IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 25 June 2023, accepted 5 July 2023, date of publication 19 July 2023, date of current version 22 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3296798

RESEARCH ARTICLE

SDN Attack Identification Model Based on CNN Algorithm

HUIMIN XUE AND BING JING
Department of Internet of Things Technology, Shanxi Vocational & Technical College of Finance & Trade, Taiyuan 030031, China
Corresponding author: Bing Jing (sxcmjb@163.com)

ABSTRACT With the complexity of network structure, the requirements for network architecture are also increasing, and Software Defined Network (SDN) technology has emerged. SDN technology has successfully simplified network management, but its open programming nature poses a risk of network attacks. In complex network environments, the recognition accuracy of traditional recognition models cannot meet the requirements of accuracy and speed. In view of this, this research proposes an attack identification model based on Convolutional neural network (CNN), hoping to solve the attack identification problems faced in the SDN environment, improve the accuracy of the model, and ensure the security of SDN. An SDN attack recognition model is constructed using the NSL-KDD dataset and the MIT LL DARPA dataset, and the CNN is used to utilize it in SDN. In the performance testing experiment of the model, the results show that the proposed model has an accuracy of 98.25% in SDN attack recognition, and its performance is significantly better than traditional CNN models. The accuracy of traditional attack recognition reaches 98.25%, and its performance is superior to the KNN-PSO model. The superiority of the model has been verified, further confirming the application value of the research model in SDN attack recognition.

INDEX TERMS Network architecture, SDN, CNN, attack identification, KNN-PSO.

I. INTRODUCTION

In recent years, the rapid development of Internet technology has greatly brought convenience to people's lives. With the increasing diversity of network structures, network management is becoming more complex and difficult. The main reason for the difficulty in network management is the coupling between the data domain and the control domain. In order to design a more reasonable network architecture and effectively manage the network, some researchers have proposed Software Defined Network (SDN). SDN can achieve centralized control in terms of logic, and can also utilize controllers to allocate link resources, thereby flexibly controlling network data, simplifying network structure, and solving network congestion problems [3].

SDN has become a key technology in cloud computing environments, effectively addressing network architecture issues. However, since SDN programming is open to the outside world, SDN also faces security risks from network attacks caused by its own architecture. With the development of technology, networks are facing more and more types of attacks, and the traditional models that rely on expert experience are no longer applicable. A large number of scholars have begun to study network attacks based on deep learning algorithms. In the field of deep learning algorithms, Convolutional neural network (CNN) plays an important role in deep learning, and has achieved great success in different fields. Compared with other deep learning algorithms, CNN can process high-dimensional data, has stronger ability to extract features, and has higher accuracy of output results [5].

Based on the research of scholars in different fields, it can be seen that the improved SDN based on CNN has a relatively ideal effect in dealing with attack recognition problems. In view of this, this study focuses on the attack recognition requirements of SDN in network environments, fully utilizing the characteristics of centralized control in SDN logic, and proposes an attack recognition model based on CNN algorithm. Firstly, the overall framework of the attack recognition model is designed, followed by data collection. Then, feature screening methods and recognition sections

The associate editor coordinating the review of this manuscript and approving it for publication was Ton Duc Do.

are selected and designed. Finally, performance testing is conducted on the model.

It is hoped to solve the network attack problem faced by SDN, ensure the security of SDN network environment, and contribute to the development of SDN. The research content is mainly divided into four parts. The first part briefly describes the research status of SDN and CNN algorithms. In the second part, the SDN attack recognition model was first constructed. Then the CNN algorithm was introduced to optimize the model. The third part is to analyze the model results based on CNN algorithm, conduct performance testing and comparative analysis experiments. The fourth part is a summary and outlook of the research content.

II. RELATED WORKS

SDN technology can improve network capacity and has been widely applied in multiple fields. Ouamari MA et al. found issues with data exchange between headquarters and local branches in the wide area network. In order to solve this problem, they proposed the SDN-WAN solution. Firstly, reconfigure network management to meet service requirements. Then, the joint optimization problem of average request latency and survivability solved the latency problem of the server. The results showed that compared with traditional methods, the research method significantly improved the performance of the system [6].

Ouamari et al. found that SDN has great potential in preventing channel congestion and providing multi-network connections, but the controller's processing power is limited. In view of this, a new method based on deep reinforcement learning strategy is proposed to optimize the balance process of the controller. The results showed that the proposed method effectively reduced load balancing and is superior to other traditional methods [7]. Garg et al. designed an SDN-based framework for the normal operation of the network, which not only improves the network communication part but also simplifies network management [8].

However, the isolated nature of SDN technology has caused many security issues, so many scholars have shifted their research direction to SDN attack recognition. Badotra and Panda proposed a method to detect DDoS attacks. First, consider setting up experiments in different network scenarios and using different tools to process data information. Then the controller is used to analyse and process the data flow. Finally, parameter analysis and identification are performed using the characteristics of attack time and attack type. The results showed that this method optimised the control performance of SDN [9].

An attack recognition model based on the active learning method of entropy was proposed by Ahmed et al. The load balancing mechanism suggested by this model optimises sensor processing power and tracks network intrusions. The outcomes of the trial demonstrate a considerable improvement in detection performance by this approach [10]. Kamel et al. considered the security vulnerabilities of SDN and proposed a method for protecting the network against DDoS attacks. This method is based on the weight distribution of the underlying network, specifies the transmitted data flow, continuously updates the capacity and number of controllers, and then uses deep learning algorithms to update the network weights. The results showed that this method effectively reduced DDoS attacks [11].

In recent years, the CNN algorithm has been widely applied in various fields and has achieved good results. Chen et al. proposed a new model that aims to accelerate the computational speed of gas detection. This model mixes Memristor based on CNN. The study suggests that the model is built on convolution cores of various sizes, employs the multi-dimensional convolution method to extract feature information of various dimensions, and makes use of memristor to increase the utilisation rate of the entire model hardware structure in order to improve operation speed [12]. Bao and Yang found in the field of radar research that the signal is unstable due to human motion and other factors. Therefore, a new personnel counting method was designed based on CNN. Researchers hope to collect clearer graphical information through this method to complete the counting task. The results verified the superiority of this method and achieved the expected effect [13].

In the field of rock exploration, Chen et al. designed a model of Laser-induced breakdown spectroscopy combined with two-dimensional CNN algorithm in view of the lack of simultaneous multi-task models at present. This model is designed with two types of outputs to simultaneously perform classification and recycling tasks. The results showed that the accuracy of this model was higher than that of traditional models [14]. Guo et al. enhanced the CNN-based algorithm to increase the effectiveness of power plant detection. Prior to input to the model for fault detection, an attention mechanism was added to the algorithm. According to the experimental findings, the defect location was annotated, which significantly increased the detection accuracy [15]. Xue et al. proposed an architecture search method to address the issue of consuming a large amount of human resources and computational time in designing CNN architectures. This method adjusts the strategy based on adaptive mutation neural structure to achieve automation of CNN architecture design. The results showed that this method achieved the expected goals, saved labor costs and reduced computational time [16].

In summary, SDN has various advantages, and Ouamari MA has found that SDN-WAN can solve network latency problems. M. A. Ouamari et al. found that it can improve the processing power of the controller and ensure the balance of control operation. However, in the process of studying SDN, Garg S found that the isolated features of SDN can easily lead to security issues. To address this issue, Badotra S and Panda S N proposed a method for detecting DDoS attacks, optimizing the control performance of SDN. Kamel AE also proposed a network protection method in the face of DDoS attacks. With the goal to increase SDN detection performance, Ua A et al. suggested an attack recognition model based on the entropy active learning method. However,

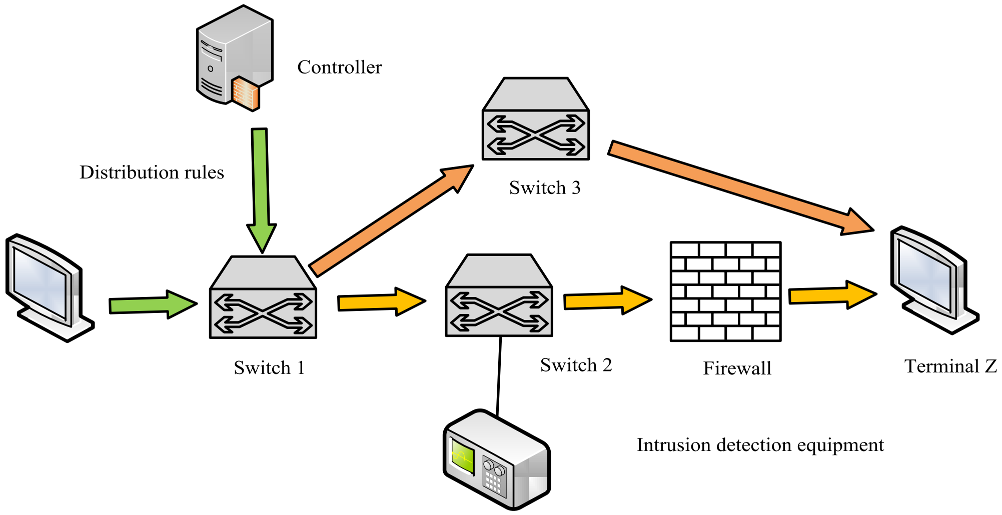


FIGURE 1. SDN framework packet transmission path changes.

research on network attacks against SDN is still in its infancy, and there isn't yet an accurate identification model.

Therefore, there is an urgent need to conduct research on SDN attack identification to ensure the security of the SDN network environment and contribute to the development and application of SDN. CNN has developed relatively maturely and has made contributions in various fields. Compared to other deep learning algorithms, CNN can sensitively use local perception to extract features. This study further optimizes the CNN algorithm and builds an SDN attack recognition model to improve the detection and recognition rate of the model, thereby improving the security performance of SDN.

III. OPTIMIZATION METHOD FOR SDN ATTACK IDENTIFICATION TECHNOLOGY BASED ON CNN

A. SDN ATTACK RECOGNITION MODEL CONSTRUCTION

Scholars such as Siddiqui et al. believe that SDN has become a paradigm that helps software controllers more effectively manage IoT infrastructure and traffic [17]. Khalid et al. proposed a solution to ensure the security of the Internet of Things based on SDN, providing the ability to design Tamper resistance and independent verification strategies [18]. SDN is a programmable network that simplifies network management and makes it more flexible. At the same time, SDN plays a central control role in attack identification systems, assisting intrusion systems in improving point deployment issues. The uniqueness of SDN gives it the advantage of a control flow table, which can also assist attack recognition systems in responding to detected network attacks [19]. In SDN networks, when data flows, data information can easily break through traditional security protection, as shown in Figure 1.

From Figure 1, it can be seen that during the connection process between two terminals X and Z, the controller received a new issuance rule, and the transmission path of data information changed from orange to yellow. Through attacks, data material can be transmitted over an unrestricted path without being detected by detecting devices or firewalls. Therefore, how toproperly deploy IDS in SDN is also a key research project. Before deployment, it is necessary to master the principles of traditional flow table attacks and assist in system deployment. As shown in Figure 2, the attack principle of general flow table information is presented.

As shown in Figure 2, the controlled terminal sends a large amount of ambiguous false information to the switch, but the switch cannot recognize it, so the ambiguous information is sent to the controller. When a lot of false information enters the controller through the switch, the controller must continue to identify and judge the false information, using up a lot of resources and time. This causes traffic jams and prevents the controller from processing the normal information requirements. When false information accumulates to a certain extent and exceeds the processing capacity of the controller, overload operation of the controller will lead to network paralysis. On this basis, if the information that the switch fails to process accumulates to a certain extent, and there is too much information in the switch flow table items, it will lead to its own failure and even crash. In addition to considering the above factors, it is also necessary to consider the consequences of bandwidth intrusion in the control domain caused by the controller itself. If the controller is attacked and fails, the attack recognition system will not be able to operate normally, and the security policy will not be executed in time, as shown in Figure 3.

Figure 3 shows the process of controller bandwidth intrusion. The attack information does not first pass through the switch, and the attacker directly sends false information and rules through the controller. The switch operates according to the new false rules, generating a large number of false information packets, leading to network congestion and even network collapse. From this, it can be seen that traditional

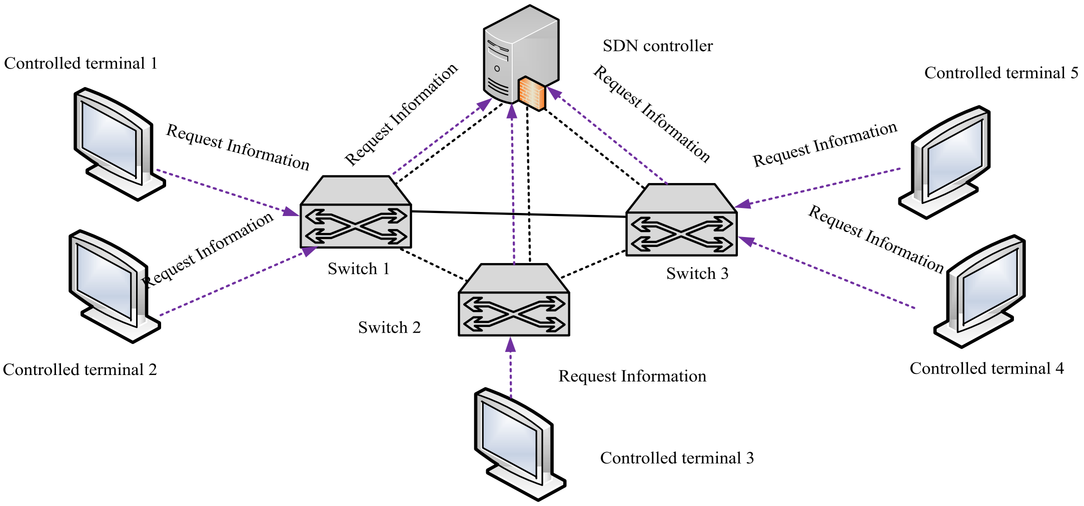


FIGURE 2. Principle of switch flow table attack.

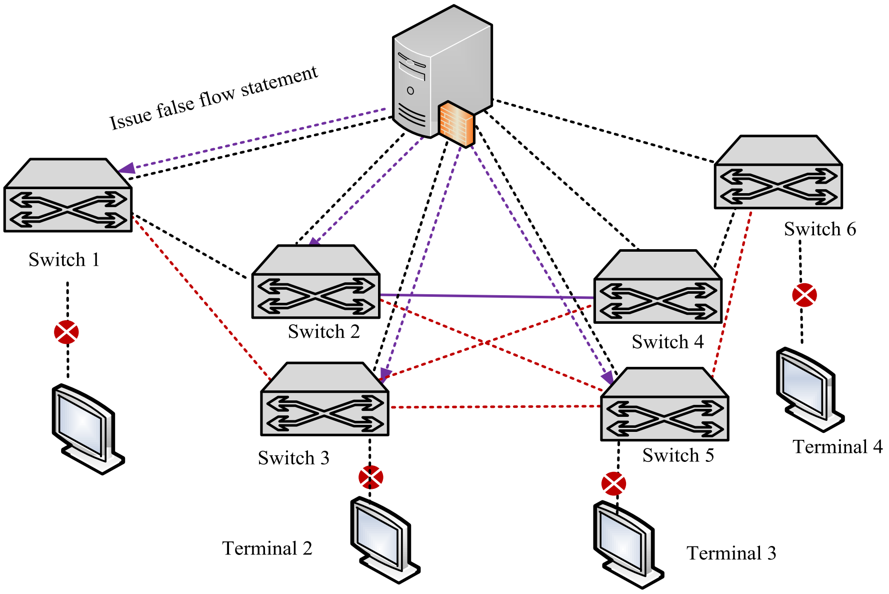


FIGURE 3. Control domain bandwidth attack principle.

network attack recognition systems have shortcomings and need to be optimized and improved. Research is considering improving attack recognition algorithms and models. Optimizing algorithms can not only improve the processing speed of information, but also enhance the sensitivity of identifying attack behaviors and detecting more attack behaviors. In addition, it efficiently addresses network paralysis brought on by the controller's high load operation while saving time, money, and resources. Optimize the attack recognition system to have the ability to identify SDN specific attack modes.This research constructs an SDN attack identification model based on neural networks to solve the attack identification problem. The architecture of the model is shown in Figure 4.

As shown in Figure 4, itis a traditional attack recognition system for SDN. Based on the traditional system, an SDN-adapted attack identification model is designed by combining it with the specific attacks of SDN, and this model is divided into four modules: data acquisition, inflow data preprocessing, feature selection and detection. For specific network attacks in SDN, existing intrusion detection methods are difficult to detect them effectively. This research constructs an SDN flow feature database [16].

By collecting flow table entry information under different network behaviours. The NSL-KDD dataset was used to test the proposed algorithm for this study to ensure the integrity of the experiment. The switch flow table attack will cause the denial of the SDN southbound channel, while the control

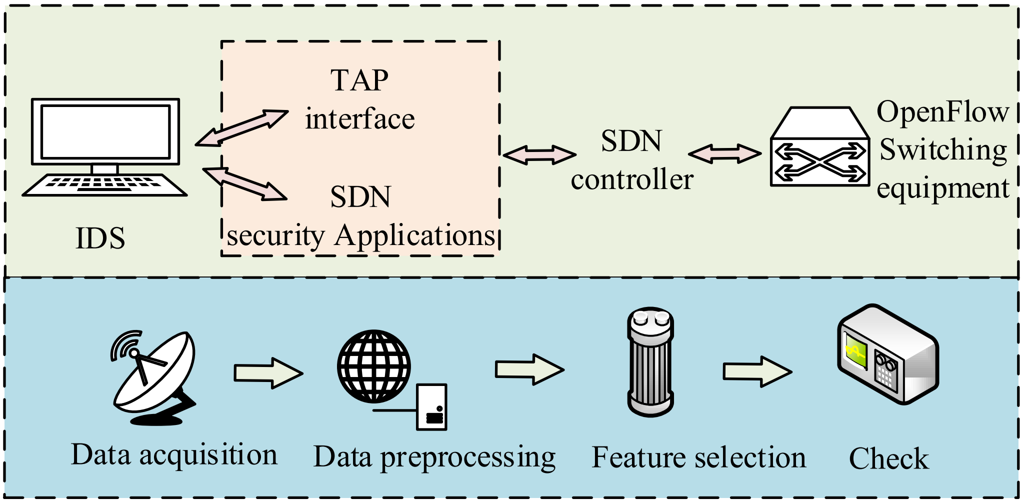


FIGURE 4. SDN attack identification model.

domain bandwidth attack will cause the resource depletion of the SDN transport layer, so both are SDN specific distributed denial of service attacks. The primary difference between the two is that attackers can quickly and randomly pseudocode each target IP, enabling the switch's requests to be processed by the control end and leading to a more dispersed dispersion of the target IP across the network. To cover up the attack, attackers can randomly simulate a large number of packets. Under normal circumstances, the traffic characteristics are relatively stable, but when attacked, their characteristics may suddenly change. After an attack begins, an SDN switch will transmit a large amount of data information to the switch. In view of this, this study will introduce features such as the average number of flow packets, the success rate of flow table matching, and the comparison flow.During a network attack, the number of data streams is usually reduced to speed up the spoofing of IPs at the source and target, so there is a difference between the number of packets under normal conditions and the number of packets under attack, as shown in equation (1).

$$APF = \frac{\sum_{i=1}^{FlowNum} packetNum_i}{FlowNum} \tag{1}$$

As shown in equation (1),$packetNum_i$ represents the first packet in the flow table at i and the total number of packets in the flow table is represented by $FlowNum$. When data reaches the switch, it automatically performs a seek and match action. The influx of stream table information into the switch will cause the switch to be overloaded with information, resulting in a sharp drop in the success rate of stream table information matching, as shown in equation (2).

$$MLR = \frac{Match}{Look\,\mathrm{sup}} \tag{2}$$

As shown in equation (2), the number of total stream table information is denoted by $Match$ and the number of matched result stream table information is denoted by $Look$ sup. During normal operation of the network, the ratio of the number of pairs to the number of single flows in the system is relatively smooth. In the case of a large number of false destination IP addresses, the stability of the network is severely compromised and the ratio of the number of pairs to the number of single flows in the system is shown in equation 3.

$$PPF = \frac{FlowNum - 2 \times pair}{FlowNum} \tag{3}$$

As shown in equation (3), the ratio of the number of pairs of flows to the number of single flows is denoted by PPF and the number of pairs of interactive flow tables is denoted by $pair$. When a network is attacked, the number of bytes in a packet is usually reduced, thus speeding up the rate at which the network sends packets. Therefore, the average number of bits of network traffic is usually a good measure in detecting networks and is calculated as shown in equation (4).

$$ABF = \frac{\sum_{i=1}^{FlowNum} ByteNum_i}{FlowNum} \tag{4}$$

As shown in equation (4), the number of packet bytes in the flow table at i is represented by $ByteNum_i$. The network port growth rate is stable during normal network operation and increases significantly during an intrusion.

$$PGS = \frac{portNum}{T} \tag{5}$$

The new increment of the port is represented by $portNum$ as shown in equation (5). In the event of an attack on the network, the number of spoofed destination IPs will increase substantially, so the rate of increase in destination IP addresses is also a good indicator of a network attack, as shown in equation (6).

$$DIGS = \frac{DIGSNum}{T} \tag{6}$$

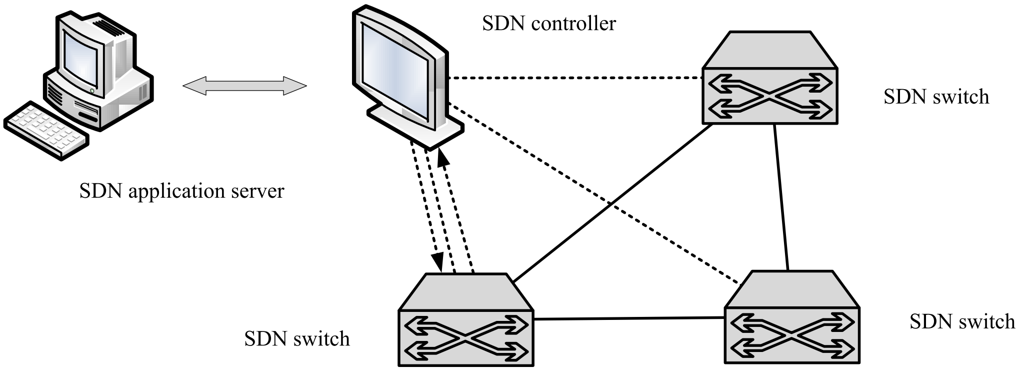


FIGURE 5. Flow table collection process.

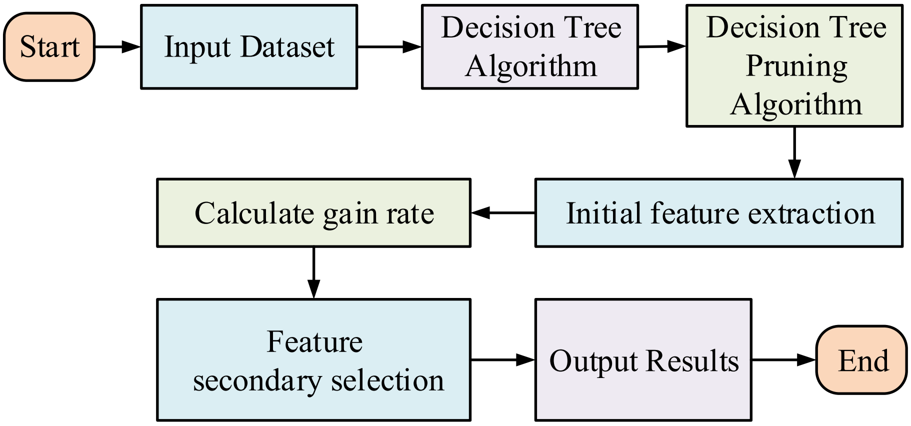

Start
Input Dataset
Decision Tree Algorithm
Decision Tree Pruning Algorithm
Initial feature extraction
Calculate gain rate
Feature secondary selection
Output Results
End

FIGURE 6. Feature selection process.

As shown in equation (6), the destination IP growth rate is represented by *DIGS* and the increment of destination IP during the cycle is represented by *DIGSNum*. It is also necessary to compare the analysis with the IP growth rate in the normal state.

$$SIGS = \frac{SIGSNum}{T} \tag{7}$$

As shown in equation (7), the original IP increment rate is represented by *SIGS* and the intra-cycle IP increment is represented by *SIGSNum*. The SDN comes with switch flow table attacks and control domain bandwidth attacks, and these two types of attacks will cause changes in the traffic flow of the southbound channel and the traffic layer of data forwarding of the SDN. In view of this, the flow table entry information can be used as the judgment basis for attack identification. The flow table entry information collection process is shown in Figure5.

As shown in Figure 5, first the SDN server sends the capture command, then the controller periodically sends the information to the SDN switch, and the switch responds to the information. Finally, the flow table is redirected and the flow table is captured. When the network is under attack, the number of flow table entry requests will increase, so the flow table entry increase rate can be chosen as a reference indicator to measure the network attack.

$$RFB = \frac{FlowNum}{T} \tag{8}$$

As shown in equation (8), the rate of increase of stream table items is denoted by *RFB* and the number of stream table requests within the period *T* is denoted by *FlowNum*. Preprocessing of the collected data is required after data collection, and this study chose to digitise the dataset with symbolic features and data normalisation, with the normalisation formula shown in equation (9).

$$x^* = \frac{x - \mu}{\sigma} \tag{9}$$

As shown in equation (9), the sample means are represented by μ, the raw data by x and the normalised data by x^*. After preprocessing the dataset, feature selection is required. This study used decision trees to screen the features. The traditional decision tree algorithm selects the information gain for feature screening, and this study also added the gain rate to screen the features repeatedly, and the screening process is shown in Figure 6

As shown in Figure 6,the whole feature selection process is divided into four steps. In the first step, a decision tree is generated based on the information of the data set.In the second

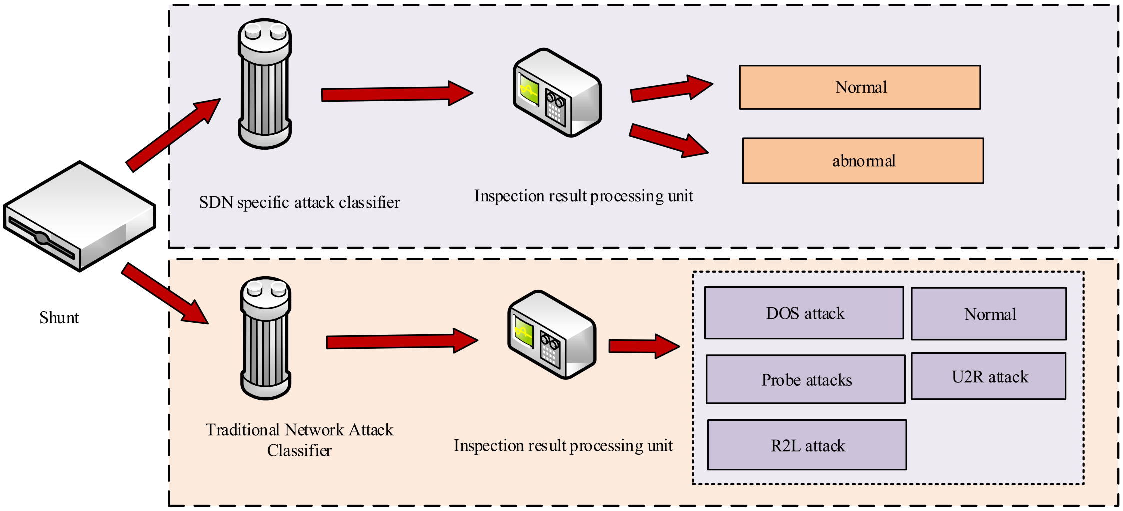

Normal
abnormal
SDN specific attack classifier
Inspection result processing unit
Shunt
DOS attack
Normal
Probe attacks
U2R attack
R2L attack
Traditional Network Attack Classifier
Inspection result processing unit

FIGURE 7. **Identification module structure design.**

step, pruning operation and initial screening are performed.In the third step, the gain rate is calculated.In the fourth step, the second screening is performed according to the gain rate and the results are output [17]. After processing and analysing the data with feature screening, it enters the recognition stage, and the recognition process is shown in Figure 7.

As shown in Figure7, the collected data is first triaged into traditional network attacks and unique network attacks using a splitter. In the traditional network attack identification segment, the detection result processing unit marks the raw data and classifies the attack behaviour into five categories. In the SDN-specific attack segment, a CNN algorithm is used for binary classification to identify both abnormal and normal behaviours.

B. OPTIMIZATION OF CNN-BASED SDN ATTACK IDENTIFICATION MODEL

Bhayo J et al. believe that SDN is an effective solution to improve network security and access control mechanisms. Faced with DDoS attacks, a detection method based on machine learning has been proposed. The study used decision tree and support vector machine algorithms to classify network data packets. The structure shows that the proposed solution can enhance the security of IoT networks and reduce the risk of DDoS attacks. Deep learning algorithms can compensate for deficiencies such as misuse detection in traditional attack detection models. CNN is a feed-forward neural network with a wide range of applications in computer and natural language processing, etc. [18]. CNN mimic the visual mechanism of living things, extracting feature information based on local perception, with the distinguishing feature of having a set of neurons with equally weighted connections. The study uses CNN algorithms for the detection part of the SDN attack recognition model, and the traditional CNN attack recognition model training process is shown in Figure 8.

As shown in Figure 8,The CNN attack recognition model is first obtained after forward transmission of prediction results by CNN, then calculating the result error, then back propagating according to the error feedback and updating the information parameters of the model, and finally after several iterations. This research combines the principles of CNN algorithms with the practical requirements in the field of attack recognition to optimise on the traditional CNN attack recognition model, and the optimisation steps will be described in detail later. The main purpose of CNN pooling layer design is to improve training speed, reduce intermediate parameters and reduce overfitting. At the same time, pooling is also a process of losing feature information [19].This research uses pooling with mean sampling to reduce the probability of losing feature information and reduce the classification error. The degree of parameter correction for back propagation will be greater during neural network training when the discrepancy between the anticipated and true values is bigger, leading to quick convergence. The convergence speed of the algorithm is governed by the choice of the loss function, and this study uses the faster convergence cross entropy as the loss function, which is calculated as shown in equation (10).

$$C = -\left[y \ln \hat{y} + (1-y) \ln(1-\hat{y}) \right] \tag{10}$$

As shown in equation (10), the predicted output is represented by $\hat{y}$ and the actual output is represented by y. From this equation, it is clear that making $\hat{y}$ be 0 results in NaN. To avoid this, an extremely small value is added to the end, as shown in equation (11).

$$\hat{y}' = \hat{y} + \sigma \tag{11a}$$

$$C = -\left[y \ln \hat{y}' + (1-y) \ln(1-\hat{y}') \right] \tag{11b}$$

Traditional CNN models are based on gradient descent, although it has good performance. However, it still has some

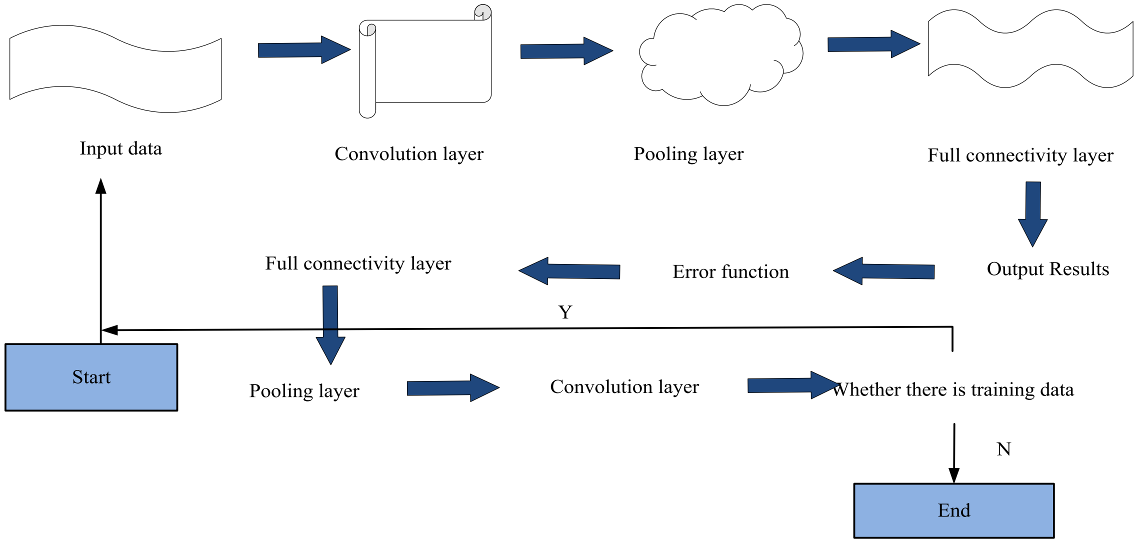


FIGURE 8. CNN identification module training flowchart.

shortcomings. Firstly, it is difficult to choose a suitable learning rate. A lower learning rate can make the convergence of the algorithm very poor. If the learning speed is too fast, it will affect the normal convergence of the algorithm, leading to oscillations at the limit point. Secondly, the same learning rate should be used every time to update all parameters of the model. Different learning rates should be chosen for different features of the data, such as for rare data features, a larger learning rate should be chosen. Then, when the objective function is non-convex, some suboptimal problems may arise, such as saddle points and local extremum points [20]. In view of this, the adaptive learning rate is generally chosen to optimise it, and the Adam algorithm, which has a better optimisation effect, is chosen for this study. The stochastic gradient descent algorithm based on the Adam algorithm is shown in equation (12).

$$m_t = \rho_1 * m_{t-1} + (1 - \rho_1) * g_t \tag{12a}$$

$$n_t = \rho_2 * n_{t-1} + (1 - \rho_2) * g_t^2 \tag{12b}$$

$$\hat{m}_t = \frac{m_t}{1 - \rho_1^t} \tag{12c}$$

$$\hat{n}_t = \frac{n_t}{1 - \rho_1^t} \tag{12d}$$

$$\Delta\theta = -\frac{\hat{m}_t}{\sqrt{\hat{n}_t} + \delta} * \eta \tag{12e}$$

$$\theta_{t+1} = \theta_t + \Delta\theta \tag{12f}$$

As shown in (12), the gradient is denoted by g_t, the first-order moment estimates and second-order moment estimates in the gradient are denoted by m_t and n_t, respectively, the corrected moment estimates are denoted by $\hat{m}_t$ and $\hat{n}_t$, δ represents the minimal constant, which can prevent the denominator from being 0, and the learning rate is denoted by η [21] This study improved the structure of the CNN attack recognition model based on the properties of the attack recognition feature vector, as shown in Figure 9.

In the field of image recognition in CNN, the input data is fed directly into the image, followed by the pooling instruction after convolution, as a means to minimise the number of intermediate parameters, as illustrated in Figure 9. The bottom side is the modern pooling process [22]. However, because its feature information is easily expressed, a large number of pooling operations for attack recognition in the network may result in muddled feature information.To reduce the pooling operation, this study performs a pooling operation after two convolutions, as shown at the bottom of Figure 9, which is an improved model flowchart with an improved optimisation of the connection between the convolution and pooling layers by adding a classifier with the *soft* max function after the fully connected layer, which transforms the multi-category result values into readily comparable and understandable relative probabilities with the fun function [23].

$$S_i = \frac{e^{V_i}}{\sum_i^C e^{V_i}} \tag{13}$$

As shown in equation (13), the total number of categories is represented by C, the category index is represented by i, the output of the prepole unit is represented by V_i and the ratio of the present element index to the index of all elements is represented by S_i. Because the ReLU function is computationally small in error calculation, it helps to train the neural network during operation and there is no loss of information; the assignment to 0 also avoids overfitting conditions when the neurons perform their output [24]. The main purpose of the classifier is to prevent the occurrence of overfitting conditions by introducing random deactivation techniques into the classifier. This technique uses the method of randomly setting some of the weights or outputs of the

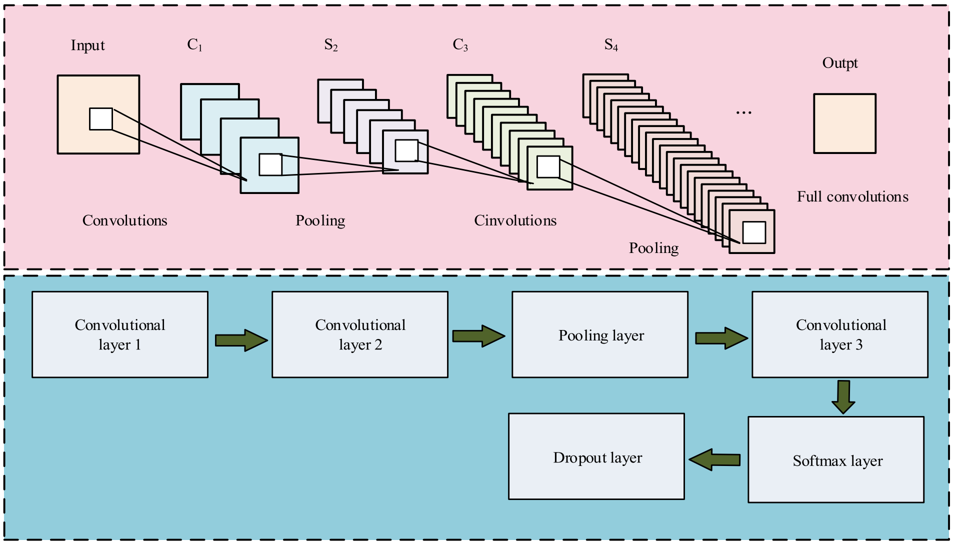


FIGURE 9. CNN attack identification model.

hidden layer to zero in order to reduce the correlation between the nodes, which in turn leads to the regularisation of the neural network, thus solving the problem of gradient disappearance and overfitting. The process of calculating neurons in a conventional network is shown in equation (14).

$$c^{(l+1)} = w^{(l+1)}a^{(l)} + b^{(l+1)} \tag{14a}$$

$$a^{(l+1)} = f(c^{(l+1)}) \tag{14b}$$

As shown in equation (14), the total number of layers in the output layer network is represented by L, $a^{(l+1)}$ represents the output of $l+1$ layer,$l+1$ input weighting and vector is represented by$c^{(l+1)}$, $l+1$ layer weights are represented by $w^{(l+1)}$, $b^{(l+1)}$ is the $l+1$ layer bias and $f(x)$ is the excitation function [25]. After using random deactivation, the improved formula is shown in equation (15).

$$r^{(l)} \sim Bernoulli(p) \tag{15a}$$

$$\tilde{a}^{(l)} = r^{(l)} * a^{(l)} \tag{15b}$$

$$c^{(l+1)} = w^{(l+1)}\tilde{a}^{l} + b^{(l+1)} \tag{15c}$$

$$a^{(l+1)} = f(c^{l+1}) \tag{15d}$$

As shown in equation (15), the selected neurons are denoted by $\tilde{a}^{(l)}$ and the subset of samples satisfying the Bernoulli distribution with probability p is denoted by $r^{(l)}$.

IV. RESULTS AND DISCUSSIONS

This study conducted performance testing on the constructed CNN-basedSDN attack recognition model, and validated the CNN algorithm-based SDN attack recognition model. The traditional CNN model and KNN-PSO model were introduced to compare and analyze their performance to verify the performance of the CNN-based SDN attack recognition model.

A. PERFORMANCE TESTING OF CNN-BASED ATTACK RECOGNITION MODELS FOR SDNS

Performance evaluations of the built CNN-based SDN attack identification model are conducted in this study. The study employed the same computer device for performance testing with an Intel(R) Core(TM) i5-10210U CPU, 20GB RAM, and Windows 10 Home operating system with 16G RAM to prevent mistakes brought on by various devices. Google's human-computer intelligence technology was selected utilising Linux as the platform. Python was utilised to create the algorithms, with Tensorflow serving as the foundation.

This study used the NSL-KDD dataset and the MIT LL DARPA dataset, among which the NSL-KDD dataset is a classic dataset in the field of attack recognition and is optimized based on the KDD-CUP99 dataset. The NSL-KDD data and MIT LL DARPA dataset remove duplicate information, avoiding classifier bias and making detection rates more accurate. NSL-KDD data records abnormal and normal information, which can be divided into four categories: Dos, Probe, R2L, U2R. The MIT LL DARPA dataset records abnormal information, which includes five categories: Dos, Probe, R2L, U2R, and Date. The study selected the same classification of Dos, Probe, R2L, and U2R from two types of data for testing.During the process of collecting and labeling datasets, some experimental biases may arise due to irregular data collection and subjective judgments.

The study employed the Friedman test to take into account the disparate knowledge held by various individuals and increase testing effectiveness. The Friedman test critical value database was queried following a Friedman rank analysis of variance. There were notable variances between the data, and the critical value exceeded the estimated statistic. Testing the training dataset led to the selection of feature vectors with various dimensions as input, as illustrated in Figure 10.

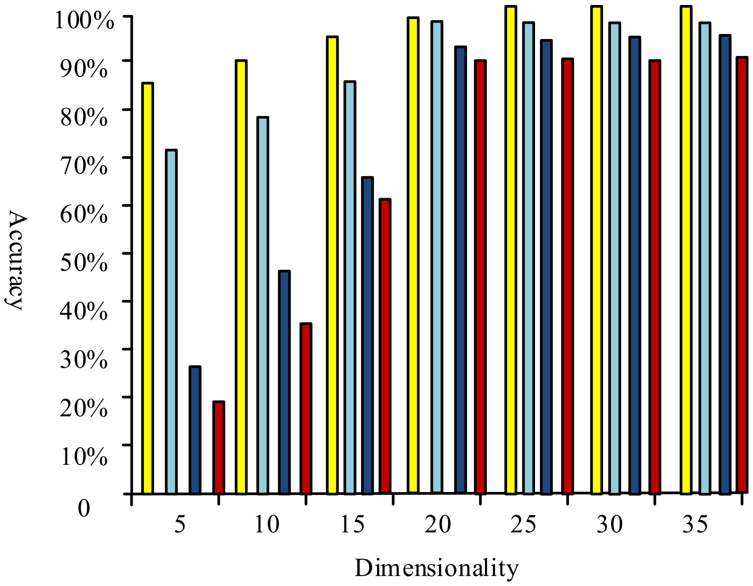


FIGURE 10. Accuracy under different dimensions.

As shown in Figure10, as the number of dimensions of the feature vector increases, the accuracy of all four attack modes shows an upward trend in the range of 5 to 20 dimensions. When the number of dimensions is greater than 20, the fluctuation amplitude of the curve decreases sharply and tends to a horizontal state. As a result, the accuracy of the four attack modes no longer increases and tends towards a stable state. Among them, the U2R detection accuracy is between 88.2% and 89.5%, the R2L detection accuracy is between 93.3% and 93.6%, the Probo detection accuracy is between 98.7% and 98.9%, and the DOS detection accuracy is between 99.5% and 99.9%.From this, it can be shown that the research model has the lowest accuracy in U2R and the best accuracy in identifying DOS attacks. The effects of various dimension numbers on training time must also be taken into account in addition to their effects on accuracy, as shown in Figure 11for various dimension numbers.

As shown in Figure 11, the training time increases as the number of dimensions increases. When the number of dimensions is 20, the average training time of the four attack recognition is 82.32s; when the number of dimensions is 40, the average training time of the four attack recognition is higher, 239.78s, and the training time of 40 dimensions is more than the training time of 20 dimensions by 157.46s. And as can be seen in Fig. 10, the change of the accuracy rate between dimensions 20 and 40 is not obvious, while the training time is increasing extremely fast. Therefore, 20 dimensions will be chosen as input features in the later test experiments. In the method part, the CNN algorithm is optimised from different aspects. In the performance test, the selection of the activation function, the gradient optimisation, the pooling layer and the overfitting are analysed and compared. Figure 12shows the change in accuracy and loss values of the two activation functions with the number of iterations.

Figure 12(a) shows the change of the accuracy of the two activation function with the increase of iteration number. It can be seen that the accuracy of the function increases with the increase of iteration number. The fluctuation amplitude of the curve reduces, no longer grows, and tends to be horizontal when the iteration number is 100. ReLU, the activation function utilised in this work, has a considerably greater accuracy rate than sigmoid. ReLU function accuracy is 98.92%, sigmoid function accuracy is 87.69%, and ReLU function accuracy is 11.23% greater than sigmoid function accuracy.The change in the loss value of the two activation functions with more iterations is depicted in Figure 12 (b). The ReLU function is more stable when the iteration number is 100 because the loss values of the two activation functions reach their lowest points, the ReLU function curve declines more quickly, and the curve volatility after convergence is reduced. It is evident that the activation function chosen for this study has a better impact on the model for recognising SDN attacks.

Figure 13indicates that as the number of iterations rises, Figure 13 (a) demonstrates the accuracy of the four gradient optimisation strategies. It can be seen that as the number of iterations increases, the accuracy of the four gradient optimization algorithms increases. When the iteration number approaches 100, the fluctuation amplitude of the curve decreases, no longer increases, and tends to be horizontal. The accuracy of the Adam algorithm used in this study is significantly higher than other algorithms. The Adam algorithm has an accuracy of 99.62%, the RMSP algorithm has an accuracy of 80.52%, and the ReLU function has an accuracy of 19.10% higher than the RMSP algorithm. Figure 13(b) shows the

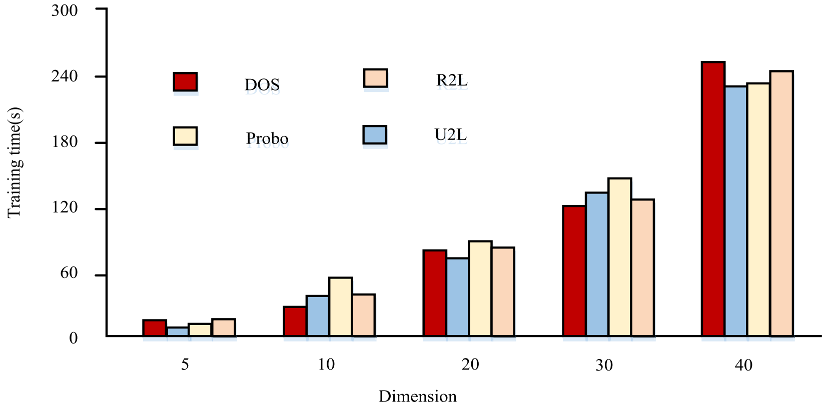
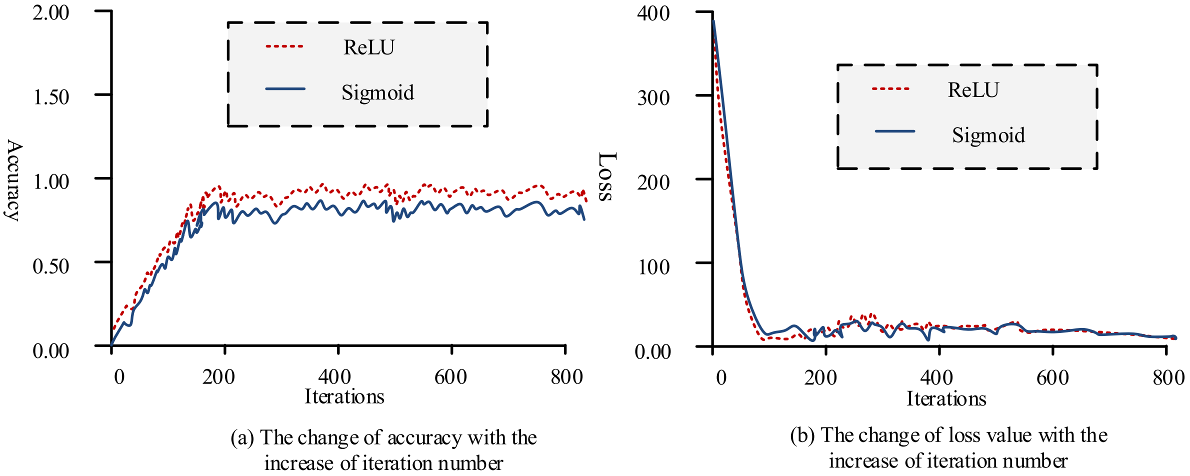


FIGURE 11. Training time under different dimensions.



(a) The change of accuracy with the increase of iteration number

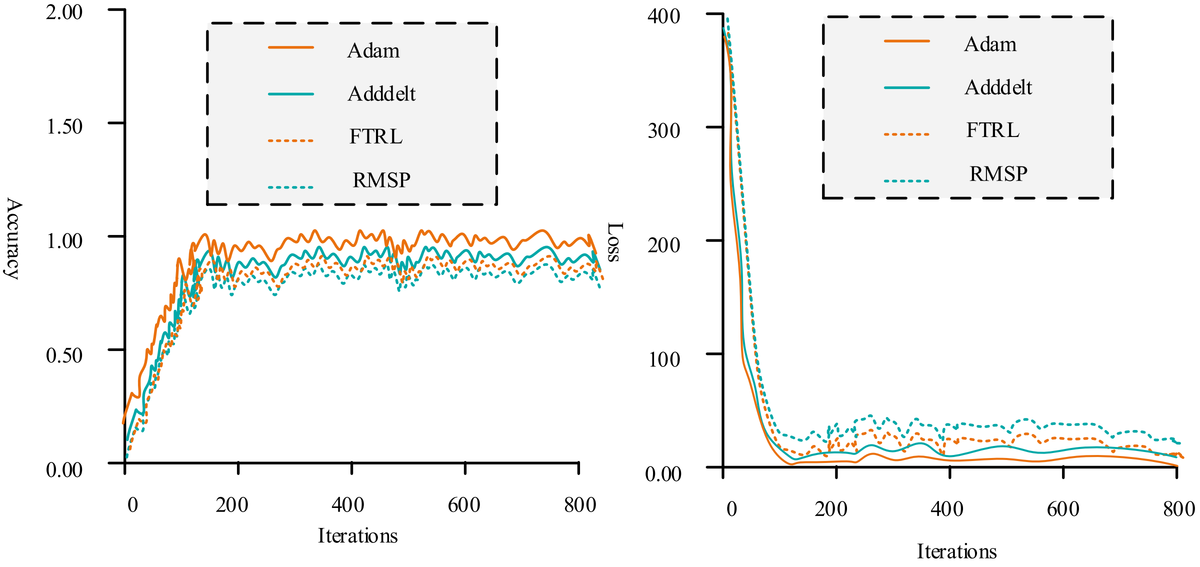
(b) The change of loss value with the increase of iteration number

FIGURE 12. Comparison of activation functions.

changes in the loss values of the four gradient optimization algorithms as the number of iterations increases. When the number of iterations is 100, the loss values of the four gradient optimisation algorithms fall to the lowest point. It can be seen that the curve of the Adam algorithm decreases faster and the curve amplitude is smoother during convergence, indicating that the Adam algorithm is more stable. This indicates that the algorithm selected in this study has better performance in SDN attack recognition models.

Figure 14 illustrates that as the number of iterations rises, Figure 14 (a) displays the accuracy of the three pooling approaches. It is clear that the accuracy of the three pooling approaches grows with the number of rounds. When the iteration number approaches 100, the fluctuation amplitude of the curve decreases, no longer increases, and tends to be horizontal. The accuracy of the mean pooling method used in this study is significantly higher than other methods. The accuracy of the mean pooling method is 98.87%, and the accuracy of the multi-pooling layer method is 85.46%. The accuracy of the mean pooling method is 13.41% higher than that of the multi-pooling layer method. Figure 14 (b) shows the changes in the loss values of the three pooling methods as the number of iterations increases. When the number of iterations is 100, the loss values of the three pooling methods drop to their lowest point. It can be seen that the mean pooling curve decreases faster, and the fluctuation amplitude of the curve after convergence is smoother, indicating that the Adam algorithm is more stable. This indicates that the mean pooling chosen in this study is more effective for SDN attack recognition models.

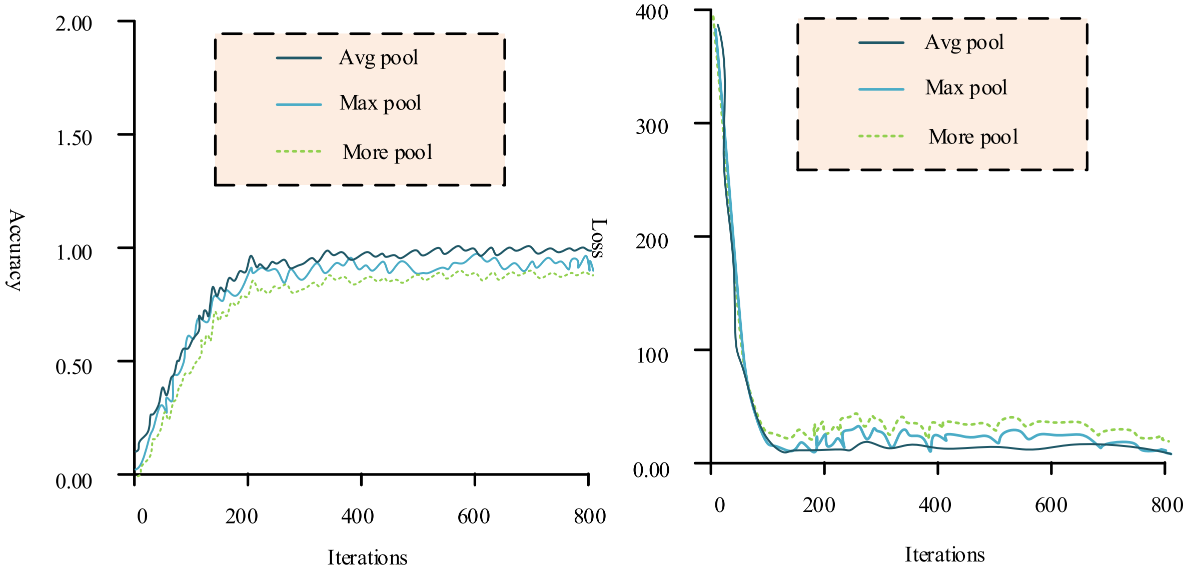
As the number of iterations rises, Figure 15's Figure 15 (a) illustrates the accuracy of the fully linked layer and random inactivation approaches. It is clear that the accuracy of the random inactivation and fully connected layer approaches grows with the number of iterations.As the iteration number approaches 100, the fluctuation amplitude of the curve decreases, stops increasing and tends to be horizontal. In this study, the random deactivation technology was used to optimise the overfitting, and the accuracy rate of this method was significantly higher than that of the full connection layer. The accuracy rate of the random deactivation technology was 99.23% and the accuracy rate of the full connection layer



(a) The change of accuracy with the increase of iteration number

(b) The change of loss value with the increase of iteration number

FIGURE 13. Comparison of gradient optimization algorithms.



(a) The change of accuracy with the increase of iteration number

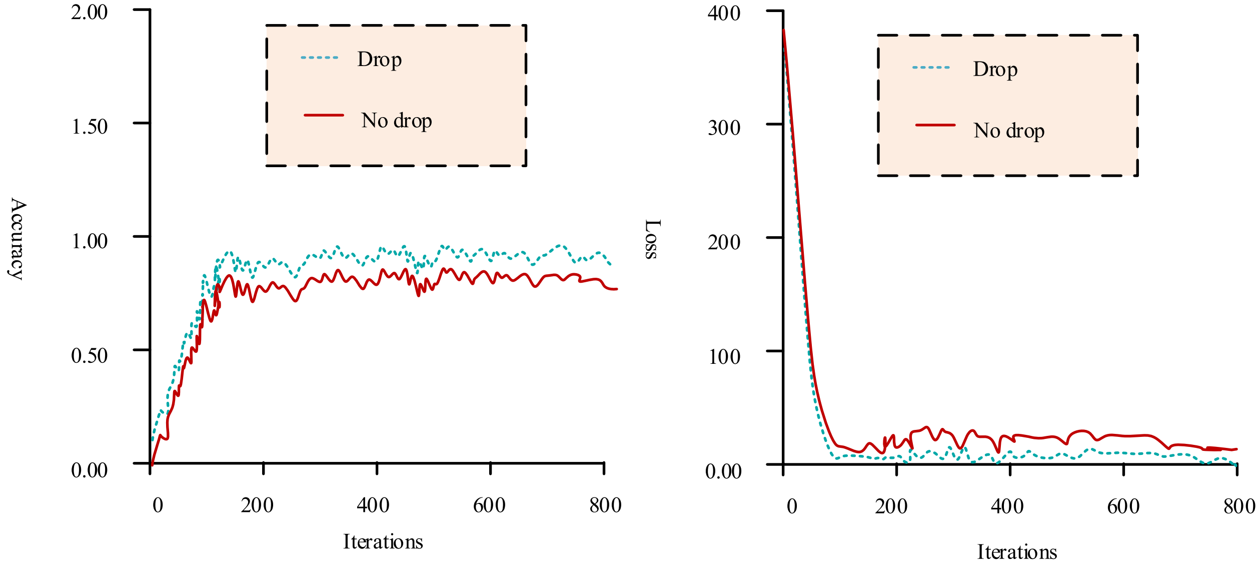
(b) The change of loss value with the increase of iteration number

FIGURE 14. Comparison of different pooling methods.

was 90.02%.The accuracy rate of the random deactivation was 9.21% higher than that of the multi full connection layer. Figure 15 (b) shows the variation of the loss value with increasing iteration number using random inactivation and fully connected layers. When the iteration number is 100, the loss value drops to the lowest point using the random inactivation and fully connected layers method. It can be seen that the random inactivation curve decreases faster, the convergence effect is faster and the fluctuation amplitude of the curve after convergence is smaller. It can be seen that the random deactivation optimisation overfitting condition selected in this study is more effective for the SDN attack recognition model.

B. PERFORMANCE COMPARISON OF CNN-BASED SDNS FOR ATTACK RECOGNITION MODELS

This study uses the NSL-KDD dataset to validate the SDN attack identification model based on the CNN algorithm, introducing the traditional CNN model and the KNN-PSO model to compare and analyse their performance. The attack



(a) The change of accuracy with the increase of iteration number

(b) The change of loss value with the increase of iteration number

FIGURE 15. Comparison of overfitting optimization methods.

TABLE 1. Performance comparison of different attack identification models.

Attack identification model	classification problem	Attack Type	Test Set			Test duration
			AC	FP	Recall	
Improved CNN model	IIclassification	-	98.25%	1.55%	99.13%	1.48s
	multiclassification	normal	98.94%	1.25%	98.25%	
		Dos		1.04%	99.89%	
		probe		3.52%	97.56%	
		R2L		5.44%	98.24%	
		U2R		7.25%	94.88%	
KNN-PSO	IIclassification	-	85.32%	2.64%	86.25%	17.28s
	multiclassification	nomal	85.24%	1.55%	92.45%	
		Dos		1.26%	83.76%	
		probe		3.87%	82.86%	
		R2L		8.57%	79.24%	
		U2R		8.47%	80.89%	
CNN	IIclassification	-	97.13%	3.45%	94.67%	5.78

identification evaluation metrics used in this study are Accuracy (AC), Recall, False Positive Rate (F False Positive and FP).

From Table 1, it can be seen that in terms of SDN attack recognition performance testing, the attack recognition model designed in this study has an AC of 98.25%, a Recall of 99.13%, and a FP of only 1.55%. The CNN model was not improved, with AC of 97.13%, Recall of 94.67%, and FP of 3.45%. The KNN-PSO model has an AC of 85.32%, a Recall of 86.25%, and a FP of 2.64%.The AC of the attack recognition model developed in this study is 1.12% higher than thetraditional CNN model, Recall is 4.46% higher than thetraditional CNN model, and FP is 1.90% lower than thetraditional CNN model. This study proposes that the model AC is 12.93% higher than the KNN-PSO model, Recall is 12.88% higher than the KNN-PSO model, and FP is 1.09% lower than the KNN-PSO model. Compared to the KNN-PSO model, this study also shows good performance in traditional attack recognition, with recalls of 98.25%, 99.89%, 97.56%, 98.24%, and 94.88%, respectively. The KNN-PSO model has recalls of 92.45%, 83.76%, 82.86%, 79.24% and 80.89% respectively. The model in this study has higher recalls than the KNN-PSO model in all five attacks. The FP of the five types of attacks developed in this study are 1.25%, 1.04%, 3.52%, 5.44%, and 7.25%, respectively. The FP of the KNN-PSO model is 1.55%, 1.26%, 3.87%, 8.57% and 8.47% respectively.The FP of the five attack types designed in this study is lower. The accuracy of the model in this study reached 98.25%, which is 13.7% higher than the KNN-PSO model. In terms of running time, the model designed in this study was significantly lower than the KNN-PSO model and the traditional CNN model with a running time of 1.48 seconds. The traditional CNN ran for 5.78 seconds, while the KNN-PSO model ran for 17.28 seconds. Compared to the KNN-PSO model, the running time decreased by 91.44%, and compared to thetraditional CNN model, the running time decreased by 74.39%. Overall, the CNN-based SDN attack identification model is more accurate, more generalizable, and runs more quickly. It not only exceeds competing recognition models in the performance of traditional network attack recognition, but it also has a high recognition rate in the specific attack recognition of SDN, demonstrating the model's superiority.

V. CONCLUSION

To address the security risks of SDN, the study designed an SDN attack recognition model based on improved CNN algorithm, which is expected to improve the model attack recognition rate and thus enhance the security performance of SDN. The experimental results of the performance test show higher accuracy and faster convergence when 20 features are selected as input features, i.e., dimension 20 is selected. In terms of algorithm optimization, the accuracy rate of ReLU function used in this study reaches 98.92%, and the accuracy rate of ReLU is significantly higher than that of sigmoid; the accuracy rate of Adam algorithm used in this study reaches 99.62%, and the accuracy rate of Adam algorithm is significantly higher than that of other algorithms; and the accuracy rate of Mean Pooling method used in this study reaches 98.87%, and the accuracy rate of Mean Pooling is significantly higher than that of other methods. The accuracy rate of the method using random deactivation technique is 99.23%, which is higher than the accuracy rate of the fully connected layer. For the validation experiments on model performance, comparing the traditional CNN model with the KNN-PSO model, the results show that the proposed model in this research has a 98.25% AC, 99.13% Recall and only 1.55% FP on SDN attack recognition, which is significantly better than the traditional CNN model. It also shows high accuracy in traditional attack recognition with 98.25%, lower FP and higher Recall in all five attack types, and its performance is significantly better than the KNN-PSO model. In terms of running time, this research design model is significantly lower than the KNN-PSO model and the traditional CNN model, only 1.48 s. Overall, the attack recognition model based on CNN for SDN has higher accuracy, better generalization ability, faster running speed, and higher recognition rate in terms of recognition of unique attacks for SDN, which is better than other models. However, due to the limitation of experimental conditions, this study still has deficiencies, itdoes not make full use of the advantages of the parallel design of CNN algorithm.In the next step of the study, the advantages of the parallel design of CNN will be taken into account to further improve the model, and to enhance the efficiency of the analysis and processing of data.

REFERENCES

[1] E. W. E. Viklund, I. Nilsson, and A. K. Forsman, "Nordic population-based study on internet use and perceived meaningfulness in later life: How they are linked and why it matters," *Scand. J. Public Health*, vol. 50, no. 3, pp. 381–388, May 2022.

[2] N. Ravi and S. M. Shalinie, "BlackNurse-SC: A novel attack on SDN controller," *IEEE Commun. Lett.*, vol. 25, no. 7, pp. 2146–2150, Jul. 2021.

[3] H. Li, J. Lu, J. Wang, H. Zhao, J. Xu, and X. Chen, "*SDM4IIoT*: An SDN-based multicast algorithm for industrial Internet of Things," *IEICE Trans. Commun.*, vol. 105, no. 5, pp. 545–556, May 2022.

[4] D. Tang, Y. Yan, S. Zhang, J. Chen, and Z. Qin, "Performance and features: Mitigating the low-rate TCP-targeted DoS attack via SDN," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 1, pp. 428–444, Jan. 2022.

[5] S. Ravikumar and D. Kavitha, "CNN-OHGS: CNN-oppositional-based Henry gas solubility optimization model for autonomous vehicle control system," *J. Field Robot.*, vol. 38, no. 7, pp. 967–979, May 2021.

[6] M. A. Ouamri, M. Azni, D. Singh, W. Almughalles, and M. S. A. Muthanna, "Request delay and survivability optimization for software defined-wide area networking (SD-WAN) using multi-agent deep reinforcement learning," *Trans. Emerg. Telecommun. Technol.*, vol. 34, no. 7, Jul. 2023, Art. no. e4776.

[7] M. A. Ouamri, G. Barb, D. Singh, and F. Alexa, "Load balancing optimization in software-defined wide area networking (SD-WAN) using deep reinforcement learning," in *Proc. Int. Symp. Electron. Telecommun. (ISETC)*, Timişoara, Romania, Nov. 2022, pp. 1–6.

[8] S. Garg, K. Kaur, G. Kaddoum, S. H. Ahmed, and D. N. K. Jayakody, "SDN-based secure and privacy-preserving scheme for vehicular networks: A 5G perspective," *IEEE Trans. Veh. Technol.*, vol. 68, no. 9, pp. 8421–8434, Sep. 2019.

[9] S. Badotra and S. N. Panda, "SNORT based early DDoS detection system using opendaylight and open networking operating system in software defined networking," *Cluster Comput.*, vol. 24, no. 1, pp. 501–513, Mar. 2021.

[10] U. Ahmed, J. C.-W. Lin, and G. Srivastava, "A resource allocation deep active learning based on load balancer for network intrusion detection in SDN sensors," *Comput. Commun.*, vol. 184, pp. 56–63, Feb. 2022.

[11] A. El Kamel, H. Eltaief, and H. Youssef, "On-the-fly (D)DoS attack mitigation in SDN using deep neural network-based rate limiting," *Comput. Commun.*, vol. 182, pp. 153–169, Jan. 2022.

[12] J. Chen, L. Wang, and S. Duan, "A mixed-kernel, variable-dimension memristive CNN for electronic nose recognition," *Neurocomputing*, vol. 461, pp. 129–136, Oct. 2021.

[13] R. Bao and Z. Yang, "CNN-based regional people counting algorithm exploiting multi-scale range-time maps with an IR-UWB radar," *IEEE Sensors J.*, vol. 21, no. 12, pp. 13704–13713, Jun. 2021.

[14] S. Chen, H. Pei, J. Pisonero, S. Yang, Q. Fan, X. Wang, and Y. Duan, "Simultaneous determination of lithology and major elements in rocks using laser-induced breakdown spectroscopy (LIBS) coupled with a deep convolutional neural network," *J. Anal. At. Spectrometry*, vol. 37, no. 3, pp. 508–516, 2022.

[15] S. Guo, Z. Wang, Y. Lou, X. Li, and H. Lin, "Detection method of photovoltaic panel defect based on improved mask R-CNN," *J. Internet Technol.*, vol. 23, no. 2, pp. 397–406, Mar. 2022.

[16] Y. Xue, Y. Wang, J. Liang, and A. Slowik, "A self-adaptive mutation neural architecture search algorithm based on blocks," *IEEE Comput. Intell. Mag.*, vol. 16, no. 3, pp. 67–78, Aug. 2021.

[17] S. Siddiqui, S. Hameed, S. A. Shah, I. Ahmad, A. Aneiba, D. Draheim, and S. Dustdar, "Toward software-defined networking-based IoT frameworks: A systematic literature review, taxonomy, open challenges and prospects," *IEEE Access*, vol. 10, pp. 70850–70901, 2022.

[18] M. Khalid, S. Hameed, A. Qadir, S. A. Shah, and D. Draheim, "Towards SDN-based smart contract solution for IoT access control," *Comput. Commun.*, vol. 198, pp. 1–31, Jan. 2023.

[19] K. Renuka, D. S. Roy, and K. H. K. Reddy, "An SDN empowered location aware routing for energy efficient next generation vehicular networks," *IET Intell. Transp. Syst.*, vol. 15, no. 2, pp. 308–319, Feb. 2021.

[20] S. Hameed, S. A. Shah, Q. S. Saeed, S. Siddiqui, I. Ali, A. Vedeshin, and D. Draheim, "A scalable key and trust management solution for IoT sensors using SDN and blockchain technology," *IEEE Sensors J.*, vol. 21, no. 6, pp. 8716–8733, Mar. 2021.

[21] M. Robinson, A. M. Johnson, L. K. Fischer, and H. M. MacKenzie, "Two symptoms to triage acute concussions: Using decision tree modeling to predict prolonged recovery after a concussion," *Amer. J. Phys. Med. Rehabil.*, vol. 101, no. 2, pp. 135–138, Feb. 2022.

[22] J. Bhayo, S. A. Shah, S. Hameed, A. Ahmed, J. Nasir, and D. Draheim, "Towards a machine learning-based framework for DDOS attack detection in software-defined IoT (SD-IoT) networks," *Eng. Appl. Artif. Intell.*, vol. 123, Aug. 2023, Art. no. 106432.

[23] D. Morales, E. Talavera, and B. Remeseiro, "Playing to distraction: Towards a robust training of CNN classifiers through visual explanation techniques," *Neural Comput. Appl.*, vol. 33, no. 24, pp. 16937–16949, Dec. 2021.

[24] X. Wu, P. Li, J. Zhou, and Y. Liu, "A cascaded CNN-based method for monocular vision robotic grasping," *Ind. Robot, Int. J. Robot. Res. Appl.*, vol. 49, no. 4, pp. 645–657, Jun. 2022.

[25] Y. Zhang, W. Yan, G. S. Hong, J. F. H. Fuh, D. Wang, X. Lin, and D. Ye, "Data fusion analysis in the powder-bed fusion AM process monitoring by Dempster–Shafer evidence theory," *Rapid Prototyping J.*, vol. 28, no. 5, pp. 841–854, May 2022.

HUIMIN XUE was born in June 1980. She received the master's degree in computer science and technology from Taiyuan Normal University, in July. She is a Lecturer. She is currently with the Shanxi Vocational &Technical College of Finance & Trade. She has published six articles and participated in projects. She mainly engages in the research of computer application technology.

BING JING was born in December 1979. She received the master's degree in computer science and technology from Taiyuan Normal University, in July. She is a Lecturer. She is currently with the Shanxi Vocational &Technical College of Finance & Trade. She has published five articles and participated in two projects. She mainly engages in the research of computer application technology.