electronics

MDPI

Article

Industrial Internet Intrusion Detection Based on Res-CNN-SRU

Zengyu Cai [1], Yajie Si [1], Jianwei Zhang [2,*], Liang Zhu [1], Pengrong Li [1] and Yuan Feng [1]

[1] School of Computer and Communication Engineering, Zhengzhou University of Light Industry, Zhengzhou 450001, China; czy@zzuli.edu.cn (Z.C.); 332107020571@email.zzuli.edu.cn (Y.S.); lzhu@zzuli.edu.cn (L.Z.); 332207030629@email.zzuli.edu.cn (P.L.); fy@zzuli.edu.cn (Y.F.)

[2] School of Software Engineering, Zhengzhou University of Light Industry, Zhengzhou 450001, China

* Correspondence: ing@zzuli.edu.cn

Abstract: Nowadays, the industrial Internet is developing rapidly, but at the same time it faces serious information security risks. At present, industrial Internet data generally have the problems of complex attack sample types, large numbers, and high feature dimensions. When training a model, the complexity and quantity of attack samples will result in a long detection time for the intrusion detection algorithm, which will fall short of the system's real-time performance. Due to the high feature dimension of the data, shallow feature extraction will be unable to extract the data's more significant features, which will render the model's overall detection capacity insufficient. Aiming at the above problems, an industrial Internet intrusion detection method based on Res-CNN-SRU is proposed. This method not only considers the temporality of network traffic data but can also effectively capture the local features in the data. The dataset used in the experiment is the gas pipeline industry dataset proposed by Mississippi State University in 2014. Experiments show that the algorithm can effectively improve the recognition rate of the system and reduce the false-alarm rate. At the same time, the training time required for this method is also greatly shortened, and it can perform efficient intrusion detection on the industrial Internet.

Keywords: intrusion detection; deep learning; industry Internet; residual connection

check for updates

Citation: Cai, Z.; Si, Y.; Zhang, J.; Zhu, L.; Li, P.; Feng, Y. Industrial Internet Intrusion Detection Based on Res-CNN-SRU. *Electronics* **2023**, *12*, 3267. https://doi.org/10.3390/electronics12153267

Academic Editor: Juan-Carlos Cano

Received: 19 June 2023
Revised: 13 July 2023
Accepted: 25 July 2023
Published: 29 July 2023

1. Introduction

Today's digital and Internet technology is profoundly changing our way of production and life. The traditional industrial control system mostly runs on the isolated Intranet, and almost does not need to consider the threat of the external network to hardware, protocols, deployment, etc. However, with the rapid development of communication, Internet of Things and other technologies, it has brought great changes to the application of industrial automation control systems and the business and technical architecture in the industrial environment and formed a new industrial platform, that is, the industrial Internet platform [1–3]. Driven by industrial Internet technology, in order to improve production efficiency, more enterprises have begun to realize the digitization, networking, and intelligence of production processes. However, with the continuous improvement in industrial equipment interconnection and intelligent technology, industrial Internet systems are also facing various attacks from the network, including computer viruses, malware, distributed denial-of-service attacks, and so on. These attacks can cause systems to crash, data to be compromised, or industrial equipment to be remotely controlled. At the same time, the industrial Internet involves a large amount of data transmission and storage, including production data, sensitive information and trade secrets. Hackers may attempt to steal these data for malicious purposes, such as industrial espionage, extortion, or theft of intellectual property. In February 2021, a water treatment plant system was attacked in Florida, the United States. The attacker remotely logged into the system by stealing credentials, obtained SCADA access, and then started an HMI program to send attack signals, destroy the liquid level control and dosage setting process, and increase the concentration of sodium hydroxide to 111 times the normal value, directly threatening the safety of citizens [4]. Therefore, industrial Internet intrusion detection has become more and more important.

Industrial Internet intrusion attacks may not only cause losses to the production and economy of enterprises but also have an irreversible impact on society. Therefore, intrusion detection for the industrial Internet is a major challenge to be solved urgently. Industrial Internet intrusion detection refers to discovering intrusion attacks by monitoring network traffic, identifying abnormal behaviors and threats, and taking corresponding measures to prevent and respond to attacks.

Nowadays, the combination of deep learning methods and network intrusion detection has become more and more popular. The convolutional neural network is a deep learning model that has been widely used in the field of image and speech processing. It can realize intrusion detection by analyzing and modeling network traffic data. In addition, due to the successful application of recurrent neural networks in the field of sequence data processing, more and more researchers have begun to apply it to network intrusion detection in recent years. Based on RNN, SRU (simplified recurrent unit) is a new recurrent neural network structure with fast operation and better representation ability.

The main contributions of this paper are as follows.

- A depth feature extraction method for high dimensional network traffic is proposed, which can extract local features without losing time features, and add residual connections, which not only alleviates the problem of gradient disappearance but also improves the convergence speed of the network.
- Combination with a parallel algorithm of SRU abnormal traffic detection. Compared with the traditional LSTM model, the SRU model has the advantages of high computational efficiency, fast training, strong sequence modeling ability, low memory utilization rate, and is able to train the accuracy of the model faster.
- Experiments show that the proposed method has high detection accuracy and low time consumption, and can effectively detect complex malicious response injection attacks, denial-of-service attacks, reconnaissance attacks and other attack types in the industrial Internet environment.

2. Related Work

In recent years, China and other nations have attached great importance to industrial Internet security and carried out a great deal of intrusion detection theory and technology research, among which the most representative is the intrusion detection method based on machine learning, especially the deep learning method, which can reduce false-positive rates and improve detection rates and adaptability.

2.1. Industrial Internet Intrusion Detection Based on Machine Learning

Machine learning has been widely used in computer vision, natural language processing, biometric identification, search engines, data mining and other fields. In the field of intrusion detection, machine learning also plays an important role, including support vector machine [5,6], K-means clustering algorithm [7] and Bayesian network model [8]. R. Zhang et al. [9] proposed a network intrusion detection scheme based on an improved SVM algorithm. This method simplifies the intrusion detection system through sample classification and selects the optimal parameters as the basis of intrusion detection through iterative processing. Simon D. Duque Anton et al. [10] used an anomaly detection algorithm based on machine learning and time series to analyze network data containing industrial operations to detect attacks that introduce the data. To detect attacks, two machine learning-based algorithms, SVM and random forest, are used. Both perform well and solve the problem of feature extraction and selection and processing of missing data.

Through sample categorization and iterative processing, this technique chooses the best parameters to serve as the intrusion detection system's foundation. However, as a nonparametric model, SVM is mainly used for the classification and verification of small datasets. With the increasing interconnection of modern heavy industry and manufacturing, the scale of integration is gradually expanding, and the traffic in the industrial Internet is also developing to a larger order of magnitude. In the face of the industrial Internet with huge data, support vector machines face problems such as high time overhead, reduced detection efficiency, and difficulty in obtaining hyperplanes. Ahsan Al Zaki Khan et al. [11] employed machine learning algorithms using WEKA to develop a misuse intrusion detection system designed to identify attacks on a SCADA system network of a gas pipeline infrastructure. They used naïve Bayes, rule-based and tree-based classifiers in supervised learning mode for classifying the attacks. Majed Al-Asiri et al. [12] presented a case study simulating a natural gas pipeline dataset to compare the effectiveness of decision tree classifiers for various types of features in SCADA systems. Oliver Eigner et al. [13] improved the K nearest neighbors algorithm and applied it to the industrial Internet, successfully detecting an attack. S. Jaiswal et al. [14] improved the KNN model by using the ant colony optimization algorithm, and conducted experiments on the KDD99 dataset. The above experiments of applying KNN to intrusion detection have indeed achieved certain results, but the KNN model is applied to industrial control systems, especially when the industry is large, because KNN has to calculate the distance from all data for each test sample and has the problem of high time overhead and poor performance.

In summary, the machine learning algorithm has good detection when dealing with fewer feature dimensions. However, due to the massive and high-dimensional characteristics of industrial Internet network data, traditional machine learning has been unable to meet this demand. Therefore, it is imperative to carry out deep learning research.

2.2. Industrial Internet Intrusion Detection Based on Deep Learning

Since traditional machine learning methods can no longer meet the needs of increasingly complex networks, many researchers use deep learning to replace traditional machine learning methods. Deep learning has been well applied in many fields, such as image, video recognition, natural language processing and robot technology. Nowadays, deep learning application scenarios are more and more extensive, and it has been proved that deep learning has certain advantages over traditional machine learning methods in industrial Internet intrusion detection. The development of deep learning has brought new possibilities to intrusion detection. Xia W et al. [15] optimized the BP neural network and used the Adaboost algorithm to obtain the optimal weight and threshold by continuously adjusting the weight of training samples, which effectively solves the problem of intrusion detection. Aiming at the security problem of the Internet of Things, Yang Aimin et al. [16] proposed an LM-BP neural network model by improving the BP network, and applied it to the intrusion detection system. However, the BP neural network model has a single structure, and a large number of parameters will be generated when fitting complex functions, which will easily lead to overfitting and performance degradation, so the detection results are not ideal. Y Li, Y Xu et al. [17] proposed a multi-CNN fusion-based intrusion detection system. The processed data showed a better training result for deep learning.

Chuanlong Yin et al. [18] proposed a deep learning approach for intrusion detection using recurrent neural networks. The RNN-IDS model improves the accuracy of intrusion detection and provides a new research method for intrusion detection. Bipraneel Roy et al. [19] presented a novel deep learning technique for detecting attacks within the IoT network using a bidirectional long short-term memory recurrent neural network. The experimental outcome showed that BLSTM RNN was highly efficient for building a high-accuracy intrusion detection model and offered a novel research methodology. Song Zhiwen [20] used a genetic algorithm to obtain the optimal selection for the training set and test set, and combined convolutional neural network and gated loop unit to propose a CNN-GRU intrusion detection method based on a genetic algorithm.

Zhou et al. [21] proposed a variational long short-term memory (VLSTM) learning model for intelligent anomaly detection based on reconstructed feature representation to solve the industrial Internet's imbalance in data distribution in high-dimensional anomaly detection for industrial applications. RH Hwang, MC Peng et al. [22] proposed an intrusion detection model based on word embedding and long short-term memory network, which can classify malicious traffic. The experimental results show that the method has a significant classification effect in normal and malicious binary classification detection. Jie Ling, Zhishen Zhu et al. [23] proposed an intrusion detection method based on a bidirectional simple recurrent unit. With skip connections employed, the optimized bidirectional structure in the SRU neural network is able to alleviate the vanishing gradient problem and improve training effectiveness. As mentioned above, these detection algorithms have achieved some success, but the RNN model they use has many parameters and the performance is not good enough. It is easy to cause gradient disappearance or gradient explosion, and compared with the convolution model, it has no advantage in the final recognition rate.

Therefore, compared with traditional machine learning methods, deep learning performs well in processing large-scale and high-dimensional data, and deep learning can automatically learn and extract features, but there are still problems such as limited feature learning ability and gradient disappearance, so further optimization is needed.

In summary, traditional machine learning methods cannot extract features very accurately in the field of industrial Internet intrusion detection. Therefore, for the industrial Internet with large network traffic, its detection accuracy is usually low. This paper proposes an industrial Internet intrusion detection model based on 1D CNN with residual structure and a simple recurrent unit algorithm to solve this problem using 1D CNN to extract features and improve the accuracy of data classification. The residual structure can make the model deeper and more powerful, which helps to improve the accuracy of intrusion detection. It also can improve the generalization ability of the model, avoid the problem of gradient disappearance through skip connection, and reduce overfitting. Compared with the traditional LSTM model, the SRU model has faster training and lower memory consumption, and can train a model with high accuracy faster.

Combined with the Mississippi natural gas pipeline dataset for experiments, it can be found that the proposed convolutional neural network and simple recurrent unit model combined with residual structure are more efficient than other algorithms in intrusion detection. While improving the detection accuracy, it also takes into account the stability of the model. After many experiments, it is proved that the improved model has better detection.

3. Proposed Method

This paper proposes an industrial Internet intrusion detection model based on Res-CNN-SRU. A deep neural network hybrid model is constructed by fusing 1D CNN and a simple recurrent unit network. One-dimensional CNN combines the direct connection structure of the residual network. The direct connection of the residual structure can avoid the disappearance of the depth gradient. The SRU will further screen the data after convolution extraction and mine the timing information. Specifically, the intrusion detection process is regarded as a classification problem, and the traffic characteristics in the network are classified to determine whether there is an attack in the network.

3.1. Intrusion Detection Model Based on Res-CNN-SRU

The method consists of three parts: one-dimensional convolutional neural network, residual connection, and simple recurrent unit.

3.1.1. One-Dimensional Convolutional Neural Network

A convolutional neural network (CNN) is a kind of feedforward neural network with convolution calculation and deep structure [24]. Among them, 1D CNN is often used in the field of natural language processing [25], while 2D CNN and 3D CNN are often used in image recognition [26], Mandarin speech recognition [27], face recognition [28] and other fields. The traditional neural network uses matrix multiplication to establish the connection by using the input data and the neural network parameter matrix. Each input unit interacts with the output unit through the parameters in the parameter matrix. However, CNN reduces the number of network model parameters by local connection and weight sharing, which not only reduces the computational complexity of the model but also makes the network easy to optimize [29].

The 1D CNN is a convolutional neural network that uses one-dimensional convolution to extract features from one-dimensional time series, which can ensure that local features are extracted without losing time series features [30].

Convolutional neural networks usually comprise three layers:

1. Convolution layer. In order to achieve the effect of feature extraction, the input features are scanned by the convolution kernel, subjected to matrix operations in the "receptive field," and superimposed with deviations [31].
2. Pooling layer. The pooling layer has a variety of different forms of nonlinear pooling functions. It divides the input image into several rectangular regions and outputs the maximum value for each subregion. The pooling layer will continuously reduce the space size of the data, so the number of parameters and the amount of calculation will also decrease, which also controls the overfitting to a certain extent.
3. Fully connected layer. The extracted features are nonlinearly combined and output to other fully connected layers. The convolution layer and pooling layer can achieve the purpose of automatically extracting local features of data, while the fully connected layer can achieve feature learning.

The one-dimensional convolution formula is shown in Equation (1).

$$\mathrm{Z}^{\mathrm{m}+1} = [\mathrm{Z}^{\mathrm{m}} * \mathrm{w}^{\mathrm{m}+1}] + \mathrm{d} = \sum\nolimits_{\mathrm{z}=1}^{\mathrm{m}} [\mathrm{Z}_{\mathrm{k}}^{\mathrm{m}}(\mathrm{s}_0 + \mathrm{x})\mathrm{w}_{\mathrm{k}}^{\mathrm{m}+1}(\mathrm{x})] + \mathrm{d} \tag{1}$$

The maximum pooling formula is shown in Equation (2).

$$\mathrm{A}_{\mathrm{i}}^{\mathrm{m}+1}(\mathrm{j}) = \max_{(\mathrm{j}-1)\mathrm{W}+1 \leq \mathrm{t} \leq \mathrm{jW}} \{\mathrm{F}_{\mathrm{i}}^{\mathrm{m}}(\mathrm{t})\} \tag{2}$$

In the formula, Z^m is the value of the convolution input of the m layer, Z^{m+1} is the value of the convolution output of the $m + 1$ layer, d is the value of the deviation, w_k^{m+1} is the value of the weight of the corresponding node of the $m + 1$ layer, f is the value of the convolution kernel size of the convolution layer, s_0 is the value of convolution step size of the convolution layer, $F_i^l(t)$ is the value of the t neuron in the i feature of the m layer, W is the value of the pooled area, A_i^{m+1} is the value of the output of the $m + 1$ layer neuron.

3.1.2. Residual Connection

It is found that with the deepening of the number of the network layer, not only will the gradient disappearance problem occur but also the network degradation will lead to the occurrence of overfitting. Residual connection can effectively solve the above problems. The idea of residual connection is derived from the gating idea of LSTM, which expresses the output as a linear superposition of a nonlinear transformation of input and input [32,33], as shown in Figure 1.

The traditional neural network layer can be expressed as $y = F(x)$, where $F()$ is the mapping function of the network layer. Suppose there is a residual block whose input is x and output is $H(x)$. In the residual network, we hope to learn the residual $F(x)$ through the residual connection so that the output can be expressed as $y = H(x) + F(x)$. Through the residual connection, we can optimize the network by learning the residual $F(x)$. If the network can successfully learn the identity mapping, that is, $f(x) = H(x)$, then the residual $F(x)$ is close to zero and the output of the network is close to the input. In this way, the network can gradually optimize the performance of the model by adjusting the residual part.

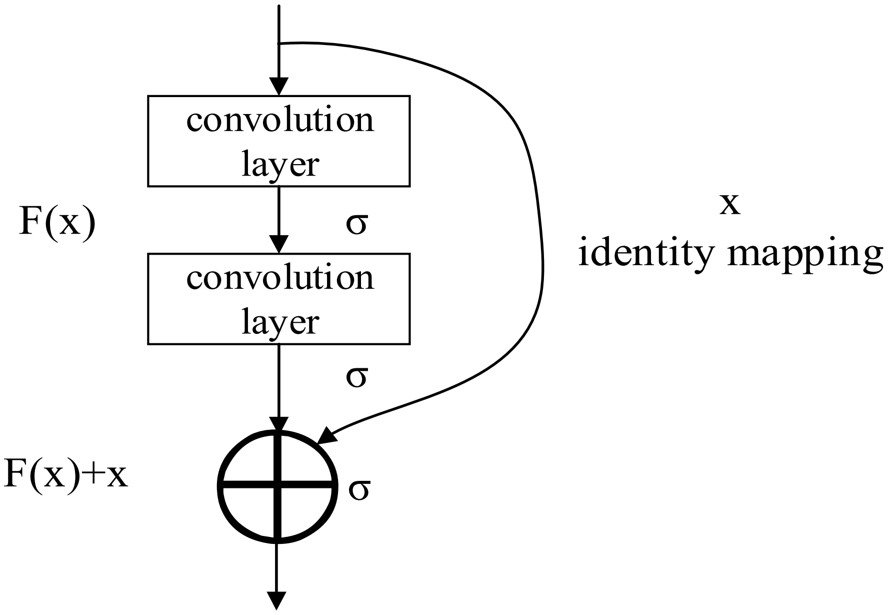

Figure 1. Residual block.

The output formula of the residual block is shown in Equation (3).

$$\mathrm{x_{m+1} = x_m + F(x_{ml}, w_m)} \tag{3}$$

In the formula, x_{m+1} is the output of the m + 1 layer, x_m is the input of the layer m, *f(*x_m*, *w_m*)* is the residual of layer m.

In summary, residual connection is a design that introduces skip connections. It allows the input of the previous layer to be added directly to the output of the subsequent layer, making it easier for the network to learn the residual part, thereby improving the performance and training effect of the network. Through residual connections, information can flow more freely in the network, and gradient signals can also spread more easily. The introduction of this structure makes it possible to train deeper networks, improve the performance of the model, and solve the problem of gradient disappearance and gradient explosion in deep neural networks.

3.1.3. Simple Recurrent Unit

Many advances in the field of deep learning come from enhanced modeling capabilities and related computing capabilities, which usually involve deeper neural networks. While the deep neural network brings significant improvements, it also has certain drawbacks, that is, it requires a lot of training time. Simplified recurrent unit is a sequence modeling method for processing time series data, such as text and voice data. It is a model similar to recurrent neural network, but it has a simpler structure and more efficient calculation. Traditional RNN has some problems, such as difficulty in capturing long-term dependencies, low computational efficiency, and difficulty in parallel computing. The SRU structure is simple and contains only two key operations: reset gate and update gate. Most importantly, SRU has parallel computing capabilities. When calculating each time step, the traditional RNN needs to rely on the results of the previous time step, which makes it difficult to perform parallel computing. SRU does not have this limitation, and can process the entire input sequence in parallel, thus speeding up the training. The structure of SRU is shown in Figure 2.

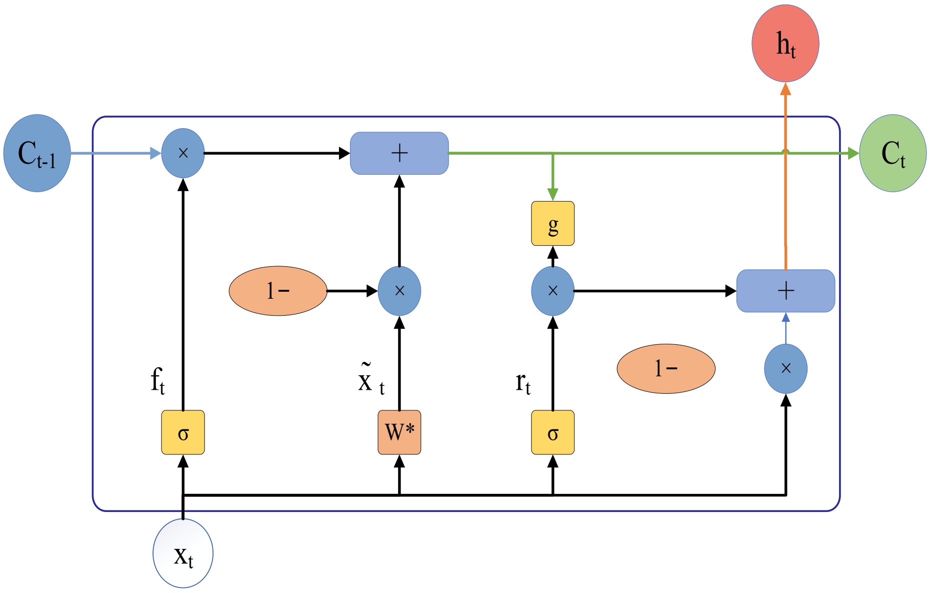

Figure 2. Structure diagram of SRU.

The SRU model is roughly divided into two parts: the light recurrence component and the highway network component. The light recurrence section computes the sequence of states ct while reading the input vectors x_t one at a time in order to collect sequential data. The input x_t and the state c_t generated by the light recurrence are adaptively combined using the reset gate r_t. The skip connection technique is used to calculate the hidden state ht of the highway network unit.

The light recurrence component is mainly manifested in the following two ways. First, in order to reduce the degree of recursion, its two gating units, the forgetting gate and the reset gate, no longer depend on the hidden state h_{t-1} at the previous moment, but depend on the intermediate state c_{t-1} at the previous moment; second, in order to reduce the amount of calculation, the Hadamard product is used instead of the matrix product. The specific implementation process is as follows.

First, the input $\mathrm{x_t}$ is linearly transformed, as shown in Equation (4).

$$\widetilde{\mathrm{x_t}} = \mathrm{Wx_t} \tag{4}$$

The forget gate of SRU is a vector that controls forgetting based on current information and past information. It obtains sequence information by reading the input vector x_t in order and calculates the state vector C_t. The calculation as shown in Equation (5).

$$\mathrm{f_t} = \sigma(\mathrm{W_f x_t} + \mathrm{v_f} \odot \mathrm{C_{t-1}} + \mathrm{b_f}) \tag{5}$$

In the formula, $\odot$ represents the element-by-element multiplication, σ represents the sigmoid function, b_f represents the offset term, and the intermediate state C_t synthesizes the information of the past state and the information of the current input. How much past information is retained depends on the calculated forgetting gate f_t, as shown in Equation (6).

$$\mathrm{C_t} = \mathrm{f_t} \odot \mathrm{C_{t-1}} + (1 - \mathrm{f_t}) \odot \widetilde{\mathrm{x_t}} \tag{6}$$

The highway network unit directly incorporates the input x_t into the calculation, which is equivalent to a crossover of the input in the residual network, as shown in Equations (7) and (8).

$$\mathrm{r_t} = \sigma(\mathrm{W_r x_t} + \mathrm{v_r} \odot \mathrm{C_{t-1}} + \mathrm{b_r}) \tag{7}$$

$$h_t = r_t \odot g(C_t) + (1 - r_t) \odot x_t \tag{8}$$

Here, b_r represents the offset term, and $(1 - r_t) \odot x_t$ is a skip connection, which can optimize the gradient propagation. When the network depth increases, the gradient will not disappear because the propagation distance is too far.

In the above formula, although the dependence of the previous moment is removed, there is still a certain bottleneck, that is, the operation of three matrix multiplications, which provides a deeper optimization strategy matrix multiplication. Batch processing can be performed at all time steps, which can significantly improve the intensity of calculation and improve the utilization of GPU. In the above formula, matrix multiplication can be combined into one, and subsequent processing can be found according to the index, as shown in Equation (9).

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

Among them, $U \in R^{n \times 3D}$ is the calculated matrix, d is the hidden state size, and n is the sequence length of the input data.

3.2. Detection Model

LSTM and GRU can suppress gradient disappearance and gradient explosion to a certain extent when capturing long-distance related information, and their effects are better than traditional SimpleRNN. However, as variants of SimpleRNN, they have the disadvantage of the RNN structure itself, that is, parallel computing cannot be performed. SRU can realize parallel computing of hidden layer dimension, with less calculation, fewer parameters and fast training. In industrial Internet intrusion detection, CNN can locally perceive network traffic data through convolution operation and capture local features and signs of attack in the data so as to realize sensitive detection of intrusion behavior. In addition, CNN can adaptively learn and optimize network weights through the backpropagation algorithm in the training process, thereby improving the generalization ability of the model and the ability to detect unknown attacks. However, as the number of layers of the model network becomes deeper and deeper, the problem of gradient disappearance will inevitably become more obvious, which will consume a lot of computing resources. Thus, we introduce the residual connection. The residual connection allows information to be transmitted across layers in the network, avoiding the problem of gradient disappearance and gradient explosion. This direct connection method can maintain the information integrity of the input data and enable the network to learn and model complex intrusion behaviors more deeply.

The detection model based on Res-CNN-SRU intrusion constructed in this paper is shown in Figure 3. The process is as follows.

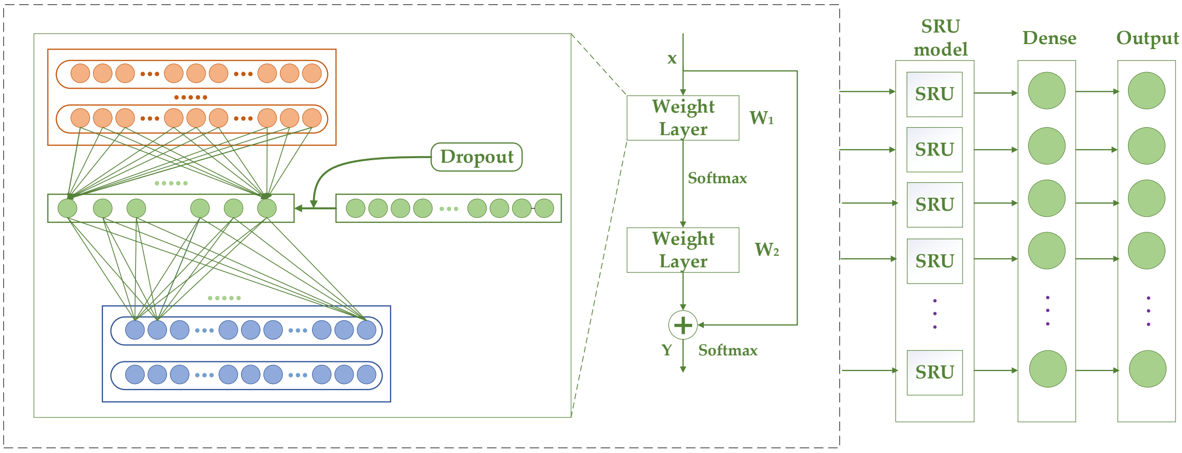

Figure 3. Intrusion detection model based on Res-CNN-SRU.

1. Firstly, the original gas pipeline traffic is preprocessed, and the preprocessed data are input into the convolution layer.
2. Feature extraction. In the 1D residual block, the input has two paths. In the first path, the data are first extracted by the convolutional layer, and then the features are convoluted to summarize the output. Within the residual structure, feature reuse is completed through weight sharing. The second path is where the data are directly output after shortcut processing, and the final output is the sum of two parts. The obtained results are then passed through 1 BN layer, Conv1D layer and maximum pooling layer, and finally the extracted features are obtained. Finally, we use the output vector as the input of the SRU to predict the subsequent features.
3. First, a time series is generated for the input data so that the input data become sequence data with a time step. After the time series traffic is generated, we set the learning rate attenuation to control the learning rate in segments, in order to achieve more efficient learning at different stages and train the neural network. We use small batches of random gradient descent for training. The output obtained by the SRU is passed through the fully connected layer. In order to prevent overfitting, a dropout layer is added. Finally, the classification is performed through the softmax layer.

Dropout function is set after the pooling layer, and some neuron nodes are randomly discarded during the training process, with a probability of 0.2. The mechanism of randomly discarding some neurons is equivalent to training different neural networks in each iteration, which can effectively suppress the occurrence of overfitting. The vector operation of the dropout function is expressed in Equation (10).

$$dropout\left(x_j^{l'}\right) = x_j^{l'} \circ m \tag{10}$$

where $x^{i}{}_{j}$ is the value of the input vector, m is the value of the random mask vector, and $\circ$ represents the product of elements, that is, the multiplication of the same elements. When the corresponding position element of the mask matrix or vector is 1, the input element is retained. When the corresponding position element is 0, the input element is discarded.

The classification stage uses the features learned by the model based on Res-CNN-SRU to mark the input instance. At this stage, a fully connected output layer maps the learned features to the output class. The output of this stage is controlled by the softmax function, as shown in Equation (11).

$$\mathrm{y(x) = softmax(\varphi)} \tag{11}$$

where φ is the output of the dropout layer.

4. Experiment

In this section, we first introduce the dataset used in the experiment, then describe the implementation and evaluation indicators of data preprocessing and carry out multiple comparative tests by constantly adjusting parameters.

4.1. Experimental Dataset

In 2014, Mississippi State University published a set of industrial control system intrusion detection standard datasets from the network layer data of a natural gas pipeline control system [34]. Compared with the KDD CUP99 dataset, the data collected in Mississippi are the data collected in the industrial network, which have higher dimensions and more types of attacks. The attack types of the dataset are shown in Table 1.

4.2. Data Preprocessing

Data preprocessing plays an important role in the experiment and testing of the industrial Internet intrusion detection model, which affects the performance and detection accuracy of the intrusion detection model. The data preprocessing in this paper is mainly divided into three steps: low-variance filter, normalization, and one-hot encoding.

Table 1. Description of datasets.

Attack Type	Description	Label	Number
Normal	Normal data	0	61,156
NMRI	Naive malicious response injection attack	1	2763
CMRI	Complex malicious response injection attack	2	15,466
MSCI	Malicious state command injection attack	3	782
MPCI	Malicious parameter command injection attack	4	7637
MFCI	Malicious function command injection attack	5	573
DOS	Denial-of-service attack	6	1837
Recon	Reconnaissance attack	7	6805

4.2.1. Low-Variance Filter

Our dataset is complex and variable, with many eigenvalues, but not every eigenvalue is well distinguished, that is, it has a very low variance. Such eigenvalues have no analytical value, so we choose to remove them directly. For example, if a feature of a column accounts for 95% of the instance value of all input samples, it can be considered not very useful. If 100% is 1, then this feature is meaningless. This paper chooses to remove the nine feature columns with the smallest variance, and finally obtains a dataset with 17-dimensional effective eigenvalues.

4.2.2. Normalization

The gas pipeline dataset has high-dimensional features, and the maximum and minimum intervals of these features are large. We set the data eigenvalues in a small specific interval. We use min–max normalization to map the features to the range [0, 1]. The normalization formula is as shown in Equation (12).

$$x'_p = \frac{x_q - min(x_p)}{\max(x_p) - min(x_p)} \tag{12}$$

4.2.3. One-Hot Encoding

The classifier cannot directly process the disordered discrete features of the natural gas pipeline dataset. We use one-hot coding to establish a mapping table for discrete feature data to make them ordered and continuous. The dataset has eight classification results. They can be encoded as (1,0,0,0,0,0,0,0), (0,1,0,0,0,0,0,0), (0,0,1,0,0,0,0,0), (0,0,0,1,0,0,0,0), (0,0,0,0,1,0,0,0), (0,0,0,0,0,0,0,1,0,0), (0,0,0,0,0,0,0,0,1,0), (0,0,0,0,0,0,0,0,0,0,0,1), as shown in Equation (13).

$$One - hot\ encoding = \begin{cases} (1,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0),\ if\ the\ result\ is\ Normal(0). \\ (0,\ 1,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0),\ if\ the\ result\ is\ NMRI(1). \\ (0,\ 0,\ 1,\ 0,\ 0,\ 0,\ 0,\ 0),\ if\ the\ result\ is\ CMRI(2). \\ (0,\ 0,\ 0,\ 1,\ 0,\ 0,\ 0,\ 0),\ if\ the\ result\ is\ MSCI(3). \\ (0,\ 0,\ 0,\ 0,\ 1,\ 0,\ 0,\ 0),\ if\ the\ result\ is\ MPCI(4). \\ (0,\ 0,\ 0,\ 0,\ 0,\ 1,\ 0,\ 0),\ if\ the\ result\ is\ MFCI(5). \\ (0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 1,\ 0),\ if\ the\ result\ is\ DOS(6). \\ (0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 1),\ if\ the\ result\ is\ Recon(7). \end{cases} \tag{13}$$

4.3. Benchmarking Metrics

Accuracy, precision, recall and F1 are used as key performance indicators to evaluate the proposed method. The calculation methods of these four indicators are as shown in Equations (14)–(17).

$$\text{Accuracy} = \frac{TN + TP}{TP + FP + TN + TP} \tag{14}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{15}$$

$$\text{Recall} = \frac{TP}{FN + TP} \tag{16}$$

$$\text{F1} = \frac{2TP}{2TP + FP + FN} \tag{17}$$

Among them, *TP* represents the abnormal flow instance of correct classification, *TN* represents the normal flow instance of correct classification, *FP* is the normal flow instance of wrong classification, and *FN* represents the abnormal flow instance of wrong classification.

4.4. Performance Comparison

The performance of the proposed algorithm is evaluated and analyzed, mainly involving detection time, detection accuracy and loss.

4.4.1. Experimental Parameter Settings

This method is compared with three traditional machine learning methods (SVM, naïve Bayes and REPtree) and three deep learning methods based on RNN. The experiment is carried out on a workstation with Intel Core i7-9700H CPU, NVIDIA GeForce GTX745 GPU, 32GB RAM and Windows 10 64-bit operating system. We use the 2.3.1 version of the Keras package to implement our model. Experiments are carried out under the same hardware, software environment and algorithm parameters. The ratio of the training set to the test set is 8:2. We conducted four experiments under different dataset partitions with an average accuracy of 98.7%, similar to the results described in the paper. The specific parameters of the simulation platform are shown in Table 2.

Table 2. Experimental parameters.

Parameter Name	Description	Value
depth	Hidden layer size	4
optimizer	Gradient descent algorithm	RMSprop
activation	Activation function	softmax
epochs	Iteration size	100
batch_size	Samples per epoch	100
unit	Hidden unit size	128
dropout	Random deactivation rate	0.2

4.4.2. Hyperparameter Optimization

In this paper, we use the hyperparameter optimization of grid search to obtain the best performance. This method can evaluate each possible permutation of the selected hyperparameters. This paper focuses on the selection of activation function, optimizer and batch size.

Activation function is an important part of neural network design that directly affects the performance of neural network. Each activation function has different effects on the overall performance and convergence of the neural network, so the choice of activation function is very important. In this paper, we choose three most commonly used activation functions for experiments, namely, rectified linear unit (ReLU), softmax, and hyperbolic tangent (Tanh).

One cycle of learning and adjusting the network weights is called an epoch, and the number of samples used in another iteration becomes the batch size. Different batch sizes affect the convergence speed and convergence effect of this model. In this paper, we choose 10 and 100 as the batch size for hyperparameter search.

In the training process, the choice of the optimizer also affects the best solution to the model parameters. A suitable optimizer can make the model fall into overfitting and achieve global optimization. In this paper, we choose three optimizers, such as Adam, SGD, and RMSprop, to conduct experiments.

Table 3 shows the performance of each hyperparameter combination. By adjusting the hyperparameters, the model with the highest accuracy of 98.79% is obtained. The activation function to achieve the optimal result is softmax, the optimization method is RMSprop, and the batch size is 100.

Table 3. The effect of different hyperparameters on model accuracy.

Activation	Optimizer	Batch_Size	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)
ReLU	Adam	10	98.5	94.06	94.01	94.05
ReLU	Adam	100	96.51	79.67	77.41	78.44
ReLU	SGD	10	90.71	62.26	62.2	62.26
ReLU	SGD	100	78.43	31.58	63.15	42.1
ReLU	RMSprop	10	97.78	88.04	86.91	87.42
ReLU	RMSprop	100	96.26	95.38	95.49	95.42
softmax	Adam	10	98.54	94.35	94.07	94.21
softmax	Adam	100	98.55	94.25	94.23	94.24
softmax	SGD	10	94.54	78.48	78.09	78.26
softmax	SGD	100	90.71	62.27	62.31	62.26
softmax	RMSprop	10	98.55	94.26	94.27	94.26
softmax	RMSprop	100	98.79	95.34	95.04	95.19
Tanh	Adam	10	98.66	94.72	94.70	94.73
Tanh	Adam	100	98.35	93.77	93.08	93.41
Tanh	SGD	10	87.51	60.15	60.22	60.19
Tanh	SGD	100	90.71	62.26	62.31	62.27
Tanh	RMSprop	10	98.42	93.09	92.63	92.83
Tanh	RMSprop	100	98.63	95.15	93.84	94.46

4.4.3. Comparison of Methods

Table 4 shows the performance comparison of our method with the other six methods, including three classical machine learning methods and three deep learning methods based on RNN. The results show that compared with the other methods, the intrusion detection method based on Res-CNN-SRU has the highest accuracy, precision, recall rate and F1 on the gas pipeline dataset, and the training time is the shortest. This means that our proposed method achieves the best intrusion detection results on the gas pipeline dataset.

Table 4. Comparison with other methods.

Paper	Method	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)	Training Time (s)
[10]	SVM	92.5	78.2	93.6	85.2	-
[11]	Naïve Bayes	71.94	70.6	71.9	71.24	-
[12]	Decision Tree	84.9	86.1	84.9	87	-
[18]	RNN	94.95	78.89	78.17	77.98	-
[19]	BLSTM	97.36	89.59	89.36	90.1	102
[20]	CNN-GRU	94.69	78.94	78.92	75.45	107
	Ours	98.79	95.34	95.04	95.38	89

Figure 4 shows the comparison of training accuracy and loss between our method and three RNN-based deep learning methods. All models train 100 epochs. In contrast, our method converges faster in the training process and can obtain higher accuracy.

Experiments are performed on normal data and various types of attack data, as shown in Figure 5. The results show that the RNN algorithm has low accuracy for CMRI and DOS data, the BLSTM algorithm has low accuracy for DOS data, and the CNN-GRU algorithm has low accuracy for NMRI, CMRI, DOS and Recon data. Compared with other algorithms, our method has better performance on all kinds of data in the gas pipeline dataset, and the accuracy of DOS data is significantly higher than other algorithms.

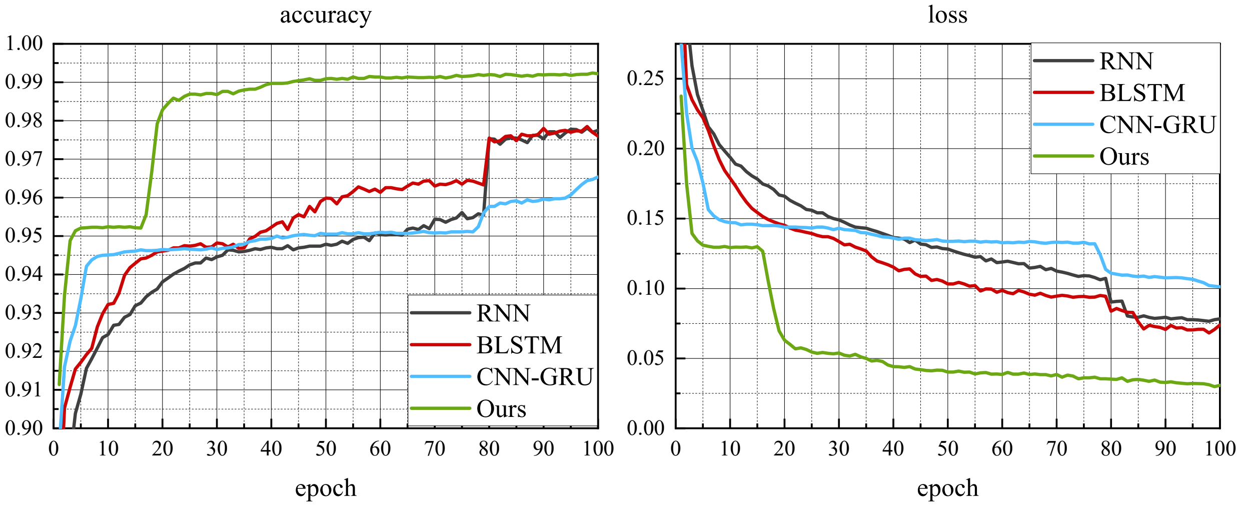

Figure 4. Training accuracy and loss.

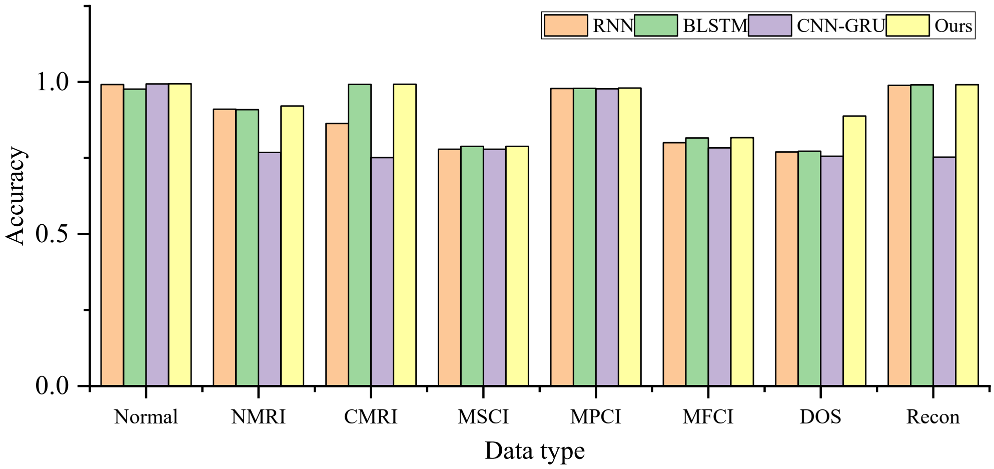

Figure 5. Accuracy of normal data and various types of attack data.

The output vector based on 1D CNN with residual connection is put into a time-varying model based on RNNs. These four models are SimpleRNN, LSTM, GRU and SRU. We use the softmax activation function and RMSprop optimizer.

In Figure 6, SRU has the highest accuracy of the methods. In Figure 7, the training time of SRU is significantly shorter than that of LSTM and GRU. SimpleRNN has the least training time because of its simplest internal structure. However, SimpleRNN is prone to gradient disappearance and gradient explosion. It can be seen from the above results that the accuracy of SRU is the highest among the models, and the training time is shorter than LSTM and GRU.

We conducted a model ablation study to verify the effect of our model. Specifically, verification improvements come from each component. Each component is removed from the Res-CNN-SRU-based model in turn and compared with the complete model based on Res-CNN-SRU.

The results of the ablation study are shown in Figure 8. This proves that whatever components are removed from the model, the final accuracy, precision recall and F1 will decline. Among them, the accuracy of the model based on Res-CNN-SRU is 0.9879, the precision is 0.9534, the recall is 0.9504, and the F1 is 0.9519. If there is no CNN, the accuracy rate is obviously lagging behind and becomes the worst result. After deleting the SRU or residual connection part, all performance indicators also decreased. This shows that the use of CNN can effectively and automatically extract the features of industrial Internet network traffic and improve the accuracy of intrusion detection.

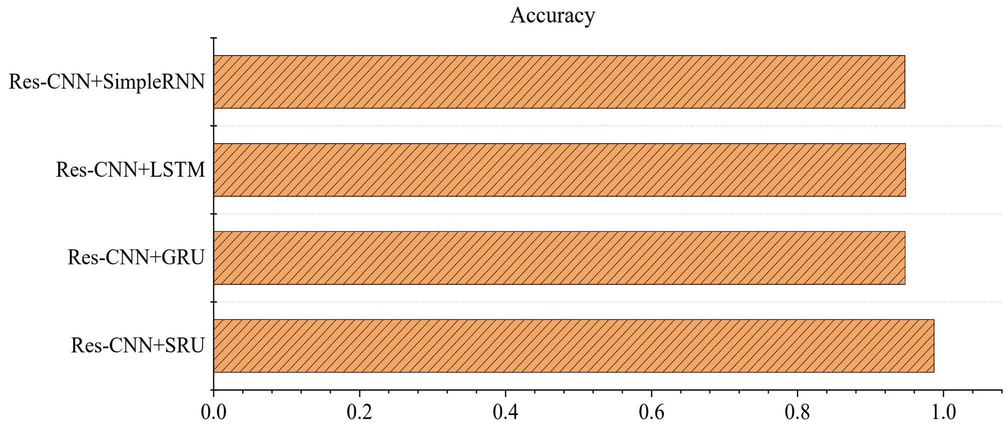

Figure 6. Comparison of accuracy.

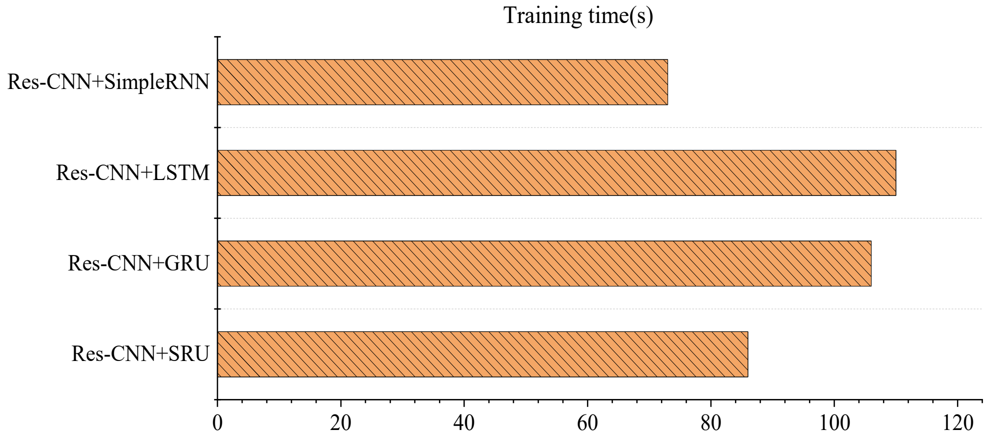

Figure 7. Comparison of training time.

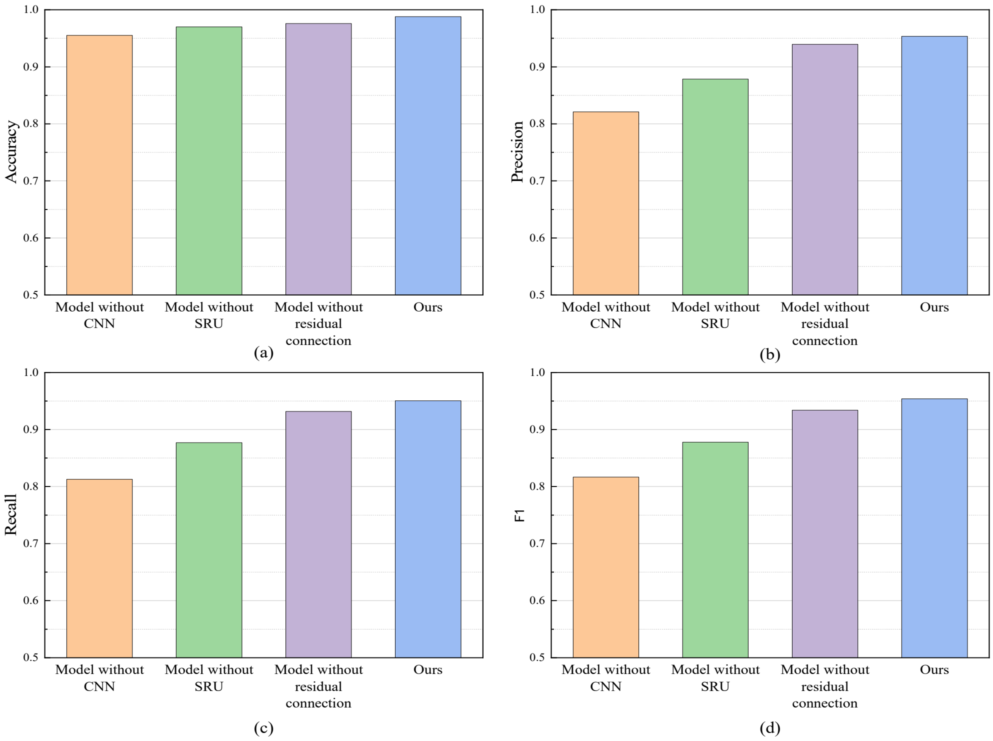

Figure 8. Ablation experimental results. (**a**) Accuracy; (**b**) precision; (**c**) recall; (**d**) F1.

5. Conclusions

Aiming at the problems of large industrial network traffic and difficult processing of features, this paper proposes an industrial Internet intrusion detection method based on Res-CNN-SRU. Our main contribution is to introduce a deep feature extraction method that combines spatial and temporal dimensions. Firstly, we propose a 1D CNN for spatial feature extraction of high-dimensional network traffic, which can extract local features without losing temporal features. At the same time, residual connection can not only alleviate the problem of gradient disappearance but also improve the convergence speed of the network. Then, a parallel computing SRU anomaly traffic detection algorithm is proposed. Compared with the traditional LSTM model, the SRU model has the advantages of efficient calculation, fast training, strong sequence modeling ability and low memory usage, and can train a model with high accuracy faster. Finally, using the gas pipeline dataset, the performance test and ablation experiment of the proposed intrusion detection model are carried out. The experimental results show that the accuracy of this method on the Mississippi natural gas pipeline dataset can reach 0.9879, the precision is 0.9534, the recall is 0.9504, and the F1 is 0.9519, giving higher accuracy and calculation efficiency than the existing method. This proves the performance advantages and effectiveness of our method on the gas pipeline dataset. In real life, the application of industrial Internet intrusion detection can detect and respond to intrusion events in time to reduce potential risks and losses. Early detection and response can prevent attackers from causing more damage to industrial systems and reduce downtime and production disruptions.

However, with the rapid development of the Internet, network intrusion behaviors are ever-changing, and many new attacks have emerged. Due to the lack of sufficient sample data to train machine learning models or detect the characteristics of new attacks, the encryption of network traffic and privacy protection measures may limit the visibility of intrusion detection systems to attack activities. The detection effect of this system against unknown attacks is not ideal. The detection of unknown type attacks is a complex and challenging problem. In the future, we will adopt a combination of supervised learning and unsupervised learning. For the attacks that cannot be identified by the classification model, unsupervised learning will be adopted to perform cluster analysis so as to enhance the detection ability of the intrusion detection system to unknown-type attacks.

Author Contributions: Conceptualization, Z.C. and Y.S.; methodology, Z.C. and Y.S.; validation, J.Z., L.Z. and P.L.; data curation, Y.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (62072416), Key Research and Development Special Project of Henan Province (221111210500), and Key Technologies R&D Program of Henan Province (232102211053, 222102210170).

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Dai, M.; Deng, H.; Chen, B.; Su, G.; Lin, X.; Wang, H. Design of binary erasure code with triple simultaneous objectives for distributed edge caching in industrial Internet of Things networks. *IEEE Trans. Ind. Inform.* **2019**, *16*, 5497–5504. [CrossRef]
2. Huang, X.; Hong, S.H.; Li, Y. Hour-ahead price based energy management scheme for industrial facilities. *IEEE Trans. Ind. Inform.* **2017**, *13*, 2886–2898. [CrossRef]
3. Zhang, Y.; Guo, Z.; Lv, J.; Liu, Y. A framework for smart production-logistics systems based on CPS and industrial IoT. *IEEE Trans. Ind. Inform.* **2018**, *14*, 4019–4032. [CrossRef]
4. Awotunde, J.B.; Chakraborty, C.; Adeniyi, A.E. Intrusion detection in industrial internet of things network-based on deep learning model with rule-based feature selection. *Wirel. Commun. Mob. Comput.* **2021**, *2021*, 7154587. [CrossRef]
5. Bhavsar, Y.B.; Waghmare, K.C. Intrusion detection system using data mining technique: Support vector machine. *Int. Emerg. Technol. Adv. Eng.* **2013**, *3*, 581–586.
6. Kuang, F.; Xu, W.; Zhang, S. A novel hybrid KPCA and SVM with GA model for intrusion detection. *Appl. Soft Comput.* **2014**, *18*, 178–184. [CrossRef]

7. Kumar, V.; Chauhan, H.; Panwar, D. K-means clustering approachto analyze NSL- KDD intrusion detection dataset. *Int. J. Soft Comput. Eng.* **2013**, *3*, 1–4.
8. Zhao, H.; Liu, I. Research on test data generation method of complex event big data processing system based on Bayesian network. *Comput. Appl. Res.* **2018**, *35*, 155–158,162.
9. Zhang, R.; Song, Y.; Wang, X. Network Intrusion Detection Scheme Based on IPSO-SVM Algorithm. In *Proceedings of the 2022 IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC), Dalian, China, 14–16 April 2022*; IEEE: Piscataway, NJ, USA, 2022; pp. 1011–1014.
10. Anton, S.D.D.; Sinha, S.; Schotten, H.D. Anomaly-Based Intrusion Detection in Industrial Data with SVM and Random Forests. In *Proceedings of the 2019 International Conference on Software, Telecommunications and Computer Networks (SoftCOM), Split, Croatia, 19–21 September 2019*; IEEE: Piscataway, NJ, USA, 2019; pp. 1–6.
11. Khan, A.A.Z. Misuse Intrusion Detection Using Machine Learning for Gas Pipeline SCADA Networks. In Proceedings of the International Conference on Security and Management (SAM), The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), Las Vegas, NV, USA, 29 July–1 August 2019; pp. 84–90.
12. Al-Asiri, M.; El-Alfy, E.S.M. On using physical based intrusion detection in SCADA systems. *Procedia Comput. Sci.* **2020**, *170*, 34–42. [CrossRef]
13. Eigner, O.; Kreimel, P.; Tavolato, P. Detection of Man-in-the-Middle Attacks on Industrial Control Networks. In *Proceedings of the 2016 International Conference on Software Security and Assurance (ICSSA), Saint Pölten, Austria, 24–25 August 2016*; IEEE: Piscataway, NJ, USA, 2016; pp. 64–69.
14. Aiswal, S.; Saxena, K.; Mishra, A.; Sahu, S.K. A KNN-ACO approach for intrusion detection using KDDCUP'99 dataset. In *Proceedings of the 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 16–18 March 2016*; IEEE: Piscataway, NJ, USA, 2016; pp. 628–633.
15. Xia, W.; Neware, R.; Kumar, S.D.; Karras, D.A.; Rizwan, A. An optimization technique for intrusion detection of industrial control network vulnerabilities based on BP neural network. *Int. J. Syst. Assur. Eng. Manag.* **2022**, *13* (Suppl. S1), 576–582. [CrossRef]
16. Yang, A.; Zhuansun, Y.; Liu, C.; Li, J.; Zhang, C. Design of intrusion detection system for Internet of Things based on improved BP neural network. *IEEE Access* **2019**, *7*, 106043–106052. [CrossRef]
17. Li, Y.; Xu, Y.; Liu, Z.; Hou, H.; Zheng, Y.; Xin, Y.; Zhao, Y.; Cui, L. Robust detection for network intrusion of industrial IoT based on multi-CNN fusion. *Measurement* **2020**, *154*, 107450. [CrossRef]
18. Yin, C.; Zhu, Y.; Fei, J.; He, X. A deep learning approach for intrusion detection using recurrent neural networks. *IEEE Access* **2017**, *5*, 21954–21961. [CrossRef]
19. Roy, B.; Cheung, H. 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 2018 28th International Telecommunication Networks and Applications Conference (ITNAC), Sydney, NSW, Australia, 21–23 November 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 1–6.
20. Cao, B.; Li, C.; Song, Y.; Qin, Y.; Chen, C. Network intrusion detection model based on CNN and GRU model. *Appl. Sci.* **2022**, *12*, 4184. [CrossRef]
21. Zhou, X.; Hu, Y.; Liang, W.; Ma, J.; Jin, Q. Variational LSTM enhanced anomaly detection for industrial big data. *IEEE Trans. Ind. Inform.* **2020**, *17*, 3469–3477. [CrossRef]
22. Hwang, R.H.; Peng, M.C.; Nguyen, V.L.; Chang, Y.L. An LSTM-based deep learning approach for classifying malicious traffic at the packet level. *Appl. Sci.* **2019**, *9*, 3414. [CrossRef]
23. Ling, J.; Zhu, Z.; Luo, Y.; Wang, H. An intrusion detection method for industrial control systems based on bidirectional simple recurrent unit. *Comput. Electr. Eng.* **2021**, *91*, 107049. [CrossRef]
24. Xu, Y.; Jin, T.; Xu, Y.; Shi, X.; Chen, S.; Sun, W.; Xue, Y.; Wu, H. Transformer image recognition system based on deep learning. *J. Shanghai Electr. Power Univ.* **2021**, *37*, 51–56.
25. Mahmoud, A.; Zrigui, M. Sentence embedding and convolutional neural network for semantic textual similarity detection in Arabic language. *Arab. J. Sci. Eng.* **2019**, *44*, 9263–9274. [CrossRef]
26. Zhang, K.; Guo, Y.; Wang, X.; Yuan, J.; Ding, Q. Multiple feature reweight densenet for image classification. *IEEE Access* **2019**, *7*, 9872–9880. [CrossRef]
27. Huang, J.T.; Li, J.; Gong, Y. An Analysis of Convolutional Neural Networks for Speech Recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, South Brisbane, QLD, Australia, 19–24 April 2015*; IEEE: Piscataway, NJ, USA, 2015; pp. 4989–4993.
28. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet Classification with Deep Convolutional Neural Networks. In Proceedings of the 2012 Advances in Neural Information Processing Systems (NIPS2012), Lake Tahoe, NV, USA, 3–6 December 2012; pp. 1097–1105.
29. Wang, H.; Shi, H.; Chen, X.; Zhao, L.; Huang, Y.; Liu, C. An improved convolutional neural network based approach for automated heartbeat classification. *J. Med. Syst.* **2020**, *44*, 35. [CrossRef] [PubMed]
30. Zhao, L.; Ma, Y. Fault diagnosis of gear box based on one dimensional convolutional neural networks. *J. Test Meas. Technol.* **2019**, *33*, 302–306.
31. Qu, J.; Yu, L.; Yuan, T.; Tian, Y.; Gao, F. Adaptive fault diagnosis algorithm for rolling bearings based on one-dimensional convolutional neural network. *Chin. J. Sci. Instrum.* **2018**, *39*, 134–143. [CrossRef]

32. Shaikh, A.; Gupta, P. Real-time intrusion detection based on residual learning through ResNet algorithm. *Int. J. Syst. Assur. Eng. Manag.* **2022**, 1–15. [CrossRef]
33. He, K.M.; Zhang, X.Y.; Ren, S.Q.; Sun, J. Deep Residual Learning for Image Recognition. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016*; IEEE: Piscataway, NJ, USA, 2016.
34. Morris, T.; Gao, W. Industrial Control System Traffic Data Sets for Intrusion Detection Research. In *Critical Infrastructure Protection VIII. ICCIP 2014. IFIP Advances in Information and Communication Technology*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 65–78.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.