Hindawi
Mobile Information Systems
Volume 2021, Article ID 1440259, 13 pages
https://doi.org/10.1155/2021/1440259

Research Article

Intrusion Detecting System Based on Temporal Convolutional Network for In-Vehicle CAN Networks

Dongxian Shi,[1,2] **Ming Xu**,[1] **Ting Wu**,[1] **and Liang Kou**[1]

[1]*School of Cyberspace, Hangzhou Dianzi University, Hangzhou 310018, China*
[2]*College of Information Technology, Zhejiang Institute of Economics and Trade, Hangzhou 310018, China*

Correspondence should be addressed to Liang Kou; kouliang@hdu.edu.cn

Received 8 July 2021; Revised 23 August 2021; Accepted 7 September 2021; Published 24 September 2021

Academic Editor: Vishal Sharma

In recent years, deep learning theories, such as Recurrent Neural Networks (RNN) and Convolutional Neural Networks (CNN), have been applied as effective methods for intrusion detection in the vehicle CAN network. However, the existing RNNs realize detection by establishing independent models for each CAN ID, which are unable to learn the potential characteristics of different IDs well, and have relatively complicated model structure and high calculation time cost. CNNs can achieve rapid detection by learning the characteristics of normal and attack CAN ID sequences and exhibit good performance, but the current methods do not locate abnormal points in the sequence. To solve the above problems, this paper proposes an in-vehicle CAN network intrusion detection model based on Temporal Convolutional Network, which is called Temporal Convolutional Network-Based Intrusion Detection System (TCNIDS). In TCNIDS, the CAN ID is serialized into a natural language sequence and a word vector is constructed for each CAN ID through the word embedding coding method to reduce the data dimension. At the same time, TCNIDS uses the parameterized Relu method to improve the temporal convolutional network, which can better learn the potential features of the normal sequence. The TCNIDS model has a simple structure and realizes the point anomaly detection at the message level by predicting the future sequence of normal CAN data and setting the probability strategy. The experimental results show that the overall detection rate, false alarm rate, and accuracy rate of TCNIDS under fuzzy attack, spoofing attack, and DoS attack are higher than those of the traditional temporal convolutional network intrusion detection model.

1. Introduction

With the development of technologies such as the Internet of Vehicles, unmanned driving, and software-defined cars, modern cars are equipped with more and more advanced sensing devices and intelligent control systems [1], making cars more intelligent and providing people with a more comfortable driving service. However, with the increase of the number of electronic control units (ECU), sensing devices, ports, etc., and the diversity of networking, the attack surface of automobiles has become more and more extensive [2] and many security researchers have demonstrated the vehicles' vulnerability to attacks. For example, Miller et al. used WiFi open ports to invade a car's in-vehicle CAN network [3] by analyzing the CAN communication protocol [4], i.e., sending protocol data to the bus to cause car brake failure and engine stop. Therefore, the in-vehicle network security problem has become the focus of automotive safety, especially the CAN network commonly used in automobiles [5].

Intrusion detection is an effective method to solve the problem of in-vehicle network security, of which the study of CAN data as a sequence is an important research field of current intrusion detection. The normal CAN ID sequence features are extracted through sequence learning, and when a nonexistent sequence appears in the network, the intrusion detection system detects it as an abnormality [6, 7]. Taylor et al. proposed an intrusion detection method based on Long Short-Term Memory (LSTM) [8], which directly inputs the original CAN data packets into the model, and predicted network traffic through a short time sequence of dozens of data packets. This method of learning sequence through recurrent neural network

effectively realized intrusion detection, but establishing subsequences and corresponding models for each independent CAN ID will cause the loss of sequence relationships between different IDs and reduce the efficiency of intrusion detection. Song et al. proposed an intrusion detection method based on a deep convolutional neural network [9], which learned normal and attack CAN ID sequence features through the convolutional network and achieved a higher detection rate while using the parallel processing capability of the convolutional network to reduce the time cost. However, it does not locate abnormal points and the abnormal detection of the message level is not realized.

To solve the above problems, an intrusion detection system based on temporal convolution network is proposed in this paper. We choose temporal convolution network because it shows excellent performance and efficiency on different tasks and data sets [10]. In our TCNIDS model, the original CAN data are directly regarded as a sequence, the probability of each CAN ID in the future sequence is predicted by word embedding encoding, and the time convolution model is learned and decoded, so as to realize the anomaly detection at the message level.

Contributions of this paper are the following:

(1) The temporal convolutional network model is applied to the intrusion detection of in-vehicle CAN network for the first time. The model has a simple structure, and effectively realizes the message-level prediction and anomaly detection.

(2) CAN IDs are encoded as words by using the word embedding method, which effectively represents the potential features between IDs and improves the performance of the model. At the same time, word embedding reduces the dimension of data and improves the computational efficiency of the model.

(3) PReLU activation function is used to improve the TCN model, and the performance of this activation function in TCNIDS model is compared and analyzed.

The remainder of this paper is organized as follows. We present the background material about CAN bus and intrusion detecting system in Section 2. The framework of the IDS is proposed and introduced in detail in Section 3. In Section 4, we present our experiment environment, evaluation metrics, and results, and give our conclusions in Section 5.

2. Background

2.1. CAN Bus and Its Features' Analysis. CAN is a field bus with high reliability, strong real-time performance, and low flexibility [4]. It is a standard bus of automobile in-vehicle control system and realizes the communication between in-vehicle electronic control units (ECUs). CAN network is an important part of the entire in-vehicle network. It is a peer-to-peer network, where each ECU node in the CAN network not only receives messages but also sends messages actively. Its main features are as follows:

2.1.1. Realize the Message Exchange between ECUs by Broadcast. Each ECU node in the CAN network sends messages by broadcast, and all ECU nodes in the CAN network receive messages. There are 5 types of messages: data frame, error frame, remote frame, inter-frame space, and overload frame. Figure 1 shows the structure of CAN standard data frame.

2.1.2. Adopt Arbitration Mechanism to Avoid Message Conflict. The CAN network provides an arbitration mechanism to avoid conflicts caused by different ECU nodes sending messages to the CAN network simultaneously. Each ECU carries out line and operation between its own messages to be sent and the ID of other messages, that is, comparing the bits of the arbitration field, if it is the dominant bit 0, it will continue to get the control of the bus; if it is the recessive bit 1, it will lose the arbitration, and turn to be the receiving state from the next bit, until the bus is idle.

2.1.3. Increase ECU and External Interfaces. With the improvement of vehicle intelligence, more and more mechanical parts are replaced by ECU. At present, the number of ECUs in some luxury cars is more than 100 [11], while the increasing demand for network communication and entertainment experience has greatly enriched the external interface of vehicles. For example, Tesla carries out remote software upgrade of ECU through OTA (Over-the-Air) [12], which is a technology to download new software update packages from a remote server through the network to upgrade its own system.

2.1.4. Implement Simple Data Check Code. In order to ensure the real-time performance and functional requirements of the vehicle to the greatest extent, the CAN network only includes a simple data check code when designing the message structure, and does not identify the identity ID of the message sender. Therefore, the protocol lacks security mechanism, such as encryption, access control, and message authentication. At the same time, this broadcast method allows all ECUs to easily obtain message information, which is easy to be sniffed by attackers.

2.2. Intrusion Detecting System. There have been many researches on intrusion detection of in-vehicle CAN networks. Hamada et al. learned the behaviour patterns under normal and attack environments by analyzing the periodicity of CAN messages [11]. Ji et al. believed that although the frequency of the ECU transceiver is fixed, the clock drift [13] would occur because the crystal oscillator was not exactly the same, so the accumulation of clock drift was used as the fingerprinting feature of the ECU [14]. Müter and Asaj applied information entropy to intrusion detection of in-vehicle network through maximum entropy estimation method [15], which can detect abnormal conditions of network traffic. However, these intrusion detection models are targeted at specific attacks, and so their application is limited.

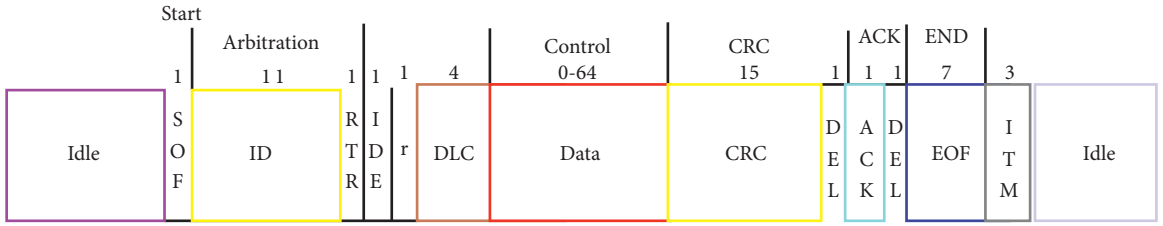

Figure 1: CAN standard data frame.

In view of these limitations, some literature studies [8, 16, 17] build intrusion detection model of in-vehicle CAN network through deep learning theory. We divide deep learning methods into 3 categories: RNN, CNN, and others. RNN, as a deep learning model for time series data processing, has been widely used in many fields. Taylor et al. proposed an intrusion detection method based on LSTM [8] to solve the problems of gradient disappearance, and short memory existed in RNN itself, which directly input original CAN packets into the model and can predict network traffic within a short time scale of dozens of packets. Another advantage of this raw traffic forecasting is that the model does not require domain knowledge. Hanselmann et al. proposed an intrusion detection system CANet based on LSTM and AutoEncoder [18]. The system introduced an independent LSTM model for each CAN ID to learn the time dynamic characteristics of each message-related signal, and then aggregated all IDs and used the AutoEncoder model to learn the interdependence between signals. The AutoEncoder included an Encoder and a Decoder. The Encoder mapped the high-dimensional input data to the low-dimensional embedding space, which could be used for dimensionality reduction. At the same time, the Decoder was used to reconstruct the low-dimensional embedding space of the representation, which could be compared with the original input data for deviation comparison, so as to effectively identify anomalies. In addition, for the first time, it used an Autoencoder to naturally process the data structure of the high-dimensional CAN bus. Wang et al. proposed a distributed anomaly detection system based on the hierarchical time memory (HTM) algorithm [19], which effectively realized the real-time prediction of the original CAN traffic data at the bit level. The method in [8, 18, 19] causes the loss of some information and relationships in the CAN network by establishing a model for an independent CAN ID or ECU [20], and the model becomes more complicated. Kang and Kang proposed a deep neural network (DNN)-based intrusion detection method [21], which used an unsupervised deep belief network (DBN) to pretrain the initialization parameters and test it on the simulation data set generated by the OCTANE platform. Usually, when training a model, it is considered that DNN and LSTM consume more time than CNN. Based on this fact, Song et al. proposed an intrusion detection method based on deep convolutional neural network [9], by simplifying the Inception-ResNet model. The method achieves a higher detection rate and reduces the time cost. However, this method cannot effectively locate the message level detection by detecting whether the sequence has an attack. In addition, some current studies do not use a single method but use the advantages of various methods to mix them. Xiao et al. combined LSTM and CNN to treat CAN network traffic data as a whole from the two dimensions of time and space [20], and proposed a convLSTM model, which can better extract the potential features of normal data flow, so as to predict the deviation attack behaviour of the time series more effectively. However, it needs to be improved in terms of threshold selection and real-time detection performance.

3. Methodology

In this section, first we present the overview of the TCNIDS model for in-vehicle CAN network. Then, we introduce each model component in detail.

3.1. Model Overview. The traffic data in the in-vehicle CAN network appears in the form of sequence. Due to the arbitration mechanism of CAN network and the periodicity of message transmission, there is a dependency on the appearance of the message sequence [22]. On the bus, each ECU in the CAN network follows the CAN protocol to send and receive messages, and there is a certain relationship between the previous message and the next message. Therefore, we convert the intrusion detection of CAN traffic data into sequence prediction for research. We learn to extract normal sequence features, and when a nonexistent sequence appears in the network, the intrusion detection system will identify it as an abnormality and determine which message is inconsistent with the predicted sequence result. At present, in the field of time series forecasting, time convolutional networks have shown excellent performance and efficiency on various data sets and tasks. Therefore, this paper chooses time convolution as the basis of the entire model. Assume that there is an input sequence $X_{t-s:\,t} = \{x_{t-s}, x_{t-s+1}, \ldots, x_{t-1}, x_t\}$ at each time interval t, the objective is that the model can predict the corresponding output sequence $\widehat{Y}_{t-s:\,t} = \{\widehat{y}_{t-s}, \widehat{y}_{t-s+1}, \ldots, \widehat{y}_{t-1}, \widehat{y}_t\}$. Formally, the model is an arbitrary function f: $X_{t-s:\,t} \longrightarrow \widehat{Y}_{t-s:\,t}$:

$$\widehat{Y}_{t-s:\,t} = f\left(X_{t-s:\,t}\right). \tag{1}$$

The goal of the model is to train the function f to minimize the loss function $\mathrm{Loss}(Y_{t-s:\,t}, f(X_{t-s:\,t}))$ between the model output sequence and the real sequence. The loss function of this model training adopts cross entropy, and the specific expression is as follows:

$$\text{Loss}\left(Y_{t-s:\,t}, f\left(X_{t-s:\,t}\right)\right) = -\frac{1}{S}\sum_{1}^{S}\sum_{i=1}^{M} y_i \log_a p_i, \quad (2)$$

where S denotes the number of messages in the sequence, M denotes the number of CAN message types, y_i denotes the true label of message category i, and p_i is the probability that the model predicts to belong to message category i.

TCN proposed the network structure shown in Figure 2. First, since the output length generated by the network in sequence prediction needs to be consistent with the input length, TCN uses a 1D fully convolutional network (FCN), and each hidden layer uses zero padding for length padding. Second, using future information to predict the past will lead to information leakage [10], so TCN introduces causal convolution [10] to ensure that the output at the current moment comes from the convolution of current and historical information. Third, having a longer historical memory requires a deeper network, but it will increase the number of parameters. Therefore, TCN uses expanded convolution to expand the receptive field of the convolution, thereby reducing the depth of the network as much as possible. Fourth, normalization can solve the problem of gradient vanishing or gradient exploding caused by the increase of network depth to a certain extent, but it will also bring about degradation problems. Therefore, the residual network [23] is introduced to solve this problem in TCN.

This paper, by extending TCN, proposes the intrusion detection model TCNIDS for in-vehicle CAN network. The overview of the model is shown in Figure 3.

The model has two stages: training and detection. The training stage learns the normal CAN data sequence and realizes the prediction of the next sequence by extracting potential sequence features, thereby learning the sequence law of normal behaviour. The detection stage checks all CAN data sequences including attack behaviours. Through observation, there is more than one possibility of the message predicted by the CAN sequence. Therefore, this paper uses the Top g probability strategy to detect anomalies in each message in the prediction sequence. If the predicted real message is in the message set with the top g probability, it is detected as normal, otherwise it is detected as abnormal. The following will introduce each component in the model in detail.

3.2. Data Preprocessing. The data set includes timestamp, CAN ID, DLC, Data, and Label. We only need to extract the two fields of CAN ID and Label to form the original ID sequence. Among them, CAN ID is extracted in the training phase, and CAN ID and Label are extracted in the anomaly detection phase for evaluating the performance of the TCNIDS model proposed in this paper.

3.3. Encoder. Since in One Hot encoding CAN ID, the distance between all IDs is the same, there is a disadvantage that the potential relationship between IDs cannot be extracted during model training; on the other hand, the word embedding coding method maps a word to a point in the semantic space, which makes the semantically similar words relatively close, and it can effectively characterize the relationship between IDs [24]. Therefore, this paper uses the word embedding method to treat each type of CAN ID in the data set as a word, uses a word vector to represent the CAN ID, and learns to extract the potential relationship between IDs, thereby improving the performance of the model. Figure 4 shows the process of CAN ID Embedding:

Step 1. Various types of IDs in the original CAN ID sequence are extracted to construct an embedding matrix. Each type of CAN ID is expressed as a word vector of the same dimension, and the initial vector is assigned a random value.

Step 2. Replace each ID in the original ID sequence according to the embedding matrix of CAN ID, which is represented by the word vector in **Step 1.**

Step 3. The embedded matrix constructed by **Step 1** and the ID sequence represented by **Step 2** are added to the corpus for the input data of model training and testing.

3.4. Temporal Convolutional Network. The TCNIDS model proposed in this paper extends on the general TCN model described in Ref. [10]. The TCN model has two main constraints. The output of the hidden layer in the middle of the model has the same length as the input, and the prediction at time t can only rely on the information before time t. For the first constraint, TCN uses a 1-D fully convolutional network (FCN) to convolve time series data, and uses zero padding to ensure the same length of the front and back network layers. Regarding the second constraint, TCN introduces causal convolution, so that the output at time t can only be convolved with time t and previous information, ensuring that the past cannot be predicted by future information, thereby causing information leakage. As shown in Figure 5, through causal convolution, one-dimensional convolution of past information is realized, and the potential features of CAN data sequence are effectively extracted.

Formally, set the convolution filter $F = (f_1, f_2, \ldots, f_k)$. For any element x_j in sequence $X_{t-k:\,t} = \{x_{t-k}, x_{t-k+1}, \ldots, x_{t-1}, x_t\}$, the causal convolution at x_j is defined as follows:

$$(F \oplus X)\left(x_j\right) = \sum_{i=1}^{K} f_i x_{j-K+i}, \quad (3)$$

where K denotes the size of the convolution kernel.

3.4.1. Dilated Convolutions. For the prediction of CAN data series, we expect the model to remember more historical information, so that the prediction performance will be more stable. However, with the above causal convolution method, to achieve a larger receptive field, it is necessary to stack many network layers to reach the goal. In order to overcome this problem, the dilated convolution is used to expand the receptive field of the convolution, which greatly reduces the number of intermediate hidden layers, which is also the biggest feature of the dilated convolution. In dilated convolution, filters are applied by skipping a certain number

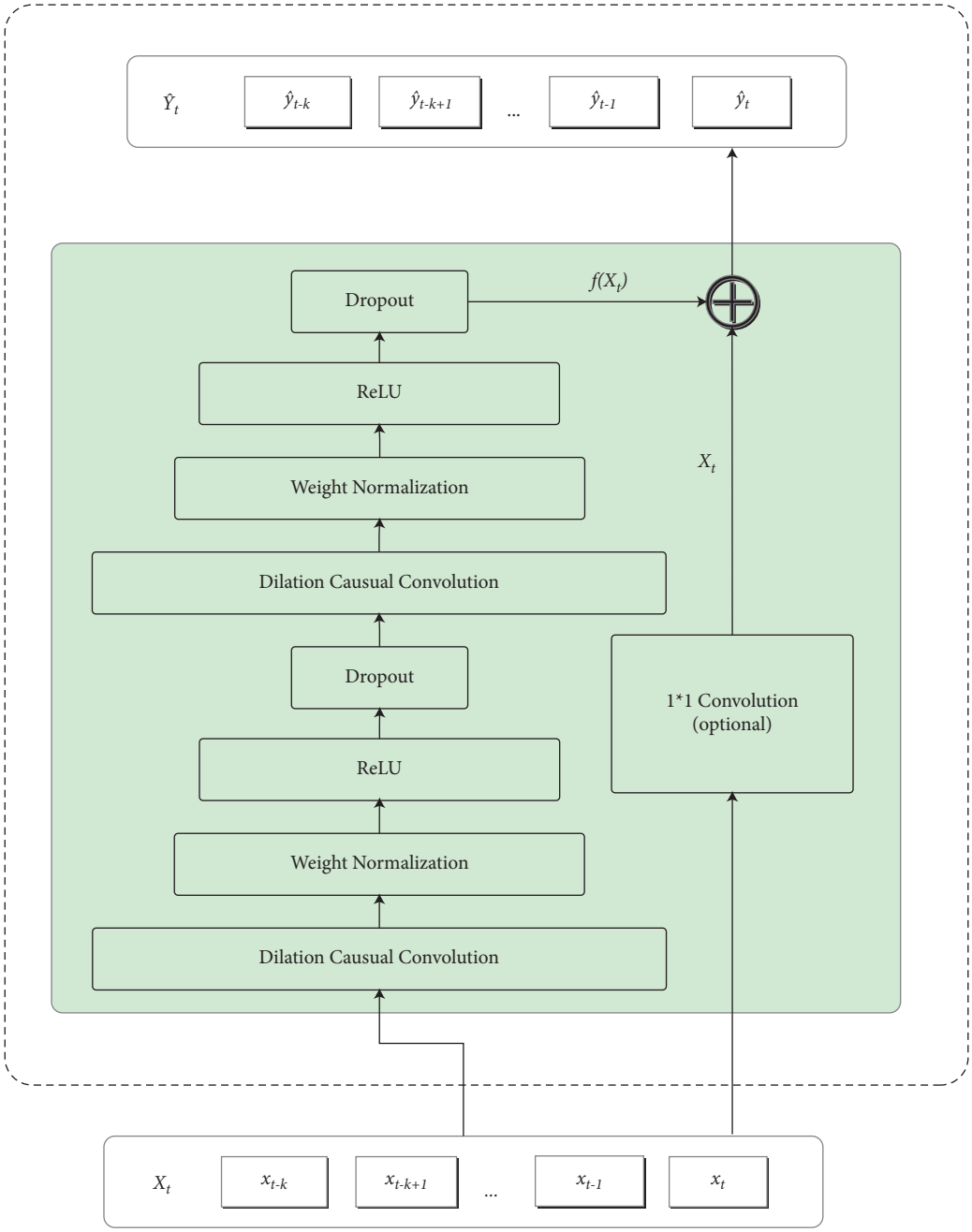

Figure 2: TCN network structure.

of steps according to the expansion factor d to achieve convolution of a larger area. As shown in Figure 6, this growth method of the receptive field is different from pooling operation, as it skips some existing elements. In general, d will increase exponentially as the network depth i increases, so the model can build a long memory.

Formally, set the convolution filter $F = (f_1, f_2, \ldots, f_k)$. For any element x_j in sequence $X_{t-k:\,t} = \{x_{t-k}, x_{t-k+1}, \ldots, x_{t-1}, x_t\}$, the causal convolution at x_j is defined as follows:

$$\left(F \oplus_d X\right)\left(x_j\right) = \sum_{i=1}^{K} f_i x_{j-(K-i)d}. \tag{4}$$

Among them, d is the expansion factor of the dilated convolution, and when $d = 1$, the convolution kernel degenerates into a general convolution operation.

3.4.2. Residual Connections. Since the receptive field of the TCN model depends on the network depth n, filter size k, and expansion factor d, making the TCN deeper and larger is the key to obtain a large enough receptive field [25]. The residual connection can simplify deep network training. The deep network through this structure has been proved to be very effective, which can speed up the training process and avoid the disappearance of gradients. As shown in Figure 2,

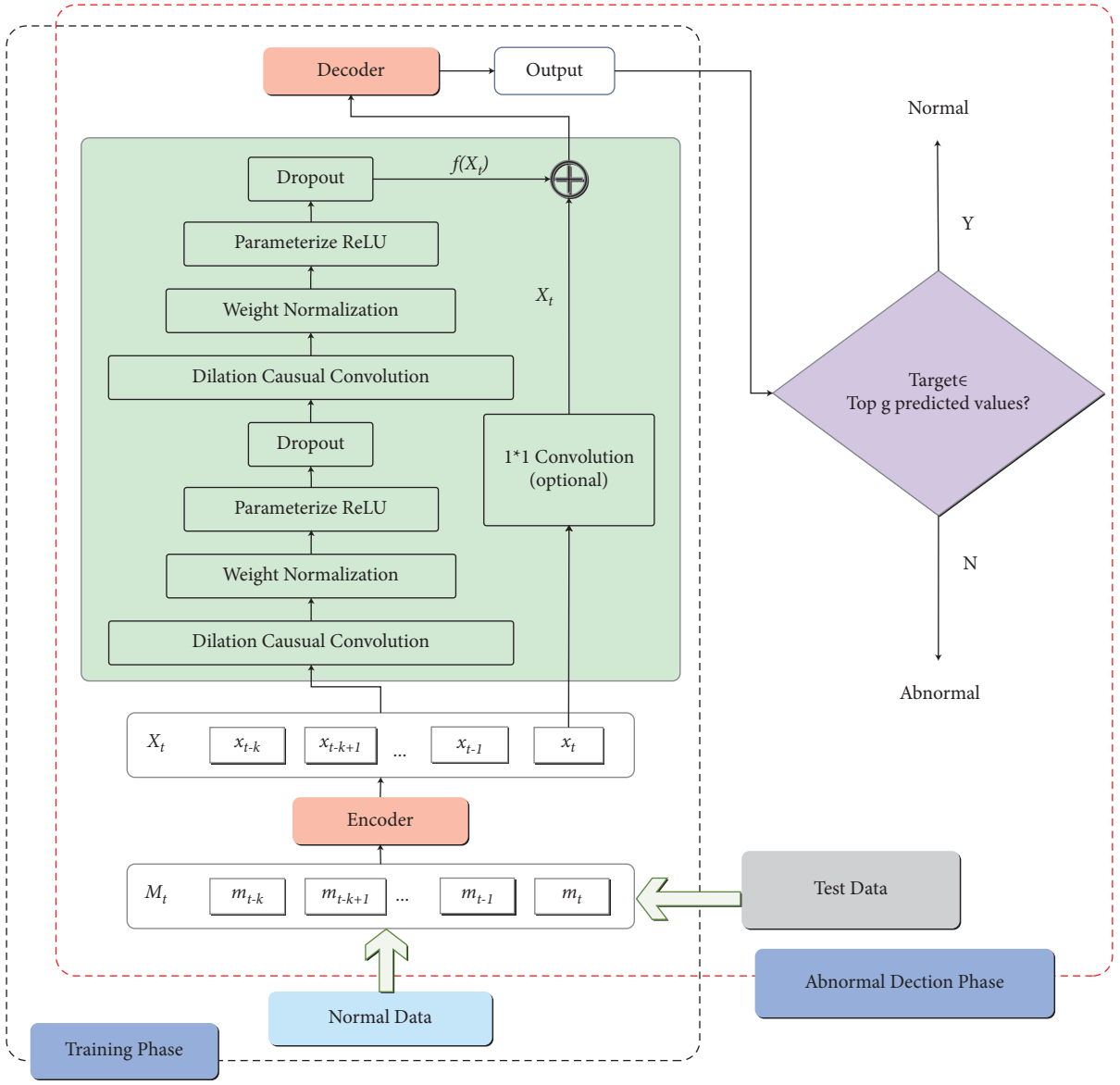

Figure 3: Illustration of the TCNIDS model.

the model is constructed by residual blocks in TCN. Each residual block contains two network layers, and each layer is composed of four parts: causal dilated convolution, normalization, activation function, and regularization. For normalization, we apply weight normalization to the convolution filter. Regularization can effectively prevent the over-fitting phenomenon of the model. In addition, in the standard ResNet [23], the input is directly added to the output of the residual function. While in TCN, the input and output may have different channel dimensions. In order to be able to perform residual operations, we use an additional $1 * 1$ convolution to ensure that the output and input of each layer have the same shape.

3.4.3. Activation Function Selection and Improvement. The original TCN model uses a one-dimensional convolutional network to extract features, and uses the Relu activation function to nonlinearly map the features [26]. As shown in Figure 7(a), when $x \geq 0$, the gradient of the Relu activation function is 1, and when $x < 0$, the gradient reduces to 0, so that the network can converge faster. This activation function is widely used in CNN. However, when $x < 0$, the output value of the convolution kernel operation is always 0, which causes many features to be masked, and the network cannot extract effective features. This phenomenon in which the Relu activation function is killed in the negative region is called "Dying" [27].

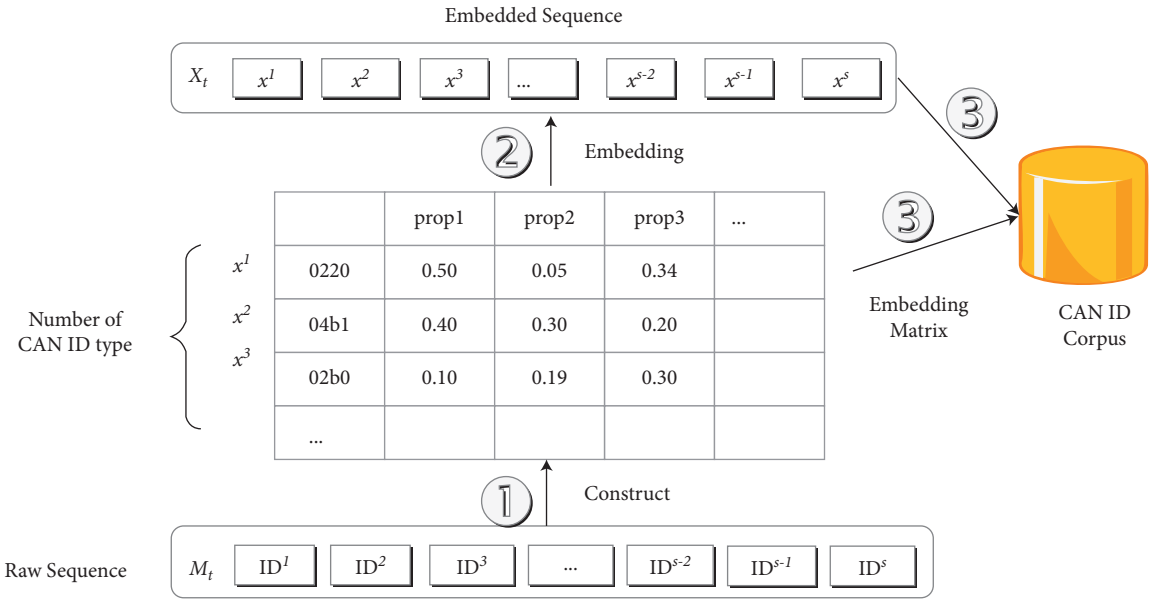

Figure 4: CAN ID encoding process.

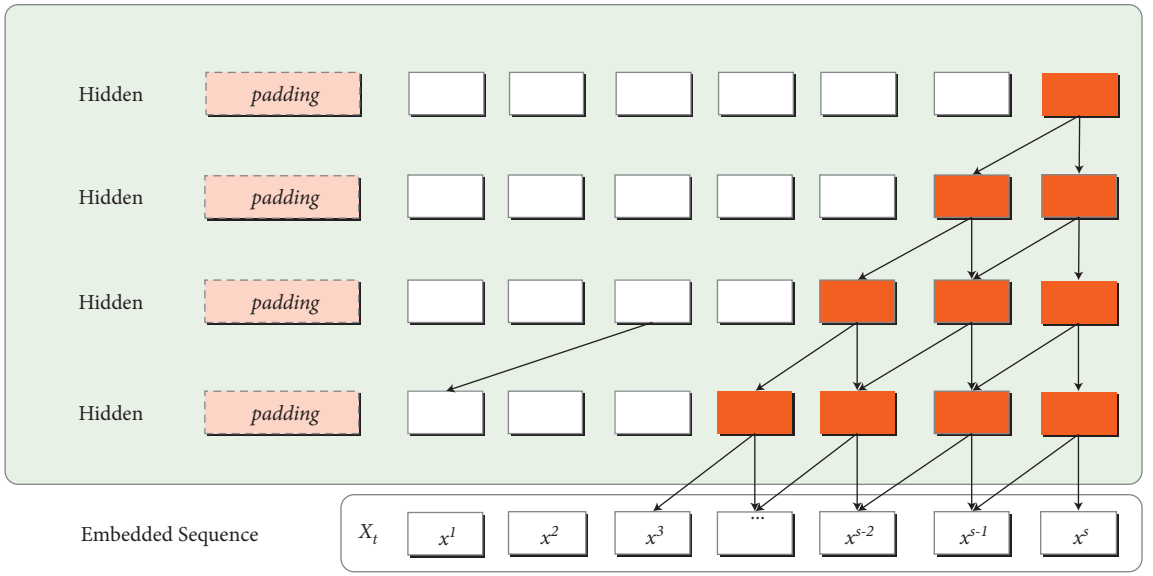

Figure 5: Causal convolutions.

In order to solve the problems of Relu, He et al. proposed a parameterized Relu function method [28], as shown in Figure 7(b). The parameter α is introduced in the parameterized Relu function. When $x < 0$, the gradient of the activation function will automatically change with the learning of the network, so as to obtain the optimal value of the model.

3.5. Decoder. One goal of the model is to predict the probability of which type of CAN ID each message in the sequence belongs to, that is, the target dimension is the number of CAN ID types, but the output dimension obtained through the time convolutional network model is different from this target dimension, so it is necessary to realize the transformation of these two dimensions through decoding. Since the number of CAN IDs in the in-vehicle network is not much, generally within 100, we adopt the simple method of full connection to directly realize the transformation of two dimensions.

4. Experiments

This section firstly introduces the CAN data sets, experimental environment, and evaluation metric, and then illustrates the optimized parameter settings for the training and detecting. Finally, the performance of the model is deeply analyzed through the experimental results.

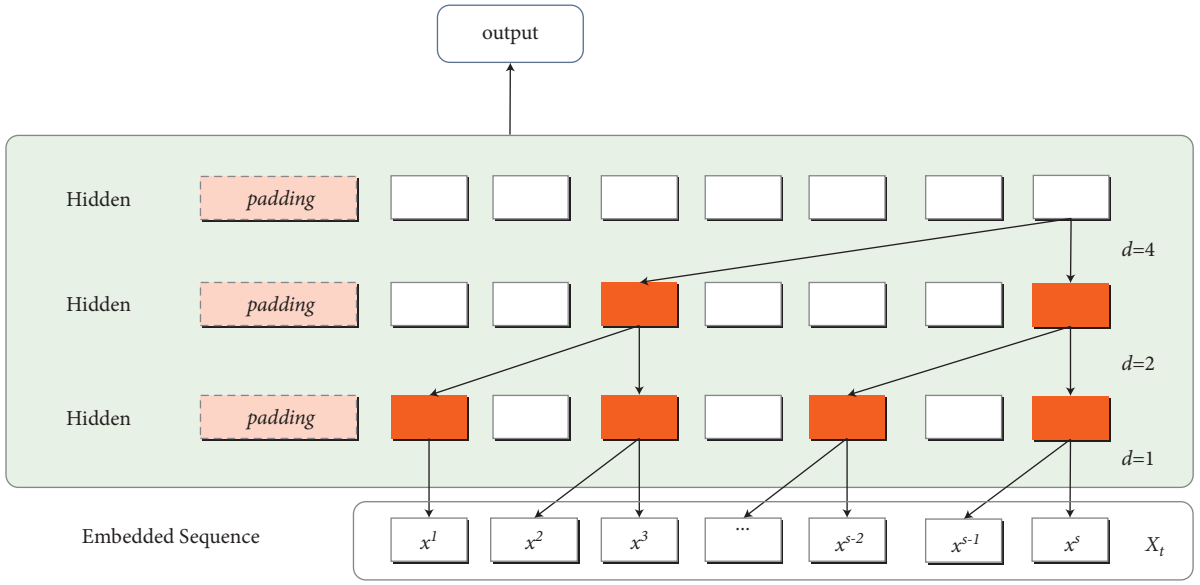

Figure 6: Dilated convolutions.

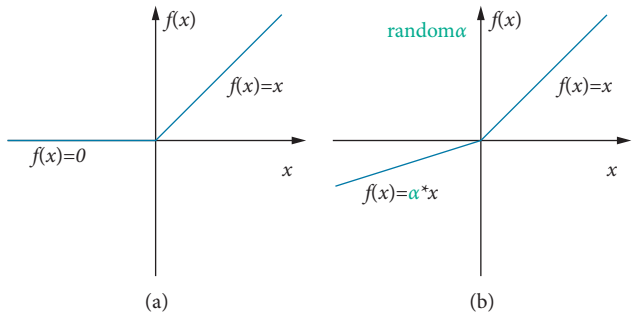

Figure 7: Activation function curve. (a) Relu. (b) Parameterized Relu.

4.1. Data Sets and Experimental Environment. This paper adopts the public CAN data sets provided in [12], which are collected by the Kia Soul test vehicle and contain 17558346 CAN messages. The data sets can be divided into Normal, Fuzzy Attack, Spoof Gear attack, Spoof RPM Attack, and DoS Attack. The information of the CAN data sets is shown in Table 1.

The length of the CAN ID of the data sets used in this paper is 11 bits. Before input to the model, CAN ID of all the data sets above will be extracted to form the corresponding ID sequence data. In this paper, the data sets are not divided according to the fixed time, but are divided according to the sequence length specified in the model parameters, and the method of sliding window is used to extract the next sequence. 80% of the normal data set is selected as the training set, and 20% of the normal data set and the other 4 attack data sets are selected as the test set.

The experimental environment in which the TCNIDS model is tested in this paper is shown in Table 2.

4.2. Evaluation Metric. In order to evaluate the detection performance of the proposed TCNIDS model, we firstly define the confusion matrix for intrusion detection shown in Table 3.

Among them, TP denotes the number of CAN messages that are abnormal and predicted to be abnormal, FN denotes the number of CAN messages that are abnormal but predicted to be normal, FP denotes the number of CAN messages that are normal but predicted to be abnormal, and TN denotes the number of CAN messages that are normal and predicted to be normal. According to this confusion matrix, three indicators are specifically defined to evaluate the ability of real CAN messages to be predicted by TCNIDS as normal or abnormal.

4.2.1. Detecting Rate. Detecting Rate is also known as True Positive Rate (TPR). This paper uses TPR to represent the detection rate, which represents the proportion of abnormal packets predicted to be the total number of abnormal packets. The higher the value, the better the performance. The specific formula is as follows:

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}. \tag{5}$$

Table 1: CAN data set information.

Data set	Total messages	Normal messages	Attack messages
Normal	988871	988871	0
Fuzzy attack	3838860	3347013	491847
Gear spoof	4443142	3845890	597252
RPM spoof	4621702	3966805	654897
DoS attack	3665771	3078250	587521

Table 2: Experiment environment.

Configuration item	Configuration parameter
CPU	Intel Xeon Gold 5118@2.3 GHz * 24
RAM	16.0 GB
GPU	NVIDIA Quadro P4000 GPU
Operating system	Windows Server 2012 R2

Table 3: Confusion matrix of TCNIDS.

Packet	Predicted attack	Predicted normal
True attack	TP	FN
True normal	FP	TN

4.2.2. False Positive. False Positive Rate represents the ratio of the number of normal packets predicted to be abnormal to the total number of normal packets. The lower the value, the better the performance. The specific formula is as follows:

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{TN} + \mathrm{FP}}. \tag{6}$$

4.2.3. Accuracy. Accuracy represents the proportion of the number of correctly predicted packets to the total number of packets. The higher the value, the better the performance. The specific formula is as follows:

$$\mathrm{accuracy} = \frac{\mathrm{TN} + \mathrm{TP}}{\mathrm{TN} + \mathrm{TP} + \mathrm{FP} + \mathrm{FN}}. \tag{7}$$

For the normal data and four types of attack data in the test set, this paper adopts the same sequence length as the training set, and inputs the sequence data to the model for intrusion detection according to the detection process. If the predicted sequence satisfies the ID of the CAN message in the first q message categories with high probability, then the model will update TN or FN according to the real message label, and otherwise update TP or FP.

4.3. Parameter Setting. In this paper, through repeated experiments with different parameter combinations, the optimal parameters of the model are determined, and subsequent experiments all use the optimal parameters for experimental evaluation and comparison. The specific parameter settings are shown in Table 4.

We apply the SGD algorithm to ensure the convergence in the experiment, and for a faster convergence, a learning rate (lr) annealing method is adopted. When the loss is greater than the loss of the previous 5 times, set $\mathrm{lr} = \mathrm{lr}/2$.

Table 4: Parameter setting.

Parameter item	Parameter value
Embedding size	200
Kernel size	5
Layer number	4
Hidden units	150
Convolution dropout	0.45
Embedding dropout	0.25
Initial learning rate	0.50
Gradient clip	0.35
Batch size	16
Sequence length	60
Top g	16

4.4. Result Analysis

4.4.1. Overall Result. The test results of TCNIDS proposed in this paper on the test data sets are shown in Table 5

It can be seen from Table 5 that TCNIDS exceeds 93% on both TPR and Accuracy indicators, and the FPR is not higher than 5%. Especially for normal behaviour, Fuzzy attack, and DoS attack, the TCNIDS model has good detection capabilities. The TPR and Accuracy detected by the model on the normal data set are close to 100%, and the FPR is close to 0. In the case of DoS attacks, TPR is also close to 100%. In addition, the model's TPR and Accuracy indicators for detecting Fuzzy attack are not less than 98%. Since DoS attack and Fuzzy attack themselves are uncommon CAN ID injections, according to the given method of word embedding, their value in the word vector will gradually differ from the normal CAN ID as the model is trained; so, TCNIDS can easily detect these two attacks. At the same time, the model also shows good performance in Gear Spoof Attack and RPM Spoof Attack. TPR and Accuracy also exceed 93%, and FPR is lower than 3%.

4.4.2. Detail Performance. In order to thoroughly analyze the performance of the TCNIDS model in intrusion detection, we collected more experimental results. Figure 8 shows the change of loss during a single epoch of training. It can be seen from Figure 8 that the loss drops and converges rapidly. In this paper, a variety of methods, such as weight normalization, regularization, time convolution network, and residual network, are used to deal with the problem of gradient dispersion and disappearance, which improves the stability of model training and further verifies the effectiveness of the model.

In order to observe the loss of the model on each data set, we trained the model for 50 epochs. Figure 9 shows the changes of loss on each data set. It can be seen from the figure that, in the training stage, with the increase of model training times, loss in the normal data set rapidly declines and converges. In the test stage, the trained model carries out loss calculation for each data set, and it is not difficult to find that the normal data set still maintains the loss similar to that in the training stage, but the loss of each attack data set is at a high level and fluctuates greatly due to the attack behaviour, especially the gear spoofing attack. Therefore, loss can be

Table 5: Detecting results using TCNIDS.

Data set (%)	TPR	FPR	Accuracy
Normal	99.999	0.001	99.999
Fuzzy attack	97.552	0.027	98.345
Gear spoof	93.526	0.039	96.290
RPM spoof	93.078	0.046	94.626
DoS attack	100.000	0.034	98.496

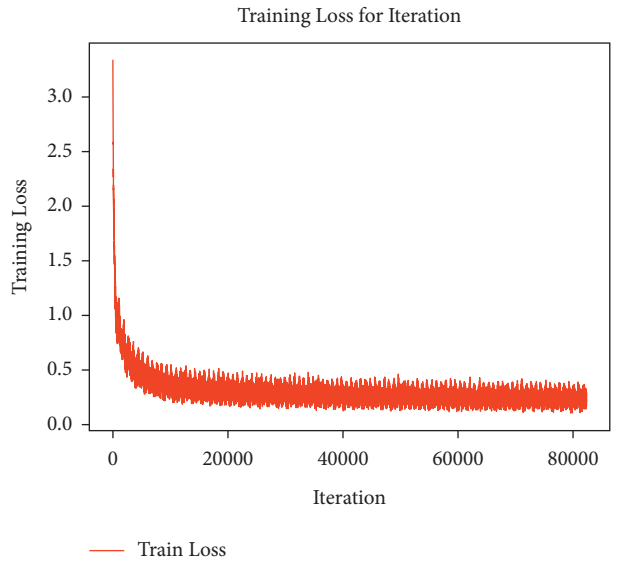

Figure 8: Loss curve for training.

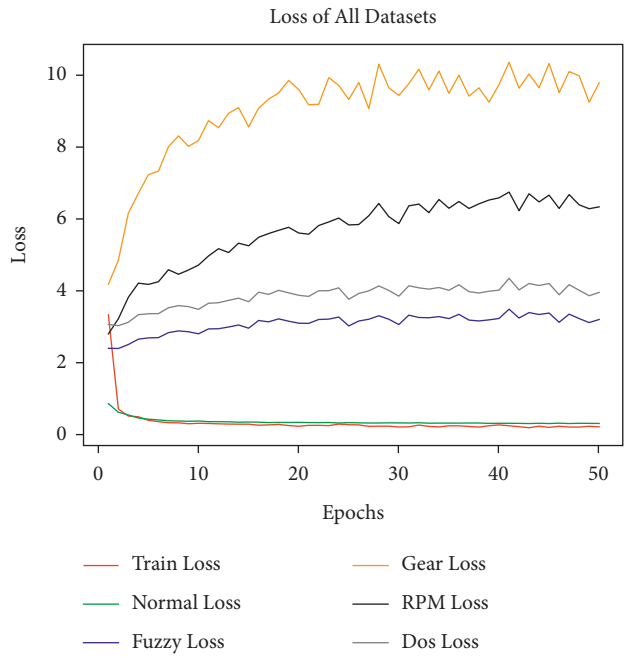

Figure 9: Loss comparison for all data sets.

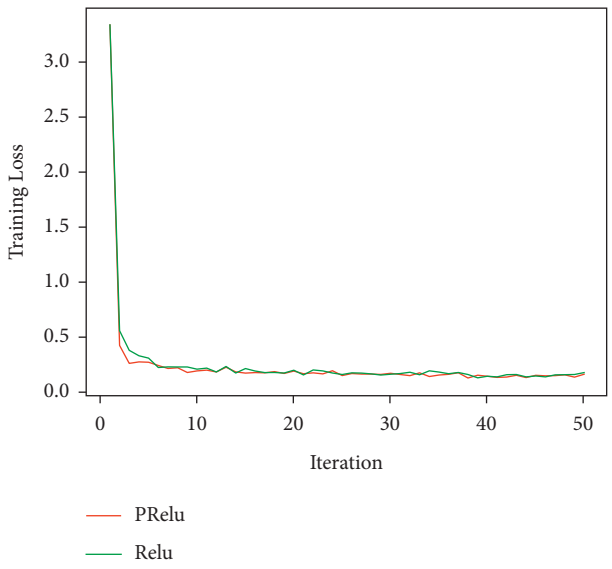

Figure 10: The influence of Relu and parameterized Relu on loss.

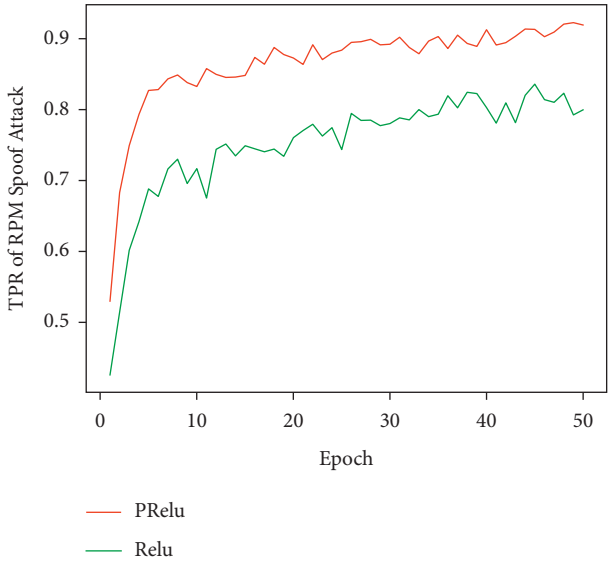

Figure 11: The influence of Relu and parameterized Relu on TPR.

used to distinguish between normal and abnormal behaviour patterns.

It can be seen from Figure 10 that after improving the model by parameterizing the Relu activation function, the convergence rate is faster. It can be seen from Figures 11 and 12 that the TPR and Accuracy have been improved to a certain extent, indicating that the existing features should be preserved as much as possible when nonlinear mapping of the features through the activation function and the full shielding method cannot be adopted when $x < 0$. This also verifies the effectiveness of the parameterized Relu method in Section 4 to improve the model.

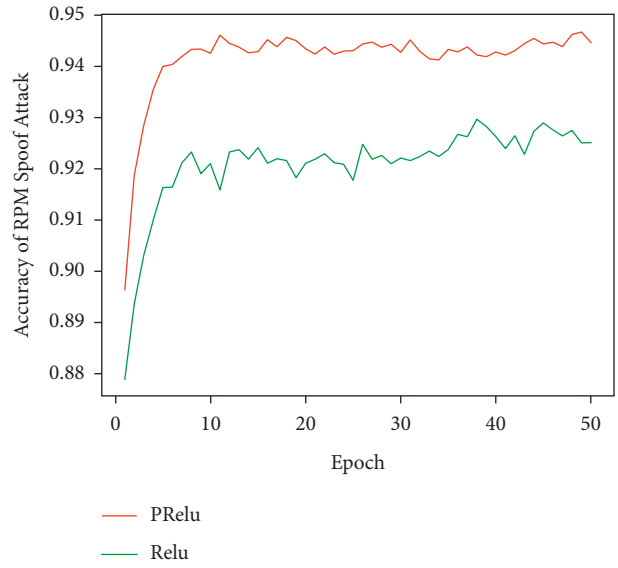

Figure 12: The influence of Relu and parameterized Relu on accuracy.

5. Conclusion

With the increase of the attack surface of modern automobiles, intrusion detection systems have become the most important technology for in-vehicle network security protection. In view of the current problems in the implementation of the in-vehicle CAN network anomaly detection through the deep learning network model, this paper proposes an intrusion detection system based on time convolution network. The structure of the model is simple, and the sequence data are predicted by word embedding encoding, time convolution network and decoding, and the intrusion detection is realized by top g strategy. In the model, the word embedding method encodes CAN ID into words, which effectively characterizes the potential features between IDs, improves the performance of the model, reduces the dimensionality of the data, and improves the computational efficiency of the model. At the same time, the TCNIDS model uses the parameterized Relu activation function to try to retain the characteristics of nonlinear mapping when $x < 0$, and optimize the performance of the model. The experimental results show that the TCNIDS model proposed in this paper has high performance in Fuzzy attack, Spoof attack, and DoS attack, especially Fuzzy attack and DoS attack. At the same time, compared with the ordinary time convolutional network model, the improved model has a certain improvement in detection rate, false alarm rate, and accuracy rate, which also proves the effectiveness of the method. Therefore, the TCNIDS proposed in this paper can strengthen the security of the in-vehicle CAN network. Since the model uses an unsupervised learning method, in the future, we will apply it to more data sets and attack scenarios, and further improve the performance of the model.

Data Availability

The data used to support the findings of the study are available at Korea University and Hacking and Countermeasure Research Lab (https://ocslab.hksecurity.net/Datasets/CAN-intrusion-dataset).

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The authors thank Korea University and Hacking and Countermeasure Research Lab for publishing the CAN intrusion data sets. This work was partially supported by the "Fundamental Research Funds for the Provincial Universities," Zhejiang Institute of Economics and Trade (Grant Number: 19SBYB06), and the Domestic Visiting Engineer Project of Universities, the Education Department of Zhejiang (Grant Number: FG2020135).

References

[1] W. Wu, R. Li, G. Xie et al., "A survey of intrusion detection for in-vehicle networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 3, pp. 919–933, 2019.

[2] Z. Abdollahi Biron, S. Dey, and P. Pisu, "Real-time detection and estimation of denial of service attack in connected vehicle systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 12, pp. 3893–3902, 2018.

[3] C. Miller and C. Valasek, *Remote Exploitation of an Unaltered Passenger Vehicle*, Black Hat, Washington, DC, USA, 2015.

[4] B. Parikh, "CAN protocol: understanding the controller area network," 2021, https://www.engineersgarage.com/can-protocol-understanding-the-controller-area-network-protocol/.

[5] Q. Luo and J. Liu, "Wireless telematics systems in emerging intelligent and connected vehicles: threats and solutions," *IEEE Wireless Communications*, vol. 25, no. 6, pp. 113–119, 2018.

[6] M. L. Han, B. I. Kwak, and H. K. Kim, "Event-triggered interval-based anomaly detection and attack identification methods for an in-vehicle network," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 2941–2956, 2021.

[7] C. Zhou and R. Paffenroth, "Anomaly detection with robust deep autoencoders," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 665–674, Halifax, Canada, August 2017.

[8] A. Taylor, S. Leblanc, and N. Japkowicz, "Anomaly detection in automobile control network data with long short-term memory networks," in *Proceedings of the 2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 130–139, Montreal, Canada, October 2016.

[9] H. M. Song, J. Y. Woo, and H. K. Kim, "In-vehicle network intrusion detection using deep convolutional neural network," *Vehicular Communications*, vol. 21, pp. 1–13, Article ID 100198, 2020.

[10] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," 2018, http://arxiv.org/abs/1803.01271v1.

[11] Y. Hamada, M. Inoue, H. Ueda, Y. Miyashita, and Y. Hata, "Anomaly-based intrusion detection using the density estimation of reception cycle periods for in-vehicle networks," *SAE International Journal of Transportation Cybersecurity and Privacy*, vol. 1, no. 11, pp. 39–56, 2018.

[12] H. Lee, S. H. Jeong, and H. K. Kim, "OTIDS: a novel intrusion detection system for in-vehicle network by using remote frame," in *Proceedings of the 2017 15th Annual Conference On Privacy, Security And Trust (PST)*, pp. 57–5709, Calgary, Canada, August 2017.

[13] H. Ji, Y. Wang, H. Qin, X. Wu, and G. Yu, "Investigating the effects of attack detection for in-vehicle networks based on clock drift of ecus," *IEEE Access*, vol. 6, pp. 49375–49384, 2018.

[14] K. T. Cho and K. G. Shin, "Viden: attacker identification on in-vehicle networks," 2017, https://arxiv.org/abs/1708.08414.

[15] M. Müter and N. Asaj, "Entropy-based anomaly detection for in-vehicle networks," in *Prceedings of the 2011 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1110–1115, Baden-Baden, Germany, June 2011.

[16] E. Seo, H. M. Song, and H. K. Kim, "Gids: gan based intrusion detection system for in-vehicle network," in *Proceedings of the 2018 16th Annual Conference on Privacy, Security and Trust (PST)*, pp. 1–6, Belfast, Ireland, August 2018.

[17] S. Tariq, S. Lee, H. K. Kim, and S. S. Woo, "Detecting in-vehicle CAN message attacks using heuristics and RNNs," in *Proceedings of the International Workshop on Information and Operational Technology Security Systems*, pp. 39–45, Heraklion, Greece, September 2018.

[18] M. Hanselmann, T. Strauss, K. Dormann, and H. Ulmer, "CANet: an unsupervised intrusion detection system for high dimensional CAN bus data," *IEEE Access*, vol. 8, pp. 58194–58205, 2020.

[19] C. Wang, Z. Zhao, L. Gong, L. Zhu, Z. Liu, and X. Cheng, "A distributed anomaly detection system for in-vehicle network using HTM," *IEEE Access*, vol. 6, pp. 9091–9098, 2018.

[20] J. Xiao, H. Wu, and X. Li, "Internet of things meets vehicles: sheltering in-vehicle network through lightweight machine learning," *Symmetry*, vol. 11, no. 11, p. 1388, 2019.

[21] M. J. Kang and J. W. Kang, "A novel intrusion detection method using deep neural network for In-vehicle network security," in *Proceedings of the 2016 IEEE 83rd Vehicular Technology Conference (VTC Spring)*, pp. 1–5, Nanjing, China, May 2016.

[22] M. Marchetti and D. Stabili, "Anomaly detection of CAN bus messages through analysis of ID sequences," in *Proceedings of the 2017 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1577–1583, Angeles, CA, USA, June 2017.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the 2016 IEEE Conference On Computer Vision And Pattern Recognition (CVPR)*, pp. 770–778, Vegas, NV, USA, June 2016.

[24] S. Deng, N. Zhang, Z. Wen, J. Chen, J. Z. Pan, and H. Chen, "Knowledge-driven stock trend prediction and explanation via temporal convolutional network," in *Proceedings of the WWW'19: Companion Proceedings of the 2019 World Wide Web Conference*, pp. 678–685, San Francisco, CA, USA, May 2019.

[25] C. Lea, R. Vidal, A. Reiter, and G. D. Hager, "Temporal convolutional networks: a unified approach to action segmentation," *Lecture Notes in Computer Science*, Springer, Berlin, Germany, pp. 47–54, 2016.

[26] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proceedings of the*

Proceedings of International Conference on Machine Learning, pp. 807–814, Haifa, Israel, June 2010.

[27] R. Yang, D. Qu, S. Zhu, Q. Yekui, and T. Yongwang, "Anomaly detection for log sequence based on improved temporal convolutional network," *Computer Engineering*, vol. 46, no. 8, pp. 50–57, 2020.

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1026–1034, Santiago, Chile, December 2015.