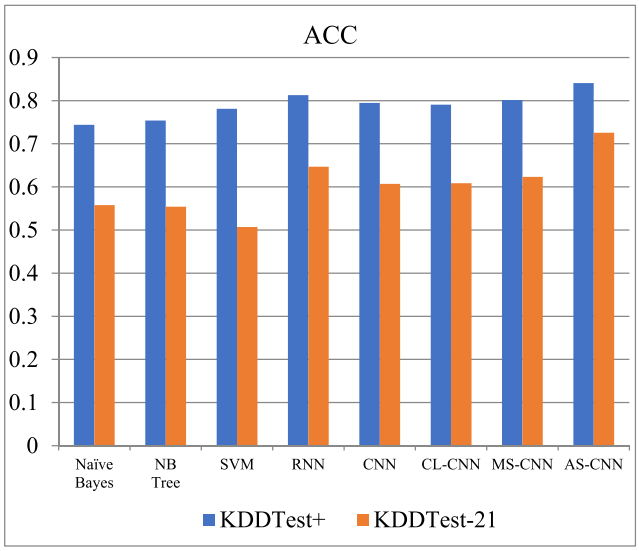

FIGURE 5. ACC of multiple models.

models in the same KDDTest+ and KDDTest-21 datasets. Obviously, the attack recognition ACC based on DL methods is generally better than the traditional ML methods. The detection performance of the traditional 1D CL-CNN model and 2D CNN model is equivalent, however, the CL-CNN based on feature coding increases the feature dimension and uses 1D convolution, which is not as efficient as the 2D

TABLE 10. DR and FAR for multiple models.

Criteria/ \ Model	RNN	CNN	CL-CNN	AS-CNN
DR(%)	69.73	68.66	68.56	**80.00**
FAR(%)	26.89	27.90	25.10	**12.32**

CNN model, and their common feature is that the CNN structure is relatively simple. The structure of the MS-CNN model is different from that of CNN and CL-CNN, which obtains the multistage features of network traffic by changing the connection mode of the convolutional layer. Generally, the diversified features contribute to the characterization ability of CNN model for intrusion samples, so the recognition performance of the classifier has a certain improvement. Different from the MS-CNN model in [35], the AS-CNN model based on channel splitting proposed in this article can not only obtain diversified features but also effectively eliminate the feature redundancy between channels, and then adopt a soft attention mechanism to rationally use features of different levels. Therefore, the comprehensive performance of AS-CNN significantly improved compared with traditional CNN and MS-CNN. Compared with the classic RNN and MS-CNN model, the ACC of AS-CNN on the KDDTest+ dataset increased by 2.79% and 3.95%, 7.87% and 10.22% on the KDDTest-21 dataset, respectively. In addition, FAR and DR are selected as the core indicators to evaluate the comprehensive ability of multiple intrusion recognition models, as shown in Table 10. Obviously, AS-CNN achieves better performance in both DR and FAR. Additionally, the DR of AS-CNN is increased by 11.34%, and the FAR is reduced by 15.58% compared with the CNN model. In summary, the comprehensive performance and generalization ability of AS-CNN proposed in this article have been significantly improved compared with other DL models.

The DR of each attack sample is also an important criterion for evaluating the IDS. Figure 6 shows the detection rate of attack categories by multiple models. Traditional CNN uses a weight-loss method to solve the unbalanced sample distribution, so the model has significant DR for small samples. The DR of CL-CNN model to Probe and U2R is better than that of traditional CNN model because character-level encoding is used in the model to discretize features. The CL-CNN model based on 1D convolution has better DR for Probe and U2R than the traditional CNN model because character-level encoding is used in the model to discretize features. The cross-layer aggregated structure increases the diversity of extracted features in the MS-CNN model, which increases the model's ability to recognize Probe and U2R compared to CNN and CL-CNN. Different from the traditional CNN model, we use the ADASYN data augmentation algorithm to eliminate the interference of the unbalanced sample distribution on model training. In addition, the SPC-CNN model based on channel splitting, hierarchical processing and soft

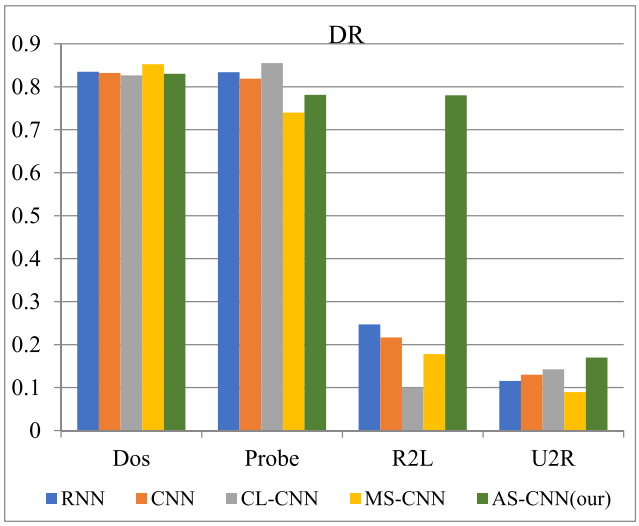

FIGURE 6. DR of multiple models to different attacks.

attention feature fusion effectively solves the problem of feature redundancy and multi-scale feature fusion between channels which are ignored in MS-CNN model. The AS-CNN model that combines the characteristics of ADASYN and SPC-CNN is used for intrusion detection tasks, therefore, the AS-CNN proposed in this article achieves good DR for the other three categories except for probe, where the detection of R2L and U2R samples has always been difficult in the research field. Particularly, Figure 6 shows that the DRs of R2L and U2R improved significantly by the hybrid AS-CNN model.

V. CONCLUSION

We aim to address the problems of unbalanced data distribution and interchannel information redundancy that are ignored by existing CNN-based intrusion detection algorithms. We apply the ADASYN method to balance the sample distribution, which can effectively prevent the model from being sensitive to large samples and ignore small samples. In addition, the split-based SPC-CNN model we designed can make three major contributions: 1) multiscale features can be extracted through different levels of convolution operations; 2) the problem of interchannel redundancy can be effectively mitigated by this complementary processing of information; and 3) a soft attention operation makes the SPC-CNN model use multiscale features more reasonably, which greatly contributes to the expression ability of the model. Finally, an AS-CNN model mixed with ADASYN and SPC-CNN is used for intrusion detection tasks. The simulation results illustrate that the hybrid AS-CNN model achieves significant improvement on the three criteria in multiclassification tasks. However, there is still much work to optimize the recognition accuracy of small samples and the execution efficiency. We intend to strengthen the identification ability of IDS through a simplified residual network.

REFERENCES

[1] Y. Zou, J. Zhu, X. Wang, and L. Hanzo, "A survey on wireless security: Technical challenges, recent advances, and future trends," *Proc. IEEE*, vol. 104, no. 9, pp. 1727–1765, Sep. 2016, doi: 10.1109/JPROC.2016.2558521.

[2] A. Kavianpour and M. C. Anderson, "An overview of wireless network security," in *Proc. IEEE 4th Int. Conf. Cyber Secur. Cloud Comput. (CSCloud)*, New York, NY, USA, Jun. 2017, pp. 306–309, doi: 10.1109/CSCloud.2017.45.

[3] H. Sallay and S. Bourouis, "Intrusion detection alert management for high-speed networks: Current researches and applications," *Secur. Commun. Netw.*, vol. 8, no. 18, pp. 4362–4372, Dec. 2015.

[4] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 1153–1176, 2nd Quart., 2016, doi: 10.1109/COMST.2015.2494502.

[5] S. Z. Lin, Y. Shi, and Z. Xue, "Character-level intrusion detection based on convolutional neural networks," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Rio de Janeiro, Brazil, Jul. 2018, pp. 1–8, doi: 10.1109/IJCNN.2018.8488987.

[6] B. Selvakumar and K. Muneeswaran, "Firefly algorithm based feature selection for network intrusion detection," *Comput. Secur.*, vol. 81, pp. 148–155, Mar. 2019.

[7] Y. Qi, M. Liu, and Y. Fu, "Research on SVM network intrusion detection based on PCA," *Inf. Netw. Secur.*, vol. 2, pp. 15–18, Feb. 2015.

[8] D. Zheng, Z. Hong, N. Wang, and P. Chen, "An improved LDA-based ELM classification for intrusion detection algorithm in IoT application," *Sensors*, vol. 20, no. 6, p. 1706, Mar. 2020.

[9] C. Iwendi, S. Khan, J. H. Anajemba, M. Mittal, M. Alenezi, and M. Alazab, "The use of ensemble models for multiple class and binary class classification for improving intrusion detection systems," *Sensors*, vol. 20, no. 9, p. 2559, Apr. 2020.

[10] H. Wang, J. Gu, and S. Wang, "An effective intrusion detection framework based on SVM with feature augmentation," *Knowl.-Based Syst.*, vol. 136, pp. 130–139, Nov. 2017.

[11] 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*, Västerås, Sweden, Jul. 2014, pp. 128–133, doi: 10.1109/COMPSACW.2014.25.

[12] A. S. Eesa, Z. Orman, and A. M. A. Brifcani, "A novel feature-selection approach based on the cuttlefish optimization algorithm for intrusion detection systems," *Expert Syst. Appl.*, vol. 42, no. 5, pp. 2670–2679, Apr. 2015.

[13] S. S. Roy, A. Mallik, R. Gulati, M. S. Obaidat, and P. V. Krishna, "A deep learning based artificial neural network approach for intrusion detection," in *Proc. Int. Conf. Math. Comput. (ICMC)*, Haldia, India, Jan. 2017, pp. 44–53.

[14] S. M. H. Bamakan, H. Wang, T. Yingjie, and Y. Shi, "An effective intrusion detection framework based on MCLP/SVM optimized by time-varying chaos particle swarm optimization," *Neurocomputing*, vol. 199, pp. 90–102, Jul. 2016.

[15] R. R. Reddy, Y. Ramadevi, and K. V. N. Sunitha, "Effective discriminant function for intrusion detection using SVM," in *Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI)*, Jaipur, India, Sep. 2016, pp. 1148–1153, doi: 10.1109/ICACCI.2016.7732199.

[16] L. I. Peng and Z. Wen-Huan, "Mixed intrusion detection algorithm based on k-means and decision tree," *Comput. Modernization*, pp. 12–16, Dec. 2017.

[17] H. M. Tahir, W. Hasan, A. Said, N. H. Zakaria, N. Katuk, N. F. Kabir, M. H. Omar, O. Ghazali, and N. I. Yahya, "Hybrid machine learning technique for intrusion detection system," in *Proc. 5th Int. Conf. Comput. Informat. (ICOCI)*, Istanbul, Turkey, Aug. 2015, pp. 2289–3784.

[18] X. Tan, S. Su, Z. Zuo, X. Guo, and X. Sun, "Intrusion detection of UAVs based on the deep belief network optimized by PSO," *Sensors*, vol. 19, no. 24, p. 5529, Dec. 2019.

[19] N. Marir, H. Wang, G. Feng, B. Li, and M. Jia, "Distributed abnormal behavior detection approach based on deep belief network and ensemble SVM using spark," *IEEE Access*, vol. 6, pp. 59657–59671, 2018, doi: 10.1109/ACCESS.2018.2875045.

[20] C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," *IEEE Access*, vol. 5, pp. 21954–21961, 2017, doi: 10.1109/ACCESS.2017.2762418.

[21] W. Wang, M. Zhu, X. Zeng, X. Ye, and Y. Sheng, "Malware traffic classification using convolutional neural network for representation learning," in *Proc. Int. Conf. Inf. Netw. (ICOIN)*, Da Nang, Vietnam, 2017, pp. 712–717, doi: 10.1109/ICOIN.2017.7899588.

[22] W. Ming and L. Jian, "Network intrusion detection model based on convolutional neural network," *J. Inf. Secur. Res.*, pp. 990–994, Nov. 2017.

[23] K. Wu, Z. Chen, and W. Li, "A novel intrusion detection model for a massive network using convolutional neural networks," *IEEE Access*, vol. 6, pp. 50850–50859, 2018, doi: 10.1109/ACCESS.2018.2868993.

[24] P.-F. Wu and H.-J. Shen, "The research and amelioration of pattern-matching algorithm in intrusion detection system," in *Proc. IEEE 14th Int. Conf. High Perform. Comput. Commun. IEEE 9th Int. Conf. Embedded Softw. Syst.*, Liverpool, U.K., Jun. 2012, pp. 1712–1715, doi: 10.1109/HPCC.2012.256.

[25] V. Dagar, V. Prakash, and T. Bhatia, "Analysis of pattern matching algorithms in network intrusion detection systems," in *Proc. Int. Conf. Adv. Comput.*, 2016, pp. 1–5.

[26] I. S. Thaseen and C. A. Kumar, "Intrusion detection model using fusion of chi-square feature selection and multi class SVM," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 29, no. 4, pp. 462–472, Oct. 2017.

[27] B. Ingre, A. Yadav, and A. K. Soni, "Decision tree based intrusion detection system for NSL-KDD dataset," in *Proc. Int. Conf. Inf. Commun. Technol. Intell. Syst.*, Ahmedabad, India, 2017, pp. 207–218, doi: 10.1007/978-3-319-63645-0_23.

[28] P. Nancy, S. Muthurajkumar, S. Ganapathy, S. V. N. S. Kumar, M. Selvi, and K. Arputharaj, "Intrusion detection using dynamic feature selection and fuzzy temporal decision tree classification for wireless sensor networks," *IET Commun.*, vol. 14, no. 5, pp. 888–895, Mar. 2020, doi: 10.1049/iet-com.2019.0172.

[29] M. A. Jabbar, R. Aluvalu, and S. S. Satyanarayana Reddy, "Intrusion detection system using Bayesian network and feature subset selection," in *Proc. IEEE Int. Conf. Comput. Intell. Comput. Res. (ICCIC)*, Coimbatore, India, Dec. 2017, pp. 1–5, doi: 10.1109/ICCIC.2017.8524381.

[30] T.-T.-H. Le, J. Kim, and H. Kim, "An effective intrusion detection classifier using long short-term memory with gradient descent optimization," in *Proc. Int. Conf. Platform Technol. Service (PlatCon)*, Busan, South Korea, Feb. 2017, pp. 1–6, doi: 10.1109/PlatCon.2017.7883684.

[31] T. Su, H. Sun, J. Zhu, S. Wang, and Y. Li, "BAT: Deep learning methods on network intrusion detection using NSL-KDD dataset," *IEEE Access*, vol. 8, pp. 29575–29585, 2020, doi: 10.1109/ACCESS.2020.2972627.

[32] P. Wei, Y. Li, Z. Zhang, T. Hu, Z. Li, and D. Liu, "An optimization method for intrusion detection classification model based on deep belief network," *IEEE Access*, vol. 7, pp. 87593–87605, 2019, doi: 10.1109/ACCESS.2019.2925828.

[33] M. Gao, L. Ma, H. Liu, Z. Zhang, Z. Ning, and J. Xu, "Malicious network traffic detection based on deep neural networks and association analysis," *Sensors*, vol. 20, no. 5, p. 1452, Mar. 2020.

[34] R. Vinayakumar, K. P. Soman, and P. Poornachandran, "Applying convolutional neural network for network intrusion detection," in *Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI)*, Udupi, India, Sep. 2017, pp. 1222–1228, doi: 10.1109/ICACCI.2017.8126009.

[35] Y. Ding and Y. Zhai, "Intrusion detection system for NSL-KDD dataset using convolutional neural networks," in *Proc. 2nd Int. Conf. Comput. Sci. Artif. Intell. (CSAI)*, 2018, pp. 81–85.

[36] H. Yang and F. Wang, "Wireless network intrusion detection based on improved convolutional neural network," *IEEE Access*, vol. 7, pp. 64366–64374, 2019, doi: 10.1109/ACCESS.2019.2917299.

[37] Q. Zhang, Z. Jiang, Q. Lu, J. Han, Z. Zeng, S.-H. Gao, and A. Men, "Split to be slim: An overlooked redundancy in vanilla convolution," 2020, *arXiv:2006.12085*. [Online]. Available: http://arxiv.org/abs/2006.12085

[38] L. Dhanabal and S. P. Shantharajah, "A study on NSL-KDD dataset for intrusion detection system based on classification algorithms," *Int. J. Adv. Res. Comput. Commun. Eng.*, vol. 4, no. 6, pp. 446–452, 2015.