Received March 9, 2021, accepted April 11, 2021, date of publication April 15, 2021, date of current version April 26, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3073413

An Ensemble Intrusion Detection Method for Train Ethernet Consist Network Based on CNN and RNN

CHUAN YUE, (Graduate Student Member, IEEE), LIDE WANG, DENGRUI WANG, RUIFENG DUO, AND XIAOBO NIE

School of Electrical Engineering, Beijing Jiaotong University, Beijing 100044, China

Corresponding author: Xiaobo Nie (xbnie@bjtu.edu.cn)

This work was supported by the Science and Technology Research and Development Program of China State Railway Group Company Ltd. under Grant N2020J007.

ABSTRACT The train Ethernet Consist Network (ECN) undertakes the task of transmitting critical train control instructions. With the increasing interactions between the train network and the outside environment, masses of network intrusions are threatening the data security of railway vehicles. The intrusion detection system has been proved to be an efficient method to detect network attacks. In this paper, a novel ensemble intrusion detection method is proposed to defense network attacks against the train ECN, in particular IP Scan, Port Scan, Denial of Service (DoS) and Man in the Middle (MITM). Thirty-four features of different protocol contents are extracted from the raw data generated from our ECN testbed to form a specific dataset. A data imaging method and a temporal sequence building method are designed to optimize the dataset. Six base classifiers are built based on several typical convolutional neural networks and recurrent neural networks: LeNet-5, AlexNet, VGGNet, SimpleRNN, LSTM and GRU. A dynamic weight matrix voting method is proposed to integrate all the base classifiers. The proposed method is evaluated based on our dataset. The experiment results show that our method has an outstanding ability to aggregate advantages of all the base classifiers and achieves a superior detection performance with the accuracy of 0.975.

INDEX TERMS Train Ethernet consist network, industrial cyber security, intrusion detection system, ensemble method.

LIST OF ABBREVIATION

Abbreviations for certain essential concepts in this article are listed as follows.

TCMS	Train Control and Management System
ETB	Ethernet Train Backbone
ECN	Ethernet Consist Network
IDS	Intrusion Detection System
CS	Consist Switch
ED	End Device
DoS	Denial of Service
MITM	Man in the Middle
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
BC	Base Classifier

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

I. INTRODUCTION

As the main pillar of transportation, railway is an important way to ensure people's daily travel in most countries. For the need of data transmissions and automation functions, Train Communication Network (TCN) is deployed on railway vehicles, known as the Train Control and Management System (TCMS). The traditional network architecture of TCMS encompasses two busses: Wire Train Bus (WTB) and Multifunction Vehicle Bus (MVB) [1]. However, due to the constraints of communication rate and expansion capacity, the architecture of WTB&MVB cannot satisfy the actual needs of the railway industry [2]. In response to these requirements, train Ethernet is introduced as the new main network architecture for TCMS. In 2014, IEC published IEC61375-2-5 (Ethernet Train Backbone, ETB) standard [3] and IEC61375-3-4 (Ethernet Consist Network, ECN) standard [4] to replace WTB and MVB. ECN can be used to create vehicle buses and also as train-wide communication network in the case of fixed consist (no dynamic train configuration).

As we know, using Ethernet technology in industrial scenarios is a double-edged sword [5]. On the bright side, it brings a faster data transfer rate and a more open information interaction to TCMS; but on the dark side, it makes TCMS more vulnerable to malicious cyber-attacks. Lots of intelligent devices are connected to each other through ECN in modern trains, which means that all the information generated by on-board devices, including critical control instructions, needs to be transmitted by ECN. The hacking of trains could cause serious risks to people's lives. For instance, when the driver issues a braking instruction, the control data will be generated by the master controller and transmitted to the braking control unit through ECN, however, if an intruder hijacks this control command through the man in the middle attack in this process, and modifies its corresponding control content, it will make the trains unable to slow down in time, which may lead to serious consequences and even loss of lives. Security has never been a major consideration in the design of Ethernet. The entire architecture reflects the objective of a cheap and easily deployable local area network. Security problems in traditional computer network are brought into TCMS with the application of the Ethernet technology in trains. Nowadays, as a key component of TCMS, ECN is facing a large number of cyber security threats which had already caused great losses in other Ethernet-applied scenarios.

Due to the reasons exposed above, it is necessary to take effective measures to deal with the potential cyber security problems that may occur in ECN. The Intrusion Detection System (IDS) was firstly proposed by Denning in 1987 [6], since then, with various improvement and optimization methods be applied to it, IDS has proven to be an efficient technology to counter with cyber-attacks. According to the placement of the IDS module in the network, IDS can be distinguished into two classes: host-based and network-based [7]. A host-based IDS monitors and analyzes system configurations and application activities for devices running on the network. The advantage of a host-based IDS is that it examines historical data to catch savvy intruders that use non-conventional methods which might be hard to detect in real-time. But there are obvious drawbacks of it, that is, it consumes processing time, storage, memory and other resources on the hosts. A network-based IDS monitors and analyzes network packets for malicious behavior. It can monitor a large network with less consumption, meanwhile, it has a fine-grained recognition ability.

Deep learning methods for intrusion detection have been studied by many researchers in recent years [8]–[13]. The large volume of network data has made intrusion detection problems amenable to deep learning methods. In this paper, we propose a novel ensemble intrusion detection method for ECN based on two kinds of efficient deep learning methods: Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN). Specifically, we build three CNN models with different structures: LeNet-5, AlexNet and VGGNet; and three RNN models with different structure: SimpleRNN, Long Short-Term Memory Network (LSTM) and Gated Recurrent Unit (GRU). Then an ensemble method was designed to integrate these six models to converge their advantages. The proposed method is able to extract both attribute correlations and temporal patterns of network packets captured in ECN, and flag attack packets with a high accuracy.

We note that, to our knowledge, we are the first to deal with intrusion detection issues on train ECN. As a matter of fact, at present, there is barely applied system to ensure cyber security of train ECN. Our research, to some extent, could be considered as the basic reference for other researchers' future work. The specific contributions of this article are as follows:

(1) Potential attacks against ECN are analyzed and a new dataset is generated from our testbed considering four kinds of network attacks.

(2) A data imaging method to transform our dataset into image form suitable for CNN models is proposed. A temporal sequence building method to transform our dataset into time sequences suitable for RNN models is proposed.

(3) A novel ensemble IDS method is proposed. It uses the proposed dynamic weight matrix voting method to aggregate six base classifiers.

(4) Experiments are conducted on our dataset. Including experiments on each base classifier and experiments on our whole ensemble method. The comparisons in terms of multi metrics are given. The results indicate that our method has a superior performance on ECN intrusion detection issues.

The remainder of this article is organized as follows. Section II introduces the related works. Section III describes the necessary background about our research objects for the convenience of understanding our work. Section IV describes the proposed method in detail. In Section V, several experiments are carried out to validate the proposed method. Finally, section VI presents the conclusions of this paper.

II. RELATED WORK

Because the large-scale application of Ethernet in trains just rises up in recent years, there is no effective ECN oriented intrusion detection method at present. However, researchers have accumulated a lot of research experiences in many other network scenarios. Aburomman and Reaz [14] compares several SVM methods aiming to identify multiclass SVM models best suited to the intrusion detection task. They developed a new approach called the weighted one-against-rest SVM based on the model selection method. Yang *et al.* [15] exploited a conditional deep belief network-based IDS to perform the wireless network intrusion detection in real time. Their experiment results show that their method has a high detection speed and accuracy with the average detection time 1.14 ms and the detection accuracy 0.974.

Convolutional neural network is mainly used in various tasks of image and video analysis (such as image classification, face recognition, object recognition, image segmentation, etc.), and its accuracy is generally far beyond that of other neural network models. In recent years, CNN has also been widely used in intrusion detection. Xiao *et al.* [16] proposed a CNN based IDS. They used CNN to automatically extract features of the dimensionality reduction data. KDD99 dataset is used to evaluate the performance of their model. The experiment results show that the accuracy, false positive rate and timeliness of their model are higher than those of some traditional algorithms. Nie *et al.* [17] designed a data-driven IDS by analyzing the link load behaviors of the Road Side Unit in the Internet of Vehicles against various network attacks. Their method consists of a traditional CNN and a fundamental error term in order to raise the convergence of the BP algorithm. Garg *et al.* [18] proposed a hybrid IDS anomaly detection model that leverages grey wolf optimization and CNN. They used improved grey wolf optimization for feature selection in the first phase and used improved CNN for anomaly detection in the second phase. Then they validated their method on KDD99 dataset and the results show that the method is superior compare to the other SOTA models.

Recurrent neural network is mainly used in various tasks of connected handwriting recognition and speech recognition. RNN has been proved to have an excellent ability of dealing with time sequence data. Network messages have obvious time correlation, so RNN has also been widely used in intrusion detection. Gao *et al.* [19] proposed an omni IDS for supervisory control and data acquisition networks. They combined LSTM and FNN through an ensemble approach and validated their method both on temporally uncorrelated attack data and temporally correlated attack data. Results show that the method has a high F1 score of 99.68±0.1%. Yin *et al.* [20] proposed a RNN based IDS. They studied the performance of their model in binary classification and multiclass classification, separately. NSL-KDD dataset was used to validate their model and the results show that the model is suitable for intrusion detection and performs better than several traditional machine learning methods. Sheikhan *et al.* [21] proposed a misuse IDS based on a three-layer RNN with categorized features as inputs and attack types as outputs. Their method offers better detection rate and cost per example comparing to similar intrusion detectors, and on the other hand, the false alarm rate of their method is not degraded significantly comparing to some recent machine learning methods.

There is no method can solve all the problems. A good way to raise the classification ability of a system is to combine different methods together. Researcher use ensemble learning methods to combine variant intelligent methods, in order to avoid the disadvantages of individual method. This ideology is also very popular in intrusion detection. Chawla *et al.* [22] proposed a computational efficient anomaly-based IDS based on RNN. They combined stacked CNNs with GRUs to improve intrusion detection ability of the system. Priya *et al.* [23] proposed a two-phase attack detection model to enhance the reliability of an IIoT network. In the first phase, they used an ensemble technique to integrate SVM and Naive Bayes and then used Random Forest and Artificial Neural Network to predict class labels. In the second phase, they chose the highest accuracy among two algorithms as the final result. Zhou *et al.* [24] designed a novel ensemble IDS based on the modified adaptive boosting with area under the curve algorithm. Their model achieved superior performance across multiple classes in both 802.11 wireless intrusion detection and traditional enterprise intrusion detection.

III. BACKGROUND

In this section, the necessary background was provided to better understand the contributions described in this article. The basic topology of ECN was firstly described based on which we built a testbed. The testbed is described in detail in section V. Then, according to the network topology characteristics and protocol vulnerability of ECN, the potential attacks against ECN are analyzed.

A. BASIC TOPOLOGY OF ECN

The logical view of ECN is shown in Fig. 1. The ECN is based on switched Ethernet and it consists of Consist Switches (CS), connectors, cables, and optional repeaters. ECN interconnects End Devices (ED) located in one consist. An ECN should be connected to the ETB via one Ethernet Train Backbone Node (ETBN) when it is connected to an ETB. Data frames between EDs, and between EDs and ETBN, are transmitted through ECN.

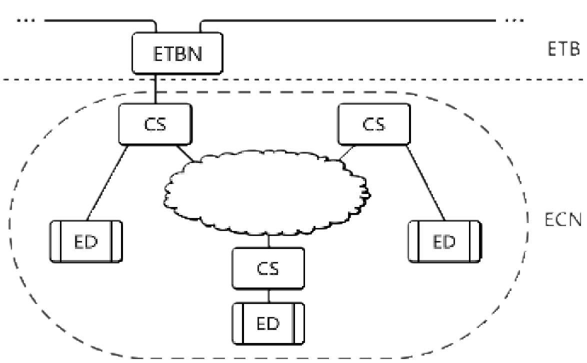

FIGURE 1. Logical view of ECN.

Any physical topology can be deployed according to the requirements from applications. Linear, ring, and ladder topologies are typical topologies as described in IEC 61375-1 [1]. Among them, ring topology is frequently applied in many operating trains such as CR300AF in China. Thus, we carry out research of intrusion detection issues based on typical ring topology that shown in Fig. 2.

In order to obtain the actual intrusion network data on the train ECN, based on the typical ring topology mentioned above, we built an ECN testbed. The details of the testbed are described in section V.

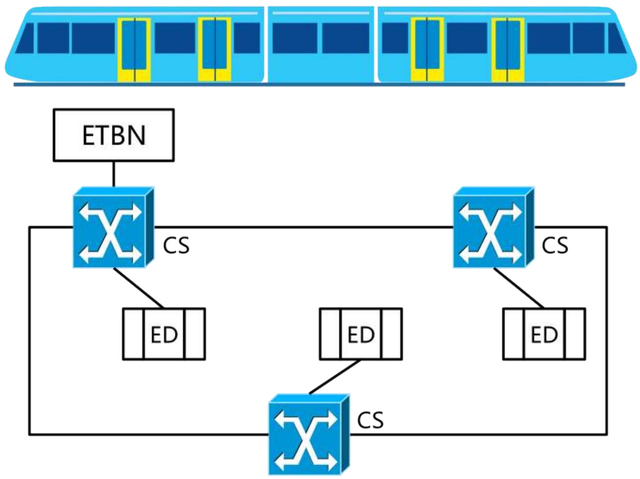

FIGURE 2. Typical ring topology of ECN.

B. POTENTIAL ATTACKS AGAINST ECN

Network information security's primary focus is the comprehensive protection of the confidentiality, integrity and availability of data, otherwise known as the CIA triad. Security professionals often evaluate threats and vulnerabilities based on the potential impact on these three core security objectives. We separately analyze the potential network attacks against ECN according to confidentiality, integrity and availability. It is impossible to find out all the possible attacks because attacking methods are always changing. We attempt to find out attacks that are mostly likely to appear in ECN. The analysis results are shown in Fig. 3.

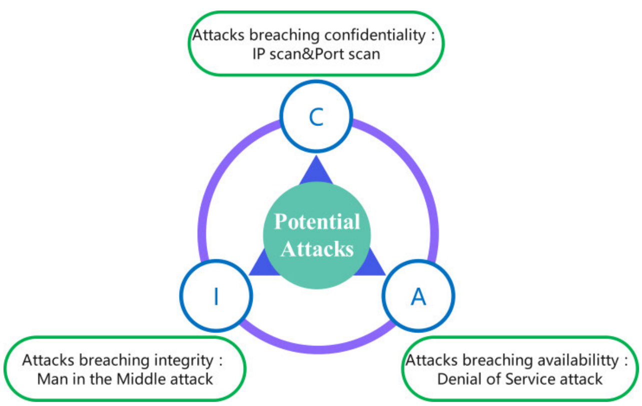

FIGURE 3. Potential attacks against ECN.

Attacks breaching confidentiality: ECN is a special-designed industrial network system for the railway use. The information of ECN except some general agreements is not commonly available. For example, different train network system vendors often formulate specific network structures and node address naming rules within the optional range described in the IEC standard. When attackers trying to intrude a system, they firstly need to obtain background knowledge of the system. An attacker can get unauthorized access to the ECN in various ways to obtain the knowledge they need. IP scan attacks can obtain the IP addresses of devices in the network [25]. An attacker sends ICMP (Internet Control Message Protocol) requests to multiple destination addresses in ECN. If a target device replies to these requests, the reply message reveals the IP address of the device to the attacker. Port scan can find out which ports on a network are open [26]. An attacker can run a port scanner on ECN to monitor which ports of a device in ECN are online.

Attacks breaching integrity: Integrity is a particularly important component of ECN. Network data transmitted in ECN mostly contain critical control instructions, such as open or close instructions of train doors, propulsion instructions of the traction unit, break instructions of the break unit and so on. Breaching the integrity of above data causes serious consequences. An attacker can breach the integrity of ECN by injecting false network packets or tempering positive messages to negative messages. Specifically, an attacker can use MITM (Man in the Middle) attacks to achieve this [27]. The MITM attack is a general term that when an attacker position himself in a middle place of device A and device B to eavesdrop or to impersonate one of A&B, making it appears as if a normal exchange of network messages is underway. Imagine when the driver sends an acceleration command, an attacker uses a MITM attack to tamper it to deceleration command and sends it to the traction unit. This may probably make accidents happen. MITM attacks have a valid syntax code as same as the normal messages, hence, spatial-domain based IDS will not be able to identify it from the message format. However, the MITM attack inevitably disrupt the original messages' temporal correlations, which means that RNN is suitable to solve this problem.

Attacks breaching availability: ECN uses the real-time Ethernet technology to ensure the real-time performance of train control messages, which means that ECN cannot tolerate a long-time delay of control messages. DoS (Denial of Service) attacks can breach the availability of a system [28]. An attacker can use DoS attacks to slow down the data transmission in ECN or even shut down the whole network. DoS attacks accomplish this by flooding the target with enormous useless messages. ECN is a mission-critical network system with high availability requirements. A DoS attack happens in civil networks may not be a big deal for most of users. Because the DoS attack usually just makes victims cannot enjoy their entertainment time, meanwhile, it just causes the victim a time loss or money loss to solve this problem. But if a DoS attack happens in ECN, where the most transmitting data are control data rather than entertainment information, it may cause train operation accidents, and could even cause casualties.

IV. PROPOSED METHOD

In this section, an ensemble IDS method was proposed, where variant CNN models and RNN models are combined together to get a high generalization ability and a high accuracy. The byte relationship of network packet data has variable density, which is a reflection of spatial features. Meanwhile, some of

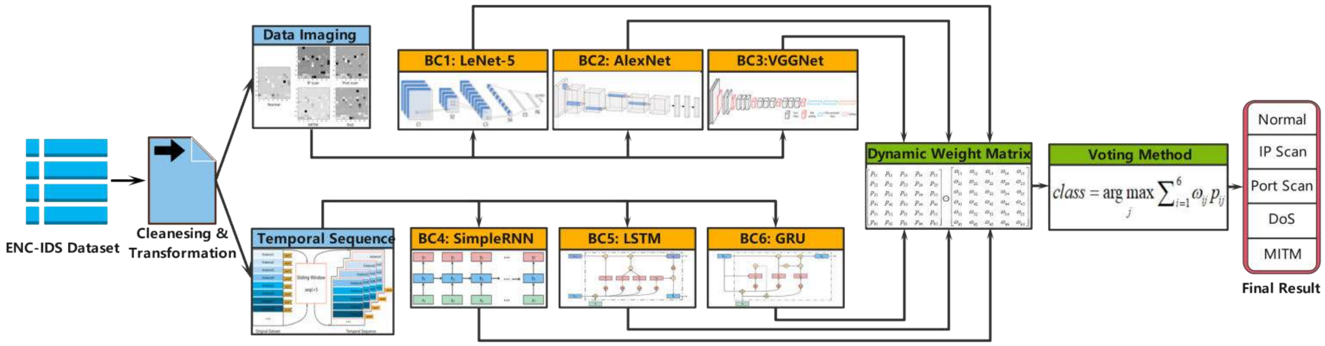

FIGURE 4. Framework of the proposed method.

the network intrusions like DoS and MITM have significant temporal patterns. That is, a certain instance has correlation to instances at the previous time, which is a reflection of time-domain features. As we know that CNN has significant ability to deal with spatial features and RNN has significant ability to deal with time-domain features. That is why we chose to combine CNN and RNN together.

The framework of our proposed method is shown in Fig. 4, where BC stands for Base Classifier. We first conduct a data cleansing and a data transformation to raise the basic quality of our dataset. For the rational use of CNN and RNN, we design a data imaging method to change the input data of CNN into the grayscale image format, and propose a temporal sequence building method to change the input data of RNN into the temporal sequence format. We then build six base classifiers based on three CNN models and three RNN models, i.e., LeNet-5, AlexNet, VGGNet, SimpleRNN, LSTM and GRU. Finally, a dynamic weight matrix voting method was designed to integrate six base classifiers. The voting method is able to assign different weights dynamically to each class per base classifier according to their classification abilities. The proposed IDS model gives the final intrusion classification based on the weighted voting results.

A. DATA PREPROCESSING

In our testbed described in Section V, we have conducted real cyber-attacks against the ECN to gather realistic dataset containing normal packets and four kinds of attack packets. For description convenience, we name our dataset as ECN-IDS dataset.

The ECN-IDS dataset contains 310537 ECN network packet instances. All instances are labeled into five categories, i.e., normal, IP Scan, Port Scan, DoS, and MITM. The percentage of each kind of instances is shown as Fig. 5.

Each of the instances has 34 features as shown in Table 1. All these features are extracted in the raw packets directly. These features can be divided into two categories. The first category consists of common Ethernet features such as source IP address, protocol type and so on. The second category consists of features of Train Real Time Data Protocol (TRDP) defined in IEC61375-2-3 [29]. This protocol is an open network protocol for upper layer communication over IP-based

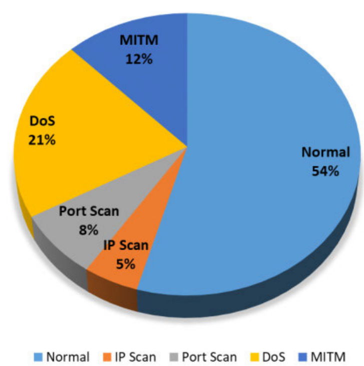

FIGURE 5. Data distribution of the ENC-IDS dataset.

networks in railway vehicles. It enables real-time transmission between Ethernet devices in trains. This means that features 24-32 are unique to train networks.

Before using the ECN-IDS dataset to train and assessment our IDS model, we firstly need to conduct data preprocessing to ensure the data can be utilized accurately and effectively.

1) DATA CLEANSING AND DATA TRANSFORMATION

It is worth noting that not all the instances have values in each feature. For example, the feature 'ComID' only appears in TRDP packets and the feature 'Flag_ICMP' only appears in ICMP packets. This means that missing values (null values) exist in the dataset. This class of missing belongs to Missing not at Random (MNAR), so we use zero imputation to cleanse our dataset [30].

The data type of features in our dataset including numerical data and nominal data. Nominal data cannot be directly recognized by deep learning model, so, we use one-hot encoding to convert nominal features into numerical features.

Differences in the scales across input values may increase the difficulty of the problem being modeled. We then use z-score normalization to normalize each feature of all the instances in our dataset. That is:

$$x_{norm} = {x - \mu}/{\sigma} \tag{1}$$

where x_{norm} stands for normalized data, x stands for unnormalized data, μ stands for the mean value of the feature, σ stands for the standard deviation of the feature.

TABLE 1. Features of our ECN-IDS dataset.

No.	Name	Description	Type	No.	Name	Description	Type
1	Protocol	Protocol type	Nominal	18	Len_TCP	Length of TCP	Numerical
2	Len	Total length	Numerical	19	Seq_TCP	Sequence number of TCP	Numerical
3	Src_IP	Source IP address	Numerical	20	Ack_TCP	Acknowledge number of TCP	Numerical
4	Dst_IP	Destination IP address	Numerical	21	Flag_TCP	Flag of TCP	Nominal
5	TTL_IP	Time to live of IP	Numerical	22	WinSize	Window size value	Numerical
6	TTL_ICMP	Time to live of ICMP	Numerical	23	RTO	Retransmission Timeout	Numerical
7	Len_IP	Length of IP	Numerical	24	Seq_TRDP	TRDP sequence number	Numerical
8	Len_ICMP	Length of ICMP	Numerical	25	Ver_TRDP	TRDP protocol version	Numerical
9	ID_IP	Identification of IP	Numerical	26	msgType_TRDP	TRDP message type	Nominal
10	ID_ICMP	Identification of ICMP	Numerical	27	ComId_TRDP	TRDP communication ID	Numerical
11	Flag_IP	Flag of IP	Nominal	28	etb_TRDP	ETB topology counter	Numerical
12	Flag_ICMP	Flag of ICMP	Nominal	29	op_TRDP	Operation topology counter	Numerical
13	SrcPort_UDP	UDP Source port	Numerical	30	Len_TRDP	Length of TRDP	Numerical
14	DstPort_UDP	UDP Destination port	Numerical	31	ReComId_TRDP	Reply ComID	Numerical
15	Len_UDP	Length of UDP	Numerical	32	ReIP_TRDP	Reply IP	Numerical
16	SrcPort_TCP	TCP Source port	Numerical	33	Src_ETH	Source MAC address	Numerical
17	DstPort_TCP	TCP Destination port	Numerical	34	Dst_ETH	Destination MAC address	Numerical

2) DATA IMAGING FOR CNN

Each instance of our dataset is a one-dimensional list with 190 elements after the one-hot encoding and the z-score normalization. In order to make full use of the advantage of two-dimensional CNN in processing image data, we design a data imaging method to transfer our dataset to image format.

We first extend every instance with a zero list of length 6 which makes all instances have 196 dimensions. Then we transfer them to grayscale images with shape of 14*14. Five image data from five different labels are shown as examples in Fig. 6. As we can see that these five classes of data have differences in images, to a certain extent, this means that they are theoretically classifiable for CNN.

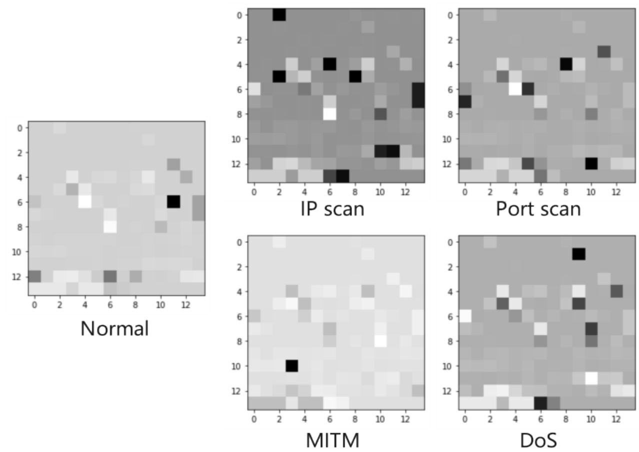

FIGURE 6. Five example images from five classes.

3) BUILD DATA TEMPORAL SEQUENCE FOR RNN

Inputs to RNN are sequences of instances. When putting a sequence $x_{1:t} = (x_1, x_2, \cdots, x_t)$ to RNN model, the model will output the classification of x_t. If we directly slice the instances at one-dimensional space to build sequences, the classification information of $x_1, x_2, \cdots, x_{t-1}$ will loss. We propose a temporal sequence building method as shown in Fig. 7 to solve this problem.

We use a changeable number (the length of the input sequence) of instances as a sliding window. This number *seqL* becomes a hyper-parameter of our model. After the one-hot encoding and the z-score normalization of ECI-IDS dataset, we use the window to slide upon the dataset with step length l to change the dataset to a two-dimensional form. The new form of the dataset has a total length l'_{total}:

$$l'_{total} = l_{total} - seqL \tag{2}$$

where l_{total} is the number of instances in ECN-IDS dataset.

In the process of sliding, the label of each window is the same as the label of the last instance within the window. This method enables our dataset can be directly input to RNN models. Compared with the traditional slicing method, only the label information of $seqL - 1$ instances at the beginning will loss, which means the loss of information will be greatly reduced. At the same time, these losses can be easily remedied by our ensemble IDS model.

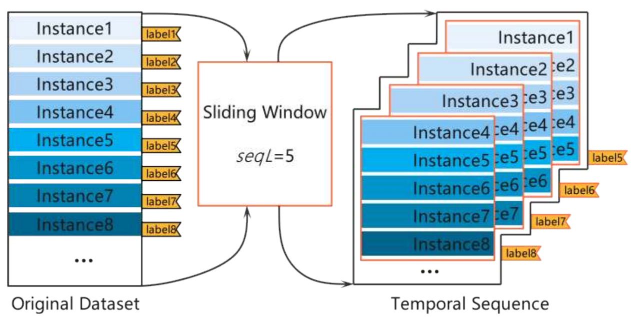

FIGURE 7. Temporal sequence building process.

B. BASE CLASSIFIER: CONVOLUTIONAL NEURAL NETWORK

CNN is a class of deep feedforward neural network with the characteristics of sparse connectivity and weight sharing. The typical structure of CNN consists of input layer, convolutional layer, pooling layer, fully connected layer and output layer.

The inputs of CNN are usually three-channel images with format of $H \times W \times C$, where H and W is height and width of one channel of the image, and C is the number of channels. $C = 3$ when the input is colored image and $C = 1$ when the input is grey scale image.

A convolutional layer can extract features of a local region from inputs and feature maps. A feature map is the features extracted from an image through filters called convolution kernels. Different convolution kernels are equivalent to different feature extractors. The size of a feature map is $M(height) \times N(width)$. Because CNN is mainly used in image processing and the image has a two-dimensional structure, in order to make full use of the local information of the image, the neuron is usually organized into a three-dimensional neural layer. Its size is $M \times N \times D$, i.e., it consists of D feature maps. In general, the architecture of a convolutional layer is as follows:

(1) Input feature maps: Three-dimensional tensor $x \in \mathbb{R}^{M\times N\times D}$, where each slice matrix $X^d \in \mathbb{R}^{M\times N}$ is one input feature map, and $1 \le d \le D$;

(2) Output feature maps: Three-dimensional tensor $y \in \mathbb{R}^{M'\times N'\times P}$, where each slice matrix $Y^p \in \mathbb{R}^{M'\times N'}$ is one output feature map, and $1 \le p \le P$;

(3) Convolution kernel: Four-dimensional tensor $w \in \mathbb{R}^{U\times V\times P\times D}$, where each slice matrix $W^{p,d} \in \mathbb{R}^{U\times V}$ is a two-dimensional convolution kernel, and $1 \le p \le P$, $1 \le d \le D$.

In order to calculate the output feature map Y^p, convolution kernels $W^{p,1}, W^{p,2}, \cdots, W^{p,D}$ are used to convolute the input feature map $X^1, X^2, \cdots, X^D$. Then the convolution results are added together. A scalar bias b is added to get a net input Z^p, again through a nonlinear activation function to get the output feature map Y^p:

$$Z^p = W^p \otimes X + b^p = \sum_{d=1}^{D} W^{p,d} \otimes X^d + b^p \quad (3)$$

$$Y^p = f(Z^p) \quad (4)$$

where $W^p \in \mathbb{R}^{U\times V\times D}$ is the three-dimensional convolution kernel, and $f(\cdot)$ is the nonlinear activation function.

The pooling layer is also called subsampling layer, which is used to summarize the outputs of neighboring groups of neurons in the same convolution kernel map. It conducts feature selection and reduces the number of features, so as to reduce the number of parameters. There are two kinds of popular pooling functions:

(1) Max pooling: For a region $R^d_{m,n}$, the maximum activity value of all neurons in the region is selected as the representation of the region. It can be described as follows:

$$y^d_{m,n} = \max_{i\in R^d_{m,n}} x_i \quad (5)$$

where x_i is the activity value of each neuron in the region.

(2) Mean pooling: It is the mean activity value of all neurons in the region:

$$y^d_{m,n} == (1/R^d_{m,n}) \sum_{i\in R^d_{m,n}} x_i \quad (6)$$

Fully connected layer and output layer together constitute the classifier. Based on the extracted features from layers ahead, classifier computes the probability distribution of all the labels by softmax function. It is calculated as follows:

$$P(y = c\,|x\,) = \text{softmax}_c(y) = \frac{\exp(y_c)}{\sum_{j=0}^{k-1} y_j} \quad (7)$$

where $0 \le c \le k-1$ and k means that there are k different labels. Then, in order to optimize the CNN model, using cross-entropy loss function as the criterion to conduct the back propagation process based on gradient descent method.

In the development process of CNN, several efficient structures were proposed. LeNet-5 [31] is a very successful neural network model, although it is one of the earliest CNN models. LeNet-5 uses convolution, parameter sharing, pooling and other operations to obtain features, and then it uses fully connected neural network for classification. The network structure of LeNet-5 is shown in Fig. 8. It has 7 layers (besides the input layer) including one input layers, three convolutional layers (C1, C3, C5), two pooling layers (S2, S4), one fully connected layer (F6) and one output layer.

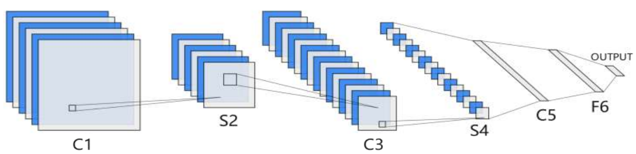

FIGURE 8. Network structure of LeNet-5.

AlexNet [32] is the first modern deep convolutional network model. It uses many modern deep convolutional network technologies for the first time, such as using GPU for parallel training, using Rectified Linear Unit (ReLU) as the nonlinear activation function, using Dropout to prevent overfitting and using data enhancement to improve accuracy of the model. Its network structure is shown in Fig. 9, including 5 convolutional layers, 3polling layers and 3 fully connected layers.

VGGNet [33] explores the relationship between the depth of neural network and its performance. By repeatedly stacking 3×3 small convolution cores and 2×2 maximum pooling layers, the convolution neural network with 11-19 layers is successfully constructed. VGGNet has a simple structure as shown in Fig.10. All convolution layers use the same size of the small convolution kernel, so that the cost of the whole network parameters is very small, at the same time, the multi-layer network structure can also ensure to learn more features.

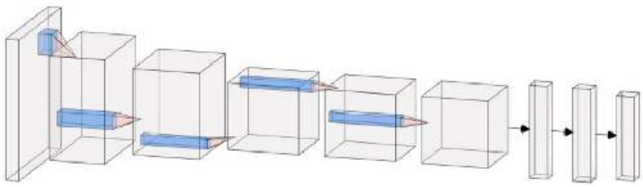

FIGURE 9. Network structure of AlexNet.

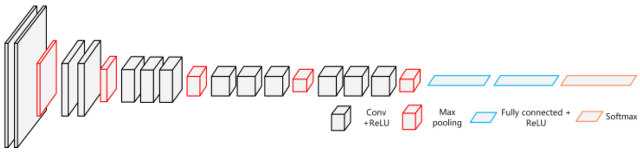

FIGURE 10. Network structure of VGGNet.

C. BASE CLASSIFIER: RECURRENT NEURAL NETWORK

RNN is a class of convolutional network with the capability of short-term memory. It is good at processing temporal sequence data. In RNN, the neurons can not only accept the information of other neurons, but also accept their own information to form a loop structure.

RNN can process temporal sequence data of any length by using the neuron with self-feedback. Given an input sequence $x_{1:T} = (x_1, x_2, \cdots, x_t, \cdots, x_T)$, the activity value of the hidden layer h_t is given by:

$$h_t = f(h_{t-1}, x_t) \tag{8}$$

where $h_0 = 0$ and $f(\cdot)$ is a nonlinear function.

Simple Recurrent Neural Network (SimpleRNN) is a basic RNN with one hidden layer. Consider the vector $x_t \in \mathbb{R}^M$ as the input of network at time t and the vector $h_t \in \mathbb{R}^D$ as the hidden layer state, then h_t is not only related to the current x_t, but also related to the hidden layer state h_{t-1} at the previous time. The update formula at time t is given by:

$$z_t = Uh_{t-1} + Wx_t + b \tag{9}$$

$$h_t = f(z_t) \tag{10}$$

where z_t is the net input of the hidden layer, $U \in \mathbb{R}^{D\times D}$ is the state-to-state weight matrix, $W \in \mathbb{R}^{D\times M}$ is the state-to-input weight matrix, $b \in \mathbb{R}^D$ is the bias vector and $f(\cdot)$ is the non-linear activation function (is usually Tanh). Fig. 11 illustrates the structure of SimpleRNN.

LSTM [34] is a variant of RNN. It can effectively solve the problem of gradient explosion or gradient disappearance of SimpleRNN. LSTM introduces the gating mechanism to control the path of information transmission. There are three gates that have been used in LSTM:

(1) Forget gate f_t: It controls how much information needs to be forgotten by the internal state c_{t-1} at the previous moment.

(2) Input gate i_t: It controls how much information needs to be saved by the candidate state $\tilde{c}_t$ at the current time.

(3) Output gate o_t: It controls how much information needs to be transmitted to the external state by the internal state c_t at the current time.

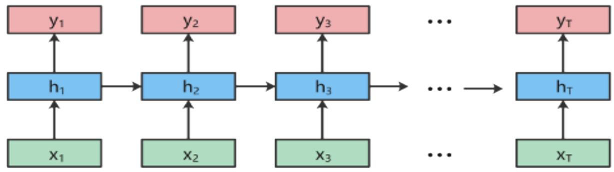

FIGURE 11. Structure of SimpleRNN.

The calculation methods of these three gates are as follows:

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{11}$$

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{12}$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{13}$$

where $\sigma(\cdot)$ is Logistic function, x_t is the current input and h_{t-1} is the external state at the previous time. The structure of LSTM is shown in Fig.12.

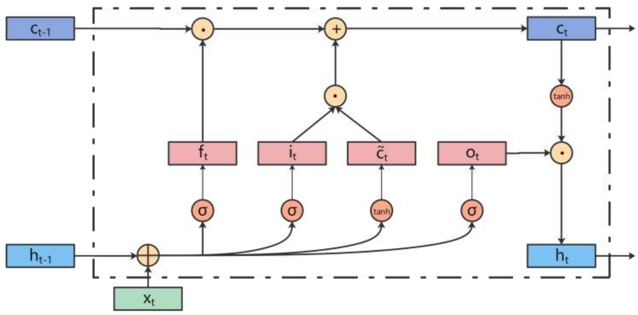

FIGURE 12. Structure of LSTM.

GRU [35] is another popular gated RNN. Differ from LSTM, GRU does not introduce additional memory units. Fig. 13 illustrates the structure of GRU. It introduces an update gate z_t to control how much historical information needs to be saved at the current state, and how much new information needs to be received from the candidate state. GRU also introduces a reset gate r_t to control whether the calculation of the candidate state $\tilde{h}_t$ depends on the state h_t at the previous moment. The update gate and reset gate are calculated as follows:

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \tag{14}$$

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \tag{15}$$

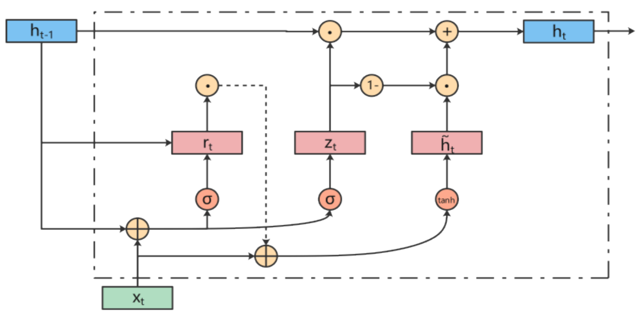

FIGURE 13. Structure of GRU.

The calculation method of how the state of GRU is updated is as follows:

$$\tilde{h}_t = \tanh(W_h x_t + U_h(r_t \odot h_{t-1}) + b_h) \tag{16}$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}_t \tag{17}$$

D. DYNAMIC WEIGHT MATRIX VOTING METHOD

Voting theory has been proven to be a particularly efficient combination method for multi-output problems [36]–[38]. We design a dynamic weight matrix voting method to converge the advantages of the six models mentioned above.

Each of our six base classifiers output the probability of 5 class at the output layer. This means that after the training, we get a probability matrix of each class per base classifier:

$$p_{ij} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} & p_{15} \\ p_{21} & p_{22} & p_{23} & p_{24} & p_{25} \\ p_{31} & p_{32} & p_{33} & p_{34} & p_{35} \\ p_{41} & p_{42} & p_{43} & p_{44} & p_{45} \\ p_{51} & p_{52} & p_{53} & p_{54} & p_{55} \\ p_{61} & p_{62} & p_{63} & p_{64} & p_{65} \end{bmatrix} \tag{18}$$

where p_{ij} stands for probability, $i \in \{1, 2, \cdots, 6\}$ stands for the number of the base classifiers, $j \in \{1, 2, \cdots, 5\}$ stands for the number of the label classes.

We give each element within the probability matrix a dynamic weight ω_{ij}. Hence, we get a dynamic weight matrix:

$$\omega_{ij} = \begin{bmatrix} \omega_{11} & \omega_{12} & \omega_{13} & \omega_{14} & \omega_{15} \\ \omega_{21} & \omega_{22} & \omega_{23} & \omega_{24} & \omega_{25} \\ \omega_{31} & \omega_{32} & \omega_{33} & \omega_{34} & \omega_{35} \\ \omega_{41} & \omega_{42} & \omega_{43} & \omega_{44} & \omega_{45} \\ \omega_{51} & \omega_{52} & \omega_{53} & \omega_{54} & \omega_{55} \\ \omega_{61} & \omega_{62} & \omega_{63} & \omega_{64} & \omega_{65} \end{bmatrix} \tag{19}$$

The dynamic weight ω_{ij} is calculated as follows:

$$\omega'_{ij} = \begin{cases} \ln(R_{ij}/1 - R_{ij}) & R_{ij} > 0.5 \\ 0 & R_{ij} \leq 0.5 \end{cases} \tag{20}$$

$$\omega_{ij} = \frac{\omega'_{ij}}{\sum_{i=1}^{6} \sum_{j=1}^{5} \omega'_{ij}} \tag{21}$$

where R_{ij} is the correct classification ratio of each class per base classifier.

Each time we train the models, the weight matrix can be calculated dynamically by formulas above. The final output label *class* of our ensemble IDS model is voted by:

$$class = \arg\max_{j} \sum_{i=1}^{6} \omega_{ij} p_{ij} \tag{22}$$

where $class \in \{1, 2, 3, 4, 5\}$ and these 5 numbers stands for five class labels.

V. EXPERIMENTS AND RESULTS

In this section, our ECN testbed was first described in detail. Then the evaluation metrics suitable to evaluate machine learning intrusion detection models are introduced. Different experiments were conducted based on all the base classifiers and our whole ensemble IDS model. The details and the results of the experiments are given.

A. ECN TESTBED

Because of commercial factors and privacy restrictions, railway operators almost never release their network data. It is hard to obtain real-world train datasets. To avoid this situation hindering our research on ECN intrusion detection, we build a physical ECN testbed as shown in Fig. 14.

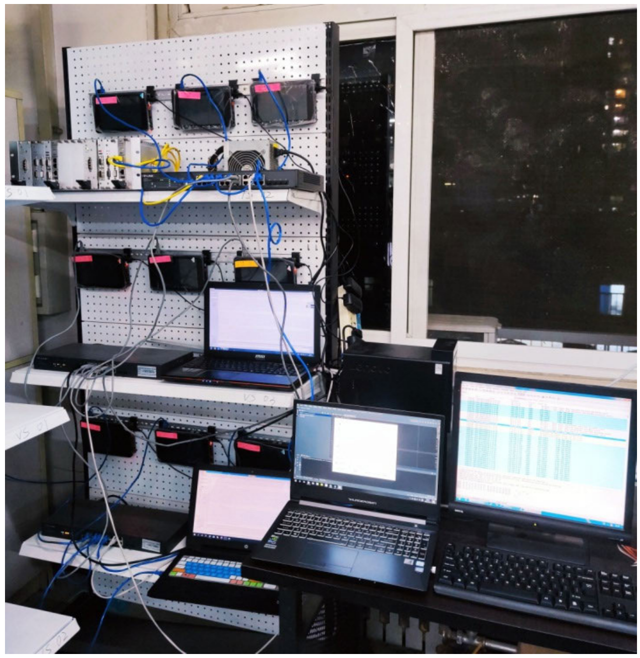

FIGURE 14. ECN testbed.

The real photo cannot clearly show the equipment distribution and cable connection of our testbed. Hence, we illustrate them in logistic topology of ECN in Fig. 15. Every node and cable in these two figures have one-to-one correspondence.

Our testbed is based on the ECN ring topology in accordance with a certain type of train which has been applied for public usage (Due to the confidentiality requirements, the specific train name is not given in this article). Our testbed consists of three kinds of equipment: three layer-3 switches simulate the consist switches in ECN; nine Linux development boards, one self-made train ethernet Control and Monitoring Server (CMS) and two PC form all the end devices; one PC simulates the attacker; one PC acquires data based on the Wireshark software and the Switched Port Analyzer (SPAN) technology.

Referencing the actual train terminal communication services, we configure the corresponding normal train communication flows in end devices. We write penetration software using python to inject four kinds of network attacks described in Section III, i.e., IP Scan, Port Scan, DoS and MITM. Using the SPAN technology and the Wireshark software, network data was captured, including normal packets and attack packets.

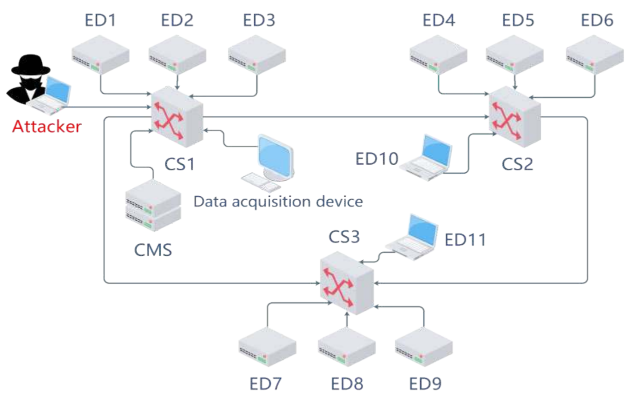

FIGURE 15. Logistic topology of ECN testbed.

B. EVALUATION METRICS

There are several metrics often used by researchers to evaluate the performance of machine learning models in intrusion detection issues [39] [40]. The description of them is as follows.

(1) Confusion matrix: The confusion matrix is a tabulation often used to describe the performance of a classification model. For binary classification, as shown in Table 2, each row of the matrix represents the instances in an actual class and each column represents the instances in a predicted class.

TABLE 2. Confusion matrix for binary classification.

Actual	Predicted	
	Attack	Normal
Attack	TP	FN
Normal	FP	TN

In the confusion matrix, true positive (TP) means attack data correctly predicted, false positive (FP) means attack data incorrectly predicted, false negative (FN) means normal data incorrectly predicted, true negative (TN) means normal data correctly predicted.

For multi-class classification, each combination of two classes generates a confusion matrix, which means that turning a multi-class classification issue into several one-to-others binary classification issues. The final result of multi-class classification can be calculated by averaging all the binary matrixes.

(2) Accuracy: The accuracy Acc is the most commonly used evaluation metric. It is the proportion of correctly predicted instances out of the total instances, that is:

$$Acc = \frac{TP + TN}{TP + FP + TN + FN} \tag{23}$$

(3) Precision: The precision P is the proportion of correctly predicted attacks out of instances classified as attacks, that is:

$$P = \frac{TP}{TP + FP} \tag{24}$$

(4) Recall: The recall R is the proportion of correctly predicted attacks out of all the actual attack instances, that is:

$$R = \frac{TP}{TP + FN} \tag{25}$$

(5) F-score: The F-score F is a comprehensive measure of precision and recall, that is:

$$F = \frac{(1 + \beta^2) \cdot P \cdot R}{(\beta^2 \cdot P) + R} \tag{26}$$

where β measures the relative importance of precision and recall.

(6) Macro average and micro average: Metrics mentioned above were used for binary classification. A multi-class classification issue can be treated as the combination of several binary classification issues. Macro and micro average are two different means to evaluate average performance of a multi-class classification issue. Macro average pays more attention to the influence of classes with small sample capacity, and micro average pays more attention to the influence of classes with large sample capacity. For IDS, attacks usually have small sample capacity, so we choose the macro average as the metric to evaluate the multi-class classification performance. The macro average is calculated as follows:

$$\begin{cases} macroP = \frac{1}{n} \sum_{i=1}^{n} P_i \\ macroR = \frac{1}{n} \sum_{i=1}^{n} R_i \\ macroF = \frac{(1 + \beta^2) \cdot macroP \cdot macroR}{(\beta^2 \cdot macroP) + macroR} \end{cases} \tag{27}$$

where n is the number of data classes (5 in the paper).

C. DETAILS OF THE EXPERIMENTS

The experiments were carried out on a PC equipped with 64-bit Intel(R)Core(M) i7-9700K CPU@3.60GHz, 16 GB RAM, Nvidia GeForce RTX 2060 Super (8GB) GPU, Windows 10 operating system, Keras 2.4.3 framework.

The total number of the instances in the ECN-IDS dataset is 310537. We used two steps to conduct sample allocation. The first step, the holdout method was used to split out dataset into two sets: 80% for step two and 20% for testing. Stratified mode was used to preserve the original class proportion. The second step, in order to obtain a reliable and stable model, we used 5-fold cross validation to split the 80% dataset into 5 mutually exclusive subsets with the same size. Each time we use 4 subsets as the training set and the rest subset as the validation set.

There are some hyperparameters need to be set before training. The learning rate is set to 0.002. The number of iteration epochs is set to 20. The batch size is set to 64. The weights and biases are initialized randomly based on Gaussian distribution. The sequence length $seqL$ is set to 30.

TABLE 3. Results for base classifiers and the ensemble model on different intrusion classes.

Table III-a IP Scan

Model	Precision	Recall	F(β=1.2)
LeNet-5	0.938	0.946	0.943
AlexNet	0.952	0.963	0.958
VGGNet	0.949	0.971	0.962
SimpleRNN	0.889	0.893	0.891
LSTM	0.925	0.928	0.927
GRU	0.919	0.930	0.925
Ensemble IDS	0.965	0.979	0.973

Table III-b Port Scan

Model	Precision	Recall	F(β=1.2)
LeNet-5	0.927	0.921	0.923
AlexNet	0.944	0.949	0.947
VGGNet	0.961	0.953	0.956
SimpleRNN	0.917	0.922	0.920
LSTM	0.909	0.919	0.915
GRU	0.921	0.918	0.919
Ensemble IDS	0.958	0.952	0.954

Table III-c DoS

Model	Precision	Recall	F(β=1.2)
LeNet-5	0.933	0.928	0.930
AlexNet	0.920	0.916	0.918
VGGNet	0.937	0.928	0.932
SimpleRNN	0.966	0.957	0.961
LSTM	0.975	0.977	0.976
GRU	0.983	0.989	0.987
Ensemble IDS	0.980	0.991	0.986

Table III-d MITM

Model	Precision	Recall	F(β=1.2)
LeNet-5	0.856	0.862	0.860
AlexNet	0.838	0.815	0.824
VGGNet	0.862	0.870	0.867
SimpleRNN	0.953	0.961	0.958
LSTM	0.972	0.981	0.977
GRU	0.977	0.980	0.979
Ensemble IDS	0.976	0.981	0.979

D. RESULTS AND ANALYSIS

In order to demonstrate the ability of our ensemble IDS model to promote the performance of base classifiers, we first list the full set of results in Table 3 for all the base classifiers and the ensemble model on different intrusion classes, separately. Then, we list macro average results of the whole model on multi-class classification to show the final detection capability, as shown in Table 4.

The value of β in F-score measures the relative importance between recall and precision. The recall has a bigger influence when $\beta > 1$. The main purpose of IDS is to detect all the network attacks as far as possible. Recall mainly reflects the ability of IDS model to recognize attack instances. So, we set $\beta = 1.2$ in the F-score.

The results in Table 3 -a and Table 3 -b indicate that CNN models and RNN models all have good performance for IP Scan and Port Scan and our ensemble IDS outperforms most base classifiers. This can be explained that IP Scan and Port Scan have effective static features and do not have obvious temporal correlation.

The results in Table 3 -c and Table 3 -d indicate that RNN models have evident better performance than CNN models and the ensemble IDS also outperforms most base classifiers. This can be explained that DoS and MITM have temporal correlation, which is suitable for RNN models. It is worth noting that CNN models have poor performance on MITM. The reason for this is that MITM have almost the same features of normal packets. It also can be seen from the data images listed in Section IV. Temporal correlation between a MITM packet and previous packets is one of the few exploitable weaknesses of MITM.

The results in Table 4 indicate that our proposed ensemble IDS model achieves a high precision value of 0.968, a high recall value of 0.975 and a high F-score ($\beta = 1.2$) of 0.972. It outperforms all the other base classifiers.

For a better intuitive display, we illustrate all the results based on precision, recall, F-score in Fig. 16, Fig.17 and Fig. 18.

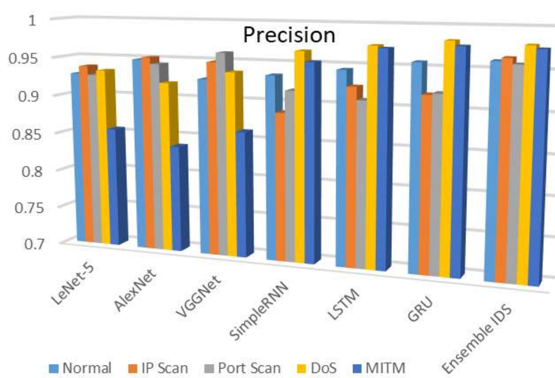

FIGURE 16. Precision comparison.

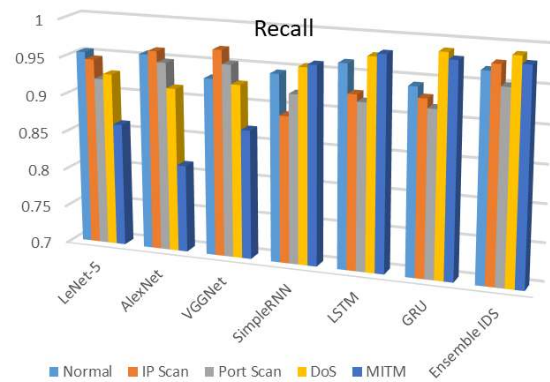

FIGURE 17. Recall comparison.

As shown in figures above, our ensemble IDS model has the ability to aggregate advantages from different

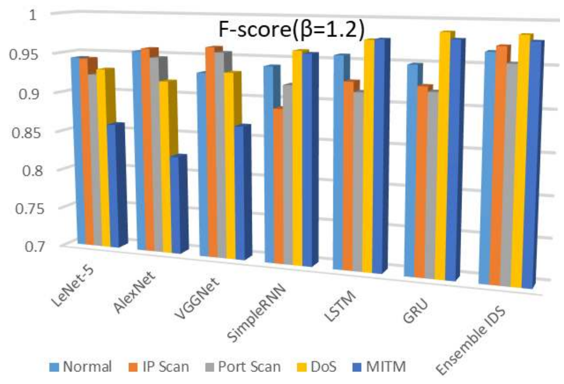

FIGURE 18. F-score($\beta = 1.2$) comparison.

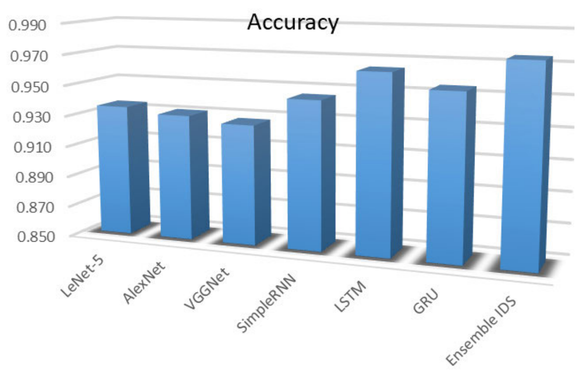

FIGURE 19. Accuracy comparison.

TABLE 4. Final results of the ensemble IDS model (macro average).

Model	macroP	macroR	marcoF
LeNet-5	0.916	0.922	0.920
AlexNet	0.921	0.920	0.920
VGGNet	0.927	0.931	0.930
SimpleRNN	0.932	0.936	0.934
LSTM	0.945	0.954	0.950
GRU	0.951	0.952	0.952
Ensemble IDS	**0.968**	**0.975**	**0.972**

base classifiers. Through our proposed method, the weaknesses of base classifiers on different attack classes are avoided. For example, although CNN models perform poor on MITM class, our ensemble IDS model still achieve a good performance in the final.

A good IDS model should detect both attack packets and normal packets accurately. The accuracy metric reflects the ability of a model to classify both positive and negative instances. We illustrate the accuracy of all the base classifiers and our ensemble IDS model in Fig. 19.

As shown in Fig. 19, CNN models achieve ordinary performances which could be caused by its poor performance on temporal correlation data. RNN models achieve better accuracy performances than CNN models. Our proposed ensemble IDS model achieves the best accuracy performance with the value of 0.975.

VI. CONCLUSION

The modern railway vehicle network is under the increasing threat of serious network intrusions. In this paper, a novel ensemble IDS method based on convolutional neural networks and recurrent neural networks for train Ethernet consist network has been proposed and discussed. In order to aggregate the abilities of CNN and RNN, we built three different CNN models and three different RNN models as the base classifiers. A dynamic weight matrix voting method was proposed to integrate all the base classifiers. We introduced the ECN testbed built by ourself and discussed the potential attacks against the ECN network. Four classes of attacks have been considered, that is: IP Scan, Port Scan, DoS and MITM. A data imaging method and a temporal sequence building method were designed to transform our dataset in appropriate forms for CNN models and RNN models, separately. The experiment results indicate that our proposed method effectively aggregate advantages of base classifiers. The proposed method yield a superior result in terms of precision, recall, F-score ($\beta = 1.2$) and accuracy.

The four classes of attacks discussed in the paper represent the main threats to ECN. However, the network attack means emerge frequently and develop rapidly. There are bound to be more network attacks that may threaten ECN. The outstanding results obtained in this paper encourage us to prove the effectiveness of the proposed method on other classes of attacks in the future. Our proposed method mainly pursues high detection ability without much discussion on the computational speed. In the future, we will continue to focus our research on how to optimize the computational speed as much as possible on the basis of ensuring an outstanding detection ability.

REFERENCES

[1] *Electronic Railway Equipment-Train Communication Network (TCN)—Part 1: General Architecture*, document IEC, 61375, 2012.
[2] X. Nie, L. Wang, B. Wang, B. Liu, and P. Shen, "A dynamic linear hashing method for redundancy management in train Ethernet consist network," *Math. Problems Eng.*, vol. 2016, pp. 1–10, Jun. 2016.
[3] *Electronic Railway Equipment-Train Communication Network (TCN)—Part 2-5: Ethernet Train Backbone*, document IEC, 61375-5, 2014.
[4] *Electronic Railway Equipment–Train Communication Network (TCN)—Part 2-5: Ethernet Consist Network*, IEC, 61375-4, 2014.
[5] D. Wu, A. Ren, W. Zhang, F. Fan, P. Liu, X. Fu, and J. Terpenny, "Cybersecurity for digital manufacturing," *J. Manuf. Syst.*, vol. 48, pp. 3–12, Jul. 2018.
[6] D. E. Denning, "An intrusion-detection model," *IEEE Trans. Softw. Eng.*, vol. 13, no. 2, pp. 222–232, Feb. 1987.
[7] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 1153–1176, 2nd Quart., 2016.
[8] S. Gamage and J. Samarabandu, "Deep learning methods in network intrusion detection: A survey and an objective comparison," *J. Netw. Comput. Appl.*, vol. 169, Nov. 2020, Art. no. 102767.
[9] Y. Xin, L. Kong, Z. Liu, Y. Chen, Y. Li, H. Zhu, M. Gao, H. Hou, and C. Wang, "Machine learning and deep learning methods for cybersecurity," *IEEE Access*, vol. 6, pp. 35365–35381, 2018.
[10] M. A. Ferrag, L. Maglaras, S. Moschoyiannis, and H. Janicke, "Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study," *J. Inf. Secur. Appl.*, vol. 50, Feb. 2020, Art. no. 102419.
[11] K. Shaukat, S. Luo, V. Varadharajan, I. A. Hameed, and M. Xu, "A survey on machine learning techniques for cyber security in the last decade," *IEEE Access*, vol. 8, pp. 222310–222354, 2020.

[12] A. Aldweesh, A. Derhab, and A. Z. Emam, "Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues," *Knowl.-Based Syst.*, vol. 189, Feb. 2020, Art. no. 105124.

[13] H. Liu and B. Lang, "Machine learning and deep learning methods for intrusion detection systems: A survey," *Appl. Sci.*, vol. 9, no. 20, p. 4396, Oct. 2019.

[14] A. A. Aburomman and M. B. Ibne Reaz, "A novel weighted support vector machines multiclass classifier based on differential evolution for intrusion detection systems," *Inf. Sci.*, vol. 414, pp. 225–246, Nov. 2017.

[15] L. Yang, J. Li, L. Yin, Z. Sun, Y. Zhao, and Z. Li, "Real-time intrusion detection in wireless network: A deep learning-based intelligent mechanism," *IEEE Access*, vol. 8, pp. 170128–170139, 2020.

[16] Y. Xiao, C. Xing, T. Zhang, and Z. Zhao, "An intrusion detection model based on feature reduction and convolutional neural networks," *IEEE Access*, vol. 7, pp. 42210–42219, 2019.

[17] L. Nie, Z. Ning, X. Wang, X. Hu, J. Cheng, and Y. Li, "Data-driven intrusion detection for intelligent Internet of vehicles: A deep convolutional neural network-based method," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 4, pp. 2219–2230, Oct. 2020.

[18] S. Garg, K. Kaur, N. Kumar, G. Kaddoum, A. Y. Zomaya, and R. Ranjan, "A hybrid deep learning-based model for anomaly detection in cloud datacenter networks," *IEEE Trans. Netw. Service Manage.*, vol. 16, no. 3, pp. 924–935, Sep. 2019.

[19] J. Gao, L. Gan, F. Buschendorf, L. Zhang, H. Liu, P. Li, X. Dong, and T. Lu, "Omni SCADA intrusion detection using deep learning algorithms," *IEEE Internet Things J.*, vol. 8, no. 2, pp. 951–961, Jan. 2021.

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

[21] M. Sheikhan, Z. Jadidi, and A. Farrokhi, "Intrusion detection using reduced-size RNN based on feature grouping," *Neural Comput. Appl.*, vol. 21, no. 6, pp. 1185–1190, Sep. 2012.

[22] A. Chawla, B. Lee, S. Fallon, and P. Jacob, "Host based intrusion detection system with combined CNN/RNN model," presented at the Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases, 2019.

[23] V. Priya, I. S. Thaseen, T. R. Gadekallu, M. K. Aboudaif, and E. A. Nasr, "Robust attack detection approach for IIoT using ensemble classifier," *Comput., Mater. Continua*, vol. 66, no. 3, pp. 2457–2470, 2021.

[24] Y. Zhou, T. A. Mazzuchi, and S. Sarkani, "M-AdaBoost—A based ensemble system for network intrusion detection," *Expert Syst. Appl.*, vol. 162, Dec. 2020, Art. no. 113864.

[25] H. S. Yang, "Honeypot using dynamic allocation technique with IP scan," presented at the IT Converg. Secur., Dordrecht, The Netherlands: Springer, 2013, pp. 197–204.

[26] C. Yuan, J. Du, M. Yue, and T. Ma, "The design of large scale IP address and port scanning tool," *Sensors*, vol. 20, no. 16, p. 4423, Aug. 2020.

[27] M. A. Al-shareeda, M. Anbar, S. Manickam, and I. H. Hasbullah, "Review of prevention schemes for Man-In-The-Middle (MITM) attack in vehicular ad hoc networks," *Int. J. Eng. Manage. Res.*, vol. 10, no. 3, pp. 153–158, Jun. 2020.

[28] H. Zhang, P. Cheng, L. Shi, and J. Chen, "Optimal DoS attack scheduling in wireless networked control system," *IEEE Trans. Control Syst. Technol.*, vol. 24, no. 3, pp. 843–852, May 2016.

[29] *Electronic Railway Equipment–Train Communication Network (TCN)—Part 2-5: TCN Communication Profile*, document IEC, 61375-5, 2015.

[30] J. Sim, J. S. Lee, and O. Kwon, "Missing values and optimal selection of an imputation method and classification algorithm to improve the accuracy of ubiquitous computing applications," *Math. Problems Eng.*, vol. 2015, pp. 1–14, Jan. 2015.

[31] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[32] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 25, 2012, pp. 1097–1105.

[33] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–14.

[34] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[35] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, *arXiv:1412.3555*. [Online]. Available: http://arxiv.org/abs/1412.3555

[36] D. Chakraborty, V. Narayanan, and A. Ghosh, "Integration of deep feature extraction and ensemble learning for outlier detection," *Pattern Recognit.*, vol. 89, pp. 161–171, May 2019.

[37] N. Yu, L. Qian, Y. Huang, and Y. Wu, "Ensemble learning for facial age estimation within non-ideal facial imagery," *IEEE Access*, vol. 7, pp. 97938–97948, 2019.

[38] F. Aydın and Z. Aslan, "The construction of a majority-voting ensemble based on the interrelation and amount of information of features," *Comput. J.*, vol. 63, no. 11, pp. 1756–1774, Nov. 2020.

[39] Z. H. Zhou, "Model evaluation and selection," in *Machine Learning*. Beijing, China: Tsinghua Univ. Press, 2016, ch. 2, pp. 24–47.

[40] H. Ji, Y. Wang, H. Qin, Y. Wang, and H. Li, "Comparative performance evaluation of intrusion detection methods for in-vehicle networks," *IEEE Access*, vol. 6, pp. 37523–37532, 2018.

CHUAN YUE (Graduate Student Member, IEEE) received the B.Eng. degree in electrical engineering from Beijing Jiaotong University, Beijing, China, in 2015, where he is currently pursuing the Ph.D. degree with the Electrical Engineering School. His research interests include machine learning, network intrusion detection, and train communication network security.

LIDE WANG received the M.Eng. degree from Southwest Jiaotong University, China, in 1986. He is currently a Professor with the School of Electrical Engineering, Beijing Jiaotong University. His research interests include control of electric traction systems, computer control networks technique, and embedded systems and applications. He is also a Senior Member of the China Railway Society.

DENGRUI WANG received the B.Eng. degree in electrical engineering and automation from the Shandong University of Science and Technology, in 2019. He is currently pursuing the master's degree in electrical engineering with Beijing Jiaotong University, Beijing, China. His research interests include ensemble learning and cybersecurity of the train communication networks.

RUIFENG DUO received the B.Eng. degree in electrical engineering and automation from Beijing Jiaotong University, Beijing, China, in 2019, where he is currently pursuing the master's degree in electrical engineering. His research interest includes cybersecurity of the train communication networks.

XIAOBO NIE received the B.Eng. and Ph.D. degrees from Beijing Jiaotong University, China, in 2005 and 2011, respectively. She is currently an Associate Professor with the School of Electrical Engineering, Beijing Jiaotong University. Her research interests include information physical system security and time-sensitive networks.