A New Intrusion Detection Method Based on Industrial Internet

Yuhong Wu[1*]*, Xiangdong Hu*[2]

[1] *College of Computer Science and Technology, Chongqing University of Posts and Telecommunications, China*
[2] *College of Automation, Chongqing University of Posts and Telecommunications, China*
269610173@qq.com, huxd@cqupt.edu.cn

Abstract

With the rapid development of the Industrial Internet, the security risks of the Industrial Internet will soon be exposed. In view of the low accuracy of the existing intrusion detection algorithms, the difficulty in adapting to the industrial Internet networking mode, and the imbalance of massive data, this paper proposes a capsule-based method. The network intrusion detection method, this method first refers to the DRN structure, introduces the residual block as the main capsule layer to extract high-quality feature maps, then uses the dynamic routing algorithm to cluster the features, and uses the Adam algorithm to optimize the learning in backpropagation rate to make the detection model stable and fast. In terms of convergence, the detection accuracy rate in the simulation test using the gas pipeline data set in this paper reaches 99.28%, and it is more robust to massive unbalanced data. The experimental results show that this method can better meet the current industrial internet security needs.

Keywords: Security risks, Intrusion detection, Unbalanced data, Encryption mode, Secure e-commerce

1 Introduction

In recent years, the Industrial Internet has emerged in many industrial fields such as intelligent manufacturing, gas supply, power and water conservancy [1], and has become an indispensable emerging industry supporting technology to promote economic development and new-generation infrastructure construction. The industrial control network was originally relatively closed and physically isolated from the outside world. The network construction itself paid more attention to the operation stability and functional safety, and lacked a comprehensive design for information security issues under the condition of network opening. With the in-depth development of industrial informatization, attacks against the Industrial Internet are becoming more frequent and destructive [2]. Typically, the "Stuxnet" virus's attack on Iran's nuclear power plant through the vulnerability of Siemens equipment sounded the alarm for industrial Internet information security. Although the security products in the traditional Internet have reached a certain maturity based on historical accumulation and iteration, due to the resource characteristics, operation mode and network attributes of the Industrial Internet itself, the existing methods cannot be directly transplanted into the Industrial Internet, and often need to be customized. Targeted solutions [3]. For example, the research on the communication mode and encryption mode of the industrial system shows that the industrial network is not compatible with the home and office network, so the ordinary Intrusion Detection System (IDS) cannot directly adapt to the industrial application [4]. At the same time, because of its relative independence, the number of attack behaviors in the Industrial Internet is much lower than that of the traditional Internet, which also brings difficulties to intrusion detection.

At the same time, the methods of constructing IDS show a diversified development trend, and new ideas and methods are constantly emerging. Reference [5] uses the entropy discretization algorithm and decision tree to build a classifier to classify multiple applications, and then sparse them. Reference [6] uses Open-plc platform and AES-256 encryption to simulate data acquisition and monitoring control (Supervisory Control and Data Acquisition, SCDA) system, on this basis, uses unsupervised k-means algorithm to attack code injection, denial of service Attacks and interceptions are detected. Reference [7] uses the perceptual hash matrix to quantify the attributes, and then uses the K-nearest neighbor voting principle to complete the intrusion detection task. The literature [5-7] completed the construction of the classifier by means of manual selection and combination of features. There are generally problems such as low detection accuracy and poor system robustness. The quality of feature selection also greatly affects the experimental results. Therefore, the literature [8] designed a hybrid IDS using support vector machines and deep belief networks for industrial control systems, but the literature used the NSL-KDD dataset for simulation. NSL-KDD is relatively old and not suitable for industrial control. system environment. Reference [9] uses the convolutional neural network based on dispersion normalization (Min-Max Normalization, MMN) to analyze the traffic of the campus network. The model has less overhead and is easy to train, and solves the problem of parameter selection well. However, the structure of this classification method is slightly different from that of LeNet-5, and the structure is simple, which makes it difficult to deal with feature learning of massive and complex data. Reference [10] uses

*Corresponding Author: Yuhong Wu; E-mail: 269610173@qq.com

DOI: https://doi.org/10.70003/160792642025012601011

a window-based instance selection algorithm to clean the training set and builds an intrusion classification model based on a recurrent neural network for the problem of unbalanced distribution of intrusion samples. Although a high accuracy rate has been achieved, the experiment uses a complex preprocessing process, which is difficult to reflect the advantages of deep learning implicit feature extraction. Reference [11] uses conditional deep belief networks to detect attacks in smart grids, which uses a bus test system for simulation and provides comparisons with methods such as artificial neural networks and support vector machines. These methods discuss different schemes of IDS from multiple perspectives, but some methods use complex means to process data, and it is difficult to solve the different requirements of different classification algorithms for feature selection. Most models have a single structure, and there are problems such as slow model convergence and poor robustness when the sample distribution is uneven

In 2017, the literature [12] proposed a vector-based capsule network (Capsule Network, Caps Net). The network introduces a vector capsule layer and a dynamic routing algorithm. Capsules are used to represent sets of neurons, and dynamic routing is used to connect capsules between different hidden layers to map the relative relationship between different features. Capsule networks improve the problem that traditional convolutional neural networks are insensitive to target location. For example, in the literature [13] on the classification of hyperspectral images, even if the number of test samples is much larger than the number of training samples, the capsule network still achieves good classification results. However, because the dynamic routing algorithm cannot share the weight of each neuron, the parameter amount of the capsule network is much larger than that of the traditional convolutional neural network.

Through the analysis of the above literature, it is known that deep learning has the potential to extract high-quality features from industrial Internet data to create better models. Inspired by the capsule network, this paper introduces a residual structure to improve it, builds a capsule network (Residual Capsule Network, RCN) fused with residual blocks to learn the correlation characteristics of industrial Internet data, and builds an intrusion detection model to realize the network. Efficient handling of traffic. This method avoids the complicated process of manual feature extraction, improves the detection accuracy under the background of uneven distribution of intrusion samples, and shortens the model training time.

2 Capsule Network Fused with Residual Blocks

The capsule network proposed in [12] is a shallow neural network. Its structure is shown in Figure 1.

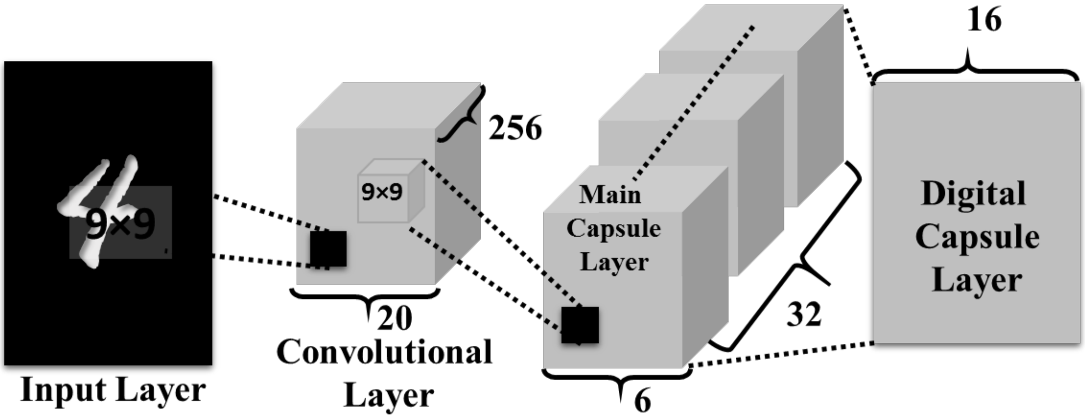


Figure 1. Basic structure of capsule network

Capsule network uses dynamic routing algorithm to iterate the relationship between capsules. Shallow neural networks have a limited number of layers and may lack the capacity to capture complex and hierarchical representations. They are less relevant for tasks involving intricate patterns. In comparison, capsule networks with dynamic routing enable efficient communication between capsules and explicitly model spatial relationships, making them more relevant for tasks requiring detailed spatial understanding and object recognition. Due to the large number of parameters, capsule network is difficult to implement in the industrial Internet. Capsule networks tend to have a larger number of parameters compared to other neural network architectures due to the presence of instantiation parameters in each capsule. These parameters encode information about the pose, orientation, and other properties of the entity being represented. The data characteristics of the industrial internet, such as high dimensionality, heterogeneity, sparsity, and noise, make the performance of traditional models poor. At the same time, the data characteristics of the industrial Internet make the performance of traditional models poor. In order to reduce the computational cost of the capsule network and improve the recognition accuracy, this paper improves the capsule network by introducing a residual structure, whose structure is shown in Figure 2. In order to improve productivity and enable quicker inference, it is critical to reduce computing cost in capsule networks. Improving recognition accuracy offers more accurate and trustworthy predictions, boosting the network's overall functionality.

RCN uses residual structure to reduce dimensionality of traffic features. The Residual Capsule Network (RCN) uses residual connections to preserve original features while adding newly learned features, reducing the dimensionality of traffic features. The features of traffic that have been reduced by RCN using the residual structure include noise, redundant information, and irrelevant details. Residual structure in RCN effectively reduces the dimensionality of traffic features by learning residual representations, focusing on capturing only essential information. The capsule network fused with residual blocks includes a residual network module composed of residual blocks, convolutional layers and pooling layers, and a capsule network module composed of a main capsule layer and a digital capsule layer. The Main capsule layer Captures low-level features, establishes communication between capsules using dynamic routing. As well, Digital capsule layer represents higher-level concepts or objects, computes instantiation parameters using dynamic routing.

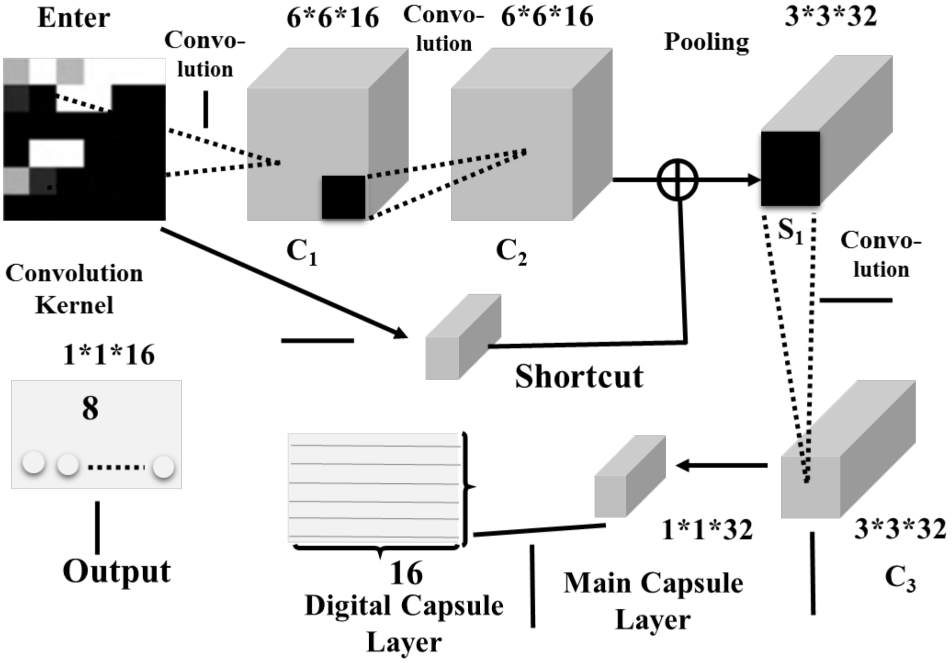


Figure 2. Capsule network structure of fusion residual block

2.1 Residual Network Module

In 2016, He [14] et al. of Microsoft Research proposed the Deep Residual Network (DRN). DRN, also known as ResNet, completes the construction of up to 152 layers of networks by introducing residual blocks, which greatly improves the performance degradation problem caused by the increase of network depth. Performance degradation has Difficulty in effectively propagating information from lower-level capsules to higher-level capsules, leading to loss of spatial relationships and hindering accurate representation of complex objects. At the same time, DRN introduces a global average pooling layer to extract the mean value of each feature map, which can reduce the dimension of the data and avoid overfitting. In this paper, a residual network module for feature extraction is constructed on the basis of the DRN structure, so as to improve the input feature quality of the capsule network.

2.1.1 Convolutional Layer

Convolutional layers are the main structure for abstracting images. The convolutional layer is composed of several different feature maps. Image abstraction is the process of simplifying and reducing the complexity of an image while retaining important information. In convolutional layers, image abstraction is achieved by using multiple feature maps that capture different levels of visual details. These feature maps range from low-level features like edges and corners to high-level features like object parts and complex patterns. By stacking these feature maps, convolutional layers create a hierarchical representation that captures a wide range of visual information. The convolutional layer performs a convolution operation on a small area of the upper layer to form the node of the next layer [15]. This area is called the convolution kernel or filter. The specific form of convolution is:

$$x_j^l = f\left(\sum_{i\in M_j} x_i^{l-1} k_{ij}^l + b_j^l\right) \quad (1)$$

In formula (1), x represents the jth position input in the lth layer after the convolution operation; x-1 represents the ith position input in the l-1th layer; k represents the connection between the l-1th layer and the lth the convolution kernel of the layer; b represents the bias of the jth position in the lth layer. In this paper, 16 convolution kernels with stride 1 are used in the C1 and C2 layers of RCN to convolve the input, and the size of the feature map before and after convolution is kept unchanged by padding 0. Because the convolution operation does not contain nonlinear components, the output data obtained after the calculation of formula (1) also needs to use the activation function for nonlinear processing, so that the network has the ability to fit complex features. Its primary role is to perform a linear transformation on the input data, capturing local features and spatial relationships. Nonlinearity is introduced separately through activation functions, allowing the network to learn complex patterns and make nonlinear predictions. In convolutional neural networks, the ReLU function can greatly improve network sparsity and improve model efficiency [16]. The ReLU (Rectified Linear Unit) function can greatly improve network sparsity and enhance model efficiency due to its ability to eliminate negative values. By setting negative activations to zero, ReLU effectively sparsifies the network, making many neurons inactive. This sparsity reduces the computational load during both forward and backward propagation, resulting in improved model efficiency.

2.1.2 Residual Block

The residual block is the main component of the residual network, and it has different representations. The common feature is that a shortcut is introduced, and the input is passed to the output as part of the result. A typical residual block structure is shown in Figure 3.

Let x be the input, y be the output, Wi be the weight matrix, and be the residual function to be learned, the residual block can be expressed as:

$$y = f\left(x, \{w_i\}\right) + x \tag{2}$$

If the dimension of x is different from the dimension of the residual function, in order to realize the fusion of input and output, the linear projection matrix Ws is used to change the dimension. The specific formula is:

$$y = f\left(x, \{w_i\}\right) + w_s x \tag{3}$$

To achieve dimensional transformation, we use a set of convolution kernels of size 1 × 1 × 16 in the shortcut to transform the input into a suitable form.

2.1.3 Pooling layer

The main function of the pooling layer is to reduce the dimension of the data and compress the feature map of the upper layer. The pooling layer’s principal functions are believed to be data dimension reduction and upper layer compression of feature maps since they minimize the number of parameters, produce translation-invariant representations, pick out key features, and allow for spatial downsampling. The performance and generalization of deep learning models are improved by these functions, which result in more effective and reliable feature extraction. And form a new feature map. This processing method can reduce the complexity of the network and retain the main feature information of the original image. The pooling layer can be expressed as:

$$x_j^l = f\left(\beta_j^l down(x_j^{l-1}) + b_j^l\right) \tag{4}$$

In formula (4), down ( ) is a sub-sampling function, which usually performs a weighted summation on the input feature map locally; β is a multiplicative parameter set on demand; b is a bias.

This paper uses two methods: maximum pooling and global average pooling. Maximum pooling preserves dominating characteristics and spatial information while extracting the most value possible from each pooling region. By calculating the average value of each feature map across all spatial dimensions, global average pooling summarizes the overall existence of features. While global average pooling offers a more condensed depiction, maximum pooling keeps more specific information. The main difference is the sub-sampling function used. In the S1 layer of the residual network module, a maximum pooling layer with a step size of 2 and a sampling kernel size of 2 × 2 is used for data dimension reduction, and then the output of the C3 layer is subjected to global average pooling to obtain a set of 32-dimensional quantity as the input to the capsule network module.

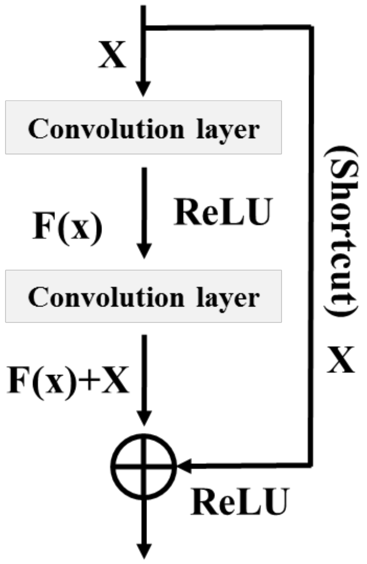


Figure 3. Residual block structure

2.2 Capsule Network Module

The traditional convolutional neural network abstracts the image features through the convolution kernel, and uses the fully connected layer to output the classification result. In a traditional CNN, convolution kernels are applied to the input image, extracting features by performing element-wise multiplications and aggregating values within a receptive field. These kernels act as filters, highlighting specific characteristics like edges, textures, or patterns. Through multiple layers, the network progressively captures complex and abstract features, enabling effective image representation. Its scalarized calculation method has poor identification of the spatial relationship of objects. The scalarized calculation method in traditional neural networks does not explicitly consider spatial relationships between objects. It treats the input data as a flattened vector, disregarding the spatial information encoded in the image. As a result, the identification of spatial relationships between objects is limited in traditional neural networks. The capsule network uses vectorized neurons (ie capsules) to replace Scalar neuron node. The information carried by each capsule increase from one-dimensional to multi-dimensional. The direction of the vector represents various attributes of a specific entity appearing in the image, such as relative position, size, texture, etc. The length of the vector represents different attributes The existence probability of. In order to meet the requirements of back-propagation between capsules, the literature [12] proposed a dynamic routing algorithm for the iterative relationship between capsules. The core idea is that the weight of the capsule is determined by the similarity between the input of the low-level capsule and the output of the high-level capsule. Higher similarity results in greater weight, indicating a stronger agreement between lower-level features and higher-level predictions. This weighting scheme prioritizes capsules that align well with the predicted entities, allowing the network to focus on more relevant and informative features during routing and inference. If the input of low-level capsules has a high similarity with the output of high-level capsules, the routes of these low-level capsules are higher-level capsules.

First, high-level capsules are computed from low-level capsules. In the initial stage of dynamic routing, the probability formula for connecting the ith capsule in layer L to capsule j in layer L+1 is

$$c_{ij} = \frac{\exp(b_{ij})}{\sum_j \exp(b_{ij})} \quad \textbf{(5)}$$

In Equation (5), bij is the prior probability that capsule i is connected to capsule j. When the route is updated, first calculate the predicted capsule uj|i of the output of the L layer capsule i to the L+1 layer capsule j, namely

$$u_{j|i} = w_{i|j} \times u_i \quad \textbf{(6)}$$

In formula (6), wi|j is the transformation matrix, and ui is the L-layer capsule i. After calculating the prediction capsule, the high-level capsule is calculated by formula (7) and formula (8), and the process of calculating vj can use the extrusion function means that

$$s_j = \sum_j c_{ij} \times u_{j|i} \quad \textbf{(7)}$$

$$v_j = \frac{\left\|s_j\right\|^2}{1+\left\|s_j\right\|^2}\frac{s_j}{\left\|s_j\right\|} = squash(s_j) \quad \textbf{(8)}$$

In formula (8), sj is the total input of L-layer capsule, and vj is the output of L+1-layer capsule j. Use vj and prediction capsule uj|i to update bij, and start a new cycle from equation (5). The iterative process of the dynamic routing algorithm is shown in Algorithm 1.

Algorithm 1. Dynamic routing algorithm

```
Input: number of iterations n, number of capsule layers l
    process:
    $b_{ij} = 0$
  1. Initialization:
  2. FOR n DO
    $c_{ij} = \frac{\exp(b_{ij})}{\sum_j \exp(b_{ij})}$
  3.
    $s_j = \sum_j c_{ij} \times u_{j|i}$
  4.
    $v_j = squash(s_j)$
  5.
    $b_{ij} \leftarrow b_{ij} + u_{j|i} \cdot v_j$
  6. Output: $v_j$
  7. END FOR
```

2.3 Loss Function and Optimization Algorithm

Capsules use the vector length to represent the probability of its representation content, and the sum of the output probabilities is not equal to 1. Therefore, different from the cross-entropy loss commonly used in traditional classification tasks, this paper uses interval loss to construct the loss function of the network. The margin loss function can be expressed as

$$L_C = T_c \max\left(0, m^+ - \left\|v_c\right\|\right)^2 + \lambda\left(1-T_c\right)\max\left(0, \left\|v_c\right\| - m^-\right)^2 \quad \textbf{(9)}$$

In formula (9), c represents the category; Tc represents the existence of the c-th type of intrusion; vc represents the length of the capsule in the output layer, that is, the probability that the sample belongs to the c-th type; m+ is the upper bound for penalizing false positives, and m- is the penalty The lower bound of false negatives; λ is the proportional correlation coefficient, which is used to adjust the proportion of the two. In this paper, the values of λ, m+ and m- are set to 0.25, 0.9 and 0.1 respectively.

The dynamic routing algorithm solves the weight update problem between capsule layers. The capsule network efficiently uses dynamic routing algorithms to iteratively establish and update relationships between capsules, allowing for effective communication and information propagation. However, dynamic routing only exists between capsules. In order to improve the convergence ability of the network, a back-propagation process needs to be introduced. Backpropagation is the process of propagating the error from the network's output layer back to the input layer. It calculates the gradients of the network's parameters using the chain rule, allowing for their adjustment through gradient descent. By iteratively updating the parameters in the direction that minimizes the error, backpropagation improves the convergence ability of the network. In this paper, the Adam method is used as the loss function optimization algorithm, and the neuron weights are updated by iteratively minimizing the loss value, so that the RCN converges smoothly.

2.4 Intrusion Detection Model Architecture

On the basis of RCN, this paper takes the industrial Internet as the object, and proposes an intrusion detection model as shown in Figure 4.

The model strengthens the relative relationship of network data mapping images through preprocessing, fully mines data information features, and performs intrusion detection tasks with RCN as the core, mainly including the following modules. Constrained intrusion detection uses RCN (Recurrent Convolutional Network) as the core model for intrusion detection tasks. The data preprocessing module preprocesses the input data to enhance its quality and relevance to the detection task. Together, RCN and the data preprocessing module improve the performance and accuracy of intrusion detection by effectively learning and classifying patterns in the data.

(1) Data preprocessing module: Standardize, normalize and map the industrial Internet data into a grayscale matrix. It is then converted to a grayscale image for easy viewing and processing.

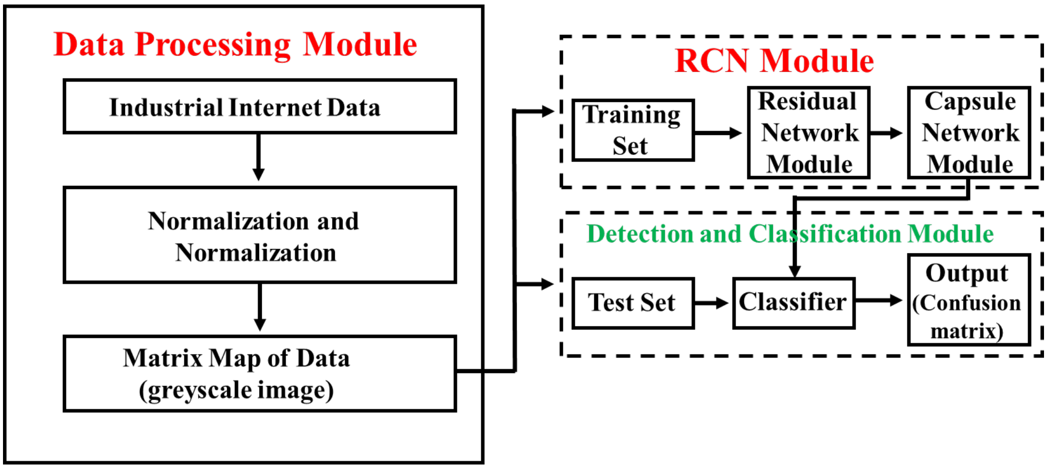


Figure 4. Intrusion detection model architecture based on RCN network

(2) RCN module: The grayscale image is used as the module input, and the data features are extracted through the residual network structure. After being processed by the extrusion function, they are sent to the capsule module to aggregate the features.

(3) Detection and classification module: Train the classifier according to the data category, use the classifier to detect the preprocessed attack samples, output a multi-dimensional confusion matrix, and observe the detection results through the confusion matrix.

3 Data Preparation and Preprocessing

3.1 Network Characteristics and Data Analysis

3.1.1 Characteristics of Industrial Internet Constructed by Constrained Intrusion Detection Scheme

At this stage, the Industrial Internet is mainly composed of two parts: the internal network of the factory and the external network. The external network is used to connect factories and customers, factories and supply chains, etc., and is mainly undertaken by the public Internet. The internal network is mainly used for industrial control and developed from the industrial control network. It mainly includes industrial production data acquisition and monitoring control systems, distributed control systems, programmable logic controllers, and internal production and operation decision support systems.

In more complex industrial systems such as modern smart factories, the controllers are usually deployed in a distributed manner, and the network includes multiple entities and a large number of sensor connections. They play a vital role in monitoring, controlling, and optimizing machines, production lines, and overall factory operations. Their distribution enables decentralized decision-making, real-time data processing, and efficient communication, enhancing productivity and flexibility in the industrial system. Various devices in an industrial control network are usually supplied by different manufacturers and use specific protocols that are different from those of the traditional Internet. Interoperability between devices with different protocols in an industrial control network is achieved through the use of gateways or protocol converters that translate data between protocols, enabling seamless communication. Its network nodes have large differences in computing power and high real-time requirements. These factors make the industrial Internet quite different from civil networks [17-18].

In addition, Industrial Internet traffic is characterized by traffic regularity and protocol specificity, with stable throughput and periodic patterns, with clear packets and predictable data flow [19]. This feature enables supervised learning and is suitable for the development and implementation of anomaly-based intrusion detection techniques. Anomaly-based intrusion detection involves detecting abnormal patterns or behaviors in network/system activities. Key advantages include detecting unknown attacks, adaptability, identifying insider threats, low false positive rates, compliance support, and detecting novel attacks. At the same time, due to the stability and relative isolation of industrial systems, the data generated every day is massive, but the proportion of cyber-attacks is relatively low, which is statistically manifested as a serious imbalance of data.

3.1.2 Analysis of Experimental Data

Based on the requirements of the typicality, wide acceptance, systematicness and suitability of the research object of the experimental data, this paper uses the gas pipeline data set (gaspipe line) [20] collected by the SCDA laboratory of Mississippi State University to verify the detection model. This Industrial Internet dataset is derived from a laboratory-scale gas pipeline platform using the Modbus/TCP protocol. The platform includes sensors such as compressors and pressure gauges and uses solenoid valves to control small gas-tight pipes, and uses a proportional-integral-derivative control scheme to maintain pipe air pressure. Modbus traffic is monitored and stored via an RS-232 based network data logger.

The platform uses line plugins to capture data logs and perform attack injection, monitor serial port communications through a C program running on a VMware virtual machine, time stamp traffic and record it in log files. Taking command injection attacks as an example, sending malicious commands to the platform will try to switch compressors or adjust the status of safety valves. The specific data of such attacks can be formed by recording network traffic characteristics, process control and sensor status. The gas pipeline data set divides data into four categories: injection attack, denial of service attack, reconnaissance attack and normal data. Among them, injection attack can be divided into 5 subcategories. The specific categories and distribution of the data are shown in Table 1.

Existing models for describing public Internet traffic cannot be directly applied to the Industrial Internet for a number of reasons, such as different day and night patterns, lack of correlation of data, and differences in distribution. Compared with KDD99 [21] and other data sets with a history of more than 20 years, the gas pipe line is more in line with the status quo of the industrial Internet, and its data construction method is more in line with the requirements of the real network environment. Data construction methods in the industrial internet involve techniques such as data fusion, preprocessing, and augmentation. These methods aim to create high-quality, diverse datasets by combining different data sources, cleaning and transforming the data, and generating synthetic data samples. They enable effective analysis and modeling in the industrial internet by providing reliable and representative data.

In order to fully verify the effectiveness of RCN in the field of industrial Internet intrusion detection, 50% of the gas pipeline data set is randomly selected for model training, and the other 50% is used for testing, and the training set is randomly divided at a ratio of 4:1 The distribution of the test set and the test set is shown in Table 2. It can be seen from Table 2 that both the training set and the test set have obvious data imbalance. The classification problem caused by data imbalance starts from the data skewness in binary classification, which will bias the detector to the majority class samples [22]. When the minority class samples are crucial in some cases, the inability to distinguish the minority class samples will cause Makes detection useless.

Table 1. Sample category distribution of gas pipeline dataset

Kind of data	Data sources	
	Training set	Test set
Normal	24513	6107
NMRI	1107	268
CMRI	6036	1622
MSCI	346	71
MPCI	3099	746
MFCI	222	56
Dos	776	169
RECO	2709	663
TOTAL	38808	9702

Table 2. Sample category distribution of gas pipeline dataset

Data type	Quantity	Tag value	Description
Normal	61156	0	Normal Data
NMRI	2763	1	Simple Malicious Response Injection Attack
CMRI	15466	2	Complex Malicious Response Injection Attacks
MSCI	782	3	Malicious Stateful Command Injection Attack
MPCI	7637	4	Malicious Parameter Command Injection Attack
MFCI	573	5	Malicious Function Command Injection Attack
Dos	1837	6	Denial of Service Attack
RECO	6805	7	Reconnaissance Attack

Taking the training set as an example, the proportion of normal data in the total data is as high as 63%, while the malicious function command injection attack MFCI with the least amount of data only accounts for 0.57% of the total data, which is very small compared to the majority of samples. Causes the relative scarcity of minority class samples.

3.2 Network Characteristics and Data Analysis

3.2.1 Standardization and Normalization of Data

There are large numerical differences in gas pipeline data samples, and a large number of outliers and outliers, which will negatively affect the convergence speed and accuracy of the intrusion detection model. Large numerical differences in gas pipeline data samples can negatively impact the performance of intrusion detection models. It can lead to difficulties in capturing meaningful patterns and relationships, resulting in decreased accuracy in detecting intrusions. Normalization and feature scaling techniques are used to mitigate this issue and improve model performance. Therefore, it is necessary to standardize and normalize the data in turn. There are 27 types of attribute instances in gas pipeline data, including 26 types of data features and one type of label attributes. To separate the labels, first assume that the data set can be composed of a matrix T with n rows and m columns, that is

$$T = \begin{bmatrix} B_{11} & B_{12} & \cdots & B_{1m} \\ B_{21} & B_{22} & \cdots & B_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ B_{(n-1)1} & B_{(n-1)2} & \cdots & B_{(n-1)m} \\ B_{n1} & B_{n2} & \cdots & B_{nm} \end{bmatrix} \quad \textbf{(10)}$$

Let Ai be a class of features, Ai can be expressed as:

$$A_i = [B_{1i}, B_{2i}, ..., B_{ni}]^T \quad \textbf{(11)}$$

Then the matrix T can be expressed as:

$$T = \left[A_1, A_2, \cdots, A_m\right] \quad \textbf{(12)}$$

Calculate the standard deviation (Standard Deviation, SD) of each type of characteristic data. If SD≥8, this kind of characteristic data needs to be standardized, let A be the processed data, Ai be the original data to be processed, the normalization process can be expressed as:

$$A_i^{'} = \arctan A_i \quad \textbf{(13)}$$

The data after standardization processing needs to be further normalized. If SD<8, skip the normalization process and directly perform normalization processing, the method is shown in formula (14):

$$A_i^{"} = \frac{A_i' - \min\left\{B_{1i}, B_{2i}, ..., B_{ni}\right\}}{\max\left\{B_{1i}, B_{2i}, ..., B_{ni}\right\} - \min\left\{B_{1i}, B_{2i}, ..., B_{ni}\right\}} \quad \textbf{(14)}$$

3.2.2 Matrix Mapping and Visualization

After standardization and normalization processing, a dataset in the range of [0, 1] is obtained. To construct a suitable input form, use a value to represent a grayscale pixel in a matrix, multiply the value by 255, and fill the 26-bit data features into a 6x6 grayscale matrix. Because the dimension of the matrix is larger than the number of data features, it is necessary to perform zero-padding operation at the end of the matrix. A group is randomly selected from each type of data, and the mapped grayscale matrix is converted into an image, and the picture set shown in Figure 5 is obtained. Among them, the smaller the data value, the closer the corresponding matrix position is to black, and vice versa. It can be seen that there are obvious differences between the pictures mapped by different types of data, while the pictures mapped by the same type of data have a certain similarity. From the visualization results, it is expected that using RCN to learn features can achieve better results.

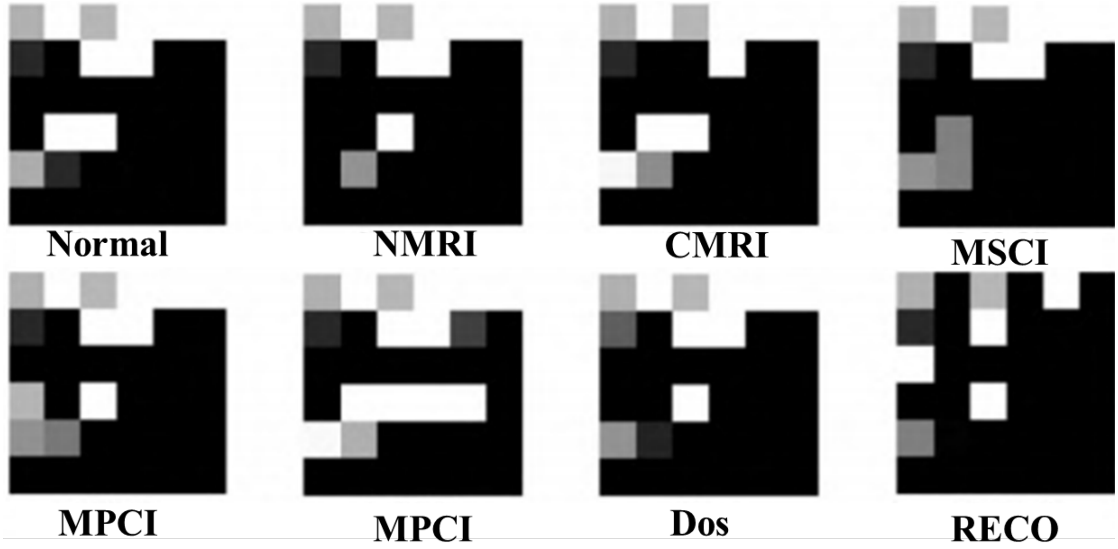


Figure 5. Visual representation of different data types

4 Experiment and Result Analysis

4.1 Experimental Environment and Hyperparameter Settings

In order to simulate the industrial Internet environment, an industrial computer is used to train the network model, and the graphics accelerator card is not used in the training process. The graphics accelerator card is not used in the training process because it is not specifically designed for the type of computations required for neural network training. The hardware and software environment configuration of the experiment is shown in Table 3. Perform parameter combination training, and determine the hyperparameter settings for model training according to the test results as follows: the amount of data selected from the training set data for each iteration training is 256, and the number of dynamic routing iterations is 3.

Table 3. Experimental environment configuration

Category	Parameter
Operating System	Centos7
Processor	IntelCorei7-8550U
Memory	2×4GBDDR42133MHz
Keras	2.2.4
Tensorflow	1.14
Python	3.6.10

4.2 Evaluation Indicators

Typical evaluation indicators of intrusion detection algorithms include Accuracy Rate (Acc), False Negative Rate (FNR) and False Positive Rate (FPR), but due to the serious imbalance of industrial Internet intrusion data, the total number of the vast majority of normal data will skew traditional metrics.

In order to ensure the comprehensiveness of the evaluation, this paper comprehensively adopts the accuracy rate, the false alarm rate, the false alarm rate and the F1 value as the evaluation indicators, where the F1 value is defined by the recall rate (Recall) and the precision rate (Precision). The index can be defined by equations (15) to (20):

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

$$FNR = \frac{FN}{FN + TP} \tag{16}$$

$$FPR = \frac{FP}{TN + FP} \tag{17}$$

$$F_1 = \frac{2Precision \times Recall}{Precision + Recall} \tag{18}$$

$$Recall = \frac{TP}{TP + FN} \tag{19}$$

$$Precision = \frac{TP}{TP + FP} \tag{20}$$

In equations (15) to (20), TP represents the true class, indicating the number of samples that belong to the attack that are correctly predicted as the attack; FN represents the false negative class, indicating the number of false positives for the attack as normal samples; FP represents The false positive class represents the number of samples that are normal samples that are wrongly predicted to be attacked; TN represents the true negative class, which represents the number of normal samples that are accurately predicted to be normal samples.

4.3 Analysis of Experimental Results

In order to evaluate the indicators of RCN in industrial Internet intrusion detection, in addition to Caps Net, this paper selects BiLSTM, GRU, MMN-CNN [9] and traditional machine learning method PSO-SVM in the field of deep learning for comparative experiments. To evaluate model training, BiLSTM, GRU, MMN-CNN use cross-entropy as the loss function, and Caps Net and RCN use the margin loss function. For fairness, all models use preprocessed data and transform the input form accordingly.

4.3.1 Data Preprocessing Analysis

The gas pipeline data set has many outliers. Taking the air pressure attribute as an example, the minimum value is -6.81×1037 and the maximum value is 6.15×1036. If normalization is carried out directly, these outliers with large extreme values will make the rest of the data too concentrated to reflect the relative magnitude of the values. In this paper, the arc tangent function is used to standardize some data in the preprocessing process, which reduces the overall discreteness of the data. Categorical or discrete data is reduced in the preprocessing process due to its overall discreteness.

Figure 6 is a comparison diagram of an NMRI attack data before and after normalization processing. It can be seen that the grayscale features of the visualized images after normalization change significantly, which is conducive to the extraction of depth features. Enhanced discrimination, improved generalization, and Consistent depth representations are advantages of large changes in grayscale features following normalization for extracting depth characteristics. Use RCN and GRU to conduct experiments on the data before and after preprocessing, and the results are shown in Figure 7. It can be seen intuitively that preprocessing can effectively improve the detection effect of intrusion data. Data preprocessing improves detection by cleaning, normalizing, selecting relevant features, and reducing dimensionality. It helps to enhance the quality and relevance of the data used for intrusion detection.

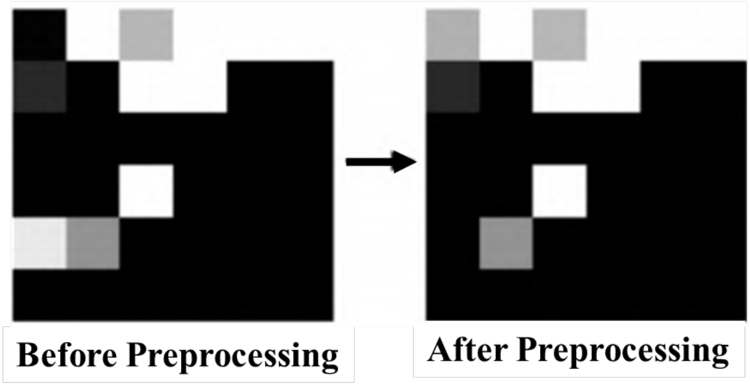


Figure 6. Comparison before and after preprocessing

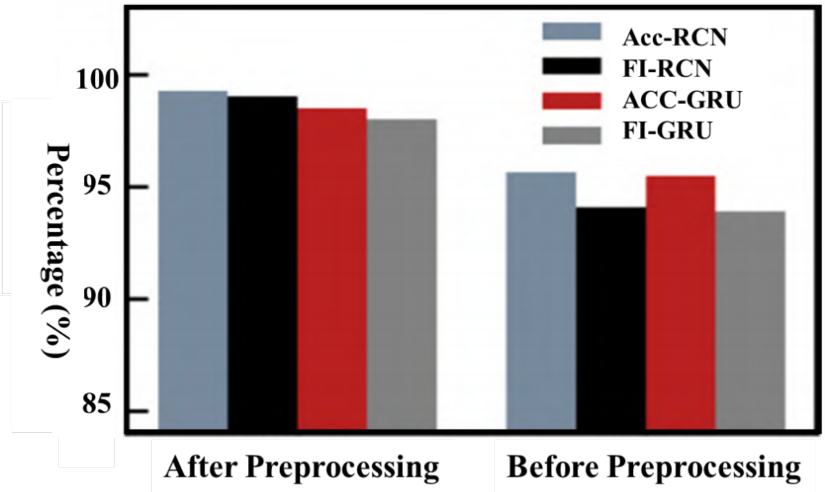


Figure 7. Changes of index values before and after preprocessing

4.3.2 Model Convergence Analysis

The previous section analyzed the effect of data preprocessing. Figure 8 shows the change of the loss value of different models with the number of iterations. Because RCN and CapsNet take the interval loss as the evaluation function, their loss values are compared separately.

From Figure 8, it can be observed that the training loss curve of RCN converges smoothly and quickly. Combined with Figure 9, it can be intuitively seen that all detection models have a certain degree of detection ability, but the method in this paper shows significant advantages in terms of convergence speed and accuracy. BiLSTM and GRU are easy to fall into local optimum, and the curve fluctuates greatly. MMN-CNN cannot process the relative positional relationship of images, and the training accuracy is generally low. MMN-CNN handles the relative positional relationship of images by utilizing multi-scale and multi-resolution features. However, it is limited in processing this information due to the lack of explicit modeling of spatial relationships and contextual information between objects. After introducing the residual network for improvement, it can be seen that the accuracy curve of RCN is more stable than that of CapNet, which shows that the introduction of residual structure has significantly improved the classification quality of capsule layer.

4.3.3 Comparison of Detection Indicators

To compare the predictive ability of each model as a whole, each model was trained and tested for classification. Because the initialization of deep learning weights is random, in order to ensure the reliability of the data, all models are trained multiple times and averaged. The experimental results are shown in Table 4. Compared with other detection models, the four indicators of RCN are the highest, and the F1 value reaches 99.03%, which is 1.03% and 1.48% higher than that of GRU and CapsNet, respectively, which shows that the residual network module significantly improves the quality of capsule layer classification, fully extracting the features of the intrusion data. BiLSTM and GRU can extract time series features, but the false negative rate and false positive rate are relatively high, indicating that more data feature information is lost during the training process. Both MMN-CNN and RCN belong to the convolutional neural network, but the accuracy rate is low. The fully trained model is used for classification test, and the final output is the 8-dimensional confusion matrix shown in Figure 10. The confusion matrix shows the classification results of each type of test sample, in which the underlined numbers indicate the number of correct predictions for each type of data. After sorting out all kinds of data, the detection accuracy of RCN for different attack categories is obtained, as shown in Table 5.

Combined with the F1 score, it can be seen that RCN has achieved excellent detection results for various types of attacks with a small amount of sample data without data enhancement, which indicates that RCN reasonably aggregates image features, effectively reduces the adverse effects of imbalanced data, and has strong generalization ability. The effect of imbalanced data includes biased model performance, low minority class detection, decreased overall accuracy, and difficulty in capturing rare events or classes.

Although RCN can achieve the expected detection effect, there is still room for improvement due to the complex dynamic routing algorithm. The main challenges associated with the complex dynamic routing algorithm in RCN are computational complexity, scalability, real-time adaptability, convergence and stability, robustness, quality of service (QoS), and the trade-off between exploration and exploitation. After testing, in the algorithm of this paper, the dynamic routing algorithm consumes 41% of the total training time, which limits the recognition efficiency of the RCN model.

Table 4. Detection results under different algorithms

Model	Accuracy/%	False negative rate/%
CapsNet	98.18	1.84
BiLSTM	98.30	1.56
GRU	98.51	1.17
MMN-CNN	98.48	1.11
PSO-SVM	96.66	5.42
RCN	99.28	1.08

Table 5. Detection accuracy of different attack categories

NMRI	CMRI	MSCI	MPCI	MFCI	Dos	RECO
91.0%	100%	95.8%	98.7%	91.1%	98.8%	100%

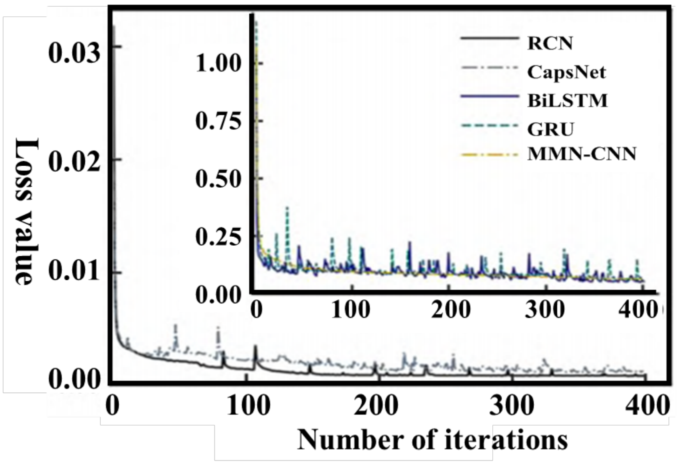


Figure 8. The curve of the loss value as a function of the number of iterations

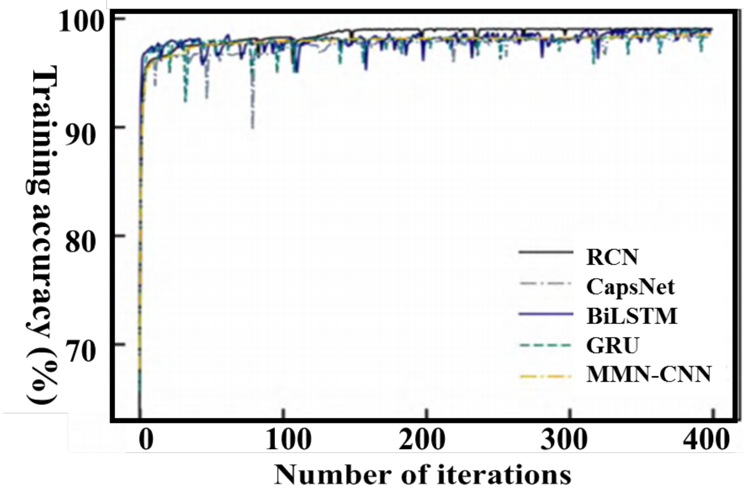
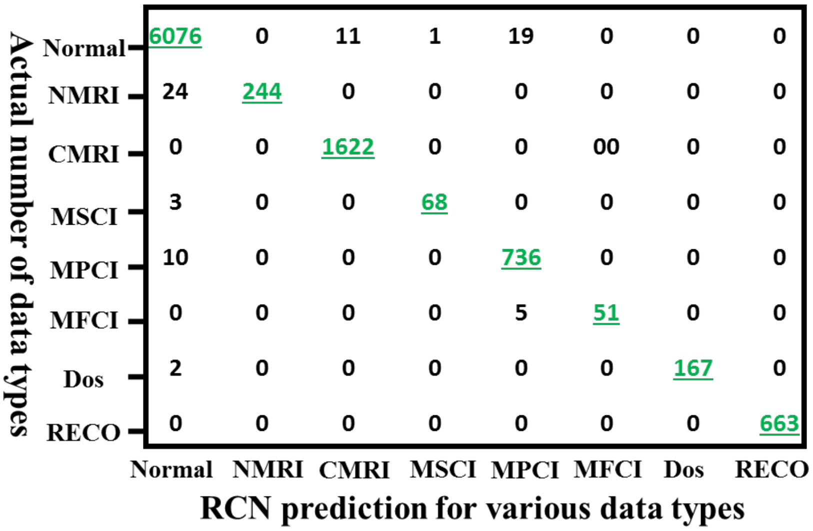


Figure 9. The curve of training accuracy with the number of iterations



	Normal	NMRI	CMRI	MSCI	MPCI	MFCI	Dos	RECO
Normal	6076	0	11	1	19	0	0	0
NMRI	24	244	0	0	0	0	0	0
CMRI	0	0	1622	0	0	00	0	0
MSCI	3	0	0	68	0	0	0	0
MPCI	10	0	0	0	736	0	0	0
MFCI	0	0	0	0	5	51	0	0
Dos	2	0	0	0	0	0	167	0
RECO	0	0	0	0	0	0	0	663

Figure 10. Confusion matrix

4.3.4 Runtime Analysis

The running time referred to in this article includes two parts: model training time and model prediction time. The running time of the model is related to the complexity of the model and the number of training iterations. The running time of each network model is recorded through experiments, and the results are shown in Table 6.

Table 6. Comparison of running time of different models

Kind of data	Time/s		
	Train	Predict	Sum
CapsNet	1446.57	0.79	1447.36
BiLSTM	391.67	0.42	392.09
GRU	296.52	0.37	296.89
MMN-CNN	161.80	0.23	162.03
PSO-SVM	1.12	0.07	1.19
RCN	517.11	0.56	517.67

The training time difference between RCN and CapsNet mainly comes from the parameters of the dynamic routing algorithm [23]. CapsNet directly feeds large-dimensional data into the main capsule layer, which greatly increases the training time. GRU has a 2-gate structure, which reduces the amount of parameters relative to BiLSTM. MMN-CNN has the fastest detection speed among the listed deep learning methods [24]. These three networks have shorter running times, but combined with Table 4, the overall detection effect is worse than that of RCN. Due to its simple structure, PSO-SVM has a low execution time cost. However, its detection index is the worst, the preprocessing process requires manual feature screening [25], and the running time is not included, which is difficult to meet the development trend of intrusion detection system intelligence.

5 Conclusions

Networking mode, low accuracy, and massive unbalanced data, this paper proposes an industrial Internet intrusion detection method based on capsule networks. This method first refers to the DRN structure, introduces the residual block as the main capsule layer to extract high-quality feature maps, then uses the dynamic routing algorithm to cluster the features, and uses the Adam algorithm to optimize the learning rate in backpropagation to make the detection model stable and fast. Convergence, and achieve a detection accuracy of 99.28% in the simulation test of the gas pipeline dataset. Even when the data distribution is seriously unbalanced, the experimental test data results show that the false negative rate, false positive rate and F1 value of the model can still reach 1.08%, 0.51% and 99.03%, which has great advantages over other comparison algorithms. It can better adapt to the industrial Internet application environment. The intrusion detection method proposed in this paper is based on the characteristics of the industrial Internet. The model is based on the supervised learning model, which requires clear packet information and traffic patterns. Although good results can be achieved, the traditional Internet cannot fully meet these conditions. At the same time, the dynamic routing algorithm the complexity is high and the calculation cost is high. The next research direction is to optimize the dynamic routing algorithm strategy to reduce the time consumption of dynamic routing.

Acknowledgement

Scientific Research Program of Chongqing Municipal Education Commission (KJ1602201); Joint Fund of MOE and China Mobile (MCM20150202).

References

[1] L. S. Dalenogare, G. B. Benitez, N. F. Ayala, A. G. Frank, The expected contribution of Industry 4.0 technologies for industrial performance, *International Journal of Production Economics*, Vol. 204, pp. 383-394, October, 2018.

[2] T. Alladi, V. Chamola, S. Zeadally, Industrial control systems: Cyberattack trends and countermeasures, *Computer Communications*, Vol. 155, pp. 1-8, April, 2020.

[3] J. D. Markovic-Petrovic, M. D. Stojanovic, S. V. B. Rakas, A fuzzy AHP approach for security risk assessment in SCADA networks, *Advances in Electrical and Computer Engineering*, Vol. 19, No. 3, pp. 69-74, August, 2019.

[4] M. Zolanvari, M. A. Teixeira, L. Gupta, K. M. Khan, R. Jain, Machine learning-based network vulnerability analysis of industrial Internet of Things, *IEEE Internet of Things Journal*, Vol. 6, No. 4, pp. 6822-6834, August, 2019.

[5] D. Tong, Y. R. Qu, V. K. Prasanna, Accelerating decision tree based traffic classification on FPGA and multicore platforms, *IEEE Transactions on Parallel and Distributed Systems*, Vol. 28, No. 11, pp. 3046-3059, November, 2017.

[6] T. Alves, R. Das, T. Morris, Embedding encryption and machine learning intrusion prevention systems on programmable logic controllers, *IEEE Embedded Systems Letters*, Vol. 10, No. 3, pp. 99-102, September, 2018.

[7] Z. T. Jiang, T. S. Z. Zhou, L. Han, Nearest neighbor intrusion detection method based on perceived hash matrix, *Acta Electronica Sinica*, Vol. 47, No. 7, pp. 1538-1546, July, 2019.

[8] S. Potluri, N. F. Henry, C. Diedrich, Evaluation of hybrid deep learning techniques for ensuring security in networked control systems, *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Limassol, Cyprus, 2017, pp. 1-8.

[9] A.A.S. Shaikh, M.S. Bhargavi, C.P. Kumar, An optimised Darknet traffic detection system using modified locally connected CNN-BiLSTM network, *International Journal of Ad Hoc and Ubiquitous Computing*, Vol. 43, No. 2, pp. 87-96, June, 2023.

[10] H. S. Chen, J. J. Chen, Recurrent neural networks based wireless network intrusion detection and classification model construction and optimization, *Journal of Electronics and Information Technology*, Vol. 41, No. 6, pp. 1427-1433, June, 2019.

[11] Y. B. He, G. J. Mendis, J. Wei, Real-time detection of false data injection attacks in smart grid: A deep learning-based intelligent mechanism, *IEEE Transactions on Smart Grid*, Vol. 8, No. 5, pp. 2505-2516, September, 2017.

[12] S. Sabour, N. Frosst, G. E. Hinton, Dynamic routing between capsules, *Proceeding of Neural Information 31st Conference on Neural Information Processing Systems (NIPS2017)*, Long Beach, California, USA, 2017, pp. 3856-3866.

[13] F. Deng, S. L. Pu, X. H. Chen, Y. Shi, T. Yuan, S. Pu, Hyperspectral image classification with capsule network using limited training samples, *Sensors*, Vol. 18, No. 9, Article No. 3153, September, 2018.

[14] K. M. He, X. Y. Zhang, S. Q. Ren, J. Sun, Deep residual learning for image recognition, *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016, pp. 770-778.

[15] Y. X. Zhao, T. Wu, Y. Han, Nearest identifying the correctness of fit of internal components based on a convolutional neural network, *Acta Electronica Sinica*, Vol. 46, No. 8, pp. 1983-1988, August, 2018.

[16] X. Y. Dong, J. S. Huang, Y. Yang, S. Yan, More is less: A more complicated network with less inference complexity, *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, 2017, pp. 5840-5848.

[17] J. Q. Li, F. R. Yu, G. Deng, C. Luo, Z. Ming, Q. Yan, Industrial Internet: A survey on the enabling technologies, applications, and challenges, *IEEE Communications Surveys & Tutorials*, Vol. 19, No. 3, pp. 1504-1526, Third quarter, 2017.

[18] Q. F. Wei, B. W. Lv, X. D. Hu, A lightweight intrusion detection method for the Internet of things based on sparse LSSVM, *Journal of Chongqing University of Posts and Telecommunications (Natural Science Edition)*, Vol. 33, No. 3, pp. 475-481, June, 2021.

[19] M. Mantere, M. Sailio, S. Noponen, Network traffic features for anomaly detection in specific industrial control system network, *Future Internet*, Vol. 5, No. 4, pp. 460-473, December, 2013.

[20] T. Morris, W. Gao, Industrial control system traffic data sets for intrusion detection research, in: J. Butts, S. Shenoi (Eds.), *Critical Infrastructure Protection VIII. ICCIP 2014. IFIP Advances in Information and Communication Technology, Springer*, Springer, Berlin, Heidelberg, 2014, pp. 65-78.

[21] M. Tavallaee, E. Bagheri, W. Lu, A. A. Ghorbani, A detailed analysis of the KDD CUP 99 data set, *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, Ottawa, ON, Canada, 2009, pp. 1-6.

[22] B. Krawczyk, Learning from imbalanced data: Open challenges and future directions, *Progress in Artificial Intelligence*, Vol. 5, No. 4, pp. 221-232, November, 2016.

[23] J. Zhang, Y. Cao, G. Han, X. Fu, Deep neural network-based underwater OFDM receiver, *IET communications*, Vol. 13, No. 13, pp. 1998-2002, August, 2019.

[24] M. Roopak, G. Y. Tian, J. Chambers, Multi-objective-based feature selection for DDoS attack detection in IoT networks, *IET Networks*, Vol. 9, No. 3, pp. 120-127, May, 2020.

[25] Y. Cao, Optimisation of classification algorithm of associated data features of large-scale network system, International *Journal of Internet Protocol Technology*, Vol. 13, No. 2, pp. 55-60, April, 2020.

Biographies

Yuhong Wu Female, born in 1981, associate professor, now a doctoral candidate. In recent years, her main research directions include industrial control system security, artificial intelligence technology, intelligent intrusion detection technology, big data, etc.

Xiangdong Hu male, born in 1971, is a professor. He has published more than 60 SCI, EI and Chinese core papers, and presided over a number of national patents, His main research interests are intelligent perception, network measurement and industrial information security.