Intelligent Automation & Soft Computing
DOI:10.32604/iasc.2021.016966
Article

Tech Science Press

Deep Learning Anomaly Detection Based on Hierarchical Status-Connection Features in Networked Control Systems

Jianming Zhao[1,2,3,4], Peng Zeng[1,2,3,4,*], Chunyu Chen[1,2,3,4], Zhiwei Dong[5] and Jongho Han[6]

[1]State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, 110016, China
[2]Key Laboratory of Networked Control Systems, Chinese Academy of Sciences, Shenyang 110016, China
[3]Institutes for Robotics and Intelligent Manufacturing, Chinese Academy of Sciences, Shenyang, 110016, China
[4]University of Chinese Academy of Sciences, Beijing, 100049, China
[5]State Grid Liaoning Electric Power Company Limited Electric Power Research Institute, Shenyang, 110016, China
[6]Korea Intelligent Automotive Parts Promotion Institute, Daegu, 43011, Korea
*Corresponding Author: Peng Zeng. Email: zp@sia.cn
Received: 16 January 2021; Accepted: 11 March 2021

Abstract: As networked control systems continue to be widely used in large-scale industrial productions, industrial cyber-attacks have become an inevitable problem that can cause serious damage to critical infrastructures. In practice, industrial intrusion detection has been widely acknowledged to detect abnormal communication behaviors. However, unlike traditional IT systems, networked control systems have their own communication characteristics due to specific industrial communication protocols. Thus, simple cyber-attack modeling is inadequate and impractical for high-efficiency intrusion detection because the characteristics of network control systems are less considered. Based on the status information and transmission connection in industrial communication data payloads, which can properly express the characteristics of industrial control logic, this paper associates industrial communication features with transmission connection payload and status payload. Furthermore, transmission connection features include device address, context, time, and packet length, while status features cover measurement, input, distributed state, control state, and more. After designing a convolutional neural network (CNN) and a long short-term memory network (LSTM) to extract status features and transmission connection features from industrial communication data, this paper proposes a hierarchical deep learning anomaly detection approach, which can integrate the advantages of CNN and LSTM to achieve high-efficiency detection. The experimental results clearly show that the proposed approach, having the advantages of strong detection capability and low false alarm rate, is a superior means of anomaly detection when compared to its peers.

Keywords: Deep learning anomaly detection; networked control system; CNN; LSTM

This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1 Introduction

With the rapid development of information and communication technologies, networked control systems have been widely used in various large-scale industrial productions, such as chemical synthesis, machinery manufacturing, and electric power generation. Automation and intelligence have been realized in control processes [1,2]. However, they still suffer from security issues, which mainly come from traditional IT systems. For example, a malicious hacker may steal or tamper with the process plan to damage the industrial control logic. As a result, severe economic losses and social consequences will be incurred [3,4]. Anomaly detection systems often stand on the opposite side of cyberattacks in offensive and defensive games, and they can prevent attackers from achieving their goals [5]. However, the intrusion detection system not only needs to detect the various cyberattacks of malicious attackers but also needs to protect the correct control logic, which is a crucial component in industrial productions. Therefore, anomaly detection systems for networked control systems should pay much more attention to the logic status information of communication data and the time or connection of the communication payload [6,7].

Compared with traditional IT systems, the characteristics of communication in networked control systems can be mainly classified into two aspects: the transmission connection characteristic and the status characteristic. To be more precise, industrial communication data must be preprocessed according to these two characteristics in order to design an outstanding and feasible anomaly detection approach in networked control systems. To better elucidate the relationship between transmission connection and status information, a novel analytical concept of deep learning to extract the feature from an industrial communication payload is introduced first. In this concept, long short-term memory (LSTM) can better analyze the transmission connection relationship [8], and a convolutional neural network (CNN) can better analyze the status information relationship from the communication payload. After that, a hierarchical deep learning anomaly detection approach is proposed, which integrates the advantages of CNN and LSTM. Finally, the efficiency and effectiveness analysis shows that our proposed method has a better performance in comparison to other models.

The rest of this paper is organized as follows: Section 2 provides an overview on related work. Section 3 describes the proposed deep learning anomaly detection approach, including dataset description, data preprocessing, and model designing. Section 4 presents the experimental results and discussions. Finally, Section 5 gives a conclusion of this paper.

2 Related Work

2.1 Anomaly Detection Techniques

Anomaly detection, an important part of network security protection architecture, identifies various malicious attack behaviors by analyzing the network traffic or key node data [9]. The main idea of anomaly detection techniques in networked control systems is to build a normal model of hierarchical data payload features and identify abnormal behaviors by comparing them with similar features of the industrial communication payload [10]. The networked control system is established with an industrial protocol that connects programmable logic controller (PLC), remote terminal unit (RTU), and human-machine-interface (HMI). One outstanding challenge in the field is how to completely and appropriately summarize industrial communication behaviors according to the specific communication characteristics.

According to the work in Wan et al. [11], existing anomaly detection approaches on transaction payload features mainly include three categories: statistical-based cases, knowledge-based cases, and machine learning-based cases. Specifically, the statistical-based and knowledge-based anomaly detection approaches mainly identify some unknown attacks by building the regular network traffic profile and utilizing a knowledge-based expert system [12]. However, it is difficult to define a high-quality model without deeply investigating the deep industrial transaction behaviors or transmission connection features.

Alternatively, the machine learning approach can be regarded as an efficient and effective measure since it can build an excellent model, which can reflect the transmission connection features and status features of the transaction payload. Furthermore, deep learning techniques have become a better choice to learn the inherent regular pattern and representation level of sample data, and they have achieved remarkable results in the fields of traditional network anomaly detection.

2.2 Deep Learning Techniques

At present, deep learning techniques are widely used in the industry: in the medical, industrial production, operation, and maintenance environments. Furthermore, deep learning techniques are used in academia: in the image, timing, and interpretable modeling environments. There are many methods, and each has its own advantages. In particular, CNNs and recurrent neural networks (RNNs), which can learn temporal and spatial features effectively, are the most commonly used models in deep learning techniques.

In common neural networks, the sum of each node in the first layer is weighted on the common neural network, and the initial value of the last layer can be regarded as a representation or function to learn the neural network from the input data [13]. In practice, a CNN has the ability of representation learning by improving the architecture of the common neural network. It can classify the input information according to hierarchical structure, and the input layer of a CNN can process multi-dimensional data. CNNs are able to learn spatial features and have achieved impressive results in many machine learning tasks [14,15]. Furthermore, CNNs have a good analysis effect on the status information of the communication payload, and many recent research results demonstrate its great potential. Reference Benkhelifa et al. [16] proposes a deep learning method based on Hybrid MLP/CNN (Multi-layer Perceptron/Chaotic Neural Network) neural network for anomaly intrusion detection. This method offers a better detection rate and a lower false alarm rate when detecting novel attacks. Reference Ponomarev et al. [17] generates Denial-of-Service (DoS) attack traffic with normal traffic that cannot be distinguished by some ordinary detection algorithms, and proposes DoS attack detection based on a CNN synthesizing the attack traces payload to improve the attack detection accuracy.

RNN is a fine algorithm that can process sequences of different lengths by using self-feedback, and it can be devoted to process time or connection series samples. Moreover, each layer in an RNN not only connects to the next layer but also outputs a hidden state for the current layer when processing next samples [18]. In practice, LSTM is used to solve the problem of the explosion and disappearance of the RNN gradient [19]. LSTM is a predictive operation model, which can carry transmission features and can use a recursive method based on a time-series back propagation predictive operation model. Additionally, it can combine real data for convergence to improve detection accuracy and detect delaying attacks. Reference Khan et al. [20] proposes that many known signatures from the attack traffic remain unidentifiable, and designs a scalable hybrid IDS (intrusion detection system) based on the convolutional-LSTM network to identify network misuses. Reference Amar et al. [21] proposes a weighted long short-term memory (WLSTM) algorithm to solve malicious behaviors in cloud computing database under high-dimensional and high-speed analysis requirements. More specifically, WLSTM realizes the attack detection of contextual malicious behaviors by considering past events, and minimizes the vanishing gradient.

In brief, CNNs can directly and comprehensively identify various state features in networked control systems and learn the existing features of each state to utilize corresponding disposal methods for different abnormal states. LSTM can effectively learn connection features from a long sequence and predict the operating state of each variable at a future time. By using the special structural features of LSTM network, it is possible to mine the data association degree between different features in networked control systems.

This paper combines the advantages of CNN and LSTM in their respective fields and constructs a deep learning anomaly detection model for the abnormal behavior, abnormal state, and abnormal transmission mode of network control systems. We propose an anomaly detection method based on deep learning technology. The effectiveness of the method is verified by training and testing the actual data set.

3 Hierarchical Deep Learning Anomaly Detection Solution

In the production control process of networked control systems, network attacks may not only cause abnormal network traffic but also cause abnormal control logics. Based on those abnormalities, we propose a deep learning anomaly detection model, which integrates the advantages of CNN and LSTM by using hierarchical status-connection features of industrial communication data. Furthermore, this model specifically includes data preprocessing, extraction of status features by CNN, extraction of transmission connection features by LSTM, and the fully-connected LSTM to realize deep learning anomaly detection in networked control systems. The main framework of the proposed model is shown in Fig. 1. The detailed process can be described as follows.

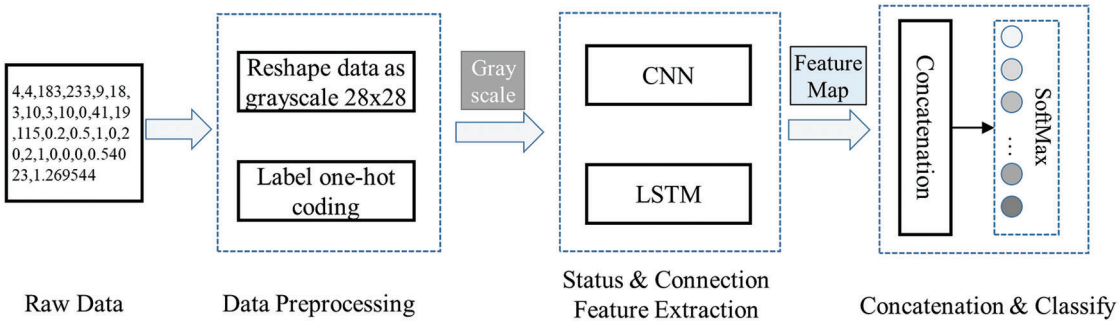

Figure 1: Main framework of hierarchical deep learning anomaly detection model

1) Data preprocessing stage: Guarantees that the training data format meets the standard format requirements of CNN and LSTM. Ensures that a deep learning model can be properly trained.

2) Status feature extraction stage: Further extracts and processes the status features from the communication payload data by creating a CNN neural network.

3) Connection feature extraction stage: Further extracts and processes the connection features from the communication data by generating a LSTM recurrent neural network [22]. This network can learn from the feature representation and models the time dependence automatically.

4) Concatenation & Classify: Classifies various communication behaviors in networked control systems by using a SoftMax layer, which takes the outputs of CNN and LSTM as inputs.

3.1 Data Preprocessing

The analyzed datasets are standard industrial control intrusion detection datasets established by the MSU Infrastructure Protection Center in 2014 [23]. Moreover, the original datasets are collected from the internal SCADA laboratory of Mississippi State University, using the MODBUS application layer protocol to achieve industrial control communication. Additionally, these datasets are generated by attacks on the natural gas pipeline system, which uses the MODBUS protocol for industrial control communication. Compared with the datasets used in IT intrusion detection, such as KDD-99 and NSL-KDD [24,25], these datasets mainly have two types of characteristics: connection payload characteristics and status payload characteristics. The connection payload characteristics describe the communication mode in the SCADA

system and can be used to extract transmission patterns for malicious industrial activity detection. The status payload characteristics describe the business status in the SCADA system, and they can be utilized to detect network attacks, which cause abnormal behaviors of some critical devices, such as programmable logic controllers and motion controllers.

In the data preprocessing step, we convert each piece of network payload into a matrix and encode all labels with one-hot encoding. First, in order to generate a suitable matrix, which can be processed by CNN and LSTM from the original data, we can expand each data line in the datasets with a random value following a random normal distribution. Consequently, the feature vector can be converted into an appropriate two-dimensional matrix. The main algorithm for converting data into a two-dimensional matrix is shown in Algorithm 1.

Algorithm 1: Matrix generation

Input: The original data set D, the number of rows m, and the number of column n;
Output: The target matrix D';
1: function $GenMatrix(D, m, n)$
2: Generate a zero padding matrix D' of $m \times n$
3: **for** $i \times n + j \leq |D'|$ **do**
4: **if** $i \times n + j \leq |D|$ **then**
5: $D'[i,j] \leftarrow D[i \times n + j]$
6: **else**
7: $D'[i,j] \leftarrow x$, where x belongs to the random normal distribution
8: **End if**
9: **End for**
10: Return D'
End function

Second, for the eight categories of behaviors in the datasets, we can apply one-hot encoding to process each behavior. One-hot encoding has a low computational cost since the independent value of each category of behavior is not too much. Tab. 1 shows the processed results for the datasets after applying one-hot encoding.

Table 1: Results of one-hot code

No.	Label name	One-hot code
1	Normal	00000001
2	NMRI	00000010
3	CMRI	00000100
4	MSCI	00001000
5	MPCI	00010000
6	MFCI	00100000
7	Dos	01000000
8	Reconnaissance	10000000

3.2 Status Feature Extraction

In the next stage of our approach, we use a CNN to learn and extract data features from the two-dimensional matrix, which has been generated in the data preprocessing stage, since the sparse connection between the convolutional layer and the pooling layer in a CNN can hide some unimportant information, reduce network training time, and accelerate network convergence.

To be specific, the proposed CNN in this paper consists of one input layer, three convolution layers of different sizes, two pooling layers, and one global pooling layer. Given the two-dimensional matrix generated by the data preprocessing step, the convolution layer extracts regional information of payload features by use of a sliding window and expresses the information abstractly. Next, the pooling operation will reduce the dimension of features, and the output of global pooling will represent all data vector information and transmit them to the connection layer. The main framework of the proposed CNN in this paper is shown in Fig. 2.

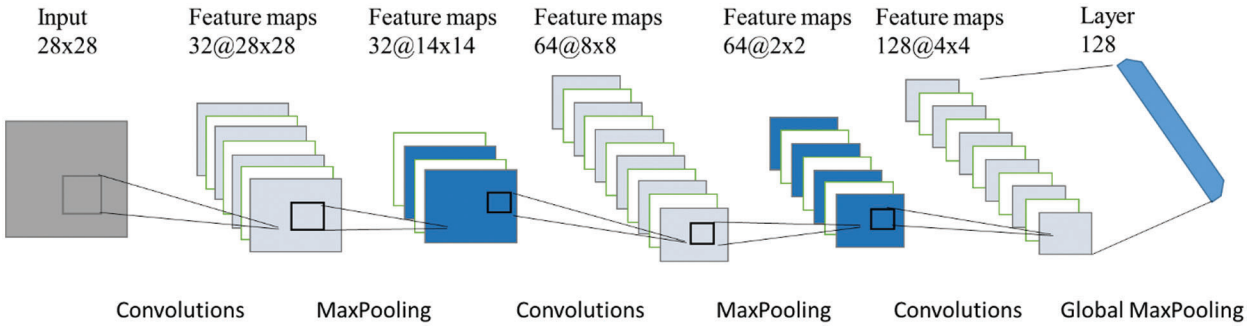

Figure 2: Main Framework of status feature extraction

Convolution calculation: set the convolution kernel weight as w and bias as b, extract the feature map of input data in the form of sliding window, and generate new feature m_i by activating the feature map with nonlinear activation function as follows:

$$m_i = f(w \cdot item_{i:i+s-1} + b), \tag{1}$$

where f denotes the activation function, and $item_{i:i+s-1}$ represents the data in the sliding window.

Pooling operation: the maximum over time polling operation is used to reduce the feature dimension of the generated features, and the maximum value m' is selected as the final data feature to compress the number of parameters and reduce over-fitting.

$$m' = Max(m_i)$$

Activation Function: we use ReLU function as the activation function, which can help us make the model converge faster and keep it continuously stable. A general ReLU function is described as Eq. (3).

$$f(x) = max(0, x) \tag{3}$$

The specific steps involved in the extraction of status features via CNN are as follows; some methods or layers refer to Algorithm 2.

Algorithm 2: Deep learning anomaly detection model

Input: The data matrix D';
Output: Target model for Status-Connection Features of payload;
Step 1. Create CNN model:
Reshape input
for i in range (2) **do**
Create convolution layer named conv_i
Set filter c_i of size s_i
Set activation as ReLUs
Add maxpooling layer of size p_i
End for
Create a dropout layer with rate = 0.25
Create convolution layer named conv_3 with activation ReLU
Add global maxpooling layer, the output of which is a temp vector V_1
Step 2. Create LSTM model:
Create LSTM layer with l_1 units, with dropout d_2
Create a Dense layer with activation ReLU and dropout d_3, the output of Which is a temp vector V_2
Step 3. Concatenate two model:
V_3 = np.concatenate (V_1,V_2)
Create a full connection Dense layer with activation ReLU and dropout d_4
Step 4. Compile and validate model:
Set optimizer as Adam
Set loss as categorical_crossentropy
Step 5. Evaluate model:
Summary target model and Test it by evaluate data sets
Return model

Step 1: We input the 28 × 28 matrix into the first convolution layer, which contains the c_1 filters. Here, the size of filter is $s_1 \times s_2$, and the activation function is set to ReLU.

Step 2: The convolution layer can output the characteristic matrix whose size is 32 × 28 × 28, and we input this matrix into a two-dimensional maxpooling layer whose size is $p_1 \times p_1$.

Step 3: The first maxpooling layer can output 32 matrices whose sizes are 14 × 14 each, and we use zero-padding to extend these matrices, whose final size is 16 × 16.

Step 4: We input these 32 matrices into the second convolution layers containing c_2 filters. Here, the size of filter is $s_2 \times s_2$, and the activation function is set to ReLU.

Step 5: By using the leaky ReLU method, we get 64 matrices whose sizes are 8 × 8 each, and input them into the second two-dimensional maxpooling layer whose size is $p_2 \times p_2$.

Step 6: Through the computation in the second maxpooling layer, we input 64 matrices whose size is 2 × 2 into the dropout layer.

Step 7: The dropout layer outputs 64 matrices whose size is 2 × 2; we use zero-padding to extend these matrices whose final sizes are 4 × 4 each.

Step 8: We input these 64 matrices into the third volume cumulant layer containing c_3 filters. Here, the size of each filter is $s_3 \times s_3$, and the activation function is set to ReLU.

Step 9: By using the leaky ReLU method, we get 128 matrices whose sizes are 4×4 each, and input them into the two-dimensional global maxpooling layer whose size is $p_3 \times p_3$.

Step 10: The status feature vector V_1 is the output of the global maxpooling layer, whose length is 128.

3.3 Connection Feature Extraction

In this step, we use LSTM to extract the transmission connection features from industrial communication data in networked control systems, since it can help us resolve the problems of gradient disappearance and gradient explosion effectively. LSTM introduces the concept of cell status, which determines the reserved and forgotten statuses using the forgetting gate, the input gate, and the output gate in the RNN training process.

The corresponding pseudo-code is shown in Algorithm 2, and the specific steps of the algorithm are listed as follows:

Step 1: We first input the 28×28 matrix into an LSTM layer containing l_1 cells.

Step 2: A vector of length 50 can be computed by the LSTM layer, and we can input this vector into the dropout d_1 layer.

Step 3: A vector of length 50 can be computed by the dropout d_1 layer, and we can input this vector into the dense layer, which also represents the full connection layer.

Step 4: A vector of length 1024 can be computed by the dense layer, and we can input this vector into the dropout d_2 layer. After that, we can get one new vector whose length is 1024, and this vector can be regarded as the connection feature vector V_2.

3.4 Concatenation and Classification

By connecting the status-connection features in the target datasets, which are extracted by CNN and LSTM, we can improve the feature accuracy in the dropout layer setting from the last step. We use SoftMax as the classifier in the final output layer to calculate the possibility of each network payload category. After that, we can further classify different industrial communication behaviors. The main framework integrating CNN and LSTM is shown in Fig. 3, and the detailed steps to realize the integration of CNN and LSTM and the classification of industrial communication behaviors are described below:

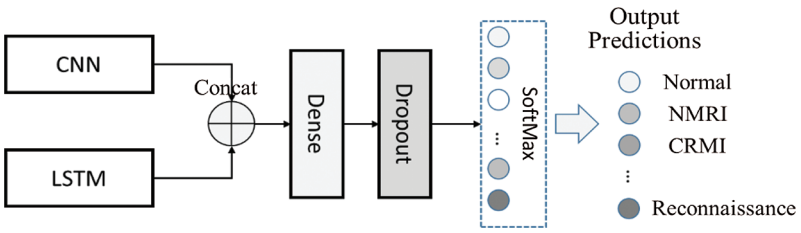

Figure 3: Main framework integrating CNN and LSTM

Step 1: We can get the vector V_3 by connecting the status features V_1 extracted by CNN and the connection features V_2 extracted by LSTM, where $V_3 = V_1 \parallel V_2$.

Step 2: We can get the vector V_4 by inputting the vector V_3 into the full dense connection layer.

Step 3: We perform the classification by inputting the vector V_4 into the activation function SoftMax, and output the classifications of different behaviors.

4 Evaluation and Analysis

To evaluate our deep learning anomaly detection approach, we use the standard intrusion detection datasets and extract the hierarchical status-connection features by analyzing the transmission and payload contents in these datasets. Furthermore, we not only give the performance discussion on the optimal evaluation results but also compare our approach with other existing detection approaches.

4.1 Experimental Environment

We designed a Python program to perform all experiments with our deep learning anomaly detection approach. The basic framework was built by using Keras 2.0 and Tensorflow 2.0. Additionally, all procedures were run on the same PC with 64 GB RAM, Intel Xeon e5-2620v4 2.10 GHz CPU, Nvidia Geforce RTX2080S GPU, and Windows Server 2016 OS.

4.2 Evaluation Indicator

The main purpose of the proposed approach is to accurately detect various abnormal industrial communication behaviors in networked control systems. To this end, we use the authoritative evaluation indicators: AC (accuracy), DR (detection rate), and FAR (false alarm rate) [26]. To be more specific, AC refers to the ratio of all correctly classified samples, which may be normal samples or malware. The DR is used to evaluate the system's performance with respect to its malware traffic detection. The FAR is used to evaluate the misclassifications of normal traffic. The main calculation formulas are listed as follows.

$$\mathrm{AC} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \tag{4}$$

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

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

Here, the true positive (TP) represents the number of attack samples, which are classified into the attack category. The true negative (TN) represents the number of normal samples, which are classified into the normal class. The false positive (FP) represents the number of normal samples, which are classified into the attack category, and the number of attack samples, which are classified into the normal category. The false negative (FN) is the number of samples that are failed to be classified into target category.

4.3 Experimental Parameter Setting

To construct an optimal model, we created different structures of convolution layer and LSTM layer. Based on the two-dimensional matrix generated in the preprocessing process, we performed the experimental tests. Specifically, when the maximum number of training times was set to be 20, the model had stabilized. The accuracy and loss comparison of different structures in the training are shown in Fig. 4.

Through the debugging of multiple experiments, we produced the most effective model, which contains three convolution layers, two pooling layers, one global pooling layer, one LSTM layer, and two full connection layers. Additionally, the SoftMax layer is considered as the final classification function. The settings of all parameters are shown in Tab. 2.

4.4 Comparison and Analysis

Based on the experimental parameter setting in Tab. 2, we were able to obtain an optimal anomaly detection model. When the training epoch was set to 30, the detection accuracy in the training

process was basically stable at 99.50%. The change of loss and detection accuracy in the training process is depicted in Fig. 5.

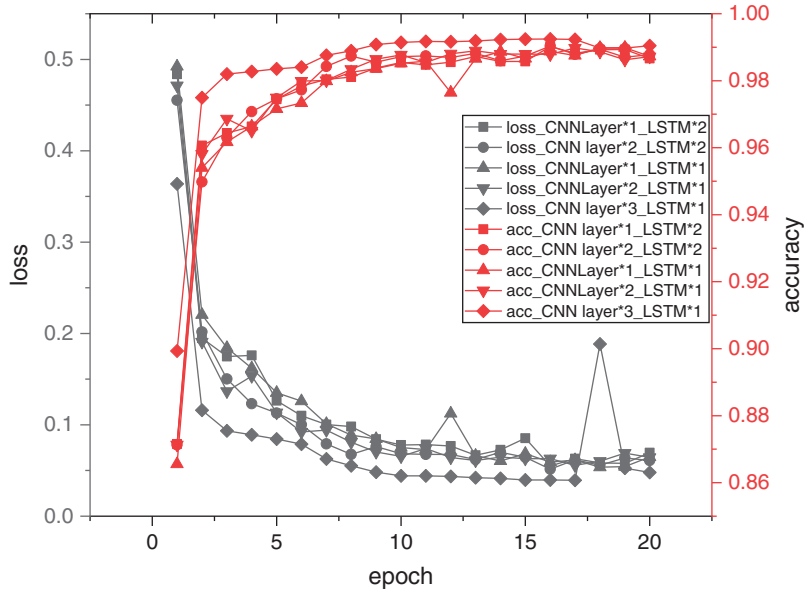

Figure 4: Change of loss and detection accuracy of different structures

Table 2: Experimental parameter setting

Layer	Type	Filters/Neurons	Stride
1	convolution layer	32	1
2	pooling layer	2	2
3	convolution layer	64	(2,2)
4	pooling layer	2	(3,3)
5	convolution layer	128	(1,1)
6	global pooling layer	1	Default
7	LSTM	28	Default
8	full connection	1024	–
9	full connection	1024	–
10	SoftMax	–	–

After we obtained the optimal anomaly detection model, we used the test datasets to evaluate the actual detection performance. The confusion matrix diagram is shown in Fig. 6, which shows the classification of the data sets by our model during the evaluation process.

From the confusion matrix, it can be seen that our proposed model has a high anomaly detection efficiency. Additionally, Tab. 3 shows the evaluation results for each behavior, including Normal, NMRI, CMRI, MSCI, MPCI, MFCI, Dos, and Reconn. AC, DR, and FAR can reach 99.8%, 98.9%, and 0.082%, respectively.

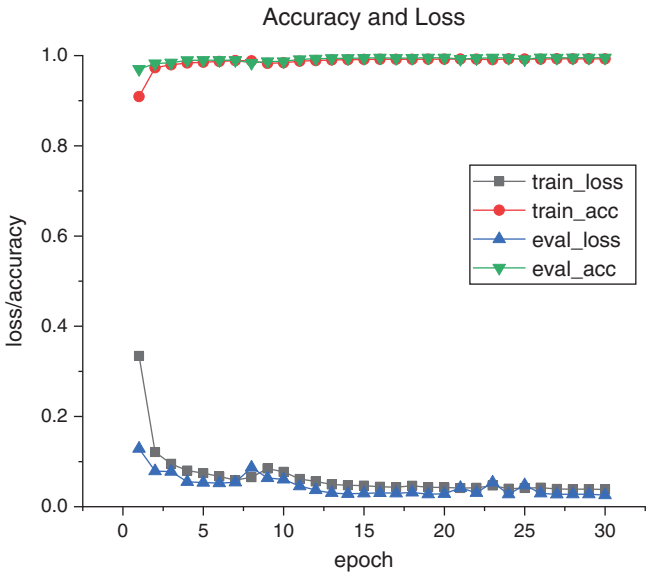

Figure 5: Change of loss and detection accuracy in the training process

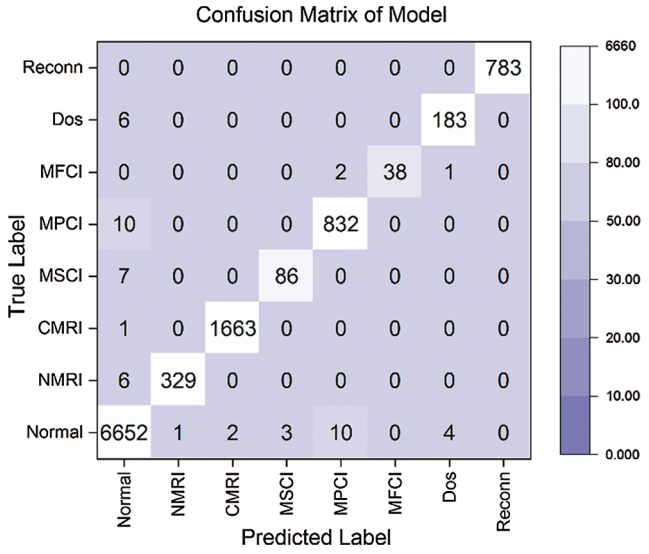

Figure 6: The classification of data sets

To further explain the superiority of our approach, we compare the detection performance with some existing anomaly detection approaches under the same test datasets. Furthermore, in order to better evaluate the superiority of the algorithm, we compare the overall evaluation index of the proposed algorithm with other published algorithms in unified datasets. As shown in Fig. 6, we can see that the classification efficiency of deep learning algorithms such as CNN and LSTM is higher than machine learning methods such as SVM.

Additionally, we construct a two-layer CNN model and apply it to the same datasets, and the accuracy can reach 98.5% [16,27]. Moreover, we also compare with other algorithms: the work in Yu et al. [28] uses the LSTM algorithm to detect the anomaly behavior, and the accuracy can be improved to 96.5%; the work in Liu et al. [29] improves the SVM algorithm to those same datasets, and the accuracy can be improved to

91.81%; the work [30] improves the C4.5 algorithm, and the accuracy can be improved to 91.30%. The accuracy of the different schemes is shown in Fig. 7. Therefore, our proposed approach has a better performance.

Table 3: The results of evaluation experiments

Data type	AC	DR	FAR
Normal	0.995	0.995	0.004
NMRI	0.999	0.996	0.003
CMRI	0.999	0.998	0.0012
MSCI	0.999	0.966	0.0337
MPCI	0.997	0.985	0.014
MFCI	0.997	1	0
Dos	0.998	0.973	0.026
Reconnaissance	1	1	0
Total	0.998	0.989	0.082

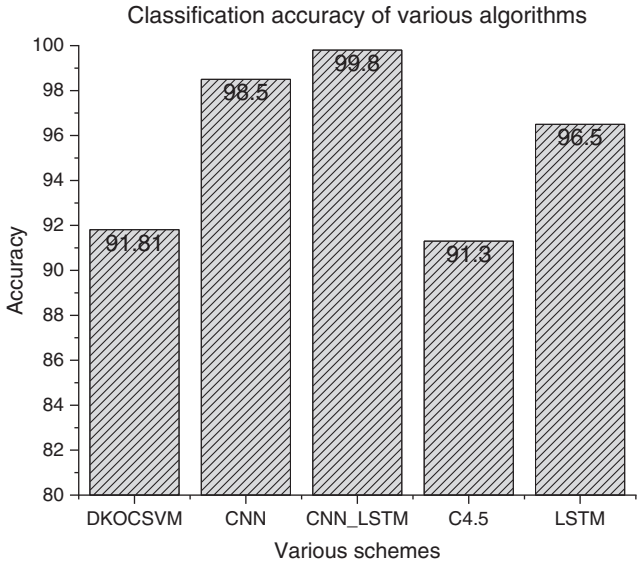

Figure 7: The accuracy of different schemes

5 Conclusion and Future Work

To detect industrial cyberattacks in networked control systems, we propose a deep learning anomaly detection approach based on hierarchical status-connection features. In the view of industrial control logic, the proposed approach generalizes industrial communication data into two types of features: transmission connection features and status features. According to the characteristics of CNN and LSTM, we use a CNN to extract status features and LSTM to extract transmission connection features from industrial communication data in networked control systems. Furthermore, the proposed approach integrates the

advantages of CNN and LSTM to achieve the high-efficiency anomaly detection. Based on the actual datasets, all experimental results show that the proposed approach, which has the advantages of strong detection capability and low false alarm rate, is a more feasible means of anomaly detection by comparing with other anomaly detection algorithms.

In future work, we will not only perform a comprehensive feature extraction from other fields of industrial communication data but also optimize and improve its detection efficiency to realize large-scale applications.

Acknowledgement: The authors gratefully acknowledge the experiment suggestions from lab mates. The authors are also grateful to the anonymous referees for their insightful comments and suggestions.

Funding Statement: This work is supported by the project: "Security Protection Technology of Embedded Components and Control Units in Power System Terminal" (2019GW-12).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] Y. Ge, X. C. Zhang and B. Han, "Complex IoT control system modeling from perspectives of environment perception and information security," *Mobile Networks & Applications*, vol. 22, pp. 683–691, 2017.

[2] Y. Liu, Y. Zhao, K. Li, S. Yu and S. Li, "Design and application research of a digitized intelligent factory in a discrete manufacturing industry," *Intelligent Automation & Soft Computing*, vol. 26, no. 5, pp. 1081–1096, 2020.

[3] D. Ding, Q. Han, Y. Xiang, X. Ge and X. Zhang, "A survey on security control and attack detection for industrial cyber-physical systems," *Neurocomputing*, vol. 275, pp. 1674–1683, 2018.

[4] Z. Baig and S. Zeadally, "Cyber-security risk assessment framework for critical infrastructures," *Intelligent Automation & Soft Computing*, vol. 25, no. 1, pp. 121–129, 2019.

[5] J. Guan, J. Li and Z. Jiang, "The design and implementation of a multidimensional and hierarchical web anomaly detection system," *Intelligent Automation & Soft Computing*, vol. 25, no. 1, pp. 131–141, 2019.

[6] E. Benkhelifa, T. Welsh and W. Hamouda, "A critical review of practices and challenges in intrusion detection systems for IoT: Toward universal and resilient systems," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 3496–3509, 2018.

[7] S. Ponomarev and T. Atkison, "Industrial control system network intrusion detection by telemetry analysis," *IEEE Transactions on Dependable and Secure Computing*, vol. 13, no. 12, pp. 252–260, 2016.

[8] H. Zhu, F. Meng, S. Rho, M. Li, J. Wang *et al.,* "Long short term memory networks based anomaly detection for kpis," *Computers, Materials & Continua*, vol. 61, no. 2, pp. 829–847, 2019.

[9] M. Wan, J. Li, J. Yao, R. Wang and L. Hao, "State-based control feature extraction for effective anomaly detection in process industries," *Computers, Materials & Continua*, vol. 63, no. 3, pp. 1415–1431, 2020.

[10] J. Cui, W. Shang, M. Wan, J. Zhao, W. Yuan *et al.,* "Intrusion detection of industrial control based on semi-supervised clustering strategy," *Information and Control*, vol. 46, no. 4, pp. 462–468, 2017.

[11] M. Wan, W. Shang and P. Zeng, "Double behavior characteristics for one-class classification anomaly detection in networked control systems," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 12, pp. 3011–3023, 2017.

[12] Y. Wang, Y. Cao, L. Zhang, H. Zhang, R. Ohriniuc *et al.,* "Yata: Yet another proposal for traffic analysis and anomaly detection," *Computers, Materials & Continua*, vol. 60, no. 3, pp. 1171–1187, 2019.

[13] T. Li, W. Xu, W. Wang and X. Zhang, "Obstacle detection in a field environment based on a convolutional neural network security," *Enterprise Information Systems*, vol. 5, pp. 1–22, 2020.

[14] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy *et al.,* "Recent advances in convolutional neural networks," *Pattern Recognition*, vol. 77, pp. 354–377, 2018.

[15] R. Chen, L. Pan, Y. Zhou and Q. Lei, "Image retrieval based on deep feature extraction and reduction with improved CNN and PCA," *Journal of Information Hiding and Privacy Protection*, vol. 2, no. 2, pp. 9–18, 2020.

[16] Y. Yu, W. Yang, F. X. Gao and Y. Ge, *Anomaly intrusion detection approach using hybrid MLP/CNN neural network*. IEEE Computer Society, pp. 1095–1102, 2006.

[17] Q. Yan, M. Wang, W. Huang, X. Luo and F. Yu, "Automatically synthesizing DoS attack traces using generative adversarial networks," *International Journal of Machine Learning and Cybernetics*, vol. 10, no. 6, pp. 3387–3396, 2019.

[18] R. Matilda, B. Pete and J. Kevin, "Early-stage malware prediction using recurrent neural networks," *Computers & Security*, vol. 77, pp. 578–594, 2018.

[19] A. Murad and J. Y. Pyun, "Deep recurrent neural networks for human activity recognition," *Sensors*, vol. 17, no. 11, pp. 2556, 2017.

[20] M. A. Khan, M. R. Karim and Y. W. Kim, "A scalable and hybrid intrusion detection system based on the convolutional-LSTM network," *Symmetry*, vol. 11, no. 4, pp. 583, 2019.

[21] M. Amar and B. E. Ouahidi, "Weighted LSTM for intrusion detection and data mining to prevent attacks," *International Journal of Data Mining, Modelling and Management*, vol. 12, no. 3, pp. 308–329, 2020.

[22] X. Hao, J. Zhou, X. Shen and Y. Yang, "A novel intrusion detection algorithm based on long short term memory network," *Journal of Quantum Computing*, vol. 2, no. 2, pp. 97–104, 2020.

[23] T. Morris and W. Gao, "Industrial control system network traffic data sets to facilitate intrusion detection system research," in *Int. Conf. on Critical Infrastructure Protection*, Heidelberg, Berlin, Germany, pp. pp 65–pp 78, 2014.

[24] L. Itzhak, "KDD-99 classifier learning contest LLSoft\'s results overview," *Acm Sigkdd Explorations Newsletter*, vol. 1, no. 2, pp. 67–75, 2010.

[25] A. Kumarshrivas and A. K. Dewangan, "An ensemble model for classification of attacks with feature selection based on KDD99 and NSL-KDD data set," *International Journal of Computer Applications*, vol. 99, no. 15, pp. 8–13, 2014.

[26] W. Wang, Y. Sheng, J. Wang, X. Zeng, X. Ye *et al.,* "HAST-IDS: Learning hierarchical spatial-temporal features using deep neural networks to improve intrusion detection," *IEEE Access*, vol. 6, no. 99, pp. 1792–1806, 2018.

[27] Z. Zhang, X. Zhou, X. Zhang, L. Wang and P. Wang, "A model based on convolutional neural network for online transaction fraud detection," *Security and Communication Networks*, vol. 2018, pp. 1–9, 2018.

[28] B. Yu, H. Wang and B. Yan, "Intrusion detection of industrial control system based on long short term memory," *Information and Control*, vol. 47, no. 1, pp. 54–59, 2018.

[29] W. Liu, J. Qin and H. Qu, "Intrusion detection algorithm of industrial control network based on improved one-class support vector machine," *Journal of Computer Applications*, vol. 38, no. 5, pp. 1360–1365+1371, 2018.

[30] K. M. Sudar and P. Deepalakshmi, "A two level security mechanism to detect a DDoS flooding attack in software-defined networks using entropy-based and C4.5 technique," *Journal of High Speed Networks*, vol. 26, no. 2, pp. 1–22, 2020.