Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 1406214, 8 pages
https://doi.org/10.1155/2022/1406214

Research Article

Intrusion Detection Model for Industrial Internet of Things Based on Improved Autoencoder

Wumei Zhang and Yongzhen Zhang

Zhejiang Tongji Vocational College of Science and Technology, HangZhou, Zhejiang 311231, China

Correspondence should be addressed to Yongzhen Zhang; zhangyongzhen@zjtongji.edu.cn

Received 15 March 2022; Revised 14 April 2022; Accepted 29 April 2022; Published 27 May 2022

Academic Editor: Le Sun

Copyright © 2022 Wumei Zhang and Yongzhen Zhang. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the gradual advancement of informatization and industrialization, the safety and controllability of industrial Internet of things (IoT) have attracted more and more attention. Aiming to improve the security of industrial IoT, a detection method using stacked sparse autoencoder network model is proposed. In this method, the basic units of the network model have been simplified and sparse, and some of basic features are combined with obtaining a higher-level abstract expression, so as to solve the problem of unbalanced network traffic data. The cascaded network structure is adopted to stack its sparse autoencoder network model, so as to improve the data ability of the detection model. In addition, the incorporation of Softmax classifier realizes the dynamic adjustment and optimization of the whole network parameters, which further ensures the efficiency of the detection method. The simulation experiment is based on NSL-KDD dataset. The experiment has proved that the proposed method has excellent network attack identification and detection performance. Its accuracy index is about 95.42%, and the detection time is about 3.42 s.

1. Introduction

The essence of Internet of things (IoT) is the integrated development of industrial automation and interconnection of all things technology [1–3]. The Industrial Internet of things (IIoT) has realized the unprecedented combination of subsystems such as production, monitoring, and management. Different systems can process all kinds of industrial data more efficiently under the unified management of the control center [4, 5]. Its high complexity and openness increase the network security risk faced by the industrial IoT.

Typical network attacks in industrial control systems are common [6]. In July 2010, the first virus "Stuxnet" targeting the Supervisory Control and Data Acquisition (SCADA) system attacked Iran's nuclear facilities. In 2012, the "Flame" virus paralyzed Iran's oil industry network. Since then, the incidents of hacker attacks on industrial control systems have been reported all over the world, and the frequency and impact have shown a rapid upward trend year by year. Industrial control security has become a complex of "network security, equipment security, control security, application security, and data security" [7]. Therefore, it is particularly urgent to propose an accurate and efficient network intrusion detection method.

Intrusion detection system is widely used in traditional industrial control system and modern industrial IoT, and it has attracted more and more attention [8, 9]. In [10], the authors detect attacks on the industrial IoT based on BiLSTM-RNN and use the UNSWNYB15 dataset to train a multilayer neural network. In [11], the authors designed a network intrusion detection system for the SCADA system based on CNN to protect the IIoT from conventional network risk such as DDoS and specific network attacks against SCADA. In [12], the authors studied the power theft attack in the smart grid and proposed a detection method using the multilayer network. However, it should be pointed out that when facing the current high real-time, high-capacity and complex multidimensional data in industrial IoT, the above methods often need a complex training process, and the accuracy needs to be improved [13].

Deep network can not only obtain the maximum reward from the high-dimensional and massive network data

environment but also have the exploration function and automatically mine more valuable information in the network environment [14–16]. Therefore, many scholars have carried out research studies and analyses using deep learning network. In [17], the authors used a context adaptive intrusion detection system, which realizes the accurate detection of network attacks through the mutual assistance of multiple agents. The IIoT detection model in [18] combines feedforward neural network and long-term and short-term memory network. In [19], the authors used an IIoT detection model based on intelligent algorithm and multilayer network, which can achieve better detection efficiency. In [20], the authors proposed a new multiagent confrontation reinforcement learning model for IIoT detection system to realize steady-state support for the network environment. However, it should be noted that the industrial IoT data has unbalanced characteristics. The current deep learning intrusion detection method cannot achieve accurate data feature extraction in the network data with too many feature dimensions, and it is difficult to support efficient and accurate intrusion attack-type mapping. At the same time, due to the deeper network structure, the deep network model also has the problem of time-consuming in intrusion detection.

Aiming at the above problems, based on the improved autoencoder (AE), a detection method for IIoT is proposed. The main innovations are as follows:

(1) In this study, the network structure unit of the multilayer network is sparse. By adding sparsity constraints to the hidden layer, some neurons are suppressed, and the problem of industrial network intrusion detection with unbalanced network traffic data is solved, so as to learn more accurate and efficient feature expression.

(2) The cascade form is used to combine the sparse autoencoder (SAE) network and construct the stacked sparse autoencoder (SSAE) network model, which can realize the continuous deep feature extraction of industrial IoT network data, so as to support the high accuracy of intrusion detection network.

2. Standard Autoencoder Model Learning Algorithm

Industrial control system network dataset presents the characteristics of more normal data, less abnormal data, and uneven data distribution [21]. Algorithms including traditional artificial neural network cannot effectively classify and identify unbalanced data.

AE network is an unsupervised feature detection model, which can learn a feature representation of input data. This model belongs to artificial neural network and is optimized by backpropagation algorithm.

The essence of the algorithm of self-encoder network is an unsupervised training and learning method. In order to make the target value input directly, it introduces the data processing model of backpropagation to maintain the consistency of data.

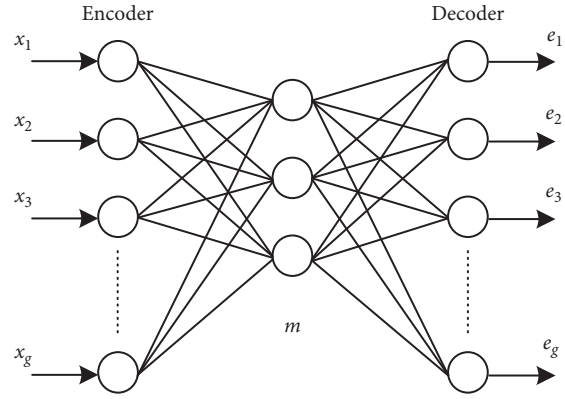

Figure 1: Autocoding network structure.

In addition to being used as the construction module of deep neural network, the AE network can also be used to extract discriminant features with lower dimension than input, so as to solve the dimension disaster.

The standard AE is a multilayer feedforward network, which expects the input and output to be consistent. It can be used to learn identity mapping and extract unsupervised features. Figure 1 is a network structure of a single-layer autoencoder, in which only one hidden layer is used to encode the input and reconstruct the input at the output through decoding. The part from the input layer to the middle layer is called encoder, and the part from the middle layer to the output layer is called decoder. Autoencoder is an unsupervised feature detection model, which can learn another feature representation of input data. Autoencoder learns to generate a hidden layer representation from the input and reconstructs the output as close to the input as possible from the hidden layer representation.

As can be seen from Figure 1, the AE network model is composed of the input layer, the hidden layer, and the output layer. Specifically, the purpose of the self-encoder is to make the output value of the model equal to or as close to the input value of the model as possible with the help of an identity function. $x_t = e_t$.

Encoding refers to the process of mapping input $x \in R$ to implicit representation $h(x) \in R$. The calculation form is

$$h(x) = \alpha_h(Wx + b), \tag{1}$$

where $W \in R$ is the encoding weight matrix, $b \in R$ is the encoding offset vector, $\alpha_h(x)$ is the vector value function, and in the case of nonlinearity, $\alpha_h(x)$ is taken as Sigmoid function.

Decoding refers to mapping the implicit representation $\alpha_h(x)$ to the output layer e, so as to reconstruct the input x. The calculation form is

$$e = \alpha_e\left(Wh(x) + b'\right), \tag{2}$$

where $W' \in R$ presents the decoding matrix, $b' \in R$ presents the decoding vector, and $\alpha_e(x)$ is similar to $\alpha_h(x)$.

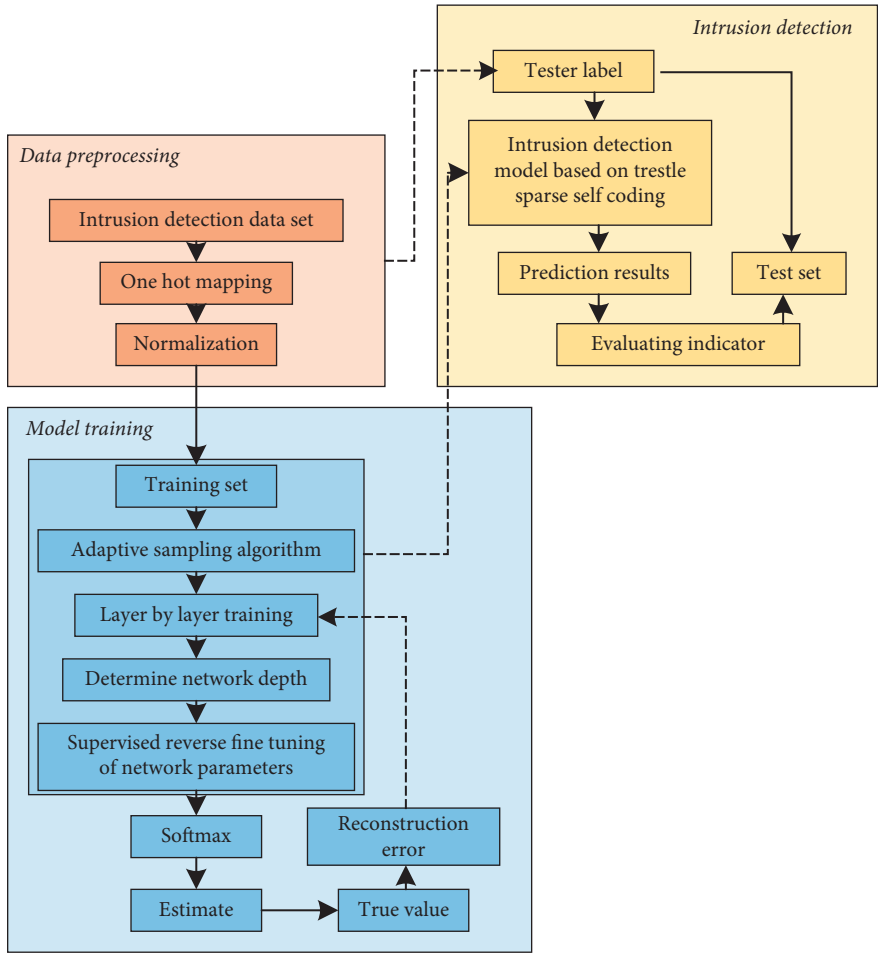

Figure 2: Intrusion detection model for IIoT.

3. Intrusion Detection Model of IIoT

Excessive feature dimension is the reason for the low efficiency of industrial control safety anomaly detection [22, 23]. Dimension reduction can be achieved by reducing high-dimensional and nonlinear attribute features. Through the sparse expression of features, a small number of basic features are combined to obtain a higher-level abstract expression.

Therefore, based on the standard AE network, this study adds sparsity constraints to the output of the hidden layer so that most neurons are suppressed and constructs a atacked sparse autoencoder (SSAE) network model.

The SSAE network is used to establish the intrusion detection model of the IIoT. On the premise of maintaining the accuracy of detection, the calculation speed and calculation memory are improved, so as to learn better feature expression.

3.1. Overall Architecture. The proposed overall architecture is shown in Figure 2.

From Figure 2, the identification of industrial IoT intrusion by this model mainly includes the following three steps:

(1) Data preprocessing: build an industrial IoT environment and capture real-time network data, including source address, target address, connection attributes, and other relevant information [24, 25]. The data are preprocessed and transformed into a format that can be processed by the stacked noise reduction convolutional autoencoder. In this study, data preprocessing is divided into three parts:

① Attribute mapping: convert character data into numerical data

② Data normalization: normalize the data to within 0 to 1 to solve the problem of dimensional inconsistency, which affects the accuracy

③ Regional adaptive oversampling algorithm: generate new samples at the algorithm level for minority samples, handle the imbalance of data distribution properly, and then carry out the next operation to optimize minority data

3.2. Stacked Sparse Autoencoder Network. SAE network suppresses most neurons by adding sparsity constraints to the output of the hidden layer, which can learn better feature expression, so as to solve the problem of industrial network

intrusion detection with unbalanced network traffic data. The specific way is to add a sparse penalty term, that is, the function of the average output activation value of neurons.

The goal of SAE is to make the output fit the input features, which is similar to AE, but SAE imposes sparsity restrictions on the middle layer in order to avoid simple mapping output to input.

The simple understanding of sparsity restriction is that when the output of neuron in each layer is 0, it indicates that the state of neuron is inhibited; when the output of neuron is 1, it indicates that the state of neuron is active, and the sparsity restriction makes the state of neuron inhibited most of the time.

The mean activation degree of hidden layer neuron i is defined as follows:

$$\widehat{\tau}_i = \frac{1}{n}\sum_{p=1}^{n}\left[c_i^{(2)}\left(v^{(i)}\right)\right], \tag{3}$$

where n indicates the total number of data sample sets and $c_i^{(2)}$ is the activation parameter of the middle layer neuron i when v is used as input. To get the sparse representation of the middle layer neuron, it should make the activation mean $\widehat{\tau}_i$ of the middle layer neuron i as 0 as possible. If making $\widehat{\tau}_i = \tau$ as a sparsity parameter, τ should be a decimal close to zero. By introducing a penalty factor into the solution of the objective, those scenarios that $\widehat{\tau}_i$ and τ are significantly different are punished, so as to realize such sparsity limitation and continuously optimize the value of the objective function. There are many ways to construct penalty factors. Here, the Kullback–Leible (KL) is used to regularize the network so that the average activation degree $\widehat{\tau}_i$ is equal to τ as much as possible:

$$KL\left(\tau\|\widehat{\tau}_i\right) = \tau\log\frac{\tau}{\widehat{\tau}_i} + (1-\tau)\log\frac{1-\tau}{1-\widehat{\tau}_i}. \tag{4}$$

The penalty factor formula is as follows:

$$\sum_{i=1}^{z^2}\tau\log\frac{\tau}{\widehat{\tau}_i} + (1-\tau)\log\frac{1-\tau}{1-\widehat{\tau}_i}, \tag{5}$$

where z^2 is the sum of neuron. The above penalty factor can also be expressed as $\sum_{i=1}^{z^2} KL(\tau\|\widehat{\tau}_i)$.

It can be seen that the loss function of the detection network is

$$\theta_{sparse}(W,b) = \theta_E(W,b) + \mu\sum_{i=1}^{z^2} KL\left(\tau\|\widehat{\tau}_i\right). \tag{6}$$

Usually, in order to avoid the overfitting problem, the L_2 weight penalty is introduced to the objective function; then,

$$\theta_{SAE}(W,b) = \theta_{\text{sparse}}(W,b) + \frac{\gamma}{2}\sum_{i=1}^{s_q}\sum_{p=1}^{s_q+1}\sum_{q=1}^{s_q-1}\left(u_{pi}^{(q)}\right)^2, \tag{7}$$

where γ represents the regularization parameter, q represents the current layer, and s_q and s_q+1 are the sum of neurons.

The formula of descent optimization is as follows:

$$W_{pi}^{(q)} = u_{pi}^{(q)} - \psi\frac{\partial}{\partial u_p^{(q)}}\theta_{SAE}(W,b), \tag{8}$$

$$b_{pi}^{(q)} = b_{pi}^{(q)} - \psi\frac{\partial}{\partial b_p^{(q)}}\theta_{SAE}(W,b), \tag{9}$$

where ψ is the learning rate. The optimal W and b can be obtained by back propagation using the SGD optimization method.

The training process of SSAE network is shown in Figure 3.

The first SAE contains layers x, m_1, and $\widehat{x}$, uses formula (6) to learn the representation of features in an unsupervised manner, and then obtains U_1 and c_1 through formulae (7) and (8) training. The second SAE contains layers m_1, m_2, and $\widehat{m}_1$. The training method of the second SAE is similar to that of the first SAE, and U_2 and c_2 are obtained through training. By repeating the above training steps, all the parameters in the stacked sparse autoencoder network can be obtained.

The way of weight assignment of neural network through pretraining is better than that of random weight assignment of neural network, and it is conducive to convergence. In the training process, the number of neurons decreases gradually, and finally, the deep sparse feature is obtained.

3.3. Detection Model Training. Softmax classifier is added in the last layer of SSAE network, and the trained parameters are used as the initial optimization parameters of the model, and then, the parameters of the whole network are fine tuned. This layer-by-layer greedy process is proved to produce a better local extremum than random initialization weights and achieves better generalization performance in some tasks.

The proposed detection model used the SSAE network model is as follows (Algorithm 1).

4. Experiment and Result Discussion

4.1. Simulation Environment. Tensorflow and OpcnAlGym are the mainstream machine learning training platforms and environments. We choose them as the software environment for simulation experiments. Meanwhile, the experimental hardware environment is CPU model: AMD Ryzen 7, CPU: NVIDIA GeForce RTX2080Ti, and RAM: 32 GB.

4.2. Data Preprocessing. At present, the public datasets of industrial IoT intrusion mainly include KDDCup99, NSL-KDD, GasPipeline Datasets, WaterDatasets, and UNSW-NB15. These datasets have the problems of redundancy and repetition of data and attributes. This study selects NSL-KDD dataset as the experimental benchmark data.

NSL-KDD dataset solves the problem of redundant data in KDDCup99 dataset. Its original training set

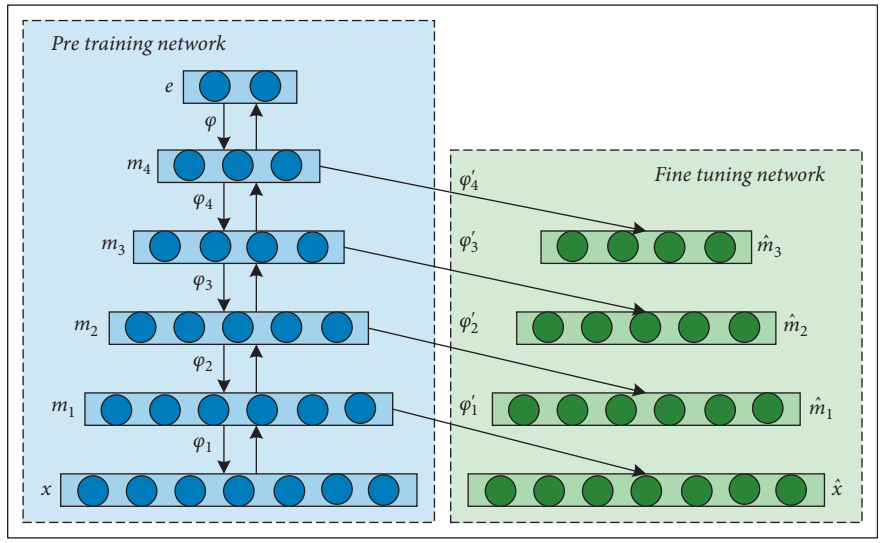

Figure 3: Stacked sparse autoencoder network training process.

Input: 256 dimensional data x after high-dimensional mapping and normalization, data $\hat{x}$ with a certain noise proportion κ.
Output: optimal network parameter values φ_1, φ_2, φ_3, φ_4, and φ_5.
Step 1: the feature extraction model based on SSAE network takes the training data x as the input. Through the SGD descent method, the input data are analyzed and processed to obtain the network parameters of the hidden layer. Finally, the output m_1 of the first hidden layer is calculated by using the original data x and parameters φ_1.
Step 2: then, combined with m_1 and φ_1, the output parameter φ_2 and output m_2 of the hidden layer can be obtained through the calculation and analysis of the second layer.
Step 3: repeat step 1 and step 2, and get the weight parameters φ_1, φ_2, φ_3, and φ_4 by layer-by-layer training. With the help of the calculation and analysis of the classifier, the parameter m_5 is obtained.
Step 4: through the above calculation, we can obtain the network parameter $\varphi_1 - \varphi_5$ of the detection model. By introducing random noise, we input it as training data, calculate the loss function between the predicted value and the target, and use various optimization methods to calculate the parameters near the minimum value.

Algorithm 1: Training algorithm of intrusion detection model based on SSAE network model.

KDDTrain contains 125973 data and the original test set KDDTest contains 22544 data. In this study, KDDTrain+20% of 25192 data are selected as experimental data.

4.2.1. Character-Type Mapping Numeric Type. "O, tcp, ftp_data, SF, 491, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 2, 0, 0, 0, 0, 1, 0, 0, 150, 25, 0.17, 0.03, 0.17, 0, 0, 0, O.OS, O, Normal" is a piece of data in the dataset. According to the analysis, the values in dimension 2, 3, and 4 of the data are character types and need to be converted into numerical types. For example, there are 3 types in dimension 2 (TCP, UDP, ICMP), 70 types in dimension 3 ("auth," "bgp," "courier," etc.), and 11 types in dimension 4 ("0TH," "REJ," "RSTO," etc.), which are processed according to the one-hot coding in Figure 4 and finally convert the 32 dimension into 256 dimension attributes.

4.2.2. Numerical Normalization. Because data order of magnitude and corresponding value range of different feature attributes are obviously different, in order to facilitate the analysis of experimental results, the Min-Max standardization method is used to uniformly map the numerical data to the [0, 1] interval so that the data is in the same order of magnitude:

$$x_{\text{normal}} = \frac{x - x_{\min}}{x_{\max} - x_{\min}}, \tag{10}$$

where x is the original eigenvalue of data, $x_{\min}$ and $x_{\max}$ represents the minimum and maximum values in the data respectively, and x_{normal} represents the new feature value after normalization of each data.

4.2.3. Low-Frequency Sample Processing. Although current industrial IoT attacks show a rapid growth trend, the individual attack categories still belong to the low-frequency category compared with the normal data flow, which makes it difficult to capture their feature records. Moreover, most AI models have obvious classification bias because they aim at the overall classification accuracy of the largest sample. Therefore, this study improves the sampling algorithm and introduces the Regional Adaptive Synthetic Oversampling algorithm (RASmote) to

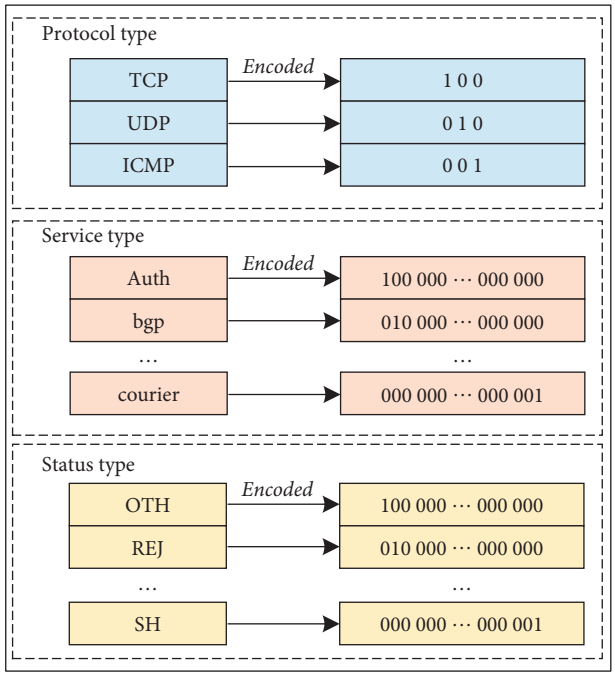

Figure 4: One-hot coding digitization.

incrementally process low-frequency samples. The algorithm formula is as follows:

$$\eta = \left|X_n - X_l\right| = \sqrt{\sum_{k=1} \left(X_{nk} - X_{lk}\right)^2}. \tag{11}$$

Euclidean distance is used to calculate the distance of low-frequency samples in the nearest neighbor radius. n is the nearest neighbor radius, X_n is the nearest neighbor sample set, X_l is the low-frequency sample, and X' is the new sample set:

$$\begin{cases} X' = 0, & 0 \le \eta \le \dfrac{n}{2}, \\ X' = X + \mu(0,1)\left(\left(\dfrac{1}{n-\eta}\sum_{i=1} X_i\right) - X\right), & \dfrac{n}{2} < \eta < n, \\ X' = X, & \eta = n, \end{cases} \tag{12}$$

where $(1/n - \eta\sum_{i=1} X_i)$ is a low-frequency sample.

4.3. Evaluation Index. The performance of the SSAE intrusion detection model can be evaluated from two aspects: model comparison and classification detection. The model comparison is mainly compared with traditional intrusion detection technology. The main indexes of system detection include accuracy Acc, precision Pre, recall Re, and F1-score F_1. It should be noted that, for these four indexes, the higher the value, the better the detection performance:

Table 1: Distribution of dataset.

Data type	Training set	Test set
Normal	9415	4034
Dos	6500	2734
Probe	1603	786
R2L	145	64
U2R	8	3

Table 2: Identification result of different types of network attacks.

Data type	Accuracy (%)
Dos	97.34
Probe	96.81
R2L	91.32
U2R	88.23

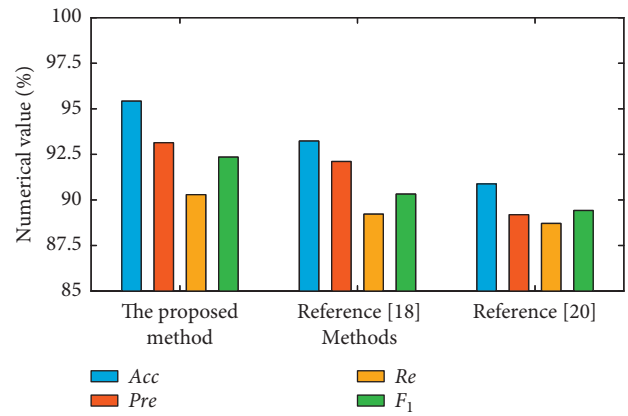

Figure 5: Intrusion detection analysis under different methods.

$$\begin{aligned} \text{Acc} &= \frac{\left|T_P\right| + \left|T_N\right|}{\left|T_P\right| + \left|F_P\right| + \left|T_N\right| + \left|F_N\right|}, \\ \text{Pre} &= \frac{\left|T_P\right|}{\left|T_P\right| + \left|F_P\right|}, \\ \text{Re} &= \frac{\left|T_P\right|}{\left|T_P\right| + \left|F_N\right|}, \\ F_1 &= 2 \times \frac{\text{Pre} \times \text{Re}}{\text{Pre} + \text{Re}}, \end{aligned} \tag{13}$$

where T_N is true negative rate, F_P is false positive rate, F_N is false negative rate, and T_P is true positive rate.

4.4. Experimental Analysis. KDDTrain+20% data are used as the experimental data, 70% as the training set, and 30% as the test set. The data distribution is shown in Table 1.

Firstly, based on the experimental dataset, the detection, analysis, and research of industrial IoT under different network attacks are carried out for the proposed model. The

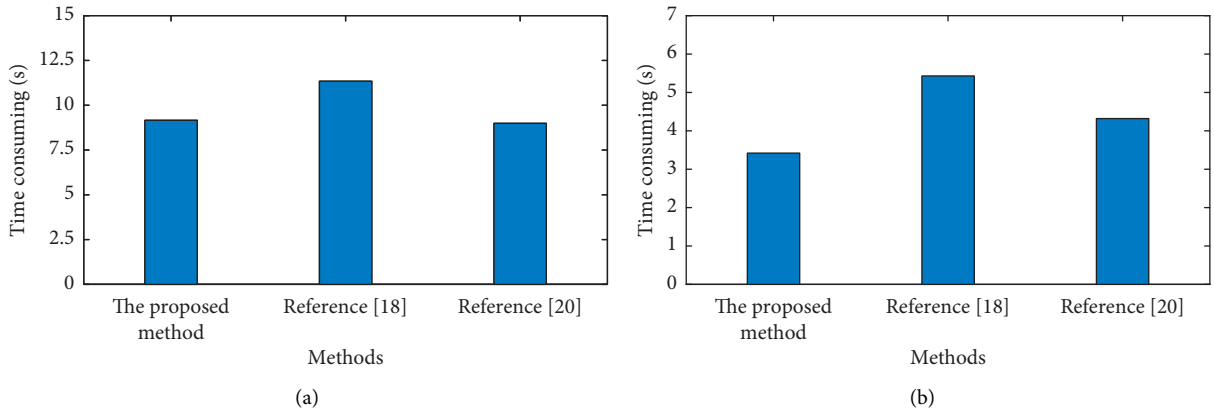

Figure 6: Comparison of intrusion detection time under different methods. (a) Training set time consuming. (b) Test set time consuming.

identification results of attacks are displayed in the under table.

From Table 2, we can see that the proposed model can better complete the task of network defense, and the detection accuracy of Dos and Probe attacks is more than 95%. For R2L and U2R attacks, because of the small volume of training data, the identification accuracy is lower than that of the first two attacks, but it is still more than 85%.

In order to further verify the performance of the proposed model, the authors [18, 20] are used as comparison methods to detect KDDTrain+20% datasets, respectively. Figure 5 shows the attack identification results under different intrusion detection methods.

From Figure 5, we can see that the proposed method is better than other comparison methods in terms of network performance. The evaluation indexes of the proposed method are as follows: the accuracy Acc is 95.42%, the precision Pre is 93.14%, the recall Re is 90.29%, and the F1-value F_1 is 92.35%. The accuracy of intrusion detection in [18, 20] is less than 95%, which is less than the detection performance of the proposed method.

The reason is that the proposed model simplifies the network and enhances the autonomous ability and can realize better feature extraction and expression of the network. Meanwhile, with the introduction of Softmax classifier, the detection network parameters can be dynamically adjusted to support accurate network attack identification and analysis. In [18, 20], LSTM network as the benchmark model is taken for modeling and analysis, without considering the imbalance of data, which is not enough to achieve more accurate and efficient intrusion identification analysis.

At the same time, the attack detection efficiency is also compared and evaluated. Figure 6 shows the analysis of detection time under different methods.

As shown in Figure 6, due to the simplification of the network unit, the unit structure of the proposed method needs more autonomous learning time to realize the accurate extraction of data features. Therefore, the training time is 9.16 s, which is 0.17 s more than that in [20]. Moreover, the time-consuming of the proposed method for network intrusion detection is only 3.42 s and that of [18] is 5.43 s and that of reference [20] is 4.32 s.

To sum up, while ensuring the accuracy of detection, the proposed method can improve the efficiency of intrusion identification and analysis and reflect its overall efficient performance.

5. Conclusion

This study proposes an intrusion detection method based on stacked sparse autoencoder network. This method constructs an intrusion network model based on autoencoder network, which can effectively improve the feature extraction of industrial Internet data. The autoencoder network is simplified and cascaded, and a small number of basic network units are used to obtain more efficient feature expression. In addition, the introduction of Softmax classifier ensures that the parameters of the detection network can be fine-tuned and optimized, which can further improve the processing and computing efficiency of the network while improving the accuracy of industrial IoT attack recognition. The experimental analysis based on NSL-KDD dataset shows that the proposed method can realize accurate and fast intrusion attack identification and can meet the safe and controllable operation requirements of industrial IoT.

Although this method improves the solution of IIoT intrusion detection, the essence of the proposed model is a centralized processing and computing model. Aiming to support the detection research in the actual complex network environment, the next step will be to study the intrusion detection method of distributed architecture mode.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

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

Acknowledgments

This work was supported by Zhejiang Water Conservancy Science and Technology Project (no. RC1974).

References

[1] X. Gming, S. Xiaorui, Z. Zhihua, and X. Bertino, "Advances in Artificial Intelligence and Security," in *Proceedings of the 27th International Conference, ICAIS 2021*, Dublin, Ireland, July 2021.

[2] A. A. Suzen, "Developing a multi-level intrusion detection system using hybrid-DBN," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 2, pp. 1913–1923, 2021.

[3] Z. Wang, Y. Lai, Z. Liu, and J. Liu, "Explaining the attributes of a deep learning based intrusion detection system for industrial control networks," *Sensors*, vol. 20, no. 14, pp. 3817–3824, 2020.

[4] A. Ayodeji, Y.-k. Liu, N. Chao, and L. Q. Yang, "A new perspective towards the development of robust data-driven intrusion detection for industrial control systems," *Nuclear Engineering and Technology*, vol. 52, no. 12, pp. 2687–2698, 2020.

[5] D. Li, L. Deng, M. Lee, and H. Wang, "IoT data feature extraction and intrusion detection system for smart cities based on deep migration learning," *International Journal of Information Management*, vol. 49, no. 1, pp. 533–545, 2019.

[6] A. Khraisat, I. Gondal, P. Vamplew, J. Kamruzzaman, and A. Alazab, "A novel ensemble of hybrid intrusion detection system for detecting Internet of things attacks," *Electronics*, vol. 8, no. 11, pp. 1210–1218, 2019.

[7] L. Lv, W. Wang, Z. Zhang, and X. Liu, "A novel intrusion detection system based on an optimal hybrid kernel extreme learning machine," *Knowledge-Based Systems*, vol. 195, no. 1, pp. 105648–105717, 2020.

[8] I. A Khan, D. C Pi, and Z. U Khan, "HML-DS: a hybrid-multilevel anomaly prediction approach for intrusion detection in SCADA systems," *IEEE Access*, vol. 7, no. 1, pp. 89507–89521, 2019.

[9] B. Li, Y. Wu, J. Song, R. Lu, T. Li, and L. Zhao, "DeepFed: federated deep learning for intrusion detection in industrial cyber-physical systems," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5615–5624, 2021.

[10] B. Roy and H. Cheung, "A deep learning approach for intrusion detection in internet of things using bi-directional long short-term memory recurrent neural network," in *Proceedings of the 28th International Telecommunication Networks and Applications Conference (ITNAC)*, pp. 1–6, Sydney, NSW, Australia, November 2018.

[11] H. Yang, L. Cheng, and M. C. Chuah, "Deep-Learning-Based Network Intrusion Detection for SCADA Systems," in *Proceedings of the 2019 IEEE Conference on Communications and Network Security (CNS)*, pp. 3–5, IEEE, Washington, DC, USA, June 2019.

[12] M. Ismail, M. F. Shaaban, M. Naidu, and E. Serpedin, "Deep learning detection of electricity theft cyber-attacks in renewable distributed generation," *IEEE Transactions on Smart Grid*, vol. 11, no. 4, pp. 3428–3437, 2020.

[13] W. Ding, J. Nayak, B. Naik, D. Pelusi, and M. Mishra, "Fuzzy and real-coded chemical reaction optimization for intrusion detection in industrial big data environment," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 6, pp. 4298–4307, 2021.

[14] I. A. Khan, D. Pi, P. Yue et al., "Efficient behaviour specification and bidirectional gated recurrent units-based intrusion detection method for industrial control systems," *Electronics Letters*, vol. 56, no. 1, pp. 27–30, 2020.

[15] W. Liang, K.-C. Li, J. Long, X. Kui, and A. Y. Zomaya, "An industrial network intrusion detection algorithm based on multifeature data clustering optimization model," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 2063–2071, 2020.

[16] W. Wang, F. Harrou, B. Bouyeddou, S.-M. Senouci, and Y. Sun, "A stacked deep learning approach to cyber-attacks detection in industrial systems: application to power system and gas pipeline systems," *Cluster Computing*, vol. 25, no. 1, pp. 561–578, 2022.

[17] K. Sethi, E. Sai Rupesh, R. Kumar, P. Bera, and Y. Venu Madhav, "A context-aware robust intrusion detection system: a reinforcement learning-based approach," *International Journal of Information Security*, vol. 19, no. 6, pp. 657–678, 2020.

[18] I. Almomani, "A multi-layer classification approach for intrusion detection in IoT networks based on deep learning," *Sensors*, vol. 21, no. 9, pp. 1–21, 2021.

[19] Y. Zhang, P. Li, and X. Wang, "Intrusion detection for IoT based on improved genetic algorithm and deep belief network," *IEEE Access*, vol. 7, no. 1, pp. 31711–31722, 2019.

[20] A. Chu, Y. Lai, and J. Liu, "Industrial control intrusion detection approach based on multiclassification GoogLeNet-LSTM model," *Security and Communication Networks*, vol. 2019, no. 1, pp. 1–11, Article ID 6757685, 2019.

[21] J. Y. Song, R. Paul, J. H. Yun, H. C. Kim, and Y. J. Choi, "CNN-based anomaly detection for packet payloads of industrial control system," *International Journal of Sensor Networks*, vol. 36, no. 1, pp. 36–49, 2021.

[22] M. T. R. Laskar, J. X. Huang, V. Smetana et al., "Extending isolation forest for anomaly detection in big data via K-means," *ACM Transactions on Cyber-Physical Systems*, vol. 5, no. 4, pp. 1–26, 2021.

[23] S. Mubarak, M. Hadi Habaebi, M. Rafiqul Islam, F. Diyana Abdul Rahman, and M. Tahir, "Anomaly detection in ICS Datasets with machine learning algorithms," *Computer Systems Science and Engineering*, vol. 37, no. 1, pp. 33–46, 2021.

[24] T. Vaiyapuri, Z. Sbai, and H. Alaskar, "Deep learning approaches for intrusion detection in IIoT networks – opportunities and future directions," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 4, pp. 86–92, 2021.

[25] X. Zhou, Y. Hu, W. Liang, J. Ma, and Q. Jin, "Variational LSTM enhanced anomaly detection for industrial big data," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 5, pp. 3469–3477, 2021.