JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 40, 107-123 (2024)
DOI: 10.6688/JISE.202401_40(1).0007

1D CNN and BiSRU Based Intrusion Detection Method for Industrial Control Systems

ZENG-YU CAI[1], HONG-YU DU[2], HAO-QI WANG[2], JIAN-WEI ZHANG[3,+] AND LIANG ZHU[1]
[1]*School of Computer and Communication Engineering*
[2]*College of Mechanical and Electrical Engineering*
[3]*College of Software Engineering*
Zhengzhou University of Light Industry
Zhengzhou, 450001 P.R. China
E-mail: mailzjw@163.com

With the integration of industrial control systems (ICSs) and modern IT networks, the security of ICSs has been threatened while increasing their efficiency. Existing intrusion detection methods based on machine learning, such as Support Vector Machine (SVM), Decision Tree, *etc*., usually rely on manually designed methods for feature learning and have low accuracy for intrusion detection of high-dimensional network traffic of ICSs. Although the detection accuracy of Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) based methods is significantly improved compared to Simple Recurrent Neural Network (SimpleRNN), there is the problem of long training time consumption. To solve the above problem, this study proposed an intrusion detection method for ICSs based on 1D Convolution Neural Networks (1D CNN) and Bidirectional Simple Recurrent Unit (BiSRU), fully learning the correlation and dependency of network traffic data of ICSs in spatial and temporal dimensions. With skip connections employed, the optimized bidirectional structure of the Simple Recurrent Unit (SRU) neural network can further alleviate the problem of gradient vanishing and improve the training effect. Mississippi State University's Gas Pipeline dataset was used to train and test the model. Experiments show that the proposed method is significantly better than other existing methods in terms of accuracy and training time.

Keywords: intrusion detection, control system, deep learning, 1D convolution neural networks, bidirectional simple recurrent unit

1. INTRODUCTION

The early ICSs were in a physical environment completely isolated from the external network, and their operating system used a dedicated communication protocol, so there was no network security problem [1]. However, with the development of computer technology, the closeness of ICSs has been broken. The proposal of "Industrial 4.0" in Germany has further promoted the process of opening up the ICSs to the outside world. At the same time, a variety of attacks on ICSs followed, among which the attacks on supervisory control and data acquisition (SCADA) systems, distributed control systems and programmable logic controllers are the majority [2]. As a large industrial country, improving the safety technology level of ICSs is of great significance for national security and social stability.

We learn that intrusion detection methods based on deep learning have a good ability for feature extraction [3]. In recent years, time series algorithms are widely used in in-

Received November 1, 2022; revised December 13, 2022; accepted December 19, 2022.
Communicated by Changqiao Xu.
[+] Corresponding author: mailzjw@163.com.

trusion detection of ICSs as models that excel in processing sequential data. However, the gradient vanishing problems of SimpleRNN [4] lead to low model detection accuracy. LSTM [5] and GRU [6] use a complex structure consisting of state vectors and gating units to control the flow of information passing between neurons. Although vanishing gradient problems are alleviated to some extent, the time dependence of the state computation prevents parallel computation. This paper proposes an intrusion detection method for ICSs based on 1D CNN and BiSRU, and use this method to detect cyberattacks on the Gas Pipeline dataset [7]. The contribution of this paper is as follows:

- Firstly, to further improve the ability of model feature extraction, a 1D CNN is proposed to learn the local spatial correlation of features.
- Secondly, this paper proposes an SRU-based BiSRU for intrusion detection of ICSs. The BiSRU extracts temporal features of network traffic data in ICSs from both positive and negative directions, which can further alleviate the gradient vanishing problem and achieve efficient parallel computation.
- Finally, we compare the proposed method with existing machine learning methods and deep learning methods. Experiments show that the proposed method outperforms existing methods in terms of accuracy and training time.

The remainder of this paper is organized as follows: Section 2 describes the related works. The problems of RNNs are introduced in Section 3. Section 4 proposes a 1D CNN and BiSRU-based approach for intrusion detection of ICSs, Section 5 presents the experimental details and analysis. Finally, we summarize our work as well as point out the next research directions in Section 6.

2. RELATED WORK

In this section, we introduce related work, including intrusion detection methods for ICSs based on machine learning and deep learning.

2.1 Machine Learning

Classical machine learning methods, including SVM [8], Decision Tree [9] and Naive Bayes [10]. Anton *et al.* [11] used SVM to detect seven different classes of attacks in the Gas Pipeline of the standard industrial dataset. Although a high accuracy rate was achieved, the precision rate was low. Al-Asiri *et al.* [12] used the Gas Pipeline of the standard industrial dataset to verify the effectiveness of the Decision Tree classifier for various features in the SCADA system using IDS with a single network metric and physical metric respectively. Khan *et al.* [13] used the original features from the Gas Pipeline dataset to formulate a new set of features for attack detection using Naive Bayes in supervised learning mode. Tian *et al.* [14] proposed a method that combines machine learning optimized by swarm intelligence algorithm and deep learning. They used a stack Autoencoder to reduce the dimension of data feature, then, Combined SVM and Artificial Bee Colony algorithm to do an intrusion detection experiment. In recent years, although machine learning-based methods can achieve good results, it can only perform shallow learning and cannot accurately identify network attacks in ICSs [15]. For example, SVM instead lead to a decrease in accuracy when the number of samples increases, Naive Bayes methods do not

handle data with correlated attributes well, and Decision Tree has poor generalization capabilities.

2.2 Deep Learning

With the increasing computing power of computers, emerging deep learning methods are rapidly emerging in various fields, especially in image detection and speech recognition [16]. At the same time, it has led many scholars in the direction of industrial Internet security to apply deep learning to intrusion detection of ICSs. Yang *et al.* [17] proposed a CNN network for intrusion detection systems (IDSs). Liu *et al.* [18] proposed a hybrid method of deep learning and population intelligence optimization algorithms. They used CNN for feature extraction and anomaly recognition, then, the features extracted by the CNN model are invoked as input to the algorithm to construct a normal state process transfer model. RNNs are widely used as a temporal deep learning model for intrusion detection of ICSs. The IDSs provide an effective way for abnormal traffic detection. Yin *et al.* [19] proposed an IDSs based on the RNN-IDS algorithm. The method was validated using the NSL_KDD dataset, and the results showed that it outperformed traditional machine learning methods. LSTM is a variant of SimpleRNN, and it alleviates the problem of gradient vanishing and gradient explosion of SimpleRNN to a certain extent. Roy *et al.* [20] proposed an Internet of Things (IoT) intrusion detection method based on Bidirectional Long Short-Term Memory Recurrent Neural Network (BLSTM RNN) to improve the problem of insufficient SimpleRNN temporal storage capacity. Sokolov *et al.* [21] used GRU for experiments on intrusion detection in the Gas Pipeline dataset and investigated the applicability of the method in various aspects of intrusion detection of ICSs. In 2018, Lei *et al.* proposed an SRU model [22]. The model used a simpler structure to solve the sequence dependence problem in previous LSTM and GRU, further alleviates the problem of RNNs gradient vanishing and gradient explosion, and enables parallel computation. SRU has been successfully applied in the field of classification and conversational systems.

Based on the above work, the traditional machine learning methods typically used for intrusion detection in ICSs are less concerned with accurate feature selection. It is usually low-accuracy for large-scale network traffic in ICSs. LSTM and GRU can suppress gradient vanishing and gradient explosion to a certain extent when capturing long-distance correlation information, and their effects are better than traditional SimpleRNN. However, as a variant of SimpleRNN, it has the disadvantage of RNNs structure itself, it cannot be computed in parallel. To solve the above problems, this paper proposes an intrusion detection method for ICSs based on 1D CNN and BiSRU. Specifically, we propose a 1D CNN to learn spatial features of network traffic data from ICSs, and BiSRU can learn bidirectional structural features of network traffic data from ICSs by forward and backward inputs to achieve accurate detection with less model training time. This study performed sufficient experiments with the Gas Pipeline to validate our proposed approach.

3. EXISTING PROBLEMS OF RNNS

3.1 Gradient Vanishing Problem of SimpleRNN

SimpleRNN performs error back propagation using a chain rule during gradient des-

cent computation to obtain the minimum bias derivative of the hidden state loss function. The key expression for the gradient derivation satisfies the following equation,

$$\frac{\partial h_t}{\partial h_i} = \prod_{j=i}^{t-1} W_{hh}^T diag(\sigma'(W_{xh} x_{j+1} + W_{hh} h_j + b)). \tag{1}$$

The gradient $\frac{\partial h_t}{\partial h_i}$ from timestamp i to timestamp t contains the concatenated multiplication operation of W_{hh}. When the maximum eigenvalue of W_{hh} is less than 1, multiple concatenation operations will cause the elemental values of $\frac{\partial h_t}{\partial h_i}$ to approach zero, causing the gradient descent algorithm to fail; When the value of $\frac{\partial h_t}{\partial h_i}$ is greater than 1, multiple multiplications will cause the value of the $\frac{\partial h_t}{\partial h_i}$ element to grow explosively.

3.2 LSTM and GRU Cannot be Parallelized

To alleviate the gradient vanishing problem of SimpleRNN in backpropagation, a complex structure consisting of state vectors and gating units are introduced, to control the flow of information passing between neurons, thus ensuring that the feedback path receives timely feedback for gradient computation. LSTM and GRU are two typical examples.

a) The LSTM adds a new state vector c_t to the SimpleRNN network and introduces a gating mechanism. Through the input gate, forget gate and output gate three gating units to control the information forgotten and refreshed, The LSTM structure is shown in Fig. 1.

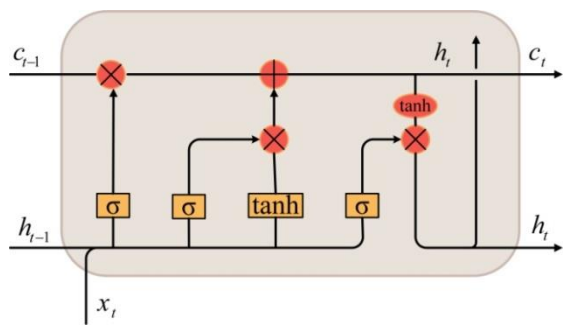

Fig. 1. LSTM structure diagram.

Where x is used as the input vector of the LSTM, c is used as the internal state vector of the LSTM, and h denotes the output vector of the LSTM.

The forgetting gate is the most important gate in LSTM. The forgetting gate acts on the LSTM state vector c to control the effect of the memory c_{t-1} of the previous timestamp on the current timestamp, thus alleviating the gradient vanishing problem when propagating backward with the time series. The control variable g_f for the forgetting gate is calculated as follows,

$$g_f = \sigma(W_f[h_{t-1}, x_t] + b_f). \tag{2}$$

W_f and b_f are the parameter tensor of the forget gate, which can be automatically optimized by the backpropagation algorithm, and σ is the activation function.

b) The GRU merges the internal state vector and the output vector based on LSTM and unifies them into a state vector h. The number of gates has been reduced to two: reset gate and update gate. The GRU structure is shown in Fig. 2.

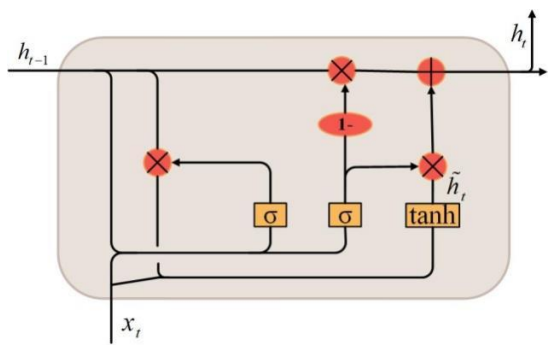

Fig. 2. GRU structure diagram.

The updating gate is used to control the last time stamped state h_{t-1} and the degree of influence of the new input $\tilde{h}_t$ on the new state vector h_t [23]. The control variable g_z for the update gate is calculated as follows,

$$g_z = \sigma(W_z[h_{t-1}, x_t] + b_z). \tag{3}$$

W_z and b_z are the parameter tensor of the update gate, it is automatically optimized by the backpropagation algorithm like LSTM.

The reliance on state vectors and gating units allows LSTM and GRU to alleviate the gradient vanishing problem to some extent. However, the relatively complex structure of LSTM and GRU leads to high computational cost. The previous moment of the network state of RNNs can be transferred to the current state, and this sequential dependence prevents parallel computation of LSTM and GRU. This leads to their long training time in performing intrusion detection of ICSs tasks and cannot meet their real-time requirements.

4. PROPOSED APPROACH

4.1 1D CNN for Series Data

In this paper, for serial data like network traffic of ICSs, we propose a 1D CNN method to extract the spatial features of the input data. Similar to the common two-dimensional convolution that extracts blocks from images and applies the same transformation to each block, 1D CNN selects sequence segments from sequence data along the time dimension and performs the same transformation on each sequence segment [24]. Our proposed 1D CNN neural network contains convolution layer and pooling layer to achieve automatic extraction and dimensionality reduction of the input data features, respectively.

Fig. 3 shows the 1D CNN feature extraction schematic. It mainly includes two phases: convolution and pooling. the convolution layer implements the automatic extraction of features from the input data. And the pooling layer performs the statistical calculation of the convolved results.

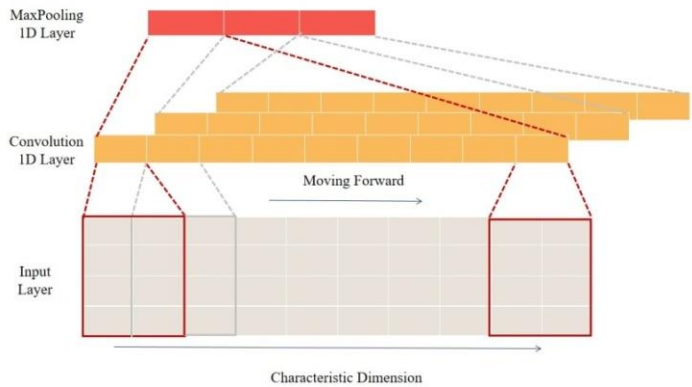

Fig. 3. 1D CNN feature extraction schematic.

The following is a brief description of the computational steps involved in a 1D CNN:

a) The input of the 1D convolution layer to $[x_1, x_2, \ldots, x_i, \ldots]$, where x_j is the feature quantity at moment j Let the output of the lth convolution layer be x^l and the output of its corresponding jth convolution kernel be x_j^l. Then the output of this input after processing by the jth convolution kernel of the convolution layer is:

$$x_j^l = s(\sum_{x_i^{l-1} \in M_j} x_i^{l-1} * W_{ij}^l + b_j^l). \tag{4}$$

s is the activation function, M_j represents the set of input features, W_j^l is the convolution kernel, $*$ is the convolution, and b_j^l is the bias term.

b) The 1D pooling layer performs statistical calculations on the feature maps generated by the convolution layer to retain the most valid information from the network traffic data of ICSs. The general form of the sampling is as follows,

$$x_j^{l'} = s(\beta_j^{l-1} D(x_j^{l-1} + b_j^l)). \tag{5}$$

D is the sampling function, β_j^l is the weight.

4.2 SRU for Accelerated Training

SRU is a variant structure of RNN. For the parallel computing problem, by analyzing the structure of models such as LSTM and GRU, an improved SRU model is proposed by analyzing and studying the structure of LSTM and other models. SRU is designed to improve the efficiency of model training in RNNs with highly parallelized implementation. Due to the efficiency of SRU, it is used to replace LSTM and GRU. The SRU structure is shown in Fig. 4.

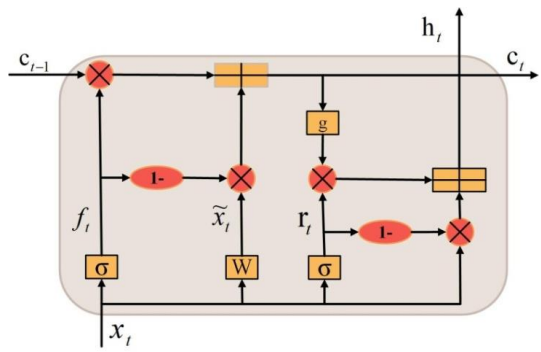

Fig. 4. SRU structure diagram.

The SRU controls the forgetting and passing of information through the forget gate and reset gate. First, the eigenvector x_t is input to the SRU cell at moment t, and the operation is performed by the forgetting gate to obtain f_t, as shown in Eq. (6),

$$f_t = \sigma(W_v x_t + b_v). \tag{6}$$

Where f_t and b_v are two parameter tensors, and σ is the activation function. The f_t obtained by the forgetting gate adaptively computes c_{t-1} to obtain the cell c_t state of the SRU at moment t. The final output state is calculated as,

$$c_t = f_t \circ c_{t-1} + (1 - f_t) \circ \tilde{x}_t. \tag{7}$$

Where $\circ$ denotes an element-wise multiplication. Reset gate r_t to adaptively combine input x_t and state c_t, reset gate r_t is set as follows,

$$r_t = \sigma(W_r x_t + b_r). \tag{8}$$

As seen from the above equations, the conversion between the gate control unit and the input no longer depends on the hidden state h_{t-1} at the previous moment, but on the intermediate state c_{t-1} at the previous moment. Thus, a large number of matrices can be computed in parallel. In addition, to solve the complex computation problem of matrix multiplication, SRU synthesizes the matrix multiplication into one, which can significantly improve the utilization of GPU, and the combined matrix is shown as follows,

$$U^T = \begin{pmatrix} W \\ W_v \\ W_r \end{pmatrix} [x_1, x_2, \cdots x_j]. \tag{9}$$

Finally, the output h_t is calculated by skip connections,

$$h_t = r_t \circ g(c_t) + (1 - r_t) \circ x_t. \tag{10}$$

The output state h_t is calculated using the skip connection trick, which directly incorporates the input x_t into the calculation. In the derivative, there is always x_t derivative of

the term with respect to x present. This method optimizes the gradient propagation without the dissipation of the gradient due to the propagation distance being too far.

4.3 Design of Detection Model

(A) Structure design

First, the original Gas Pipeline traffic is preprocessed to obtain a vector with an input dimension of 17 as the input to the 1D CNN. Our 1D CNN network consists of two convolution layers and one pooling layer, two convolution layers are stacked before the pooling layer. As shown in the lower part of Fig. 5. Among them, the convolution layer implements the automatic extraction of features from the input data, and the pooling layer performs the statistical calculation of the convolved results. The features can be dimensionally reduced while retaining the local optimal features [25]. In particular, the activation function tanh is sandwiched in the middle of the convolution layers by superimposing convolution layers. The superposition of nonlinear functions increases the nonlinear expressiveness of the activation function, which enables it to learn well the spatial feature information of complex and high-dimensional network traffic data of ICSs.

Traditional temporal models usually read the sample data from front to back only to learn the forward information of the sample sequence. In order to fully learn the temporal characteristics of network traffic data of ICSs, a bi-directional SRU structure is proposed to obtain information on forward and backward temporal characteristics of sequences in network traffic data of ICSs. As shown at the top of Fig. 5, $\vec{x}$ and $\overleftarrow{x}$ are the forward and reverse readings of the sample sequence, respectively. $\vec{c}$ and $\overleftarrow{c}$ are the memory units of the forward and reverse SRU, respectively. $\vec{h}$ and $\overleftarrow{h}$ are the output states of the forward and reverse SRUs. The prediction feature vector is obtained through the fully connected layer, and train the detection model by comparing the loss between the predicted result $y(H)$ and the actual label value y.

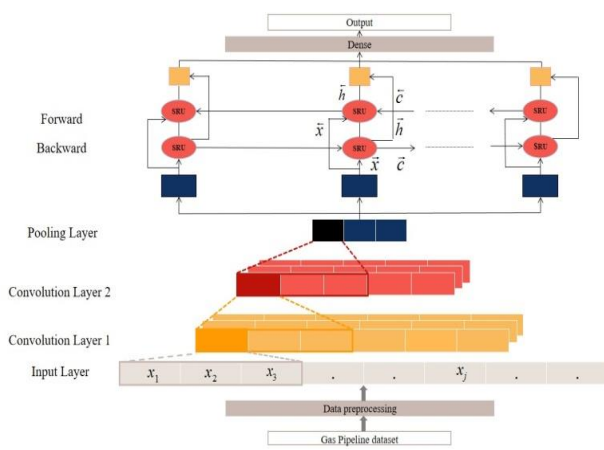

Fig. 5. Proposed model for intrusion detection of ICSs Using 1D CNN and BiSRU.

(B) Algorithm design

First, Convolution is performed on the input data $X = [x_1, x_2, \ldots, x_i, \ldots]$ or feature mapping by a convolution layer. Then, the output of the previous layer is mapped using the

function $\tanh\left(x_j^l\right) = \frac{\sinh x_j^l}{\cosh x_j^l}$. In order to extract sufficient spatial features of network traffic data of ICSs, the number of filters and the size of convolution kernels of the two convolution layers are set to {64,3}. The motivation for this is to balance the contradiction between extracted features and overfitting by pairing too many filters with smaller convolution kernels. Next, a pooling layer is deployed to pick out the optimal feature maps. we set the dropout function after the pooling layer to randomly drop a fraction of neuron nodes with a probability of 0.2 during training. The mechanism of randomly discarding some neurons, which is equivalent to training a different structure of the neural network in each iteration, can effectively suppress the occurrence of overfitting. The vector operation of the dropout function is represented as,

$$dropout(x_j^{l'}) = x_j^{l'} \circ m, \tag{11}$$

where x denotes an input vector, m denotes a random mask vector.

Using the output vector of the 1D CNN [26] as the input to the BiSRU neural network to predict the subsequent traffic features, denoted as $X' = [x'_1, x'_2, \ldots, x'_j, \ldots]$. The first layer of BiSRU is set to 64 units and the next BiSRU layer to 128 units. The reason for this choice is to mimic the use of coarse-grained to fine-grained learning and thus more fully understand the relevance of long-range time-dependent features in the sample. BiSRU consists of two models, positive and negative SRU. It summarizes the forward information $\vec{x}$ and the backward information $\overleftarrow{x}$ to enhance feature extraction abilities. positive SRU learns the forward information of the data and outputs $\vec{h}$, negative SRU learns the backward information of the data and outputs $\overleftarrow{h}$.

$$\vec{h} = SRU(\vec{x})$$

$$\overleftarrow{h}. = SRU(\overleftarrow{x}) \tag{12}$$

$H = (\vec{h}, \overleftarrow{h})$ is the vector on forward and backward temporal characteristics of the data output by BiSRU. H as the fully connected layer of the input, the fully connected output layer maps the learned features to the output classes. The output of this stage is controlled by the softmax function. The softmax function can be described as,

$$y(H) = soft\max(W_d H + b_d). \tag{13}$$

Where W_d is the weight matrix and b_d is the bias vector. In this paper, by using the back propagation algorithm, through the end-to-end way, the cross-entropy loss function is used to measure the gap between the target and the predicted value. Adam optimization method was used to optimize the experimental parameters, so as to select the optimal model parameters. The key steps of the proposed are illustrated in Algorithm 1.

Algorithm 1: The Intrusion Detection Algorithm of 1D CNN and BiSRU

Input of 1D CNN: Gas Pipeline dataset $X = [x_1, x_2, \ldots, x_i, \ldots]$

1: **For** each training epoch $n = 1, 2, \ldots, N$ **do**

2: Compute the output of convolution kernel $x_j^l = s(\sum_{x_i^{n-1} \in_{M_j}} x_i^{l-1} * W_{ij}^l + b_j^l)$

3: Feature mapping

4: Compute the activation function $\tanh(x_j^l) = \frac{\sinh x_j^l}{\cosh x_j^l}$

5: Compute the output of Pooling layer $x_j^{l'} = s(\beta_j^{l-1} D(x_j^{l-1} + b_j^l))$.
6: Dropout neurons with a probability of 0.2
Input of BiSRU: The output vector of the 1D CNN $X' = [x'_1, x'_2, \ldots, x'_j, \ldots]$.
7: Compute forget gate $f_t = \sigma(W_v x_t + b_v)$
8: Compute the current cell state based on the previous cell state c_{t-1} and forget gate f_t.

$$c_t = f_t \circ c_{t-1} + (1 - f_t) \circ \tilde{x}_t$$

9: Compute reset gate $r_t = (W_r x_t + b_r)$
10: The output of the SRU $h_t = r_t \circ g(c_t) + (1 - r_t) \circ x_t$
11: The output of the BiSRU $H = (\overrightarrow{h}, \overleftarrow{h})$
12: Execute the softmax classifier $y(H) = soft\max(W_d H + b_d)$
13: **End**
Output: Classification results $y(H)$

5. EXPERIMENTS

5.1 Dataset

All experiments used the standard industrial dataset for Gas Pipeline presented by Mississippi State University in 2014. It has been widely used in recent years for simulation experiments of intrusion detection of ICSs [27, 28], which is collected from a set of Modbus TCP-based natural gas pipeline systems, and its structure is similar to that of SCADA systems in real production environments. The Gas Pipeline dataset contains normal data and seven types of attack data. The details can be seen in Table 1.

Table 1. Description of datasets.

Type of Attacks	Abbreviation	Number	the Proportion
Normal	Normal(0)	61156	63.04%
Naïve Malicious Response Injection	NMRI(1)	2763	2.85%
Complex Malicious Response Injection	CMRI(2)	15466	15.94%
Malicious State Command Injection	MSCI(3)	782	0.81%
Malicious Parameter Command Injection	MPCI(4)	7637	7.87%
Malicious Function Code Injection	MFCI(5)	573	0.59%
Denial of Service	DOS(6)	1837	1.89%
Reconnaissance	Recon(7)	6805	7.01%

5.2 Data Preprocessing

(A) Low variance filter

Gas Pipeline dataset is complex and variable, with many eigenvalues, but not every eigenvalue is well differentiated, *i.e.* it has a very low variance. Such eigenvalues have no value for analysis, and it is chosen to remove them directly. For example, if a column has a feature that takes a value of 1 in 95% of the instances in all input samples, then it can be assumed that the feature is not very useful. If 100% of them are 1, then this feature has no

meaning. In this paper, we chose to remove the 9 feature columns with the smallest variance, and finally obtained a dataset with 17-dimensional effective feature values.

(B) Normalization

The Gas Pipeline dataset has high-dimensional features and the maximum and minimum value intervals of these features are large, so the data feature values are set in a small specific interval. Mapping features to the range [0,1] using min-max normalization. The normalized formula is shown below,

$$x'_P = \frac{x_q - \min(x_p)}{\max(x_p) - \min(x_p)}. \tag{14}$$

5.3 Experimental Hyperparameter Setting

The experimental environment of this paper is as follows: Intel Core i7-9700H CPU, NVIDIA G-eForce GTX745 GPU, python3.6, tensorflow1.5.1 and 32 GB RAM. The network parameters of the method proposed in this study are set as shown in Table 2.

Table 2. Experimental parameters.

Parameter Name	Value
1D CNN	
Number of Filters	64
Number of Convolution Kernels	3
Activation Function	tanh
Dropout Rate	0.2
BiSRU	
Depth	2
Optimizer	Adam
Activation	softmax
Batch Size	132
First Hidden Unit Size	64
Second Hidden Unit Size	128
Dropout Rate	0.2

5.4 Benchmarking Metrics

The Accuracy, Precision, Recall, and F1 are used as key performance indicators to evaluate the proposed method. The calculations of the four metrics are as follows,

$$Accuracy = \frac{TN + TP}{TP + FP + TN + TP} \tag{15}$$

$$Precison = \frac{TP}{TP + FP} \tag{16}$$

$$Recall = \frac{TP}{FN + TP} \tag{17}$$

$$F1 = \frac{2TP}{2TP + FP + FN} \tag{18}$$

where *TP* represents the anomaly traffic instances correctly classified. *TN* denotes the normal traffic instances correctly classified. *FP* is the normal traffic instances wrongly classified, and *FN* indicates the anomaly traffic instances wrongly classified.

5.5 Performance Comparison

We compare the experimental results of three machine learning methods and a deep learning optimized by swarm intelligence algorithm in the original paper with our approach. Three deep learning methods based on RNNs are selected for experiments, and the experimental parameters are set in the original paper. As shown in Table 3, Table 3 shows our proposed method has the highest Accuracy, Precision, Recall and highest F1 on Gas Pipeline dataset with the shortest training time. This shows that our proposed method of extracting temporal and spatial features successfully improves the accuracy of intrusion detection in ICSs, and the simplified SRU significantly saves the training time on the basis of ensuring the accuracy.

Table 3. Benchmarking metrics for the different algorithms.

Ref.	Technique	Accuracy(%)	Precision(%)	Recall(%)	F1(%)	Trainging Time(s)
[8]	SVM	92.5	78.2	93.6	85.2	–
[9]	Decision Tree	84.9	86.1	84.9	87	–
[10]	Naïve Bayes	71.94	70.6	71.9	71.24	–
[15]	CNN + Process State Transition	94.1	70.7	90.2	82.1	–
[16]	RNN-IDS	94.5	77.99	79.84	78.9	–
[17]	BLSTM RNN	97.56	90.24	89.97	90.1	178
[18]	GRU	98.21	93.83	92.87	93.34	117
Ours	1D CNN + BiSRU	98.86	95.76	95.81	95.78	49

As shown in Fig. 6, the accuracy of our method for MSCI, MFCI, and DOS data in the Gas Pipeline dataset is significantly higher than that of other algorithms, which indicates that our method has stronger learning ability for a small number of classes of samples.

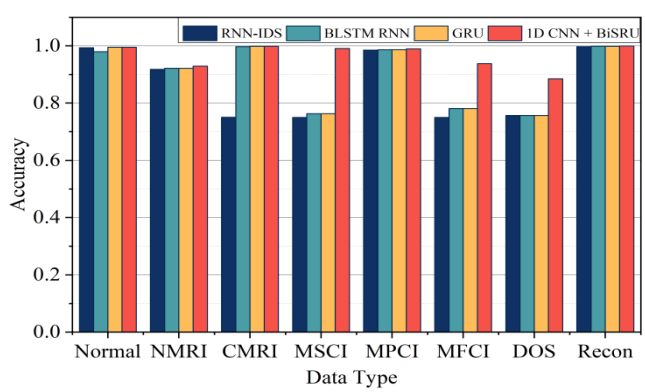

Fig. 6. Detection accuracy for normal data and various types of attack data.

The training accuracy and loss of our method are compared with three deep learning methods based on RNNs. Fig. 7 shows that our method is not only the first to converge, but also achieves the highest accuracy. This means that our intrusion detection method has the advantages of stability, high efficiency and high precision. Between 0 and 30 epochs, the accuracy of GRU and BLSTM RNN are basically the same, but after 30 epochs, the accuracy of GRU is slightly higher than that of BLSTM RNN.

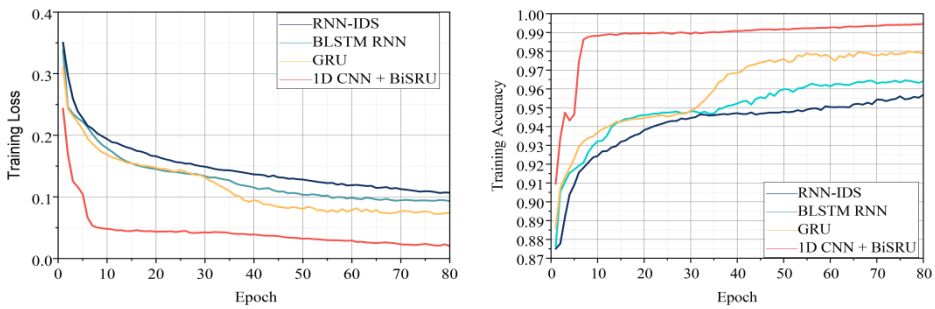

Fig. 7. Compare the loss convergence and accuracy of the proposed model with other three deep learning methods based on RNNs.

5.6 Influence of Residual Structure on Model Accuracy

This research compares models with the same parameters but no residual structure to verify the efficiency of the detection model in this paper.

The ablation study results are shown in Table 4. Comparing metrics such accuracy, precision, recall and f1 score reveals that our proposed detection model has the best results. Without the 1D CNN part, the accuracy of the model is 0.9786, the precision is 0.9158, the recall is 0.9505, and the F1 is 0.9131. When the BiSRU part is deleted, the accuracy of the model is 0.9699, the precision is 0.8787, the recall is 0.8771, and the F1 is 0.8779. The accuracy of the proposed 1D CNN+BiSRU model is 0.9896, the precision is 0.9576, the recall is 0.9515, and the F1 is 0.9538. This shows that the spatial feature extraction method based on 1D CNN and the time feature extraction method based on BiSRU have important contributions to the intrusion detection of industrial control systems.

Table 4. The ablation study results on testing dataset.

Model	Accuracy	Precision	Recall	F1
Model w/o 1D CNN	0.9786	0.9158	0.9105	0.9131
Model w/o BiSRU	0.9699	0.8787	0.8771	0.8779
1D CNN + BiSRU	0.9896	0.9576	0.9515	0.9538

6. CONCLUSIONS

This paper proposes an intrusion detection method for ICSs based on 1D CNN and BiSRU. Our main contribution is to introduce a deep feature extraction method that combines spatial and temporal dimensions. First, a 1D CNN for spatial feature extraction of high-dimensional network traffic is proposed, and then propose a parallel-computing Bi-

SRU anomaly traffic detection algorithm that uses a bidirectional structure to better extract the contextual features of network traffic. The experimental results show that our method has higher accuracy and computational efficiency compared with the latest methods. In the future, we will jointly use a population intelligence optimization algorithm to train 1D CNN and BiSRU, in order to improve the ability of model parameter seeking.

ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China under Grant No. 62072416, the Key Research and Development Special Project of Henan Province under Grant No. 221111210500, the Technologies R&D Program of Henan Province under Grant No. 232102211053, Nos. 222102210170 and 222102210322.

REFERENCES

1. H. C. Altunay, Z. Albayrak, A. N. Ozalp, and M. Cakmak, *et al*., "Analysis of anomaly detection approaches performed through deep learning methods in SCADA system," in *Proceedings of the 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications*, 2021, pp. 1-6.
2. M. Noorizadeh, M. Shakerpour, N. Meskin, D. Unal, and K. Khorasani, "A cyber-security methodology for a cyber-physical industrial control system testbed," *IEEE Access*, Vol. 9, 2021, pp. 16239-16253.
3. A. R. Khan, M. Kashif, R. H. Jhaveri, R. Raut, T. Saba, and S. A. Bahaj, "Deep learning for intrusion detection and security of internet of things (IoT): current analysis, challenges, and possible solution," *Security and Communication Networks*, Vol. 2022, 2022, pp. 1-13.
4. N. Madi, and H. Al-Khalifa, "Error detection for Arabic text using neural sequence labeling," *Applied Sciences*, Vol. 10, 2020, pp. 5279.
5. X. Shu, L. Zhang, Y, Sun, and J. Tang, "Host-parasite: graph LSTM-in-LSTM for group activity recognition," *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 32, 2021, pp. 663-674.
6. Y. Liu, A. Pei, F. Wang, Y. Yang, and X. Zhang, "An attention-based category-aware GRU model for the next POI recommendation," *International Journal of Intelligent Systems*, Vol. 36, 2021, pp. 3174-3189.
7. T. Morris and W. Gao, "Industrial control system traffic data sets for intrusion detection research," *Critical Infrastructure Protection VIII*, Vol. 441, 2014, pp. 65-78.
8. R. R. Reddy, Y. Ramadevi, and K. V. N. Sunitha, "Effective discriminant function for intrusion detection using SVM," in *Proceedings of International Conference on Advances in Computing, Communications and Informatics*, 2016, pp. 1148-1153.
9. D. Moon, H. Im, I. Kim, and J. H. Park, "DTB-IDS: an intrusion detection system based on decision tree using behavior analysis for preventing APT attacks," *Journal of Supercomputing*, Vol. 73, 2017, pp. 2881-2895.
10. M. O. Mughal and S. Kim, "Signal classification and jamming detection in wide-band radios using Nave Bayes classifier," *IEEE Communications Letters*, Vol. 22, 2018, pp. 1398-1401.

11. S. D. D. Anton, S. Sinha, and H. D. Schotten, "Anomaly-based intrusion detection in industrial data with SVM and Random Forests," in *Proceedings of the 27th International Conference on Software, Telecommunications and Computer Networks*, 2019, pp. 465-470.
12. M. Al-Asiri and E. El-Alfy, "On using physical based intrusion detection in SCADA systems," *Procedia Computer Science*, Vol. 170, 2020, pp. 34-42.
13. A. A. Z. Khan and G. Serpen, "Misuse intrusion detection using machine learning for Gas Pipeline SCADA networks," in *Proceedings of International Conference on Security and Management*, 2019, pp. 84-90.
14. Q. Tian, J. Li, and H. Liu, "A method for guaranteeing wireless communication based on a combination of deep and shallow learning," *IEEE Access*, Vol. 7, 2019, pp. 38688-38695.
15. P. Ding, J. Li, M. Wen, L. Wang, and H. Li, "Efficient BiSRU combined with feature dimensionality reduction for abnormal traffic detection," *IEEE Access*, Vol. 8, 2020, pp. 164414-164427.
16. X. Liao, K. Li, X. Zhu, and K. Liu, "Robust detection of image operator chain with two-stream convolutional neural network," *IEEE Journal of Selected Topics in Signal Processing*, Vol. 14, 2020, pp. 955-968.
17. H. Yang, L. Cheng, and M. C. Chuah, "Deep-learning-based network intrusion detection for SCADA Systems," in *Proceedings IEEE Conference on Communications and Network Security*, 2019, pp. 1-7.
18. J. Liu, L. Yin, Y. Hu, S. Lv, and L. Sun, "A novel intrusion detection algorithm for industrial control systems based on CNN and process state transition," in *Proceedings of the 37th International Performance Computing and Communications Conference*, 2018, pp. 1-8.
19. C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural network," *IEEE Access*, Vol. 5, 2017, pp. 21954-21961.
20. B. Roy and H. Cheung, "A deep learning approach for intrusion detection in internet of things using bi-directional long short-term memory recurrent neural network," in *Proceedings of the 28th International Telecommunication Networks and Applications Conference*, 2018, pp. 57-62.
21. A. N. Sokolov, S. K. Alabugin, and I. A. Pyatnitsky, "Traffic modeling by recurrent neural networks for intrusion detection in industrial control systems," in *Proceedings of International Conference on Industrial Engineering, Applications and Manufacturing*, 2019, pp. 1-5.
22. T. Lei, Y. Zhang, S. I. Wang, H. Dai, and Y. Artzi, "Simple recurrent units for highly parallelizable recurrence," in *Proceedings of Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 4470-4481.
23. H. Ma, J. Cao, B. Mi, D. Huang, Y. Liu, S. Li, C. Chen, and C. Chen, "A GRU-based lightweight system for CAN intrusion detection in real time," *Security and Communication Networks*, Vol. 2022, 2022, pp. 1-11.
24. D. Nedeljkovic and Z. Jakovljevic, "CNN based method for the development of cyber-attacks detection algorithms in industrial control systems," *Computers & Security*, Vol. 114, 2022, p. 102585.

25. J. Naranjo-Torres, M. Mora, R. Hernández-García, R. J. Barrientos, C. Fredes, and A. Valenzuela, "A review of convolution neural network applied to fruit image processing," *Applied Sciences*, Vol. 10, 2020, p. 3443.
26. D. Peng, Z. Liu, H. Wang, Y. Qin, and L. Jia, "A novel deeper one-dimensional CNN with residual learning for fault diagnosis of wheelset bearings in high-speed trains," *IEEE Access*, Vol. 7, 2019, pp. 10278-10293.
27. I. A. Khan, D. Pi, Z. U. Khan, Y. Hussain, and A. Nawaz, "HML-IDS: a hybrid-multilevel anomaly prediction approach for intrusion detection in SCADA systems," *IEEE Access*, Vol. 7, 2019, pp. 89507-89521.
28. L. Haghnegahdar and Y. Wang, "A whale optimization algorithm-trained artificial neural network for smart grid cyber intrusion detection," *Neural Computing and Applications*, Vol. 32, 2020, pp. 9427-9441.

Zeng-Yu Cai (蔡增玉) received his master degree in Computer Application Technology from Northeast Normal University, Changchun, China, in 2006. He is currently an Associate Professor at Zhengzhou University of Light Industry, Henan, China. His research interests include artificial intelligence, future network and information security.

Hong-Yu Du (杜鸿钰) received her bachelor's degree from Zhengzhou University of Light Industry, Zhengzhou, China, in 2019. She is currently pursuing her master's degree at Zhengzhou University of Light Industry, Henan, China. Her research interests include industrial internet security, deep learning and artificial intelligence.

Hao-Qi Wang (王昊琪) received his Ph.D. degree from Beijing Institute of Technology, Beijing, China, in 2018. He is currently a Lecture at Zhengzhou University of light industry, Henan, China. His research interests include model based systems engineering and digital twin.

Jian-Wei Zhang (张建伟) received his Ph.D. degree in Computer Application Technology from PLA Information Engineering University, Zhengzhou, China, in 2010. He is currently a Professor at Zhengzhou University of Light Industry, Henan, China. His research interests include machine learning, broadband information network and network security.

Liang Zhu (朱亮) received his Ph.D. degree from Beijing University of Posts and Telecommunication, Beijing, China, in 2017. He is currently a Lecturer at Zhengzhou University of Light Industry, Henan, China. His research interests include mobile social networks and privacy preserving.