TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 04/13 pp821–832
DOI: 10.26599/TST.2020.9010041
Volume 26, Number 6, December 2021

Anomaly Detection of Industrial Control Systems Based on Transfer Learning

Weiping Wang*, Zhaorong Wang, Zhanfan Zhou, Haixia Deng, Weiliang Zhao, Chunyang Wang, and Yongzhen Guo*

Abstract: Industrial Control Systems (ICSs) are the lifeline of a country. Therefore, the anomaly detection of ICS traffic is an important endeavor. This paper proposes a model based on a deep residual Convolution Neural Network (CNN) to prevent gradient explosion or gradient disappearance and guarantee accuracy. The developed methodology addresses two limitations: most traditional machine learning methods can only detect known network attacks and deep learning algorithms require a long time to train. The utilization of transfer learning under the modification of the existing residual CNN structure guarantees the detection of unknown attacks. One-dimensional ICS flow data are converted into two-dimensional grayscale images to take full advantage of the features of CNN. Results show that the proposed method achieves a high score and solves the time problem associated with deep learning model training. The model can give reliable predictions for unknown or differently distributed abnormal data through short-term training. Thus, the proposed model ensures the safety of ICSs and verifies the feasibility of transfer learning for ICS anomaly detection.

Key words: anomaly detection; transfer learning; deep learning; Industrial Control System (ICS)

1 Introduction

Modern Industrial Control Systems (ICSs) have higher production efficiency than traditional industrial systems and can well process big data.

However, increases in the type and frequency of network attacks and hacking incidents threaten the security of ICSs based on data transmission. The National Institute of Standards and Technology has proposed the main sources of security issues for modern ICSs[1], which include nonsecure communication protocols, poor network isolation and access controls[2], and the lack of an ICS anomaly detection system[3]. Intrusion detection technology is an important research direction in the field of network security. The original flows of network equipment and servers have been comprehensively analyzed[4]. When industrial control networks are invaded or traffic data are abnormal, intrusion detection technology can effectively predict and take active defensive measures in a timely manner. Deep learning has shown great research significance

• Weiping Wang and Chunyang Wang are with School of Computer and Communication Engineering, the Beijing Key Laboratory of Knowledge Engineering for Materials Science, and the Institute of Artificial Intelligence, University of Science and Technology Beijing, Beijing 100083, China, and with Shunde Graduate School, University of Science and Technology Beijing, Guangzhou 528399, China. E-mail: weipingwangjt@ustb.edu.cn; chunyang.wang.china@gmail.com.
• Zhaorong Wang is with School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing 100083, China. E-mail: 41804349@xs.ustb.edu.cn.
• Zhanfan Zhou and Weiliang Zhao are with School of Mechanical Engineering, University of Science and Technology Beijing, Beijing 100083, China. E-mail: 41804237@xs.ustb.edu.cn; 41804277@xs.ustb.edu.cn.
• Haixia Deng is with the Donlinks School of Economics and Management, University of Science and Technology Beijing, Beijing 100083, China. E-mail: 41804351@xs.ustb.edu.cn.
• Yongzhen Guo is with School of Automation, Beijing Institute of Technology, Beijing 100081, and also with China Software Testing Center, Beijing 100048, China. E-mail: yzguo@cstc.org.cn.
* To whom correspondence should be addressed.
Manuscript received: 2020-09-02; accepted: 2020-09-23

in intrusion detection technology. Feature values are extracted through a great amount of data training, parameters are constantly changed, and a system that can identify abnormal traffic data is constructed.

Deep learning and traditional machine learning show certain similarities. The core aim of traditional machine learning is to map features to the target space. In traditional machine learning algorithms, the recognition rate increases with increasing data size; however, because a bottleneck period is often encountered during processing, these models cannot handle massive amounts of data. Machine learning performs well in intrusion detection in closed environments. However, machine learning will be exposed when entering an open-world scenario with various random traffic or noise, which could adversely affect its availability[4]. Therefore, traditional machine learning algorithms are unsuitable for detecting abnormal traffic in ICSs, and finding abnormal data quickly and implementing active measures with high accuracy are quite challenging.

Compared with traditional machine learning, deep learning has a strong generalizability for extracting high-dimensional data. Deep learning uses back-propagation algorithms to change and adjust parameters continuously to achieve optimal results. This learning method can handle large amounts of data; indeed, the larger the data size, the better the resulting effect. Unfortunately, although deep learning has good generalizability in processing images, it relies on labeled data and cannot handle unknown abnormal data types[5]. In this article, we solve some of the problems of traditional machine learning by using a residual Convolution Neural Network (CNN) structure to model the source dataset and modify the relevant parameters by transfer learning. We then apply the transfer learning algorithm using the relevant information of the source domain and predicting the target domain[6]. Transfer learning is finally employed to train the model quickly and detect differently distributed or unknown datasets.

ICS flow data can usually be processed with one-dimensional data sequences through preprocessing; in this work, however, we use mapping to convert ICS flow data to an image format suitable for CNNs to take full advantage of the features of the latter. Fine-tuning is utilized during transfer learning to ensure timeliness. After building an eight-layer residual neural network, only the three deepest layers of the neural network are fine-tuned.

The residual structure, which effectively prevents gradient explosion or gradient disappearance while ensuring the depth of the model effect, is introduced. The detection ability of the model in unknown domain datasets is excellent. After training the source domain, that is the KDDCUP99 dataset, the model is used on a gas pipeline dataset through transfer learning. The model shows good anomaly detection effects on the gas pipeline dataset[7], and its precision, recall, and F1-score are fairly high.

This article is organized as follows. Section 2 introduces the background of this study and the related work. Section 3 describes the method. Section 4 introduces the evaluation index. Section 5 describes the experimental process and results. Section 6 provides the conclusions and directions for future work.

2 Background and Related Work

2.1 Research status of industrial control system anomaly detection

Given rapid developments in informatization and industrialization, ICS has been widely used in national infrastructures. However, platform hardware and software vulnerabilities and the openness of the network environment render ICSs vulnerable to security attacks. Therefore, ICS anomaly detection is very important. Anomaly detection has a significant effect on the active defense process of ICSs[8]. The anomaly detection approaches of ICSs mainly include three types, namely, knowledge-based, statistics-based, and machine learning-based. Reference [9] proposed an anomaly detection method based on state recognition to detect attacks in ICSs by using a data-driven clustering method to identify the normal and critical states of a system. A statistical model for traffic detection in the time domain has also been introduced to detect network anomalies and evaluate the performance of the method in different scenarios[10]. Results show that the model can detect network anomalies in all scenarios faster than other methods. Considering their consequences, network attacks aimed at ICSs are very serious. More importantly, they are difficult to detect[11]. In the context of industrial control environments, anomaly detection based on machine learning does well in improving the accuracy of finding abnormal behavior and is of great importance in the establishment of efficient and intelligent intrusion detection models[8].

2.2 Anomaly detection based on machine learning

2.2.1 Traditional machine learning algorithms

Decision tree, random forest, Support Vector Machine (SVM), and logistic regression are traditional machine learning methods. An earlier study used an intelligent Markov model based on statistical learning to establish a multimodel intrusion detection system for industrial process automation that could effectively detect actual attack operations[12]. Other researchers used SVM to establish a data detection model utilized in an industrial control communication protocol[13].

Decision tree is a machine learning method with a tree structure and high efficiency. It is easy to understand and highly effective for processing discrete data. SVM is based on the principle of structural risk minimization. In this method, the optimal classification is found by learning the classification model of data samples in the feature space.

2.2.2 Deep learning model

Traditional machine learning methods have a number of disadvantages, such as low efficiency in processing large-scale data and inability to solve samples with uneven distributions. Compared with traditional machine learning methods, deep learning models have more complex architectures and multiple layers. The most important advantages of deep learning over traditional machine learning are that it can learn features directly and automatically from the original data and has good performance[14].

Deep learning models are quite effective in the field of detecting industrial process anomalies. Almalawi et al.[15] proposed two novel techniques that are an automatic identification of inconsistent states of SCADA data and an automatic extraction of proximity detection rules from identified states. Gao[16] developed an anomaly-based intrusion detection system for the SCADA network and found a combined Intrusion Detection System (IDS) which includes signature-based IDS and anomaly-based IDS. These studies show that deep learning methods have good performance in anomaly detection and attack classification.

CNNs are suitable for image classification[17]. Compared with other image recognition algorithms, CNN uses not only deep learning methods but also some special structures for feedforward neural networks and has relatively little data to preprocess.

2.2.3 Transfer learning based on model fine-tuning

Transfer learning is an effective approach to exploit deep neural networks on small datasets. The essence of transfer learning is to transfer and reuse knowledge in other fields. Mathematically, transfer learning includes two concepts, namely, domain and learning task. Model-based transfer learning methods are usually combined with deep learning models to transfer the structure and parameters of models that have been trained on large-scale datasets (e.g., AlexNet, VGGNet, and ResNet) to new tasks and use the weights trained on the large dataset as the initial weights for the new task[18]. In contrast to deep learning, transfer learning can detect unknown information. Fine-tuning pretrained CNNs on images is an effective strategy to achieve transfer learning. This technique is widely used in the field of image recognition and provides new insights into the recognition of small-scale datasets. In transfer learning, the deep network structure is trained on a large natural image dataset, after which the model is transferred to a small dataset by fine-tuning its parameters. The features extracted from the pretrained deep neural network are universal and applicable to other datasets. Figure 1 shows a schematic of the model.

2.2.4 Fine-tune based on residual neural network

Kaiming He, a researcher at Microsoft Research Asia, designed residual neural networks with a deeper network structure and a simpler network structure[19]. The residual network consists of multiple residual blocks, and each residual block comprises a convolutional layer and a pooling layer. The blocks of the convolutional layer are skipped by using shortcut connections. The use of identity shortcuts requires the same input and output sizes[20]. In this case, the problem of model attenuation caused by the disappearance of gradients is avoided by the superposition of gradients. This deep residual neural network won five championships in two major technical competitions, namely, ImageNet and MS COCO. A unique feature of this network is that it includes a network depth greater than 152 layers, which

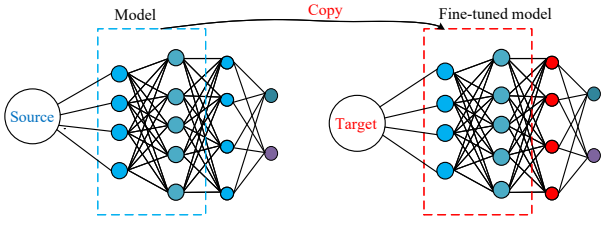


Fig. 1 Transfer learning by model fine-tuning.

had never been achieved before. In previous experiments, the gradient disappeared as the number of network layers increased and the error rate of such a network was higher than that of a neural network with a lower number of layers. The emergence of the residual neural network solves these issues well. Experiments showed that higher accuracy could be obtained as the number of network layers trained by the residual network increased, which means the residual network can allow deeper network layer training, and the performance of the model was greatly enhanced[21].

Because traditional machine learning algorithms have poor generalizability, deep learning algorithms present greater time costs and may be prone to gradient disappearance or explosion. Although fine-tuning technology based on deep neural networks can solve the problem of high time cost and detect unknown attacks, model pretraining remains subject to gradient disappearance or explosion during deep learning. Fine-tuning based on a deep residual neural network can solve these problems simultaneously.

3 Method

In this section, we will describe the processing flow in detail. Because of the correlations among the features of industrial control flow data, we convert the one-dimensional data stream into a two-dimensional matrix and then convert this matrix into a Mahalanobis Distance (MD) matrix. The obtained matrix is converted into one-dimensional data, saved, normalized, and then mapped into a black-and-white image. After building an eight-layer CNN with a residual structure, we use the KDDCUP99 dataset for pretraining and then train the obtained model on the gas pipeline dataset by fine-tuning through the transfer learning method.

3.1 Data preprocessing

This experiment uses a deep migration learning model and the KDDCUP99 dataset as the source domain to perform migration learning on the gas pipeline dataset. Because the data may be disturbed by noise, missing values, and inconsistent data, the presence of low-quality data is inevitable. We improve the credibility of the data by preprocessing them and then improve the performance of model recognition. A preliminary exploration of the data reveals the presence of attributes with exactly the same feature values in the training data; these attributes are not beneficial to the establishment of the model and affect its construction[8]. Therefore, the redundant features are removed. Some normalization methods are used to process the training data, improve the convergence speed of deep learning, and complete the task of anomaly detection.

3.1.1 Target domain: Gas pipeline dataset

The steps are as follows:

(1) Data cleaning: A large amount of abnormal data will greatly affect the normalized results by affecting the data distribution. Data cleaning is used to clean duplicates, erroneous data, and useless features, thus improving the reliability and integrity of the data as well as the accuracy of the analysis results. The gas pipeline dataset includes some negative and unreasonably large values of key measurement data, which means cleaning is necessary. Next, the attributes "commandlength", "commwritefun", "reset", "gain", "deadband", "cycletime", "rate", and "crcrate" are redundant attributes in the dataset[22]. We delete these attributes from the dataset because they interfere with data classification.

(2) Feature mapping: This experiment uses MD to perform feature mapping on the data. MD was proposed by Mahalanobis[23] as a distance measurement method and refers to the covariance distance of the data. In contrast to the Euclidean distance, MD ignores differences in measurement units and considers the relationship between features, thus aligning the relationship between features with the actual situation[24]. Therefore, MD is not affected by the measurement scale and the interference of correlations between variables can be eliminated. Figure 2 shows the pseudo code for feature mapping of data using MD.

3.1.2 Source domain: KDDCUP99 dataset

We standardize the source domain, that is the KDDCUP99 dataset. Standardized data are subtracted from the mean and then divided by the variance (or standard deviation). When this data standardization

Main code of feature mapping method based on MD

```
Input: industrial control network data stream
Output: transformed feature matrix
Do while X_i:
Diag = convert_to_diag(X_i)
# The function is responsible for converting the data stream into a diagonal
  matrix, and the mapped matrix is Map_Matrix=map_matrix(diag)
# The function is responsible for converting the diagonal matrix into a matrix
  Save_matrix(Map_Matrix)
# The function saves the transformed matrix
X_i=X_{i+1}
End while
```

Fig. 2 Pseudo code for feature mapping of data using the MD.

method is completed, the data are converted into a standard normal distribution. In general, the standard deviation is 1 and the mean is 0. The conversion function is in the following:

$$x = \frac{x - \mu\sigma}{x} \tag{1}$$

where x is the data, μ is the mean, and σ is the variance.

Standardizing the dataset can accelerate the search for optimal solutions. Standardization is conducive to process initialization, avoids numerical problems when updating the gradient value, and helps adjust the learning rate. It can also ensure that small values in the output data are utilized.

3.2 Data visualization

3.2.1 Gas pipeline dataset

One-dimensional industrial flow data are transformed into a two-dimensional matrix via the feature-mapping method. This section introduces the feature matrix visualization method employed in this article. In this paper, every element in the MapMatrix matrix is regarded as a pixel, and the element value corresponds to the gray value of the pixel.

3.2.2 KDDCUP99 dataset

The 41-dimensional feature samples are converted into 8-bit-depth grayscale images measuring 7 pixel × 7 pixel in size, the pixels number is from 0 to 255, and each feature corresponds to a pixel.

3.3 Eight-layer residual convolution neural network

The existing residual CNN (layers $\leqslant$ 34) is shown in Fig. 3.

Although the residual neural network increases the accuracy of predicting labels as the network deepens[18], it also leads to a longer training time, which is unfavorable for anomaly detection in ICS. Compared with colored pictures, the data flow has fewer features and does not require a deep network structure for feature extraction. Thus, we constructed an eight-layer CNN with a residual structure.

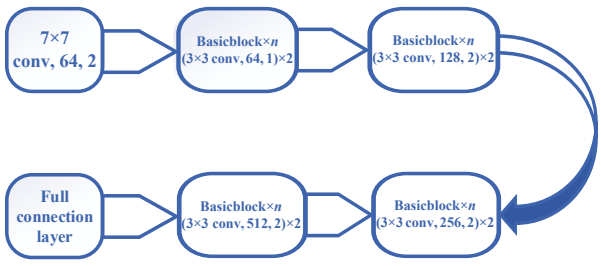


Fig. 3 Original residual convolutional neural network model structure.

3.3.1 Input layer

After data visualization, the data stream from the KDDCUP99 dataset is processed into a 7 × 7 grayscale image, and the data stream from the gas pipeline dataset is processed into an 18 × 18 grayscale image. These two input sizes are relatively small. The stride of the input layer is set to 1, the kernel size is set to 3 × 3, and the number of input channels is set to 3 (the algorithm automatically converts the grayscale image into the RGB model) to utilize the data completely.

3.3.2 Residual blocks

Three residual blocks are utilized in the model. Each residual module is composed of two weight layers and two Relu activation functions. The weight layer is composed of a convolutional layer and a batch-normalization layer.

The batch-normalization layer transforms the input value distribution of any neuron in each layer of neural network into a standard normal distribution via a certain normalization method. Therefore, the batch-normalization layer prevents the model from gradient vanishing and greatly accelerates its training speed[25].

The Relu function performs a nonlinear transformation on the input. The input is not a linear combination of the outputs of the previous layer but can be approximated to any function, thus ensuring the significance of the deep neural network[26].

The size of the kernel of the convolutional layer is 3 × 3, and the stride is 1. As the network deepens, the number of kernels varies from 64 to 256. A schematic of each residual module is shown in Fig. 4. The equation of the module is in the following:

$$\boldsymbol{y} = F(\boldsymbol{x}) + \boldsymbol{x} \tag{2}$$

where $\boldsymbol{x}$ is the input matrix and $F(\boldsymbol{x})$ is the output after the two-layer convolution operation. $\boldsymbol{y}$ is the input of the next residual module.

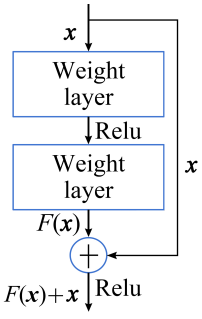


Fig. 4 Residual block structure.

3.3.3 Pooling layer

Commonly used pooling operations include maximum down-sampling, average down-sampling, and spatial down-sampling. Down-sampling is used in CNN to reduce model parameters. Among these pooling operations, maximum down-sampling has been proven to have the best information retention capability.

Because the number of data features in ICSs is small, we only add a maximum pooling layer prior to the fully connected layer to reduce information loss. The stride is 2, the kernel size is 3×3, and the number of in-channels is 256.

3.3.4 Fully connected layer

The fully connected layer is implemented by using a linear transformation function, which acts as a classifier for the entire neural network. Assuming that the output image size of the previous layer is $M \times N$, the number of kernels is K. Because we are studying a two-class problem, the fully connected layer transforms the $M \times N \times K$-dimensional data into two-dimensional data, that is the predicted probability of each label. The algorithm outputs the predicted label by finding the greatest possibility of the label being obtained.

Figure 5 shows the structure of the entire model.

3.4 Fine-tuning

Fine-tuning is performed according to the neural network. As the network deepens, the extracted features become more abstract. For two similar domains, the previous layer for extracting common features can be retained after source domain training, and the target domain only needs to train the deepest several layers of the network.

In this study, because the KDDCUP99 and gas pipeline datasets are anomaly detection datasets with fixed-dimensional data features and certain correlations between features, we can use transfer learning on the basis of the data features of the datasets described above. The KDDCUP99 dataset has a sufficient sample number and contains over 490 000 datasets, while the gas pipeline dataset is relatively small and contains only over 90 000 datasets. Employing the premise initially introduced in this subsection, we use the KDDCUP99 dataset to pretrain the model completely, retain the parameters of the convolutional layer that could extract low-dimensional features, and then train and adjust the last three layers of the neural network through fine-tuning.

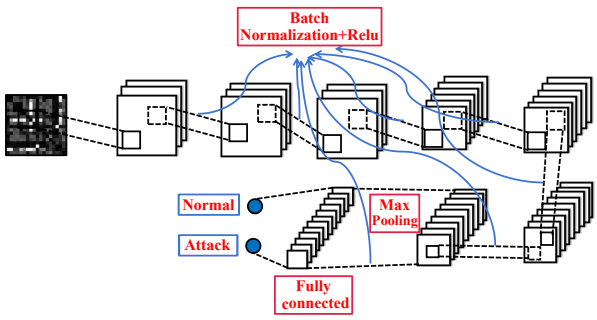


Fig. 5 Model structure used in this article.

4 Evaluation Index

We utilize recall, precision, F1-score, False Positive Rate (FPR), and accuracy to evaluate the experimental results. The percentage of positive samples in the data predicted by the model to be positive is reflected by precision, recall reflects the proportion of real positive samples that are predicted to be positive, F1-score combines precision and recall, and FPR reflects the proportion of negative samples that are incorrectly classified as positive[27].

$$\begin{aligned}
\text{precision} &= \frac{\text{TP}}{\text{TP}+\text{TN}};\\
\text{recall} &= \frac{\text{TP}}{\text{TP}+\text{FN}};\\
\text{FPR} &= \frac{\text{FP}}{\text{FP}+\text{TN}};\\
\text{F1-score} &= \frac{2\text{TP}}{2\text{TP}+\text{FP}+\text{FN}};\\
\text{accuracy} &= \frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FP}+\text{FN}}
\end{aligned} \tag{3}$$

We assume that normal samples in the actual samples are positive samples and that abnormal attack samples are negative samples. The total number of positive samples predicted to be correct is True Positive (TP), and the total number of errors is False Negative (FN). The total of negative samples predicted to be correct is True Negative (TN), and the total of errors is False Positive (FP).

5 Experiment

5.1 Dataset description

The datasets used in this experiment are the gas pipeline and KDDCUP99 datasets. The gas pipeline dataset is an industrial control network laboratory-scale ICS dataset based on Modbus application layer protocol published by Professor T. Morris of Mississippi State University[28]. The KDDCUP99 dataset is a network connection dataset obtained from a simulated US Air Force LAN costing 9 weeks[28].

The KDDCUP99 dataset is a public dataset used

to verify network anomaly detection algorithms. This dataset is employed in the present work to verify the effectiveness of the proposed anomaly detection algorithm. The dataset contains 41-dimensional data samples and includes 22 attack types divided into four categories, namely, Denial of Service attack (DoS), probing, R2L, and U2R[29]. The gas pipeline dataset contains 26 features and a category label. The number of attack categories in the training and test sets is equal, and no unknown attack category is present. This dataset contains seven types of attacks, namely Original Malicious Response Injection (OMRI), Malicious Status Command Injection (MSCI), Complex Malicious Response Injection (CMRI), Malicious Parameter Command Injection (MPCI), DoS, Malicious Function Command Injection (MFCI), and Reconnaissance Attack (RA)[27].

Because the KDDCUP99 dataset has a total of 5 million items, which is massive, we take only 10 % of these items for experimentation. The experimental data include approximately 100 000 items, which accounts for approximately 20% of this dataset. The gas pipeline dataset has a total of 97 019 items, of which 61 156 are normal samples.

5.2 Experimental settings

In view of the different sample sizes of the source and target domain data, we divide the datasets randomly as follows: 90% of the KDDCUP99 dataset is used for the training set, and 10% is used for the test set. Moreover, 80% of the gas pipeline dataset is used for the training set, and 20% is used for the test set.

We use PyTorch to construct the Resnet8 model, multiply the cross-entropy loss function by 1.5–5 and use the result as a loss indicator, and apply the stochastic gradient optimizer. The learning rate of the source domain training is set to 0.001, the batch size is set to 128, the duration is set to 4 epochs. The learning rate of the target domain is set to 0.0003, the batch size is set to 64, and the duration is set to 5 epochs.

The experimental procedure is as follows.

(1) Read the dataset samples of the target and source domains and then digitize, standardize, and normalize the source domain data. Next, digitize the target domain data, remove redundant features, delete the outliers of individual sites, perform MD calculations, and then normalize the data column by column. After processing into new samples, randomly divide the test and training set by percentage. The test set of KDDCUP99 dataset accounts for 10%, and the test set of gas pipeline dataset accounts for 20%.

(2) Visualize the new data samples obtained.

(3) Use KDDCUP99 to pretrain the Resnet8 model. Then, save the model and model parameters after testing the model performance.

(4) Load the pretrained model and model parameters, use the gas pipeline dataset to fine-tune the last three layers of the neural network of Resnet8, and obtain the model test indicators.

The experiments were performed on a computer with an i7-8550U CPU processor, 1.8 GHz frequency, and 8 GB RAM.

5.3 Experimental results and analysis

This section introduces the results of the pretraining model, describes the effects of fine-tuning different numbers of layers, and discusses the effects of model fine-tuning and random initialization parameter training with the target domain. After obtaining the results, we explain the benefits of using transfer learning and why the three-layer method of fine-tuning is used. We also demonstrate the superiority of the proposed algorithm by comparing the results with those of other existing algorithms.

5.3.1 Data preprocessing and visualization

Every data stream in the target is processed into 324 pieces of data and source domains are processed into 41 pieces of data by preprocessing each set of traffic data. Figure 6 shows the results of data visualization. Each dataset in the source domain is processed into a grayscale image of 7 pixel × 7 pixel by a 6-bit 0 supplement, and each dataset in the target domain is processed into a grayscale image of 18 pixel × 18 pixel pixels. Processing samples in this form to the residual CNN is clearly

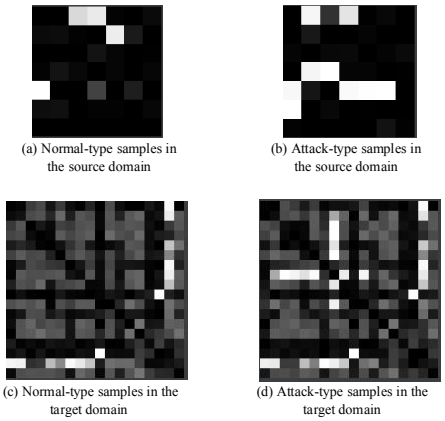


Fig. 6 Data visualization results.

feasible.

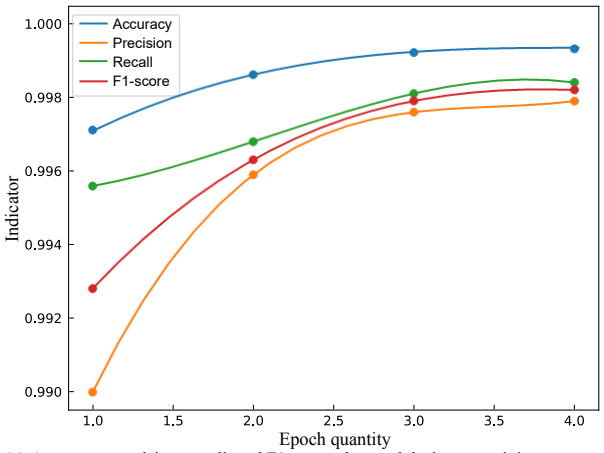
5.3.2 Model pretraining

Figure 7 shows the change curve of the evaluation index on the training and test sets during model pretraining.

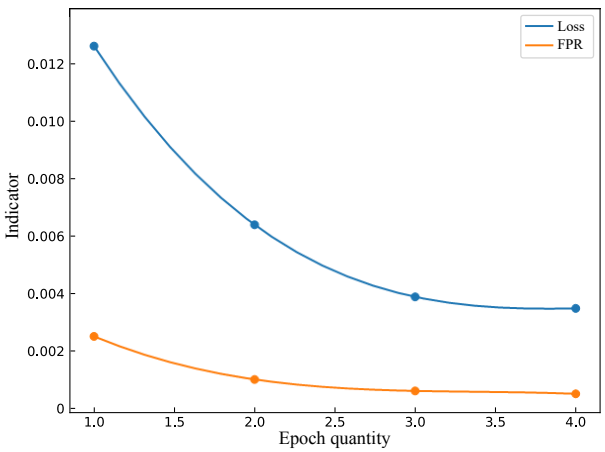
The graph shows that the loss and FPR of the source domain continuously decrease during model pretraining until the values stabilize. F1-score, recall, precision, and accuracy steadily rise until values of 100% are obtained. This result means the model converges well on the source domain, and the evaluation index indicates that the model can be used for training in the target domain.

5.3.3 Model fine-tuning and deep learning

In this experiment, all eight layers of the model are fine-tuned by utilizing transfer learning. The deep learning method used in this article initializes the model randomly and then optimizes all model parameters without transfer learning. The result in Fig. 8 shows that the pretrained model using fine-tuning converges faster and has a smaller loss and higher accuracy than the model using deep learning when the gas pipeline

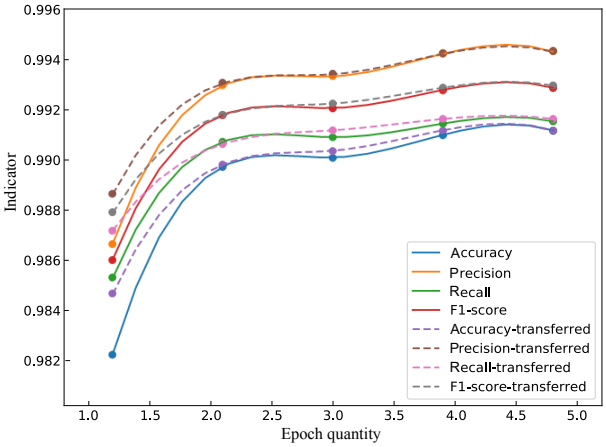


(a) Accuracy, precision, recall, and F1-score observed during pretraining process

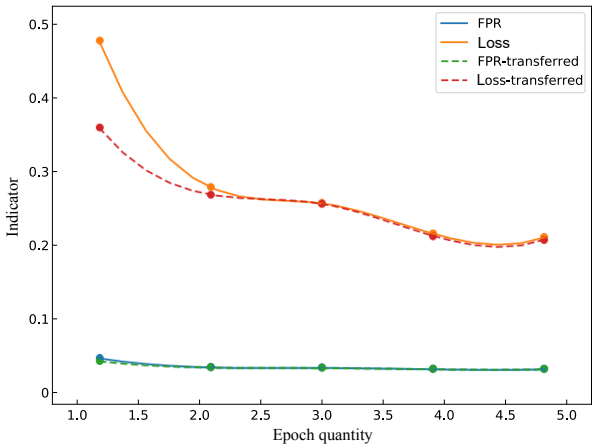


(b) Loss and FPR observed during pretraining

Fig. 7 Pretraining process index change curves.



(a) Accuracy, precision, recall, and F1-score of fine-tuning and deep learning observed during pretraining process



(b) Loss and FPR of fine-tuning and deep learning observed during pretraining process

Fig. 8 Fine-tuning and training of the model using deep learning convergence comparison curves.

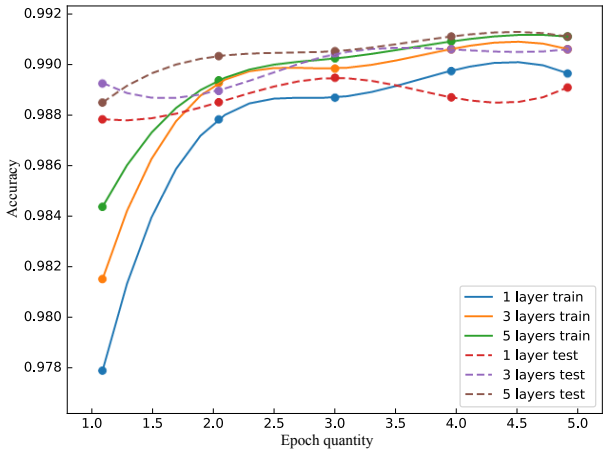
dataset is used for training. This finding indicates that transfer learning is significant in this environment. Both training methods are completed in approximately 81 minutes. The comparison shown in Table 1 reveals that the score of the method of fine-tuning three layers in the ICS flow anomaly detection index is close to the first two methods, which also greatly reduces the training time of the model.

5.3.4 Fine-tuning of the different layers of the model

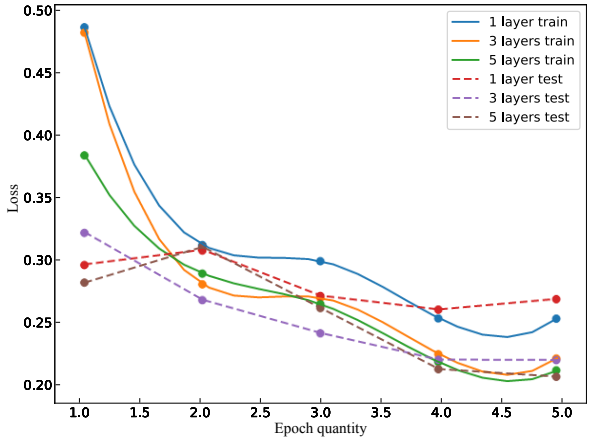
Figure 9 shows the changes in loss and accuracy

Table 1 Transfer learning effect verification form.

	Recall	Precision	F1-score	FPR	Accuracy	Training time
Deep learning	0.9915	0.9955	0.9935	0.0085	0.9915	80 min 44 s
Fine-tuning the model	0.9929	0.9953	0.9941	0.0088	0.9923	81 min 13 s
Fine-tuning three layers	0.9906	0.9955	0.9931	0.0085	0.9909	51 min 58 s



(a) Accuracy observed during the fine-tuning of different numbers of layers



(b) Loss observed during the fine-tuning of different numbers of layers

Fig. 9 Index change curves obtained during the fine-tuning of different numbers of layers.

observed during training. Other indicators are listed in Table 2. Figure 10 shows a visualization of the prediction results.

The results show that the indicators improve as the number of fine-tuning layers increases. In terms of model convergence, the effects of fine-tuning three and five layers are not much different, but the effect of fine-tuning one layer is relatively unsatisfactory. In the

Table 2 Effect of fine-tuning of different numbers of layers.

Number of layers	Recall	Precision	F1-score	FPR	Training time
1	0.9902	0.9940	0.9921	0.0113	37 min 5 s
3	0.9906	0.9949	0.9929	0.0087	51 min 58 s
5	0.9915	0.9954	0.9934	0.0087	65 min 56 s

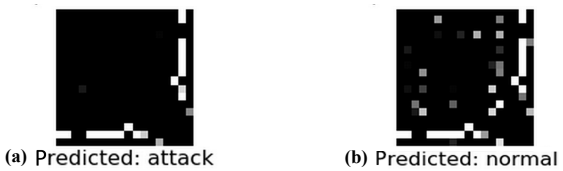


Fig. 10 Visualization of the prediction results.

anomaly detection of ICS flow data, the model training time and FPR are subject to stringent requirements. Given comprehensive consideration, the method of fine-tuning three layers appears to be the most appropriate. Such fine-tuning results in the precision and other indicators exceeding 99% and FPR decreasing to 0.85%, which indicates that the abnormal attack types OMRI, MSCI, CMRI, MPCI, DoS, MFCI, and RA, have been effectively detected.

5.3.5 Algorithm comparison

The comparison results in Table 3 show that the precision indicators of reciprocal data and AutoEncoder (AE)+One Class Support Vector Machine (OCSVM) are higher than those of other machine learning algorithms. However, the recall indicator of these algorithms is low, which means the detected positive samples are actually not all positive. The recall index of Generative Adversarial Networks (GAN) is high but its other indices are low, thus indicating that its comprehensive performance is poor. Although the F1-score of AE+OCSVM is high, which indicates that it has good overall performance, its recall value is quite low, which means some negative samples (abnormal types) may be misclassified as positive samples. These indicators are unsuitable for abnormal detection in ICSs. The algorithms proposed in this paper are clearly superior to those algorithms in terms of the indicators of interest.

6 Conclusion and Future Work

Network security is a popular and important topic. The network security of ICSs is of great importance for a country. This paper uses data visualization to convert flow data into images. Specifically, we build an eight-layer residual neural network and use fine-tuning technology for transfer learning to detect abnormal datasets of ICSs.

Experimental results show that transfer learning for residual CNNs is effective in this field. The depth of the model also ensures that it has a certain generalizability. The residual structure effectively prevents gradient

Table 3 Performance comparison of different algorithms.

Algorithm	Recall	Precision	F1-score
GAN	0.9973	0.7498	0.8621
AE+OCSVM	0.8747	0.9907	0.9284
DEC	0.8821	0.8893	0.8909
RDA	0.7301	0.9913	0.8411
Fine-tune+Resnet8	0.9906	0.9955	0.9931

Note: DEC means deep embedded clustering.

explosion or gradient disappearance. The model can provide reliable predictions for unknown or differently distributed abnormal data through short-term training by transfer learning. Compared with other anomaly detection algorithms, the algorithm proposed in this paper results in superior indicators. The method we proposed not only solves the problem associated with training time for deep learning models by transfer learning, but also meets the requirements of ICSs in terms of evaluation indicators.

At present, the model we constructed solves the two-classification problem, but a refined classification of abnormal traffic data is still desirable. In the future work, we will perform multiclassification of abnormal traffic data, track the characteristics of different abnormal data types, and then reliably classify them to further ensure network security in ICSs.

Acknowledgment

This work was supported in part by 2018 industrial Internet innovation and development project "Construction of Industrial Internet Security Standard System and Test and Verification Environment", in part by the National Industrial Internet Security Public Service Platform, in part by the Fundamental Research Funds for the Central Universities (Nos. FRF-BD-19-012A and FRF-TP-19-005A3), in part by the National Natural Science Foundation of China (Nos. 81961138010, U1736117, and U1836106), and in part by the Technological Innovation Foundation of Shunde Graduate School, University of Science and Technology Beijing (No. BK19BF006).

References

[1] A. R. Sadeghi, C. Wachsmann, and M. Waidner, Security and privacy challenges in industrial Internet of Things, in *Proceedings of the 2015 52nd ACM/EDAC/IEEE Design Automation Conference* (*DAC*), San Francisco, CA, USA, 2015, pp. 1–6.

[2] L. Obergon, InfoSec reading room secure architecture for industrial control systems, *SANS Institute InfoSec, GIAC*(*GSEC*) *Gold Certification*, vol. 1, pp. 1–27, 2014.

[3] C. Markman, A. Wool, and A. A. Cardenas, A new burst-DFA model for SCADA anomaly detection, in *Proceedings of the 2017 Workshop on Cyber-Physical Systems Security and PrivaCy*, Dallas, TX, USA, 2017, pp. 1–12.

[4] M. Mantere, I. Uusitalo, M. Sailio, and S. Noponen, Challenges of machine learning based monitoring for industrial control system networks, in *Proceedings of the 2012 26th International Conference on Advanced Information Networking and Applications Workshops*, Fukuoka, Japan, 2012, pp. 968–972.

[5] R. Zhao, R. Q. Yan, Z. H. Chen, K. Z. Mao, P. Wang, and R. X. Gao, Deep learning and its applications to machine health monitoring: A survey, *Mechanical System and Signal Processing*, vol. 115, pp. 213–237, 2019.

[6] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Q. Zhou, W. Li, and P. J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1–67, 2020.

[7] S. N. Shirazi, A. Gouglidis, K. N. Syeda, S. Simpson, A. Mauthe, I. M. Stephanakis, and D. Hutchison, Evaluation of anomaly detection techniques for SCADA communication resilience, in *Proceedings of the 2016 Resilience Week* (*RWSr*), Chicago, IL, USA, 2016, pp. 140–145.

[8] Y. Lai, J. Zhang, and Z. liu,, Industrial anomaly detection and attack classification method based on convolutional neural network, *Security and Communication Networks*, doi: 10.1155/2019/8124254.

[9] J. Hurley, A. Munoz, and S. Sezer, ITACA: Flexible, scalable network analysis, in *Proceedings of the 2012 IEEE International Conference on Communications* (*ICC*), Ottawa, Canada, 2012, pp. 1069–1073.

[10] G. Thatte, U. Mitra, and J. Heidemann, Parametric methods for anomaly detection in aggregate traffic, *IEEE/ACM Transactions On Networking*, vol. 19, no. 2, pp. 512–525, 2010.

[11] A. Terai, S. Abe, K. Shoya, Y. Takano, and I. Koshijima, Cyber-attack detection for industrial control system monitoring with support vector machine based on communication profile, in *Proceedings of the 2017 IEEE European Symposium on Security and Privacy Workshops* (*EuroS&PW*), Paris, France, 2017, pp. 132–138.

[12] C. Zhou, S. Huang, N. Xiong, S. Yang, H. Li, Y. Qin, and X. Li, Design and analysis of multimodel-based anomaly intrusion detection systems in industrial process automation, *IEEE Transactions on Systems, Man, and Cybernetics*: *Systems*, vol. 45, no. 10, pp. 1345–1360, 2015.

[13] M. Zhang, B. Y. Xu, and J. Gong, An anomaly detection model based on one-class SVM to detect network intrusions, in *Proceedings of the 2015 11th International Conference on Mobile Ad-hoc and Sensor Networks* (*MSN*), Shenzhen, China, 2015, pp. 102–107.

[14] S. C. Zhang, X. Y. Xie, and Y. Xu, Intrusion detection method based on a deep convolutional neural network, *Tsinghua Science and Technology*, vol. 59, no. 1, pp. 44–52, 2019.

[15] A. Almalawi, X. H. Yu, Z. Tari, A. Fahad, and I. Khalil, An unsupervised anomaly-based detection approach for integrity attacks on SCADA systems, *Computers & Security*, vol. 46, pp. 94–110, 2014.

[16] W. Gao, Cyberthreats, attacks and intrusion detection in supervisory control and data acquisition networks, PhD dissertation, Department of Electronic & Computer Engineering, Mississippi State University, Mississippi, MS,

USA, 2013.
[17] J. Liang, J. H. Chen, X. Q. Zhang, Y. Zhou, and J. J. Lin, One-hot encoding and convolutional neural network based anomaly detection, *Tsinghua Science and Technology*, vol. 59, no. 7, pp. 523–529, 2019.
[18] Y. Wang, C. Wang, L. Luo, and Z. Zhou, Image classification based on transfer learning of convolutional neural network, in *Proceedings of the 2019 Chinese Control Conference* (*CCC*), Guangzhou, China, 2019, pp. 7506–7510.
[19] K. He, X. Zhang, S. Ren, and J. Sun, Deep residual learning for image recognition, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognitio*, Las Vegas, NV, USA, 2016, pp. 770–778.
[20] E. Rezende, G. Ruppert, T. Carvalho, F. Ramos, and P. de Geus, Malicious software classification using transfer learning of resnet-50 deep neural network, in *Proceedings of the 2017 16th IEEE International Conference on Machine Learning and Applications* (*ICMLA*), Cancun, Mexico, 2017, pp. 1011–1014.
[21] Z. Chen, Z. Xie, W. Zhang, and X. Xu, ResNet and model fusion for automatic spoofing detection, in *Proceedings of the Interspeech*, Stockholm, Sweden, 2017, pp. 102–106.
[22] 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, 2018.
[23] P. C. Mahalanobis, On the generalised distance in statistics, in *Proceedings of the National Institute of Science of India*, Calcutta, India, 1936, pp. 49–55
[24] S. Xiang, F. Nie, and C. Zhang, Learning a Mahalanobis distance metric for data clustering and classification, *Pattern Recognition*, vol. 41, no. 12, pp. 3600–3612, 2008.
[25] S. Ioffe and C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, arXiv preprint arXiv: 1502.03167, 2015.
[26] A. F. Agarap, Deep learning using rectified linear units (Relu), arXiv preprint arXiv:1803.08375, 2018.
[27] G. J. Wang, J. Feng, M. Z. A. Bhuiyan, R. X. Lu, *Security, Privacy and Anonymity in Computation, Communication and Storage*. Berlin, Germany: Springer, 2019.
[28] X. Zhang, H. Zeng, and L. Jia, Research of intrusion detection system dataset-KDDCUP99, *Computer Engineering and Design*, vol. 31, no. 22, pp. 4809–4812, 2010.
[29] I. S. Thaseen and C. A. Kumar, Intrusion detection model using fusion of chi-square feature selection and multi class SVM, *Journal of King Saud University-Computer and Information Sciences*, vol. 29, no. 4, pp. 462–472, 2017.

Weiping Wang received the the PhD degree in telecommunications physics electronics from Beijing University of Posts and Telecommunications, Beijing, China in 2015. She is currently an associate professor at the School of Computer and Communication Engineering, University of Science and Technology Beijing. She received the support from National Key Research and Development Program of China, the State Scholarship Fund of China Scholarship Council, the National Natural Science Foundation of China, the postdoctoral fund, and other basic scientific research projects. Her current research interests include auto-driving vehicle formation control, brain-like computing, memristive neural network, associative memory awareness simulation, complex network, and network security and image encryption.

Zhaorong Wang is currently an undergraduate student at the School of Automation and Electrical Engineering, University of Science and Technology Beijing. His current research interests include auto-driving vehicle formation control, brain-like computing, intelligent control, machine learning, and anomaly detection.

Zhanfan Zhou is currently an undergraduate student at the School of Mechanical Engineering, University of Science and Technology Beijing. His current research interests include auto-driving vehicle formation control, brain-like computing, intelligent control, machine learning, and anomaly detection.

Haixia Deng is currently an undergraduate student at the Donlinks School of Economics and Management, University of Science and Technology Beijing. Her current research interests include auto-driving vehicle formation control, brain-like computing, intelligent control, machine learning, and anomaly detection.

Weiliang Zhao is currently an undergraduate student at the School of Mechanical Engineering, University of Science and Technology Beijing. His current research interests include auto-driving vehicle formation control, brain-like computing, intelligent control, machine learning, and anomaly detection.

Yongzhen Guo received the master degree in control theory and control engineering from Tianjin University, Tianjin, China in 2010. He is now a PhD candidate at the School of Automation, Beijing Institute of Technology (BIT). He is also the general manager of Industrial Control System Evaluation and Certification Department of China Software Testing Center. He received the National Science and Technology Major Projects and National Key Research and Development Programs. His research interests include security and cryptography, safety and reliability, and system evaluation and certification. As a member of SAC/TC124/SC10, SAC/TC196, ISO/TC 199/G8, and IEC/TC65/SC65C/WG18, he is participating in a number of international standards and national standards setting and revising.

Chunyang Wang received the BS degree from Shandong Agricultural University, China in 2019. He is currently a master student at the University of Science and Technology Beijing. His current research interests include auto-driving vehicle formation control, brain-like computing, intelligent control, machine learning, and anomaly detection.