Azerbaijan Journal of High Performance Computing, Vol 5, Issue 2, 2022, pp. 212-235
https://doi.org/10.32010/26166127.2022.5.2.212.235

A New Approach to Improve CNN Performance in Anomaly Detection for IoT Networks Based on the Algorithm AdaBoost

Zahra Jahangiri[1], Nasser Modiri[1] and Zahra Tayyebi Qasabeh[2]

[1]*Islamic Azad University, Zanjan, Iran, z.ahangiri.1997@gmail.com, nassermodiri@yahoo.com*
[2]*Payame Noor University of Guilan, Guilan, Iran, tayyebi.shiva@gmail.com*

*Correspondence: Zahra Jahangiri, Islamic Azad University, Zanjan, Iran, z.ahangiri.1997@gmail.com

Abstract

Since the increase in internet attacks brings much damage, it is essential to take care of the security of network activities. networks must use different security systems, such as intrusion detection systems, to deal with attacks. This research proposes a reliable approach for intrusion detection systems based on anomaly networks. The network traffic data sets are large and unbalanced, affecting intrusion detection systems' performance. The imbalance has caused the minority class to be incorrectly identified by conventional data mining algorithms. By ignoring the example of this class, we tried to increase the overall accuracy, while the correct example of the minority class protocols is also essential. In the proposed method, network penetration detection based on the combination of multi-dimensional features and homogeneous cumulative set learning was proposed, which has three stages: the first stage, based on the characteristics of the data, several original datasets of raw data or datasets criteria are extracted. Then, the original feature datasets are combined to form multiple comprehensive feature datasets. Finally, the same basic algorithm is used to train different comprehensive feature datasets for the multi-dimensional subspace of features.

An initial classifier is trained, and the predicted probabilities of all the basic classifiers are entered into a meta-module. In this research, an AdaBoost meta-algorithm has been used for unbalanced data according to a suitable design. Also, various single CNN models and multi-CNN fusion models have been proposed, implemented, and trained. This evaluation is done with the NSL-KDD dataset to solve some of the inherent problems of the KDD'99 dataset. Simulations were performed to evaluate the performance of the proposed model on the mentioned data sets. This proposed method's accuracy and detection rate obtained better results than other methods.

Keyword: Intrusion Detection (anomaly), Internet of Things, CNN, and Adaboost Algorithm.

1. Introduction

Intrusion detection systems are designed to prevent intrusion and protect programs, data, and unauthorized access to computer systems. Intrusion detection systems can classify internal and external intrusions in an organization's computer networks and raise the alarm if there is a security breach in an organization's network. One notable definition for infiltration is that it causes malignant and active external functional disturbances. The primary purpose of intrusion detection systems is to detect a wide range of intrusions, so far detected and unknown attacks, to detect and adapt to unfamiliar attacks and to detect and detect intrusions in a fast pattern (Rincy N, T., & Gupta, R., 2021). The Internet of Things is an interconnected device system that facilitates integrated information exchange between physical devices. These devices can be medical and health devices, driverless vehicles, industrial robots, smart TVs, wearables, and smart city infrastructure, which can be monitored and adjusted remotely. IoT devices are expected to become more common than mobile devices and have access to the most sensitive information, such as personal information. This state leads to an increase in the attack level and increases the probability of attacks. Since security is a critical supporting element for most IoT applications, IoT intrusion detection systems must also be developed to secure communications enabled by such IoT technologies.

In the past few years, advances in artificial intelligence, such as machine learning and deep learning techniques, have been used to improve IoT intrusion detection systems. The current need is to carry out an up-to-date, complete classification and critical review of this recent work. Several related studies used different machine learning and deep learning techniques through different datasets to validate the development of an IoT intrusion detection system. However, it still needs to be determined which dataset, machine learning, or deep learning techniques are more effective for creating an efficient IoT IDS. Second, the time spent building and testing an IoT intrusion detection system is not considered in evaluating some techniques, despite being a critical factor for the effectiveness of "online" intrusion detection systems (Khraisat, A., & Alazab, A., 2021).

Internet of things networks produce multi-dimensional, multimodal, and temporal data due to their heterogeneous structures. It is possible to discover previously unseen trends, reveal similarities, and gain new insights. Artificial intelligence has become more popular in big data processing. Profound learning methods have shown their ability to work with heterogeneous data. It can also analyze dynamic and large-scale data to gain insights, detect data dependencies, Use big data analysis, and learn from past attack patterns to detect current and unknown attack patterns. Heavy functions such as processing big data and building learning models should be loaded on cloud and fog servers because IoT computers are limited in space and have minimal storage and computing capacity. As a result, computational offloading helps to minimize task execution delays and save resources on handheld and battery-powered IoT computers but also increases security issues. Many deep learning methods have been pro-

posed for intrusion detection systems, some directly focused on the Internet of Things. However, there are still a large number of research gaps that still need to be identified from previous solutions. Some of these are (Aljumah, A., 2021):

• Some of these include limited work combining deep learning techniques for intrusion detection systems, focusing on temporal aspects of data.

• Minimal work has been done for heterogeneous data elements to detect attacks.

• The presented approach needs to include the energy efficiency aspect of the intrusion detection system.

• Although researchers have investigated the predictive aspects of the intrusion detection system, there needs to be more focus on the temporal variability of the intrusion detection system.

The term deep learning refers to the accumulation of multiple layers. The input layer is the first layer processed through the final layer to produce the output. Meanwhile, hidden layers are also added. Each layer consists of a group of units called neurons. The dimension of the input data determines the size of the input layer.

In contrast, the output layer consists of N nodes corresponding to N categories in a classification task. This paper proposes five architectural principles that should be considered when designing an accurate and efficient deep learning intrusion detection system for the Internet of Things (TCNN presents a convolution-based version of CNN) (Aljumah, A., 2021). The beneficial aspect of CNN includes minimal dependence on preprocessing, thus reducing the need for human effort to develop its functions. In addition, it is easy to understand and quick to implement. In addition, it has the highest accuracy among all algorithms. Data balancing and practical performance engineering were integrated with TCNN. Key contributions are detailed below (Aljumah, A., 2021).

• The main design concepts can be defined, including overfitting management, creating an IoT intrusion detection system should dataset balance, performance engineering, algorithm optimization, and testing on IoT datasets.

• Time complexity neural networks can be developed, and a deep learning platform can be tested for intrusion detection.

• Practical performance engineering can be used, which includes the following:

(1) Reducing the feature space: It helps to reduce memory usage.

(2) Function transformation: This transforms the skewed data into a Gaussian-like distribution by applying the log transformation and regular scaler to continuous numerical functions. Label-encoding, which replaces a classification column with a specific integer value, is often used in classification properties (Aljumah, A., 2021).

Therefore, using the deep learning approach and the Adaboost algorithm, in this research, an Internet of Things intrusion detection system is proposed, which provides a new method for processing attribute data with weak one-dimensional correlation, and the processed data has a better educational result. They give results for deep learning. Also, the AdaBoost algorithm is used to increase the machine learning algorithm's

performance. A multi-convolutional neural network fusion algorithm based on the AdaBoost algorithm is proposed, and a new research method for intrusion detection is presented using this method.

2. Related Works

Zarpelão et al. (2017)." A survey of intrusion detection in Internet of Things."The Internet of Things (IoT) is a new paradigm that integrates the Internet and physical objects belonging to different domains, such as home automation, industrial process, human health, and environmental monitoring. It deepens the presence of Internet-connected devices in our daily activities, bringing challenges related to security issues and many benefits. For more than two decades, Intrusion Detection Systems (IDS) have been an essential tool for protecting networks and information systems. However, applying traditional IDS techniques to IoT is difficult due to its particular characteristics, such as constrained-resource devices, specific protocol stacks, and standards. In this paper, They present a survey of IDS research efforts for IoT. Our objective is to identify leading trends, open issues, and future research possibilities. They classified the IDSs proposed in the literature according to the following attributes: detection method, IDS placement strategy, security threat, and validation strategy. They also discussed the different possibilities for each attribute, detailing aspects of works that either propose specific IDS schemes for IoT or develop attack detection strategies for IoT threats that might be embedded in IDSs (Zarpelão, B. B., Miani, R. S., Kawakani, C. T., & de Alvarenga, S. C., 2017).

Hajiheidari et al. (2019)." Intrusion detection systems in the Internet of things: A comprehensive investigation." Recently, a new dimension of intelligent objects has been provided by reducing the power consumption of electrical appliances. Daily physical objects have been upgraded by electronic devices over the Internet to create local intelligence and make communication with cyberspace. The Internet of things (IoT), a new term in this domain, is used to realize these intelligent objects. Since the objects in the IoT are directly connected to the unsafe Internet, the resource-constraint devices are easily accessible by the attacker. Such public access to the Internet causes things to become vulnerable to intrusions. The purpose is to categorize the attacks that do not explicitly damage the network. However, by infecting the internal nodes, they are ready to carry out attacks on the network, called internal attacks. Therefore, the significance of Intrusion Detection Systems (IDSs) in the IoT is undeniable. However, despite this topic's importance, there needs to be a comprehensive and systematic review discussing and analyzing its effective mechanisms. Therefore, the current paper presents a Systematic Literature Review (SLR) of the IDSs in the IoT environment. Then detailed categorizations of the IDSs in the IoT (anomaly-based, signature-based, specification-based, hybrid), (centralized, distributed, hybrid), (simulation, theoretical), (denial of service attack, Sybil attack, replay attack, selective forwarding attack, wormhole attack, black hole attack, sinkhole attack, jamming attack, false data attack) have also

been provided using standard features. Then the advantages and disadvantages of the selected mechanisms are discussed. Finally, the examination of the open issues and directions for future trends are also provided (Hajiheidari, S., Wakil, K., Badri, M., & Navimipour, N. J., 2019).

Nimbalkar et al. (2021)." Feature selection for intrusion detection system in Internet-of-Things (IoT)." The Internet of Things (IoT) suffers from different attacks due to device vulnerabilities. Due to many IoT network traffic features, machine learning models take time to detect attacks. This paper proposes a feature selection for intrusion detection systems (IDSs) using Information Gain (IG) and Gain Ratio (GR) with the ranked top 50% features for the detection of DoS and DDoS attacks. The proposed system obtains feature subsets using insertion and union operations on subsets obtained by the ranked top 50% IG and GR features. The proposed method is evaluated and validated on IoT-BoT and KDD Cup 1999 datasets, respectively, with a JRipclassifier. The system performs better than the original feature set and traditional IDSs on IoT-BoT and KDD Cup 1999 datasets using 16 and 19 features, respectively (Nimbalkar, P., & Kshirsagar, D., 2021).

Atul et al. (2021)." A machine learning-based IoT for providing an intrusion detection system for security." Digital communication is provided an effective communication platform for sharing and transferring information. The emergence of the Cyber-Physical System (CPS) is a platform incorporated with electronic devices that enables services through a digital platform. The considerable challenges of this system are security issues, abnormality, and service failure. Hence, the requirement to provide an effective system should overcome these issues. This paper analyzes these problems and provides an enhanced communication paradigm, specifically proposing Energy Aware Smart Home (EASH) framework. This work analyzes the problem of communication failures and types of network attacks in EASH. With the utilization of the machine learning technique, the abnormality sources of the communication paradigm are differentiated. We analyze the proposed work based on its accuracy, performance, and efficiency to evaluate the performance. Hence, we obtain better results, especially when the result shows an 85% accuracy rate. In the future, we will enhance a high accuracy rate for further development (Atul, D. J., Kamalraj, R., et al., 2021).

Sicari et al. (2015)." security, privacy and trust in Internet of Things: The road ahead." The Internet of Things (IoT) is characterized by heterogeneous technologies, which concur with providing innovative services in various application domains. In this scenario, the satisfaction of security and privacy requirements plays a fundamental role. Such requirements include data confidentiality and authentication, access control within the IoT network, privacy and trust among users and things, and the enforcement of security and privacy policies. Due to the different standards and communication stacks, traditional security countermeasures cannot be directly applied to IoT technologies. Moreover, the high number of interconnected devices raises scalability issues; therefore, a flexible infrastructure is needed to deal with security threats in such a dy-

namic environment. In this survey, we present the main research challenges and the existing solutions in the field of IoT security, identifying open issues and suggesting some hints for future research (Sicari, S., Rizzardi, A., Grieco, L. A., & Coen-Porisini, A., 2015).

Costa and et al (2019). "Internet of Things: A survey on machine learning-based intrusion detection approaches." In the world scenario, concerns with security and privacy regarding computer networks are constantly increasing. Computer security has become necessary due to the proliferation of information technologies in everyday life. The increase in the number of Internet accesses and the emergence of new technologies, such as the Internet of Things (IoT paradigm, are accompanied by new and modern attempts to invade computer systems and networks. Companies are increasingly investing in studies to optimize the detection of these attacks. Institutions are selecting intelligent techniques to test and verify by comparing the best accuracy rates. This research, therefore, focuses on rigorous state-of-the-art literature on Machine Learning Techniques applied in Internet-of-Things and Intrusion Detection for computer network security. The work aims to conduct recent and in-depth research of relevant works that deal with several intelligent techniques and their applied intrusion detection architectures in computer networks, emphasizing the Internet of Things and machine learning. More than 95 works on the subject were surveyed, spanning different themes related to security issues in IoT environments (da Costa, K. A., Papa, J. P., et al., 2019).

Li et al. (2020). "Robust detection for network intrusion of industrial IoT based on multi-CNN fusion." A robust intrusion detection system plays a critical role in network security. Traditional machine learning methods must be revised due to complex network data and diverse intrusion methods. They cannot meet the requirements of the current network environment. Existing deep learning-based methods are far from fully exploiting their potential in dealing with such one-dimensional feature data, and their performance still needs to improve in detecting unknown intrusions. This paper proposes a deep learning approach for intrusion detection using a multi-convolutional neural network (multi-CNN) fusion method. According to the correlation, the feature data are divided into four parts, and then the one-dimensional feature data are converted into a grayscale graph. Using the flow data visualization method, CNN is introduced into the intrusion detection problem, and the best of the four results emerge. The experimental results successfully demonstrate that the multi-CNN fusion model is suitable for providing a classification method with high accuracy and low complexity on the NSL-KDD dataset. Furthermore, its performance is superior to traditional machine learning methods and other recent deep learning approaches for binary and multi-class classification. This work contributes to the data security of industrial IoT (Li, Y., Xu, Y., et al., 2020).

Kodyš and et al (2021). "Intrusion Detection in Internet of Things using Convolutional Neural Networks." The Internet of Things (IoT) has become a popular paradigm to fulfill the needs of the industry, such as asset tracking, resource monitoring, and

automation. As security mechanisms are often neglected during the deployment of IoT devices, they are more easily attacked by complicated and large-volume intrusion attacks using advanced techniques. Artificial Intelligence (AI) has been used by the cyber security community in the past decade to identify such attacks automatically. However, deep learning methods have yet to be extensively explored for Intrusion Detection Systems (IDS) specifically for IoT. Most recent works are based on time-sequential models like LSTM, and there needs to be more research on CNNs as they are not naturally suited for this problem. In this article, They propose a novel solution to the intrusion attacks against IoT devices using CNNs. The data is encoded as the convolutional operations to capture the patterns from the sensors data for a long time that are useful for attack detection by CNNs. The proposed method is integrated with two classical CNNs: ResNet and EfficientNet, where the detection performance is evaluated. The experimental results show significant improvement in both actual positive and false favorable rates compared to the baseline using LSTM (Kodyš, M., Lu, Z., Fok, K. W., & Thing, V. L., 2021, December).

3. Our Model

An intrusion detection system (IDS) is a network security device that monitors network traffic in real time and can alert or take proactive actions when suspicious transmissions are discovered. It differs from other network security devices in that an IDS can identify an invasion, which could be an ongoing invasion or an intrusion that has already occurred. Intrusion detection is usually modeled as a binary classification problem that identifies whether the network traffic behavior is normal or anomalous or as a multi-class classification problem that identifies network traffic behavior and determines the type of network attack. In short, the primary motivation of intrusion detection is to improve the accuracy of classifiers in effectively identifying intrusive behavior. In recent years, the application of deep learning to solve the network intrusion detection problem has been a relatively new area of research. Deep learning has the potential to extract better representation from massive data and get much better results.

Furthermore, convolutional neural networks (CNNs) have recently generated significant developments in deep learning. We propose a multiCNN fusion-based intrusion detection system using the deep learning approach. The proposed method proposes a new method for processing feature data with a one-dimensional weak correlation. The processed data has a better training result for deep learning. Using the above processing method, a multi-CNN and AdaBoost fusion algorithm are used on the NSLKDD benchmark data set to evaluate and classify features and presents a new research method for intrusion detection in which AdaBoost algorithms are used to achieve a high detection rate (DR) with a low false positive rate (FPR).

The widespread use of information technology and the ever-increasing development of cyberspace have enriched amateur life and broadened the horizons of vision. Still, the large amount of network traffic information that results from people's heavy

reliance on cyberspace brings security and management issues. Many cyberspace security problems appear, bringing many potential threats to our online life. In particular, attackers commonly exploit widespread software vulnerabilities to attack computer systems on the network. The damage caused by these attacks may cause serious problems, such as service interruption or even significant financial losses. In traditional neural network models, data flows from the input layer to the hidden layer to the output layer. Layers are fully connected, and there are no connections between nodes in the same layer. Therefore, traditional neural networks have many problems that need to be solved (Li, Y., Xu, Y., et al., 2020).

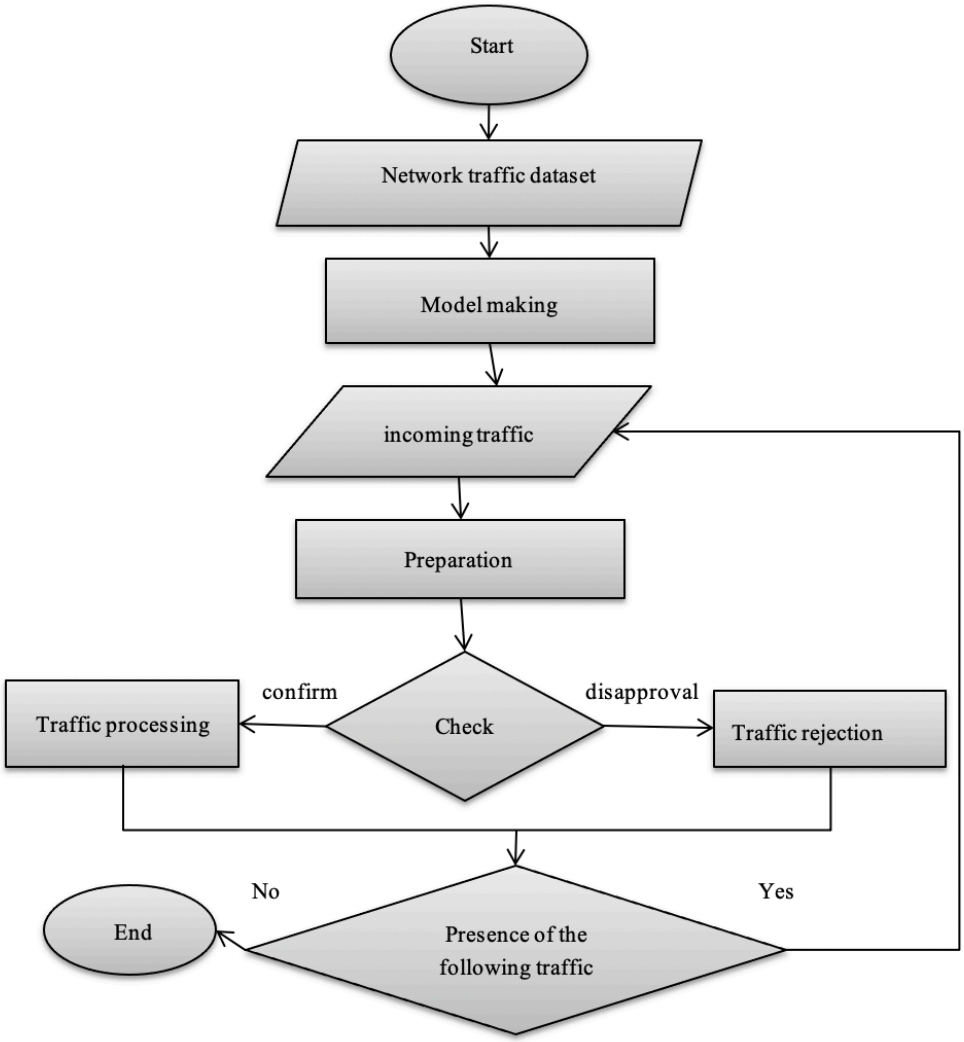

Fig.1: Flowchart of the proposed method

Convolutional neural networks, which improve the typical neural network architecture, have made significant achievements in speech analysis and image classification in recent years. A CNN consists of one or more convolution layers, pooling layers on top, and thoroughly connected layers and output layers that act as regularization layers. This structure enables the convolutional neural network to exploit the two-dimensional structure of the input data. Therefore, an image can be directly used as the network's input, thus avoiding the complex feature extraction and data reconstruction operations involved in traditional recognition algorithms. Through sparse connection, joint weights, and integration, the difficulty of manual data processing can be effectively reduced, and the modeling efficiency can be significantly improved. CNN can learn multiple levels of features from a large amount of unlabeled data. Therefore, the application perspective of CNN in network intrusion detection is vast (Li, Y., Xu, Y., et al., 2020).

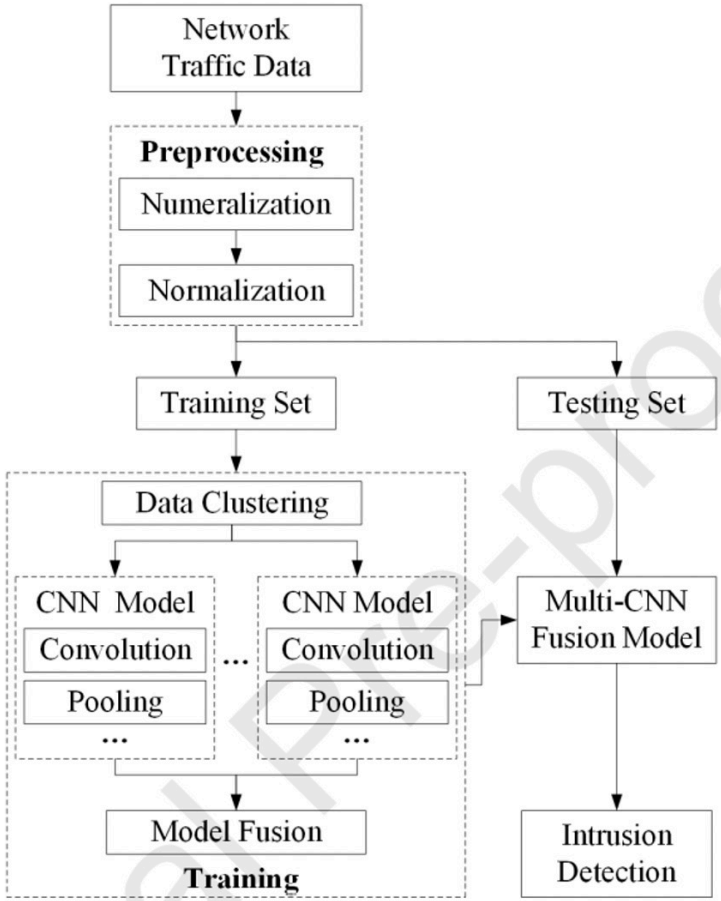

Fig.2: Block diagram of the proposed IDS (Li, Y., Xu, Y., et al., 2020).

The flowchart of the proposed method is shown in Figure 1. Based on this flowchart, the proposed method, first using network traffic data, its model is built based on combined CNN algorithms with AdaBoost. Each incoming traffic data is adapted to this model. If this traffic is accepted, it is accepted, and if Non-acceptance of this traffic data is rejected, the following incoming traffic, if there is one, be processed for it. This section describes in detail the hybrid multi-CNN architecture used in this study and the methodology used to develop intrusion detection models. The diagram of the proposed method is shown in Figure 2. The steps involved in this chapter include a description of the dataset, data preprocessing, specific methodology, and evaluation criteria (Li, Y., Xu, Y., et al., 2020).

Dataset description: The NSL-KDD dataset was generated in 2009 and is widely used in network intrusion detection experiments. In the latest literature, all the researchers use the NSL-KDD dataset as an adequate baseline dataset that can help researchers compare different intrusion detection methods. It addresses the inherent problems of the KDD CUP 99 dataset [30], which was generated in 1999 from the DARPA98 network traffic (Li, Y., Xu, Y., et al., 2020).

Numeralization The NSL-KDD dataset contains 34 numeric features and 7 character features. The features need to be numerical to convert one-dimensional feature data into a grayscale image.

Normalization Although the processed features are already trainable, the numerical differences in the records are significant, affecting the model's convergence speed and training effect. Therefore, the dataset needs to be normalized so that the data in the sample falls within the range of [0, 1]. Since the datasets contained both normal and anomalous traffic, we need to avoid the negative influence of the sample mean and variance. A simple linear normalization process can be used for general numerical features, as shown in (1) (Li, Y., Xu, Y., et al., 2020).

$$X' = (X - X_{min})/(X_{max} - X_{min}) \quad (1)$$

Xmax denotes the maximum value, and Xmin denotes the minimum value from all data for each feature. Logarithmic normalization is required for the features' duration', 'src_bytes', and 'dst_bytes' where the data ranges are significant (Li, Y., Xu, Y., et al., 2020).

Data clustering The existing deep learning-based intrusion detection methods usually directly map the preprocessed one-dimensional numerical features into corresponding two-dimensional matrices and fill the redundant parts with zeros [33]. Although this method is simple and straightforward, it ignores a fundamental issue – the added relevance (Aljumah, A., 2021). The transformed two-dimensional matrix is similar to a grayscale image and inevitably imposes a correlation in the vicinity of the matrix elements. This state seriously affects the model training and weakens the model's adaptability (Li, Y., Xu, Y., et al., 2020).

Convert to matrix Since the convolutional neural network has a better image pro-

cessing ability, this paper converts the inputs into the form of images. The advantages of CNNs can be better exploited by transforming intrusion detection problems into image classification problems

3. Training model For the different parts of the dataset, we use the same CNN structure. The architecture of the CNN model implemented for intrusion detection in the binary classification and the dimensions of each layer are shown in Fig. 3, taking the first part of the dataset as an example.

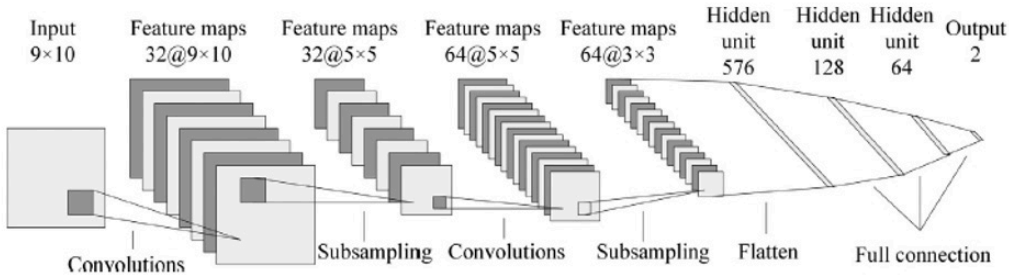

Fig. 3: An example of the architecture of the single CNN model(Li, Y., Xu, Y., et al., 2020).

Evaluation of a feature subset can be done using one of the following methods: filter method or coverage method. In summary, a reliable approach for IDS based on an anomaly network is proposed. Network traffic datasets are large and unbalanced, thus affecting IDS performance. The imbalance causes the minority class misidentified by conventional data mining algorithms. By ignoring the instance of this class, they try to increase the overall accuracy, while the correct instance of minority class protocols is also essential. Therefore, in the proposed approach, the AdaBoost meta-algorithm is used for unbalanced data according to a suitable design.

On the other hand, various single CNN models and a multi-CNN fusion model were proposed, implemented, and trained. These models are trained using the KDDTrain+ dataset, which can be used in optimizing IDS problems. The proposed algorithm has been used for detecting network connections due to the high ability of these algorithms to select the best subset of relevant features (Li, Y., Xu, Y., et al., 2020). The NSL-KDD dataset contains the KDDTrain+ dataset as the training set for model learning and the KDDTest+ and KDDTest-21 datasets as the testing sets for the performance evaluation of trained models. The KDDTest-21 contains records for attack types not in the KDDTrain+ and KKDDTest+ datasets, making classification more difficult. All the models in this paper are trained using the KDDTrain+ dataset and tested using the KDDTest+ and KDDTest-21 datasets, respectively. There are five categories in the NSL-KDD dataset: regular, denial of service (DoS) attacks, remote to local (R2L) attacks, user-to-root (U2R) attacks, and probing (Probe) attacks. The numbers of records for the different attack categories in the dataset are shown in Table 1.

Table 1: Numbers of records in the NSL-KDD dataset

	Total	Normal	Dos	Probe	R2L	U2R
KDDTrain+	125973	67343	45927	11656	995	52
KDDTest-21+	22544	9711	7548	2421	2754	200
KDDTest-21	11850	2152	4342	2402	2754	200

Evaluation metrics: In this paper, the most critical performance indicator (Accuracy) of network intrusion detection is used to measure the performance of the multi-CNN fusion model. In addition, the precision, recall, false positive rate, and F-score are also used in this paper. The True Positives (TPs) are the number of anomalous records identified as anomalies. The True Negatives (TNs) are the number of regular records identified as usual. The False Positives (FPs) are the number of regular records identified as an anomaly. The True Negatives (TNs) are the number of anomalous records identified as usual. The performance indicators that are used in this paper are defined as follows.

Accuracy: the percentage of correctly classified records to the total number of records, as shown in (2) (Li, Y., Xu, Y., et al., 2020).

$$Accuracy=(TP+TN)/(TP+TN+FP+FN) \quad (2)$$

call: equal to the true positive rate (TPR), which is the percentage of the number of correctly identified records divided by the total number of anomalous records, as shown in (3)

$$Recall=TP/(TP+FN) \quad (3)$$

False positive rate (FPR): equal to the false alarm rate (FAR): the percentage of the number of incorrectly identified records divided by the total number of regular records, as shown in (4).

$$FPR=FP/(FP+TN) \quad (4)$$

The precision measures the number of correct classifications and is penalized by the number of incorrect classifications, as shown in (5).

$$Precision=TP/(TP+FP) \quad (5)$$

The F-score is used to measure the harmonic mean of the precision. It also recalls, which serves as a derived effectiveness measurement, as shown in (6).

$$F-Score=\frac{2\times Precision\times Recall}{Precision+Recall} \quad (6)$$

Results

The proposed method has been evaluated in this previous section. This evaluation is done with the NSL-KDD dataset, a dataset proposed to solve some of the inherent problems of the KDD'99 dataset mentioned in [20]. This new version of the KDD dataset still needs some of the problems discussed by McHugh [21]. It may not be fully representative of the actual networks available due to the lack of public datasets for network-based IDSs; It is believed that it can still be used as a compelling benchmark dataset to help researchers compare different intrusion detection methods. Moreover, the number of records in the NSL-KDD train and test sets is reasonable. This advan-

tage makes it cost-effective to run tests on the entire set without having to select a small portion randomly. As a result, the evaluation results of different research works are consistent and comparable. The evaluation results are discussed in the next section.

The software used in the experiment consists of one of the latest and most straightforward deep learning frameworks – Keras on the backend of Tensorflow. The experiment is performed on a Dell Inspiron 3670 personal computer with an Intel Core i7- 8700 CPU @ 3.20 GHz, 8 GB of memory, and an NVIDIA GeForce 1050 Ti GPU for graphics acceleration. The experiments have been designed to study the performance of the multi-CNN fusion model for binary classification (standard and anomaly) and multi-class classification (regular, dos, probe, r2l, and u2r). In the binary classification experiments, we first show the performance of a single CNN with four separate pieces of data and compare the result with those of conventional machine learning methods. Then, we perform the multi-CNN fusion model and compare the result with the latest algorithms and methods. We also give detailed performance and comparisons for our model in the multi-class classification experiments.

Deep Learning Toolbox™ It provides a framework for designing and implementing deep neural networks with algorithms, pre-trained models, and programs. You can use convolutional neural networks (ConvNets, CNN) and short-term memory (LSTM) networks to perform classification and regression on images, time series, and text data. Using automatic differentiation, custom training loops, and joint weights, you can build network architectures such as adversarial networks (GANs) and Siamese networks. With the Deep Network Designer application, you can graphically design, analyze and train networks. The Experiment Manager app helps you manage multiple deep-learning experiments, track training parameters, analyze results, and compare code from different experiments. You can visualize the activation of layers and monitor the training progress graphically. Convolutional Neural Networks (ConvNets) are widely used tools for deep learning. They are particularly suitable for images as input, but they are also used for other applications such as text, signals, and continuous responses. They differ from other types of neural networks in several ways: Convolutional neural networks are inspired by the biological structure of the visual cortex, which consists of an arrangement of simple and complex cells (Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A., 2009, July).

It has been found that these cells are activated based on subregions of a visual field. These sub-areas are called receptive fields. Inspired by the findings of this study, neurons in a convolutional layer are connected to subregions of previous layers instead of being fully connected like in other types of neural networks. Neurons do not respond to regions outside these subregions in the image. These subregions may overlap, so neurons in a ConvNet produce spatially correlated outputs, whereas, in other types of neural networks, neurons are uncorrelated and produce independent outputs.

Furthermore, in a neural network with fully connected neurons, the number of pa-

rameters (weights) can increase rapidly as the input size increases. A convolutional neural network reduces the number of parameters by reducing the number of connections, joint weights, and downsampling. A ConvNet consists of several layers, such as convolution, max-pooling, or average-pooling, and fully connected layers.

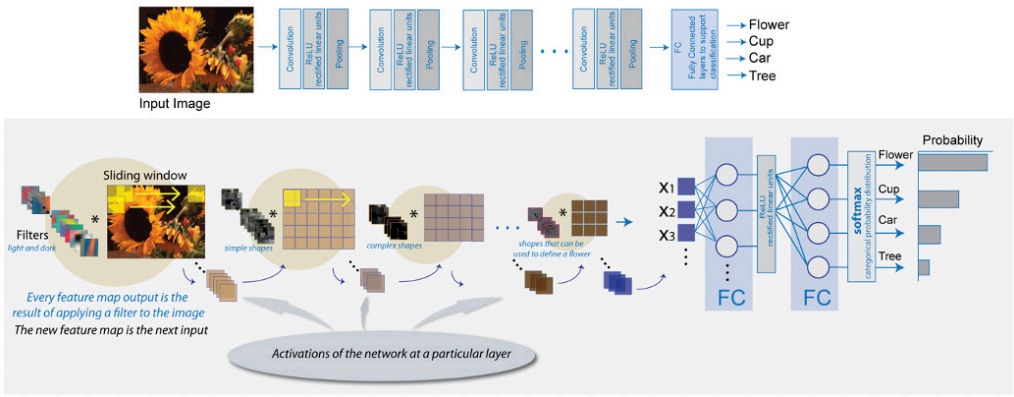

Fig.4: Structure of layers and layout of CNN

Neurons in each layer of ConvNet are arranged in 3D and transform a 3D input into a 3D output. For example, for image input, the first layer (input layer) holds images as 3D inputs with dimensions of height, width, and image color channels. Neurons in the first convolution layer connect to the regions of these images and convert them into a 3D output. Each layer's hidden units (neurons) learn the nonlinear combinations of the main inputs, called feature extraction. These learned features, also known as activations, are transformed from one layer to the input of the next layer. Finally, the learned features become input to the classifier or regression function at the network's end. The architecture of a ConvNet can be different depending on the type and number of layers. The types and number of layers included depend on the specific application or data. For example, suppose you have categorical responses. In that case, you should have a classification function and a classification layer, while if your response is continuous, you should have a regression layer at the end of the network. A smaller network with only one or two convolutional layers may be sufficient to learn a small amount of grayscale image data. On the other hand, for more complex data with millions of color images, you may need a more complex network with multiple fully connected convolutional layers. You can connect the layers of a convolutional neural network in MATLAB in the following way:

```
layers = [imageInputLayer([28 28 1])
          convolution2dLayer(5,20)
          Realplayer
          maxPooling2dLayer (2, 'Stride,' 2)
          fully-connected layer (10)
```

```
softmax layer
classificationLayer];
```

After defining the layers of your network, you must specify the training options using the training options function. For example,

```
options = trainingOptions('sgdm');
```

Then, you can train the network with your training data using the train network function. Data, layers, and training options become inputs to the training function. For example:

```
convnet = train network(data,layers,options);
```

We have mapped 41-dimensional features into 121-dimensional features and then divided them into four parts. We train and test these data using the single CNN model separately. The CNN model has 90 input nodes and 2 output nodes for the first part of the data. For the second part of the data, the CNN model has 121 input nodes and 2 output nodes. For the third part of the data, the CNN model has 81 input nodes and 2 output nodes. The CNN model has 100 input and 2 output nodes for the fourth part data. The number of epochs is given as 1000. For a better description, CNN1, CNN2, CNN3, and CNN4 are used to represent the CNN models trained using the data's first, second, third, and fourth parts. Moreover, CNN0 represents the CNN model that is trained using all the data. To better train the model, the learning rate is set as 0.001, and we use the KDDTrain+ dataset in the NSL-KDD for training.

Then, the classification accuracies for the NSL-KDD dataset using the KDDTest+KDDTest+ and KDDTest21 are shown in Table 2. The experiments show that the CNN model works and has a reasonable detection rate when trained using only part of the data. The CNN1 model obtains 82.62% accuracy for the KDDTest+ dataset and 67.22% accuracy for the KDDTest-21 dataset, which are better than that of the CNN0 model, which is trained using all the data. This state proves that better classification results can be obtained using more correlated partial data than all data with low-level relationships between the global features. Note that the sharp difference in the performance between the KDDTest+ and KDDTest-21 datasets is because the KDDTest-21 contains some attack types that do not exist in the KDDTrain+ and KDDTest+KDDTest+ datasets.

Table 2. The accuracies of the different single CNN models on the KDDTest+ and KDDTest21 datasets.

Model	Accuracy of KDDTest+	Accuracy of KDDTest-21
CNN0	78.29%	58.82%
CNN1	82.62%	67.22%
CNN2	77.48%	62.12%
CNN3	76.55%	55.76%
CNN4	76.59%	56.04%

It is obtained by the J48, Naive Bayesian, NB Tree, Random Forest, Random Tree, Multi-layer Perceptron, and SVM in intrusion detection. Fortunately, these results are based on the same benchmark – the NSL-KDD dataset. The performance of the CNN1 model that uses the first part of the data is superior to other machine learning classification algorithms in binary classification, as shown in Fig. 5. Surprisingly, by using only the other part of the data, we have also achieved almost the same accuracy that traditional machine learning can achieve. This state proves the massive potential for convolutional neural networks to detect network intrusion. It could use fewer features from the data to get better results.

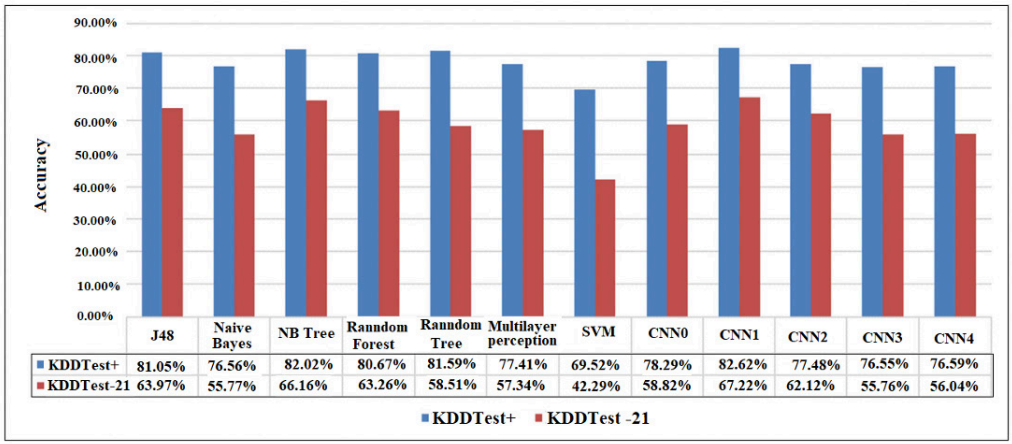

Fig. 5: Performance of the single CNN model and other traditional machine learning models in the binary classification.

We merge the multi-CNN fusion model to obtain better results and test the result. This state is the method we provided in Section III; the optimal prediction is obtained by fusing the results of the four single CNN models. In our experiment, the model obtains a higher accuracy. Table 3 and Table 4 show the confusion matrix of the multi-CNN fusion model on the KDDTest+ and KDDTest-21 testing sets in the binary classification experiment. We obtain 86.95% accuracy for the KDDTest+ dataset and 76.67% for the KDDTest-^{21} dataset.

Table 3: The confusion matrix of the multi-CNN fusion model on the KDDTest-$^{21+}$ testing set in binary classification.

Predicted Class	Actual Class	
	Anomaly	Normal
Anomaly	11198	1306
Normal	1635	8405

Table 4: The confusion matrix of the multi-CNN fusion model on the KDDTest-21 testing set in binary classification.

Predicted Class	Actual Class	
	Anomaly	Normal
Anomaly	8067	1134
Normal	4631	1018

RNN and the DCNN results are given in (Naseer, S., Saleem, Y., et al., 2018). Fortunately, all of these models are trained using the KDDTrain+ dataset and tested using both the KDDTest+ and KDDTest-21 datasets. While the models STL-IDS (Al-Qatf, M., Lasheng, Y., Al-Habib, M., & Al-Sabahi, K., 2018), Fuzziness (Ashfaq, R. A. R., Wang, X. Z., et al., 2017), and auto-encoder (AE) (Yousefi-Azar, M., Varadharajan, V., Hamey, L., & Tupakula, U., 2017, May) only provide their performance using the KDDTest+ datasets. As shown in Table 5, the accuracy of the multi-CNN fusion model is superior to the other latest classification algorithms based on deep learning in binary classification.

Table 5:The accuracies of the multi-CNN fusion model and the other latest algorithm models in binary classification.

Model	Accuracy of KDDTest-21	Accuracy of KDDTest+
DCNN	85.00%	70.00%
RNN	83.28%	68.55%
STL-IDS	84.96%	-
Fuzziness	84.12%	-
AE	83.34%	-
Base model	86.95%	876.67%

The performance of our model has reached or exceeded the average levels of the other state-of-the-art approaches and methods in binary classification. In addition, Fig. 6 shows the detailed performance metrics of our model.

Note that the number of different categories of attacks is very different and seriously affects the multi-class classification. Therefore, in the five-category classification experiment, we use the practical method of weighting the loss function generated by the various attack categories. Specifically, the reciprocal of the proportion of the sample size of each attack category is used as the weight. The overall accuracies of the CNN1, CNN2, CNN3, and CNN4 models are shown in Fig. 2. The results of some standard machine learning algorithms are also provided to compare their performances using the same benchmark dataset for the multi-class classification experiments. The CNN1 model achieved an accuracy of 78.30% using the KDDTest+ and 61.15% using the KDDTest-21 dataset, which is superior to the other algorithms. Meanwhile, we also noticed that several other single CNN models' performance could be better. The main

reason is that the partial features need to be more comprehensive to distinguish the attack types effectively, and the proportions of attack types in the training and testing

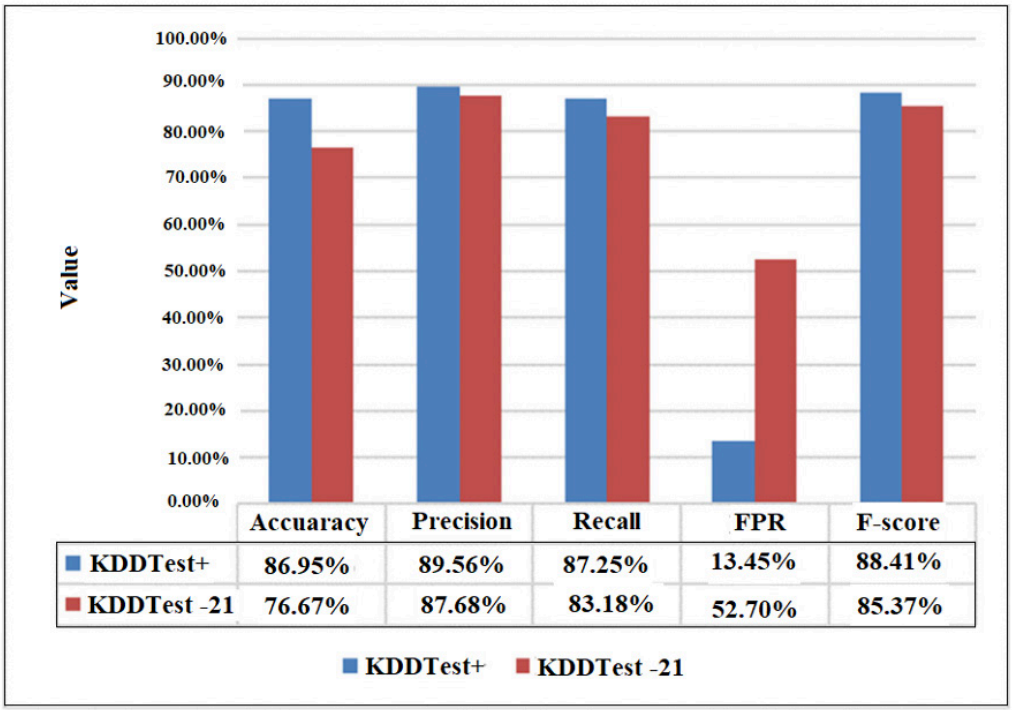

Fig. 6: The detailed performance of the multi-CNN fusion model in the binary classification.

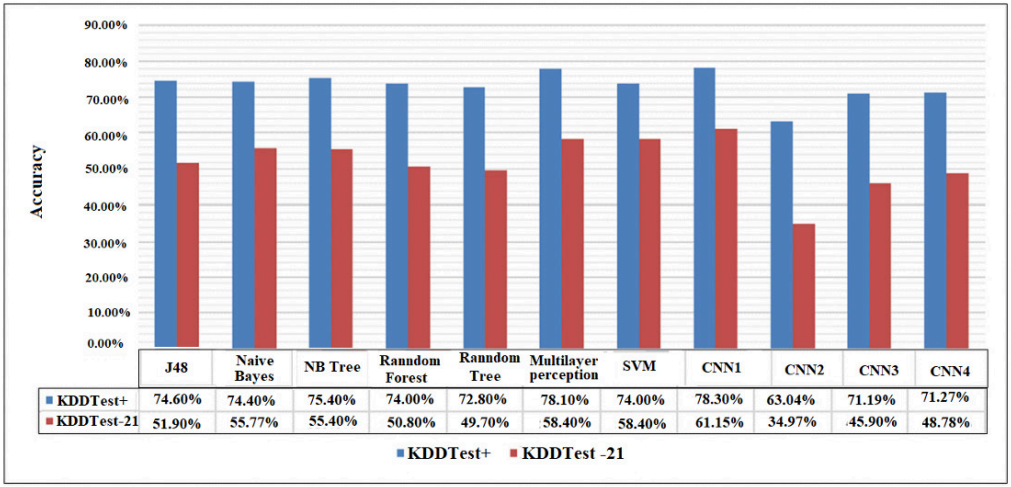

Fig. 7: Performance of the single CNN model and other traditional machine learning models in multi-class classification.

set also significantly differ.

Further, we test the multi-CNN fusion model in a multi-class classification experiment, and the confusion matrices on the KDDTest+ and KDDTest-21 testing sets are shown in Tables 6 and 7, respectively. The experiments result in an accuracy of 81.33% for the KDDTest+ dataset and an accuracy of 64.81% for the KDDTest-211 dataset.

Table 6: The confusion matrix for the multi-class classification using the KDDTest+ data

Predicted class	Actual class				
	Normal	Dos	Probe	R2L	U2R
Normal	872	872	117	1587	36
Dos	473	6461	177	1	0
Probe	288	47	2003	27	109
R2L	56	0	107	968	8
U2R	39	78	17	171	47

Table 7: The confusion matrix for the multi-class classification using the KDDTest-21 data

Predicted class	Actual class				
	Normal	Dos	Probe	R2L	U2R
Normal	1336	872	117	1587	36
Dos	473	3345	177	1	0
Probe	268	47	1984	27	109
R2L	42	0	107	968	8
U2R	33	78	17	171	47

In the same way, we compare the performance of the multi-CDN fusion model with the latest methods in the five-category classification experiment. The ANN algorithm in (Ingre, B., & Yadav, A., 2015, January) resulted in an overall accuracy of 79.90% using the KDDTest+ dataset, and the authors did not provide a result using the KDDTest-21 dataset. Yin et al. claimed their RNN model obtained 81.29% and 64.67% detection accuracies using the KDDTest+ and KDDTest-21 datasets, respectively. Majjed et al. also claimed that their self-taught learning (STL)-IDS model achieved an accuracy of 80.48% for KDDTest+. Javaid et al. provided an overall accuracy of 79.10% using a sparse auto-encoder (SAE) on the KDDTest+ dataset. As shown in Table 8, our model achieves the best result on the testing sets. We note that references do not provide their performances using the KDDTest-21 dataset.

Table 8: The accuracies of the multi-CNN fusion model and the other latest algorithm models in multi-class classification.

Model	Accuracy of KDDTest-21	Accuracy of KDDTest+
RNN	81.29%	64.67%

SAE	79.10%	-
STL-IDS	80.48%	-
ANN	79.90%	-
Base model	81.33%	64.81%

Traditional machine learning methods are better than traditional machine learning methods, but the advantages are not outstanding compared to other latest deep learning-based methods. The main reason is that the number of samples of different attack categories in the training set varies greatly, especially R2L and U2R, as shown in Table 1. Therefore, the performance of the multi-class classification of different algorithms on the NSL-KDD data set is not particularly desirable, and the advantages of the proposed method still need to be fully reflected.

In addition, due to the significant differences between the numbers of the five attack categories in the testing set, the performances cannot be well described using the total accuracy alone. References (Ingre, B., & Yadav, A., 2015, January; Yin, C., Zhu, Y., Fei, J., & He, X., 2017) provided the detection rate of each classification in the RNN model and ANN model. As shown in Fig. 8, although the overall accuracies of the three methods are not much different, the performance of our model is better concerning the detection rate for each attack category. Our model can detect attack categories more effectively with few training samples, such as R2L and U2R.

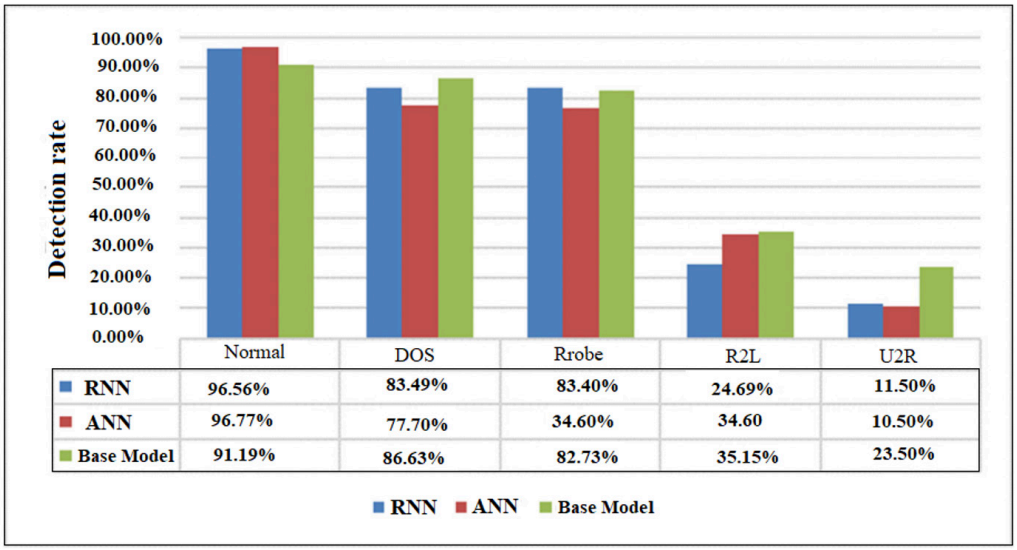

Fig. 8: Performance of five categories classification for the RNN model, ANN model, and the base model

The performance of our model has reached or exceeded the average levels of the other state-of-the-art approaches and methods. In addition, like with the binary clas-

sification, the detailed performance metrics of our model for multi-class classification on the KDDTest+KDDTest+ and KDDTest-21 testing sets are shown in Table 9 and Table 10, respectively.

Table 9: The detailed performance of the multi-CNN fusion model on the KDDTest+ dataset

Label	Accuracy	Precision	Recall	False alarm	F-score
Normal	84.62%	77.22%	91.19%	20.35%	83.62%
Dos	92.69%	90.85%	86.63%	4.32%	88.69%
Probe	96.06%	80.96%	82.73%	2.34%	81.84%
R2L	91.32%	84.99%	35.15%	0.01%	49.73%
U2R	97.97%	13.35%	23.50%	1.37%	17.03%

Table 10: The detailed performance of the multi-CNN fusion model on the KDDTest-21 data

Label	Accuracy	Precision	Recall	False alarm	F-score
Normal	71.07%	33.84%	62.08%	26.93%	43.80%
Dos	86.09%	83.71%	77.04%	8.67%	80.24%
Probe	92.67%	81.48%	82.60%	4.77%	82.03%
R2L	83.60%	86.04%	35.15%	1.73%	49.91%
U2R	96.19%	13.58%	23.50%	2.57%	17.22%

The performance of the Adaboost part is shown as an example in Figures 9 and 10 on the KDD Test and KDD Test 21 datasets, respectively.

```
Time taken to build model: 14.04 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       20374               90.3744 %
Incorrectly Classified Instances      2170                9.6256 %
Kappa statistic                          0.8051
Mean absolute error                      0.1379
Root mean squared error                  0.2627
Relative absolute error                 28.1191 %
Root relative squared error             53.0465 %
Coverage of cases (0.95 level)          99.1262 %
Mean rel. region size (0.95 level)      73.6493 %
Total Number of Instances            22544

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.918     0.107      0.867     0.918     0.891      0.968    normal
                 0.893     0.082      0.935     0.893     0.914      0.968    anomaly
Weighted Avg.    0.904     0.093      0.905     0.904     0.904      0.968

=== Confusion Matrix ===

     a     b   <-- classified as
  8913   798 |     a = normal
  1372 11461 |     b = anomaly
```

Fig. 9: Implementation of the improved Adaboost algorithm on the KDDTest+ dataset

```
Time taken to build model: 6.22 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       10733               90.5738 %
Incorrectly Classified Instances      1117                9.4262 %
Kappa statistic                          0.6555
Mean absolute error                      0.1487
Root mean squared error                  0.2665
Relative absolute error                 50.0089 %
Root relative squared error             69.1216 %
Coverage of cases (0.95 level)          98.9873 %
Mean rel. region size (0.95 level)      72.9367 %
Total Number of Instances            11850

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.638    0.035      0.802     0.638     0.711      0.925    normal
                 0.965    0.362      0.923     0.965     0.944      0.925    anomaly
Weighted Avg.    0.906    0.302      0.901     0.906     0.901      0.925

=== Confusion Matrix ===

    a    b   <-- classified as
 1374  778 |    a = normal
  339 9359 |    b = anomaly
```

Fig. 10: Implementation of the improved Adaboost algorithm on the KDDTest[21] dataset

As shown in Figures 9 and 10, this method has been performed on the test sets and has worked with accuracy and other parameters of about 90%, according to these results compared to the primary method, which is based on the basic CNN method and It has been used without any improvement. The results have been around 81, and an improvement of 9% has been recorded. In Figure 11, the results of the proposed method are compared with the primary method. As shown in Figure 11, the proposed method has recorded a 9% improvement in the KDDTest+ test set and a 26% improvement in the KDDTest-21 set compared to the primary method. According to this test, unlike the primary method of various training, there was no significant change in the final result. In both sets, the method's accuracy was 90%, with a tiny difference. However, in the primary method, the difference between the two sets was 16%, which according to the proposed method This issue has also been resolved.

In this paper, deep neural network (DNN) was considered as the base model of adaptive boosting (AdaBoost) to consider the ability of deep learning to extract data features and the ability of AdaBoost to detect anomalies with nominal value. Possibilities. AdaBoost allows assigning weights to each training sample, actively adapts to samples with different classification problems, and slightly adjusts their weights. One advantage is that the base models dealing with easy data classification and the hard-to-classify data models are eventually combined into a meta-learning model. Voting by all models determines the result.

This way, a new deep learning method (i.e., adaptive deep-boost neural network

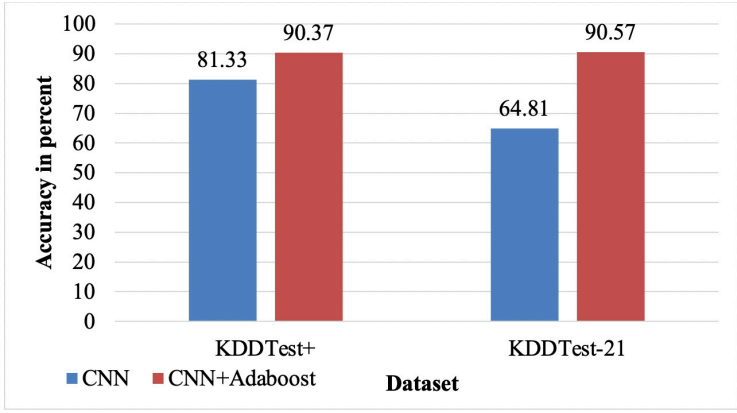

Fig. 11: Comparison of the proposed method and the base method on the NSL-KDD dataset Conclusion and Recommendations

optimized by AdaBoost) is obtained with stronger robustness and higher prediction accuracy. In this research, a new data-driven model (AdaBoost-CNN) is proposed. In addition, the imbalanced data related to the anomaly is analyzed and processed from several perspectives, including data preprocessing, feature extraction, algorithm improvement, and parameter optimization. Through experiments, it is found that the detection accuracy and robustness of the method are better than other methods. Machine learning methods can accurately detect abnormal behavior and thus reduce harm. The proposed method still partially relies on the misjudgment of a minimal number of flows with less than prominent features. For example, abnormality appears well-hidden in cases involving a few features, and regular users may be confused with those engaged in harm due to typical abnormalities. In the future, better detection performance can be realized by combining more advanced feature extraction technologies and superior meta-parameter optimization methods. This process can also be extended to scenarios with data imbalances.

References

Aljumah, A. (2021). IoT-based intrusion detection system using convolution neural networks. *PeerJ Computer Science, 7*, e721.

Al-Qatf, M., Lasheng, Y., Al-Habib, M., & Al-Sabahi, K. (2018). Deep learning approach combining sparse autoencoder with SVM for network intrusion detection. *Ieee Access, 6,* 52843-52856.

Ashfaq, R. A. R., Wang, X. Z., et al. (2017). Fuzziness based semi-supervised learning approach for intrusion detection system. *Information sciences, 378*, 484-497.

Atul, D. J., Kamalraj, R., et al. (2021). A machine learning based IoT for providing an intrusion detection system for security. *Microprocessors and Microsystems, 82*, 103741.

Costa, K. A., Papa, J. P., et al. (2019). Internet of Things: A survey on machine learning-based intrusion detection approaches. *Computer Networks, 151*, 147-157.

Hajiheidari, S., Wakil, K., Badri, M., & Navimipour, N. J. (2019). Intrusion detection systems in the Internet of things: A comprehensive investigation. *Computer Networks, 160*, 165-191.

Ingre, B., & Yadav, A. (2015, January). Performance analysis of NSL-KDD dataset using ANN. In *2015 international conference on signal processing and communication engineering systems* (pp. 92-96). IEEE.

Khraisat, A., & Alazab, A. (2021). A critical review of intrusion detection systems in the internet of things: techniques, deployment strategy, validation strategy, attacks, public datasets and challenges. *Cybersecurity, 4*(1), 1-27.

Kodyš, M., Lu, Z., Fok, K. W., & Thing, V. L. (2021, December). Intrusion Detection in Internet of Things using Convolutional Neural Networks. In *2021 18th International Conference on Privacy, Security and Trust (PST)* (pp. 1-10). IEEE.

Li, Y., Xu, Y., et al. (2020). Robust detection for network intrusion of industrial IoT based on multi-CNN fusion. *Measurement, 154,* 107450.

Naseer, S., Saleem, Y., et al. (2018). Enhanced network anomaly detection based on deep neural networks. *IEEE access, 6,* 48231-48246.

Nimbalkar, P., & Kshirsagar, D. (2021). Feature selection for intrusion detection system in Internet-of-Things (IoT). *ICT Express, 7*(2), 177-181.

Rincy N, T., & Gupta, R. (2021). Design and development of an efficient network intrusion detection system using machine learning techniques. Wireless Communications and Mobile Computing, 2021.

Sicari, S., Rizzardi, A., Grieco, L. A., & Coen-Porisini, A. (2015). Security, privacy and trust in Internet of Things: The road ahead. *Computer networks, 76*, 146-164.

Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009, July). A detailed analysis of the KDD CUP 99 data set. In *2009 IEEE symposium on computational intelligence for security and defense applications* (pp. 1-6). Ieee.

Yin, C., Zhu, Y., Fei, J., & He, X. (2017). A deep learning approach for intrusion detection using recurrent neural networks. *Ieee Access, 5*, 21954-21961.

Yousefi-Azar, M., Varadharajan, V., Hamey, L., & Tupakula, U. (2017, May). Autoencoder-based feature learning for cyber security applications. In *2017 International joint conference on neural networks (IJCNN)* (pp. 3854-3861). IEEE.

Zarpelão, B. B., Miani, R. S., Kawakani, C. T., & de Alvarenga, S. C. (2017). A survey of intrusion detection in Internet of Things. *Journal of Network and Computer Applications, 84,* 25-37.

Submitted: 22.09.2022
Accepted: 03.11.2022