Hindawi
Security and Communication Networks
Volume 2019, Article ID 6757685, 11 pages
https://doi.org/10.1155/2019/6757685

Research Article

Industrial Control Intrusion Detection Approach Based on Multiclassification GoogLeNet-LSTM Model

Ankang Chu,[1] Yingxu Lai ,[1,2] and Jing Liu [1,3]

[1]*College of Computer Science, Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China*
[2]*Science and Technology on Information Assurance Laboratory, Beijing 100072, China*
[3]*Shaanxi Key Laboratory of Network and System Security, Xidian University, Xi'an 710071, China*

Correspondence should be addressed to Yingxu Lai; laiyingxu@bjut.edu.cn

Received 1 August 2019; Revised 17 November 2019; Accepted 21 November 2019; Published 13 December 2019

Academic Editor: Clemente Galdi

Copyright © 2019 Ankang Chu et al. 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.

Intrusion detection is essential for ensuring the security of industrial control systems. However, conventional intrusion detection approaches are unable to cope with the complexity and ever-changing nature of industrial intrusion attacks. In this study, we propose an industrial control intrusion detection approach based on a combined deep learning model for communication processes that use the Modbus protocol. Initially, the network packets are classified as carrying information and noncarrying information based on key fields according to the communication protocol used. Next, a template comparison approach is employed to detect the network packets that do not carry any information. Furthermore, an approach based on a GoogLeNet-long short-term memory model is used to detect the network packets that do carry information. This approach involves network packet sequence construction, feature extraction, and time-series level detection. Subsequently, the detected intrusions are classified into multiple categories through a Softmax classifier. A gas pipeline dataset of the Modbus protocol is used to evaluate the proposed approach and compare it with existing strategies. The accuracy, false-positive rate, and miss rate are 97.56%, 2.42%, and 2.51%, respectively, thus confirming that the proposed approach is suitable for intrusion detection in industrial control systems.

1. Introduction

As any industry develops, it uses industrial field equipment extensively, and these pieces of equipment are generally widely distributed. Various communication protocols, such as Profibus and CAN bus, have been proposed to ensure stable communications between the field equipment and their centralised management. However, these protocols exhibit several problems, such as poor compatibility for Profibus and the lack of an error response mechanism for CAN bus, and hence do not satisfy the requirements for industrial control. The Modbus protocol [1] was originally proposed by Modicon and has become the most popular communication protocol in the field of industrial control because of its ease of integration. In the Modbus protocol, communication occurs via a request-response approach, and there are no additional overheads. The protocol allows for efficient communication between the interconnected devices using only three distinct protocol communication units: a Modbus request unit, a response unit, and an exception response unit. The Modbus protocol is suitable for transferring monitoring data in industrial control systems (ICS) because of its low processing overhead. The communication formats of the Modbus protocol include the remote terminal unit (RTU), American Standard Code for Information Interchange (ASCII), and Transmission Control Protocol (TCP). However, the Modbus protocol also has several serious security-related issues:

(1) The protocol does not have an authentication mechanism. Hence, verification is performed based only on the address, function code, and other fields, making it vulnerable to flooding attacks.

(2) The protocol uses plaintext communication without an information verification mechanism. This makes the data vulnerable to tampering, stealing, and forging.

(3) The protocol units are programmable. Hence, malicious code can be readily injected into the RTU and programmable logic controller.

These intrusions cannot be detected based on the network traffic. It is necessary to perform a deep analysis of the packets to detect intrusion-related behaviours based on the protocol format, state, and event/event sequences generated by the protocol. Therefore, in this study, we propose an industrial control intrusion detection approach based on a combined deep learning model for communication processes that use the Modbus protocol.

The rest of the paper is organised as follows. Section 1 presents the background information for the study. Section 2 describes related works. The proposed intrusion detection approach is described in Section 3. The results of evaluations of the proposed approach are presented in Section 4 along with a comparison with other similar approaches. Finally, the conclusions of the study are presented in Section 5.

2. Related Works

Given the rapid developments in ICS networking and informatisation technologies, the use of standardised protocols and open software is increasing. This exposes them to numerous security threats and an ever-increasing number of information security problems [2]. The Stuxnet virus in 2010 and the Black Energy virus in 2015 [3] have sounded the alarm for information security in the field of industrial controls. Conventional industrial control security technologies, such as user authentication, firewalls, and data encryption, have been unable to cope with the continuous innovations in network intrusion methods [4]. Thus, the field of intrusion detection has become a research hotspot, as it is now the second line of defence for industrial control information security. It involves the extraction of data features that reflect the system behaviour and the classification of the network packets using detection algorithms [5]. Conventional machine learning methods and algorithms, such as decision trees (DT) [6] and support vector machines (SVMs) [7], have achieved some degree of success in industrial control intrusion detection. However, most of these approaches can be considered shallow learning strategies and have limited efficacy in complex industrial control system environments. For example, the data generated during production processes have a lot of noise. In such cases, the DT algorithm is prone to overfitting and exhibits low classification accuracy. On the other hand, SVM consumes considerable computational resources when processing massive sample datasets. Therefore, intrusion detection approaches based on conventional machine learning algorithms cannot meet the requirements for industrial control intrusion detection [8].

Deep learning techniques are better at learning data features given their deep structure and have a greater ability to analyse high-dimensional data [9]. In recent years, several researchers have applied deep learning for intrusion detection. Javaid et al. [10] proposed an intrusion detection approach based on a sparse autoencoder. This approach was used for feature dimensionality reduction to improve detection and exhibited a higher detection accuracy. However, the deep learning approach was only used for dimensionality reduction and, hence, the improvement in the detection rate was not significant. Tang et al. [11] designed an intrusion detection approach based on deep neural networks (DNNs). Their approach exhibited higher detection accuracy using only six features of the NSL-KDD dataset. However, their model is hard to train, as DNNs have a large number of parameters. Hence, deep learning network structures should be used directly for intrusion detection in order to improve the detection accuracy.

Convolutional neural networks (CNNs) [12] are hierarchical structures with a good ability to extract local features. A typical CNN model consists of an input layer, a convolution layer, a pooling layer, and a fully connected layer. It combines a local receptive field, a shared weight, and spatial/temporal sampling and thus has a unique advantage when it comes to the processing of statistically stable and locally correlated data. The process for industrial control intrusion detection based on a CNN model can be divided into the following steps. To begin with, the original network packets are encoded into a two-dimensional array using the one-hot encoding method. Next, the two-dimensional array is fed as an input into the CNN, which outputs the calculation results as a feature vector after performing a number of operations, including convolution, pooling, and full connection. Finally, the network packets are classified using the extracted feature vector with the appropriate classification method to obtain the detection results. This process of industrial control intrusion detection based on a multi-classification CNN model is shown in Figure 1.

This approach combines the two tasks of feature extraction and classification and can achieve higher detection accuracy. Liang et al. [13] proposed an intrusion detection approach based on a deep CNN. They were able to achieve a detection accuracy higher than that of the conventional DNN-based approach. However, their approach has the following problems.

(1) Conventional CNNs have a poor feature extraction ability when used with industrial control data, which is multidimensional and exhibits complex variations. This is because of the simple structure of CNNs. As a result, they exhibit high miss rates during intrusion detection.

(2) The network packets generated during industrial control communication processes invariably have different lengths, and the zero-padding method has to be used during the encoding process. As a result, feature extraction using the same CNN for different kinds of network packet is inaccurate.

(3) This approach detects a single network packet, and the time-series relationships of the data are not considered. As a result, the approach is unable to detect intrusions against the communication process.

Time-series detection is an important means of intrusion detection in industrial control and is implemented using networks that are capable of time-series data processing. The long short-term memory (LSTM) network [14] is a type of recurrent neural network with a special structure called the

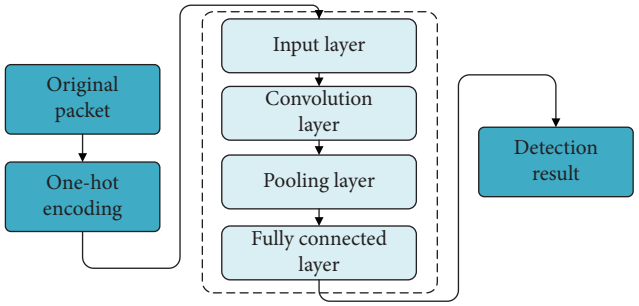


Figure 1: Workflow diagram for industrial control intrusion detection based on multiclassification CNN model.

long short-term memory module. This module consists of an input gate, a forgetting gate, and an output gate that are responsible for transferring the memory information from the initial position to the end of the sequence. In this module, the input gate determines how much input information is to be added to the memory information flow at the current instant. The forget gate controls the internal circulation of the memory cells and determines the choice of information in the cells. Finally, the output gate determines how much of the memory information will be used in the next stage. This gating mechanism of the LSTM keeps the error relatively constant and avoids the problem of gradient disappearance and gradient explosion. The calculations performed in the LSTM module can be described by the following equations:

$$\begin{aligned} f_t &= \text{sigmoid}\left(w_f \times [h_{t-1}, x_t] + b_f\right), \\ i_t &= \text{sigmoid}\left(w_i \times [h_{t-1}, x_t] + b_i\right), \\ \tilde{c}_t &= \tanh\left(w_c \times [h_{t-1}, x_t] + b_c\right), \\ c_t &= f_t \times c_{t-1} + i_t \times \tilde{c}_t, \\ o_t &= \text{sigmoid}\left(w_o \times [h_{t-1}, x_t] + b_o\right), \\ h_t &= o_t \times \tanh\left(c_t\right), \end{aligned} \tag{1}$$

where h_{t-1} is the upper hidden state; x_t is the current input; w_f, w_i, w_c, and w_o are the weight matrixes; and b_f, b_i, b_c, and b_o are the bias vectors. The process of industrial control intrusion detection based on the LSTM model can be divided into the following steps. To begin with, several historical network packets are extracted to construct the detection sequence. Next, a feature extraction method to obtain the feature vectors of the network packets in the detection sequence is used. Following this step, the feature vectors are fed as input into the network based on the time steps and calculations are performed using the LSTM module. Finally, the network packets are classified by passing the network output through a Softmax classifier to obtain the detection results. The process of industrial control intrusion detection based on a multiclassification LSTM model is shown in Figure 2.

This approach considers the time-series changes in the network packets and can detect more intrusion behaviours during the communication process. Cheng et al. [15] designed an intrusion detection approach based on an LSTM network. In their approach, a network packet sequence was

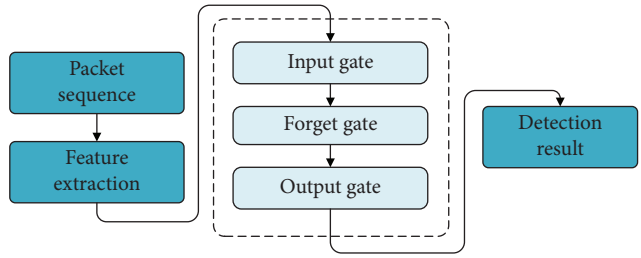


Figure 2: Workflow diagram of industrial control intrusion detection based on multiclassification LSTM model.

first constructed, and the LSTM network was used for time-series level detection. The predicted signature was output by a Softmax classifier and compared with the real signature from a signature database to detect intrusions. While the detection accuracy of this approach is high, it has the following problems:

(1) The approach is sensitive to data variations wherein many packets with normal data variations are detected as intrusion packets. An incomplete learning of the data variation characteristics leads to higher false positive rates.

(2) The approach relies heavily on feature extraction methods, which weaken the role of the network itself to some extent. Furthermore, conventional feature extraction methods always involve complex operations, such as data preprocessing and feature correlation analysis, to name a few. Thus, this method cannot extract features accurately in the case of data with high dimensions and complex variations.

(3) The conventional time-series level detection approach for the Modbus protocol is network packet oriented, and the feature vector input of each time step has different dimensions. This affects the detection accuracy to some extent.

Based on the above-described studies, it can be concluded that performing a deep analysis of the communication processes that use the Modbus protocol and exploiting the advantages of different networks are the key to improving the intrusion detection accuracy of industrial control processes.

3. Approach Based on the Combined Deep Learning Model

It can be seen from the above-described approaches that there are several limitations in the use of any single type of network for industrial control intrusion detection. The CNN can perform feature extraction with accuracy but does not consider the time-series variation characteristics of the industrial control data. On the other hand, the LSTM network can perform time-series detection but needs more accurate feature vectors to achieve high detection accuracy. Thus, in this study, a deep learning model that combines the advantages of these two types of networks was designed for industrial control intrusion detection. To begin with, the

network packet categorisation method is used to classify the network packets into basic types. Next, the template comparison method is used to detect packets without information and the approach based on GoogLeNet-LSTM model is used for packets carrying information. This section introduces the industrial control intrusion detection approach based on this combined deep learning model.

3.1. Network Packet Classification Method. It is not necessary to use a complex network model to detect the network packets that do not carry any information in the case of communication processes using the Modbus protocol. Furthermore, different detection approaches should be used for different types of network packets in order to improve the detection efficiency. However, this would require an analysis of the communication processes using the Modbus protocol.

The Modbus protocol uses a master-slave communication mechanism to ensure that the communication processes occur normally [16]. In industrial control communication processes, the RTU returns a packet carrying information regarding the system's operating status after receiving a request packet from the master terminal unit (MTU). Subsequently, the MTU sends a packet carrying control information to the RTU to maintain the stable operation of the system, and the RTU returns a confirmation packet. Different network packets implement different functions based on different key fields. Thus, different detection approaches should be used for different types of network packets.

The network packets involved in the communication processes using the Modbus protocol can be categorised as command read (CR) packets, response read (RR) packets, command write (CW) packets, and response write (RW) packets. The CR and RW packets do not carry any information and have fixed field information. The RR packets carry the information regarding the system's operating status returned from the RTU to the MTU. The CW packets carry the control information sent from the MTU to the RTU. Intrusion detection can be performed by using the template comparison approach for the CR and RW packets. However, a specific intrusion detection approach needs to be adopted for the RR and CW packets.

3.2. Template Comparison Approach. With respect to the network packets generated during communication using the Modbus protocol, the CR and RW packets do not carry any information and only perform the tasks of request and confirmation. These packets have fixed functions wherein the field values remain constant. For example, CR packets are used to request the pressure measurement data from gas pipelines. Their function code field value is 0x03, and length field value is 0x0b, with the other field values being fixed. A CR packet template is set based on these normal field values. The network packets can be detected by comparing their field values with those in the template when a new CR packet is received. The template comparison approach results in high detection speed and accuracy for network packets that do not carry information.

3.3. Approach Based on GoogLeNet-LSTM Model

3.3.1. GoogLeNet-LSTM Model. A few problems can arise when a CNN or LSTM network is used individually for intrusion detection. The CNN can extract more accurate feature vectors because of its strong feature-learning ability. However, it cannot perform time-series network packet detection. On the other hand, the LSTM network can readily perform time-series detection but relies on accurate feature vectors for effective detection. Therefore, the CNN-LSTM model has been especially designed for effective intrusion detection, wherein the CNN is initially used to obtain the feature vectors that are then used for time-series detection by the LSTM network.

Conventional CNNs are not effective at feature extraction when used with industrial control data that have a large number of features and exhibit complex variations. Hence, in the proposed approach, GoogLeNet is used instead of a conventional CNN to obtain more accurate feature vectors. GoogLeNet is a type of CNN with a special structure called an inception module [17]. It performs calculations using different types of kernels in a single layer; in contrast, conventional CNNs have only one type of kernel. The commonly used kernels in GoogLeNet are 1×1, 3×3, 5×5, and 7×7, and the calculation results from these kernels are combined into the final output. The features are better represented for calculating the different scales. Multiple inception modules can be stacked to perform calculations in different layers. The width and depth of the network are greater than those of a conventional CNN. As a result, GoogLeNet can extract richer features from industrial control data. In addition, GoogLeNet can perform operations to handle the very large number of parameters arising from the use of several types of kernels and multiple layers in the network. For example, two 3×3 convolution kernels may be used instead of a single 5×5 convolution kernel. A $1 \times n$ convolution kernel and an $n \times 1$ convolution kernel may be used to replace an $n \times n$ convolution kernel for higher-dimensional feature processing. Therefore, GoogLeNet is suitable for use as a lightweight feature extraction network in the proposed approach.

Conventional LSTM networks are not good at dealing with long sequence tasks. This is because it is difficult for these networks to learn the information present early in the sequence. Furthermore, the gradient vanishing problem occurs readily during model training. An attention mechanism [18] had been added to the LSTM network to solve these problems. The primary idea behind this mechanism, which simulates the attention mechanism of the human brain, is to assign more attention to the key parts of the input sequence that affect the output. This mechanism is not only better at learning the information in the input sequence but also reduces the information loss in the case of long sequences. The LSTM network based on the attention mechanism retains the intermediate output of the input

sequence, which is then used for selective learning by a trained model and is associated with the network output. The implementation of the attention mechanism can be represented as shown in the following equations:

$$\begin{aligned} u_i^t &= \tanh\left(w_h h^i + w_z z^t\right), \\ a_t &= \operatorname{softmax}\left(u_i^t\right), \\ c^t &= \sum_i a_t^i h^i, \end{aligned} \tag{2}$$

where h_i is the output vector of the hidden layer of the LSTM network, z_t is the matching vector, w_h and w_z are the matching parameters, and c_t is the attention vector.

Initially, GoogLeNet is used to extract each network packet in the detection sequence, and the extracted feature vectors are input into the LSTM network for time-series detection to obtain the final detection result. The model has the following characteristics:

(1) GoogLeNet is used for feature extraction, as it can extract more accurate features for the inception module

(2) The attention mechanism is added in the LSTM network in order to allow the network to process longer-sequence tasks and yield more accurate classification results

(3) A few parameter reduction methods are used to ensure a high efficiency of intrusion detection

3.3.2. Industrial Control Intrusion Detection Approach Based on GoogLeNet-LSTM Model. Before using the model for detection, a detection sequence needs to be constructed. The detection sequence should reflect the normal system operation status as well as the variations in the data [19]. The historical RR packets and CW packets are extracted from the log file or a database when a new network packet carrying information is received, in order to construct the old sequence. The intrusion packets in the sequence should be replaced by nearly normal packets of the same type. The information corresponding to a complete communication process that uses the Modbus protocol is carried in different categories of packets. The extracted feature vector is not complete if feature extraction is performed using the RR packets or CW packets separately. The adjacent packets in the old sequence are spliced to obtain a new packet that carries all the information for the complete communication process. This sequence construction method can not only guarantee the dimensional consistency of feature vectors extracted by the same GoogLeNet model but also detect intrusions with greater accuracy by using all the information. We use this method to obtain the detection sequence $\{p_1, p_2, \ldots, p_t\}$. The method for constructing the detection sequence is shown in Figure 3.

Each packet in the sequence must be preprocessed before it can be input into GoogLeNet for feature extraction. The one-hot encoding method is used for this operation, wherein each field value of the packets is converted into a one-hot code. In addition, only the effective information is used for encoding, while the key field

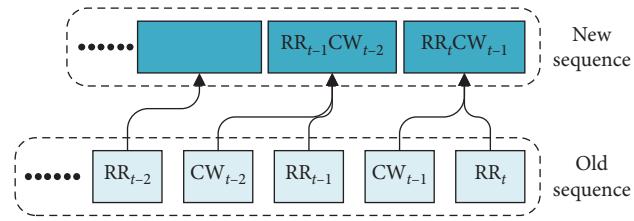


Figure 3: Detection sequence construction method.

information, which is of no use in classification, is used for categorising the network packets. This reduces the dimension of the feature vector and improves the robustness of the model. Specifically, each field value of the packets corresponding to the communication process using the Modbus protocol consists of two hexadecimal digits and varies from 0x00 to 0xff. These field values are first converted into decimal digits between 0 and 255 and then encoded to a 256-dimensional one-hot code of 0 or 1. The network packets with the key field information removed have a length of 54, and each packet in the sequence can be converted into an array consisting of 0s and 1s and having the dimensions of 54×256. The detection sequence is converted into $\{c_1, c_2, \ldots, c_t\}$ after the encoding operation. The network packet encoding method is shown in Figure 4.

The array is then input into GoogLeNet to obtain the feature vector. Multiple convolution kernels are used for convolution operations. The original array is sparse with many zeros; hence, its dimensions need to be reduced. The 7×7 convolution kernel, which has a bigger receptive field, is used first to quickly find the valid bits in the array. Here, we use three cascade 3×3 convolution kernels instead of a 7×7 convolution kernel. On the one hand, three nonlinear activation layers are integrated to increase the discriminating ability; on the other hand, the model parameters are greatly reduced. Furthermore, a 3×3 convolution kernel and a 1×1 convolution kernel are used to construct the inception module for simultaneous convolution. The convolution operation can be expressed by the following equation:

$$f_i^{n\times n} = \operatorname{ReLU}\left(w^{n\times n} \cdot c_i + b^{n\times n}\right), \tag{3}$$

where n is the convolution kernel size, $w^{n\times n}$ is the convolution kernel, $b^{n\times n}$ is the bias, and ReLU is the activation function. The rectified linear unit (ReLU) is a type of activation function that has many advantages over other activation functions, including Softmax and tanh. It helps solve the problem of gradient vanishing, it is faster, and has a higher convergence speed. The pooling operation is also used to construct the inception module for calculation. The commonly used pooling methods are average pooling and maximum pooling. Here, 3×3 average pooling is used to retain more information. The pooling result is then cascaded with other convolution results as the current inception layer output. Next, the output is input into the next inception layer for further calculation. The calculation operation can be represented by the following equation:

$$f_i^m = \operatorname{Cas}\left(f_i^{7\times 7}, f_i^{3\times 3}, f_i^{1\times 1}, f_i^{\text{pooling}}\right), \tag{4}$$

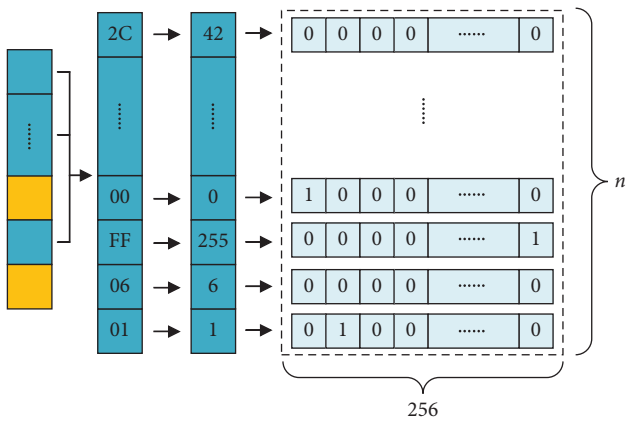


Figure 4: Network packet encoding method.

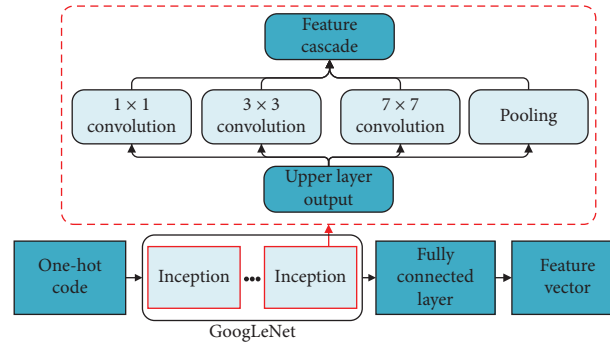


Figure 5: Feature extraction method for proposed approach.

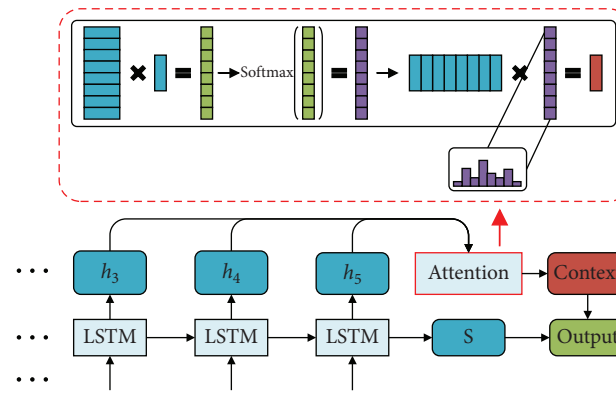


Figure 6: LSTM network based on the attention mechanism.

where f_i^{pooling} is the pooling result, Cas is the cascading operation, and m is the count of the inception module. The total number of inception modules is set as six after the experimental certification. The last inception module output is converted to the required dimension by being passed through the full connection layer, and this final output is the extracted feature vector. The feature extraction method is shown in Figure 5.

Each packet is extracted by GoogLeNet, and the detection sequence is converted into a feature vector sequence $\{v_1, v_2, \ldots, v_t\}$. The detection sequence is next input into the LSTM network for time-series detection. In order to better learn the time-series variation characteristics of industrial control data, it is necessary to use more network packets for detection. Therefore, the sequence consisting of the network packets is long; this is to ensure that the network can learn the variations seen regularly in industrial control data. However, conventional LSTM networks cannot learn the earlier information when processing tasks with a long sequence. Furthermore, each time step input contains all the information of a complete communication process in our method. It is believed that inputs with similar data are more useful in classifying the current input. The LSTM network with an attention mechanism can help overcome these problems. To begin with, the hidden states of the network are stored for further calculation, which greatly limits the loss of information. Next, the attention mechanism ensures that the model focuses on the input with similar information in the sequence by calculating the similarity between all the hidden states and the current state to make a more accurate classification. The hidden states $\{h_1, h_2, \ldots, h_t\}$ are used in the calculation together with the output o_{LSTM} of the last step to obtain an attention vector a_i. Next, parallel calculations of o_{LSTM} together with a_i yield the final output o_i, which is the detection result. The time-series detection is shown in Figure 6.

3.4. Multiclassification of Detection Results. Being able to classify the detection results into multiple classes is of great significance with respect to industrial control intrusion detection. This is because it allows the operator or host to take the appropriate countermeasures based on the intrusion type to ensure the efficiency and stability of the industrial production activities.

The detection results for the CR and RW packets can be obtained by using the template comparison approach. The specific intrusion types can be determined based on the changes to the key field values because there is no information in these packets. For example, a CR packet with a function code field value of 0x2b is indicative of a reconnaissance (recon) intrusion. On the other hand, a network packet with a function code field value of 0x08 is indicative of a malicious function code injection (MFCI) attack.

The detection results for the RR and CW packets can be obtained by using the GoogLeNet-LSTM model. The Softmax classifier is added at the end of the model to output the detection results with the dimension of the intrusion type. The Softmax classifier is trained during the model training process. During the intrusion detection process, the Softmax classifier is used to output a set of probability values such that the largest value corresponds to the intrusion type.

The overall framework for the proposed industrial control intrusion detection method is shown in Figure 7.

As shown in the figure, the industrial control intrusion detection method uses the following three parts: a network packet categorisation module, an intrusion detection module, and a data read-write module. The network packet categorisation module divides the network into different types. The intrusion detection module performs detection using different approaches based on the packet type. Finally,

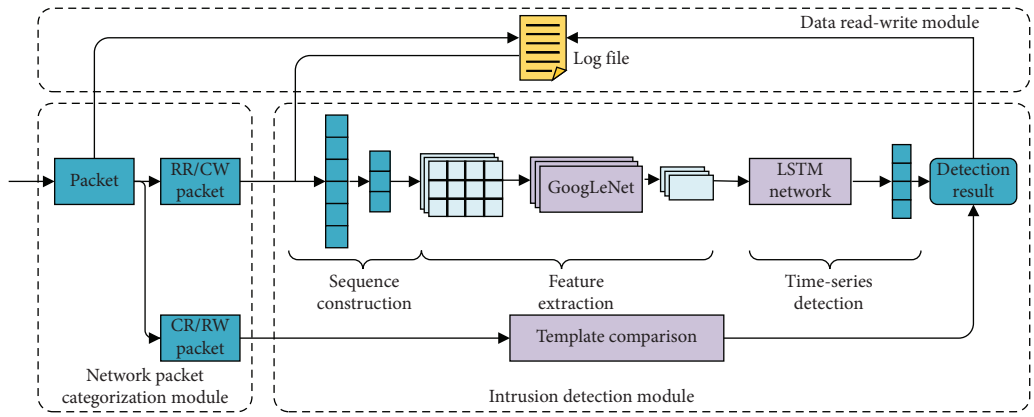


Figure 7: Overall framework of proposed industrial control intrusion detection method.

the data read-write module is responsible for network packet extraction and storage as well as for recording the detection results.

4. Evaluation and Analysis

4.1. Dataset Used. To evaluate the proposed method, we used the industrial control intrusion detection standard dataset [20] put together by Mississippi State University in 2014. The data in this dataset are the network layer data of a gas pipeline control system that uses the Modbus RTU protocol. The data consist of a hexadecimal value and a decimal value. Each network packet includes a header, the Modbus payload, the category label, and the time stamp. The breakup of the dataset is summarized in Table 1.

The dataset contains a total of 274,628 normal and intrusion packets; these were divided into a training set, validation set, and test set in a ratio of 6 : 2:2. This is because only about half of the packets are RR packets and CW packets and splitting at this ratio ensures that there are enough validation and test sets for model validation and test. Furthermore, it is necessary to keep enough intrusion packets for validation and test sets to detect the ability of the model to classify various intrusion behaviours. The network packet distribution of the three datasets is summarized in Table 2.

4.2. Experiment Design. There are two improvements over previous approaches in our strategy. First, the feature extraction method is implemented by GoogLeNet, which is considered to have better feature extraction abilities than does the traditional method based on data mining. Second, our method uses multiple network packets to conduct intrusion detection through the LSTM network and adds the attention mechanism to guarantee the validity of long sequence detection. It considers the time-series variation characteristics of data, which can greatly improve the accuracy of intrusion detection.

Therefore, we first compared the two feature extraction methods, using GoogLeNet and traditional feature extraction methods to extract feature vectors of the same dimension, following which we input the feature vectors into the LSTM network for time-series detection. By comparing the change of cross-entropy loss function, we compared the effectiveness of the two types of feature vectors while performing classification. The cross entropy is commonly used in multiple classification problems and is calculated as the loss function using the following equation:

$$\text{cross entropy} = \sum_{i=1}^{n} -y_i \log(h_i) - (1 - y_i)\log(1 - h_i), \quad (5)$$

where n is the number of classification types, y_i is the predicted value, and h_i is the label value. The gradient descent algorithm is used to minimize the loss function to train the model. Furthermore, multiple control experiments were conducted to compare the intrusion detection performance of the proposed approach with those of three other approaches, which are described below:

(1) Intrusion detection based on the LSTM network: the difference between this approach and the proposed one lies in the feature extraction method used. This approach used the conventional method while the proposed approach uses GoogLeNet to extract the features.

(2) Intrusion detection based on GoogLeNet: the difference between this approach and the proposed method is related to the consideration of the time-series relationships of the data. This approach used GoogLeNet to extract the features of the encoded packets, after which their dimensions are converted into the dimensions required by using the full connection layer. Subsequent to this step, the Softmax classifier is used to output the result.

(3) Intrusion detection based on the random forest (RF) algorithm: the RF algorithm shows a better classification performance when used for conventional machine learning tasks. This approach uses the conventional method to extract the features and the RF algorithm to classify the packets.

The commonly used metrics for evaluating the performance of industrial control intrusion detection methods are the accuracy (ACC), false-positive rate (FPR), and miss rate

Table 1: Breakup of dataset used.

Type of packet	Label	Description	Number
Normal	0	Normal	214580
NMRI	1	Inject random response packets	7753
CMRI	2	Hide real state of controlled process	13035
MSCI	3	Inject malicious state commands	7900
MPCI	4	Inject malicious parameters commands	20412
MFCI	5	Inject malicious function code commands	4898
DoS	6	Denial of service targeting communication link	2176
Recon	7	Pretend to read from devices	3874

Table 2: Network packet distribution of three datasets.

Dataset	Normal	NMRI	CMRI	MSCI	MPCI	MFCI	DoS	Recon
Training	129194	4505	7781	5046	12342	2770	1090	2272
Validation	42678	1593	2673	1502	4416	836	524	778
Test	42708	1655	2581	1352	3654	1292	562	824

(MR) [21]. These evaluation indicators are calculated as shown in the following equations:

$$\begin{aligned} \mathrm{ACC} &= \left(1 - \frac{\mathrm{sum}_{\mathrm{error}}}{\mathrm{sum}}\right) \times 100\%, \\ \mathrm{FPR} &= \frac{\mathrm{attack}_{\mathrm{error}}}{\mathrm{sum}_{\mathrm{normal}}} \times 100\%, \\ \mathrm{MR} &= \frac{\mathrm{normal}_{\mathrm{error}}}{\mathrm{sum}_{\mathrm{attack}}} \times 100\%, \end{aligned} \tag{6}$$

where $\mathrm{sum}_{\mathrm{error}}$ is the total number of packets detected incorrectly, sum is the total number of samples, $\mathrm{attack}_{\mathrm{error}}$ is the total number of normal packets detected incorrectly, $\mathrm{sum}_{\mathrm{normal}}$ is the total number of normal packets, $\mathrm{normal}_{\mathrm{error}}$ is the total number of intrusion packets detected incorrectly, and $\mathrm{sum}_{\mathrm{attack}}$ is the total number of intrusion packets. We will use these evaluation indicators for method validation and comparison.

The hardware device used for the evaluation tests is a computer with the Ubuntu 16.04 as the operating system that has a memory of 8 GB and includes a GTX1050ti graphics card. We complete the coding and experiment based on the TensorFlow platform and Python language.

During the model training process, the training of the two networks separately will result in local optimisation, and the classification results of the model will be affected. Hence, the two networks in the GoogLeNet-LSTM model need to be trained together to solve this problem. The parameters for the two networks and the learning rate are adjusted to ensure global optimisation. Because the model is complex and has too many parameters, it can easily lead to overfitting. The dropout method is added to the model to make some neurons useless in the training process and make all neurons work in the testing process. This method can improve the generalization ability of the model. The dropout ratio is set to 0.8 after adjusting on the validation set.

Models based on complex neural networks have many parameters, of which the time step and learning rate have the greatest influence on the results. Multiple control tests were performed on the same training set, and the indicators for different models for the same validation set were determined. The results of these control tests are listed in Table 3.

During intrusion detection, the value of MR should be as low as possible in the case of a high ACC in order to ensure the effective detection of intrusions. As can be seen from the table, the FPR and MR were both relatively low when the time step of the LSTM network was set to 50 and the learning rate was set to 0.05, even though the ACC was not the highest in this case. In particular, the MR was only 2.08%, which was the lowest of these parameter values.

4.3. Comparison of Feature Extraction Methods. In the proposed intrusion detection approach, GoogLeNet is used to extract the features instead of a conventional method. Control tests were conducted using different feature extraction methods for the same parameters and training set. The parameters for the conventional feature extraction method were taken from the literature [13]. The loss function used was cross-entropy loss, which was recorded during the training process. The cross-entropy losses of the two approaches are shown in Figure 8.

As can be seen from the figure, the cross-entropy loss converges to zero after approximately 200 iterations in the case of the conventional feature extraction method. In contrast, it converges quickly to zero after only approximately 100 iterations when GoogLeNet is used. This indicates that the model based on GoogLeNet is more sensitive to the extracted features and easier to train. The final cross-entropy loss using GoogLeNet is lower than that for the conventional method, indicating that the features extracted using GoogLeNet would be more useful for classification. This confirmed that the intrusion detection approach based on feature extraction using GoogLeNet would exhibit better performance.

4.4. Comparison of Intrusion Detection Approaches. Next, multiple control experiments were conducted to compare

Table 3: Results of control tests.

Parameters	ACC (%)	FPR (%)	MR (%)
Time step = 20, learning rate = 0.01	97.97	1.33	4.56
Time step = 20, learning rate = 0.05	97.15	2.87	2.77
Time step = 20, learning rate = 0.10	97.13	2.64	3.72
Time step = 50, learning rate = 0.01	97.87	1.45	4.60
Time step = 50, learning rate = 0.05	97.88	2.13	2.08
Time step = 50, learning rate = 0.10	97.56	2.37	2.69
Time step = 80, learning rate = 0.01	97.03	2.68	3.98
Time step = 80, learning rate = 0.05	97.01	3.12	2.52
Time step = 80, learning rate = 0.10	97.11	2.88	2.92

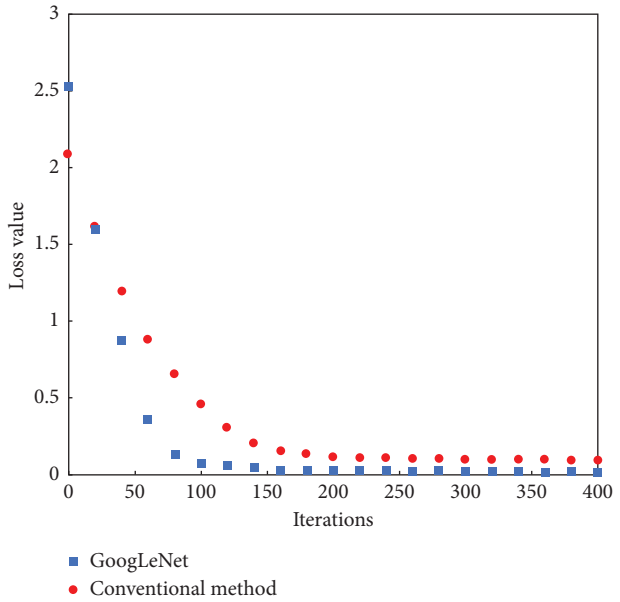


Figure 8: Cross-entropy losses of two approaches.

the intrusion detection performance of the proposed approach with those of three other approaches, which are described below:

(1) Intrusion detection based on LSTM network: the difference between this approach and the proposed scheme lies in the feature extraction method used. This approach used the conventional method while the proposed approach uses GoogLeNet to extract the features.

(2) Intrusion detection based on GoogLeNet: the difference between this approach and the proposed strategy is related to the consideration of the time-series relationships of the data. This approach used GoogLeNet to extract the features of the encoded packets, while the proposed approach uses the Softmax classifier to output the detection results directly at the end of the network.

(3) Intrusion detection based on the random forest (RF) algorithm: the RF algorithm shows better classification performance when used for conventional machine learning tasks. This approach uses the conventional method to extract the features and the RF algorithm to classify the packets.

All four approaches used the same key parameters to build the models as well as the same training dataset for model training. The evaluation indicators corresponding to the same dataset were recorded for comparison and are shown in Table 4.

As can be seen from the table, the approach based on the RF algorithm showed poor detection performance, exhibiting an FPR of 5.71% and MR of 6.52%. In contrast, the detection performance of the approach based on GoogLeNet was significantly better, as it could deep-mine the data features because of its complex structure. This approach showed an FPR of 3.12%, MR of 4.06%, and an ACC of 96.67%. Furthermore, the approach based on the LSTM network exhibited a higher MR at 3.38% because it used a packet sequence to perform time-series level detection. As a result, it could detect a greater number of intrusions. However, its other indicators were slightly lower (FPR of 3.56% and ACC of 96.48%) because it used the conventional method for feature extraction. On the other hand, the proposed approach, which is based on GoogLeNet and an LSTM network, could not only detect a greater number of intrusions but also did so with greater accuracy. The evaluation indicators for this approach were the highest with an FPR of 2.42% and MR of 2.51%. Moreover, the final ACC of this approach was nearly one percentage point higher than those of current deep learning approaches at 97.56%.

We use the detection ratios to verify the detection ability of the approach for different types of intrusions. The detection rate is defined as the proportion of detected samples among all intrusion samples. The detection ratios of the four approaches for seven different types of intrusions are plotted in Figure 9 to highlight their performances.

As can be seen from the figure, the detection ratios of the four approaches for MFCI and Recon intrusions were the highest (greater than 98%) because these two types of intrusions merely change the function code. Furthermore, the detection ratios of the proposed approach and GoogLeNet for malicious state command injection (MSCI) and malicious parameter command injection (MPCI) attacks were also high (greater than 96%). This was because of their deep mining features, while those of the approaches based on the LSTM network and the RF algorithm were lower than 96% because they used the conventional feature extraction method. The detection ratios of all four approaches for naïve malicious response injection (NMRI), complex malicious response injection (CMRI), and denial of service (DoS) attacks were much lower than those for the other intrusions. This was because the former interferes with the normal system operations and cannot be distinguished from single network packets. The proposed approach and that based on the LSTM network exhibited higher detection ratios (greater than 96%) for these intrusions because they considered the time-series relationships of the data. In contrast, the approaches based on GoogLeNet and the RF algorithm had detection ratios of less than 94% for these intrusions. Thus, the proposed approach combines the advantages of the other

Table 4: Evaluation indicators for four approaches.

Model	ACC (%)	FPR (%)	MR (%)
Proposed approach	97.56	2.42	2.51
LSTM network	96.48	3.56	3.38
GoogLeNet	96.67	3.12	4.06
RF algorithm	94.11	5.71	6.52

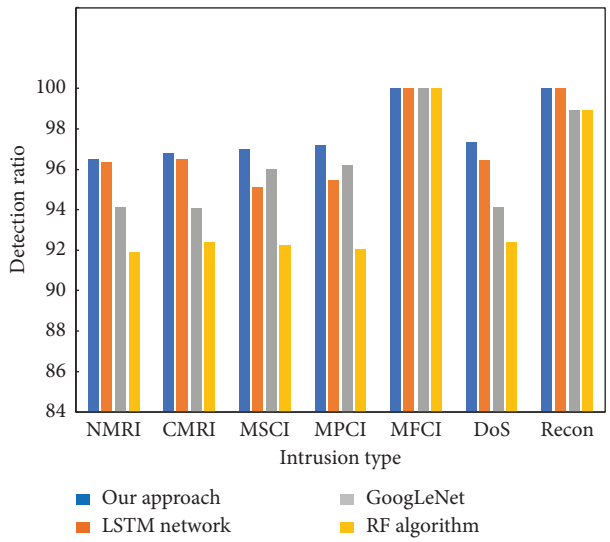


Figure 9: Detection ratios of four approaches for seven types of intrusions.

approaches and therefore has significantly better detection performance. The detection ratios of the proposed approach for NMRI, CMRI, MSCI, MPCI, MFCI, DoS, and Recon intrusions were 96.50%, 96.78%, 96.97%, 97.21%, 100%, 97.33%, and 100%, respectively.

5. Conclusions

Deep learning can significantly improve the accuracy of intrusion detection. An industrial control intrusion detection approach based on a multiclassification GoogLeNet-LSTM model was proposed and evaluated in this study. To begin with, the network packet categorisation method was used to classify the packets in order to determine which detection method to use. Next, the template comparison approach was used to detect an intrusion in the case of network packets that do not carry any information. In the case of the packets carrying information, a packet sequence reflecting the normal system operation status was constructed, and GoogLeNet was used for feature extraction. The obtained feature vectors were input into an LSTM network for process-oriented time-series level detection while a Softmax classifier was used for the multiclassification of the detection results. This approach has the following advantages:

(1) Different intrusion detection methods are used based on the type of network packets that employ the Modbus protocol. This improves the detection accuracy and efficiency.

(2) GoogLeNet is used instead of conventional methods for feature extraction. The former can extract more accurate features and has a lower MR. Furthermore, it has fewer parameters and can serve as a lightweight feature extraction method for industrial control intrusion detection methods.

(3) An attention mechanism is used in the LSTM network for time-series level detection. It improves the learning ability of the model in the case of long sequences and reduces the FPR. As a result, the model performs detections that are more accurate.

(4) The classification of the detection results into multiple categories can help the operators or hosts determine the intrusion type and take the appropriate countermeasures.

Data Availability

Reference [20] refers to the dataset used in this research. The dataset can also be accessed from the webpage https://sites.google.com/a/uah.edu/tommy-morris-uah/ics-data-sets.

Conflicts of Interest

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

Acknowledgments

This study was supported by the National Natural Science Foundation of China (61872015), the Qinghai Province Natural Science Foundation (2017-ZJ-91), the Foundation of Science and Technology on Information Assurance Laboratory (614211204031117), the Beijing Polytechnic Research Fund (2017Z004-008-KXZ), the Industrial Internet Innovation and Development Project (Typical application and promotion project of the security technology for the electronics industry) of the Ministry of Industry and Information Technology of China in 2018, the Foundation of Shaanxi Key Laboratory of Network and System Security (NSSOF1900105), and the International Research Cooperation Seed Fund of Beijing University of Technology (2018-B9).

References

[1] The Modbus Organization, *Modbus Application Protocol Specification*, The Modbus Organization, Hopkinton, MA, USA, 2006.

[2] W. L. Shang, P. F. An, M. Wan et al., "Research and development overview of intrusion detection technology in industrial control system," *Application Research of Computers*, vol. 34, no. 2, pp. 328–333, 2017.

[3] A. A. Cárdenas, S. Amin, and S. Sastry, "Research challenges for the security of control systems," in *Proceedings of the Conference on Hot Topics in Security*, USENIX Association, Berkeley, CA, USA, 2009.

[4] M. A. Ambusaidi, X. He, P. Nanda, and Z. Tan, "Building an intrusion detection system using a filter-based feature

selection algorithm," *IEEE Transactions on Computers*, vol. 65, no. 10, pp. 2986–2998, 2016.

[5] D. K. Sadhasivan and K. Balasubramanian, "A fusion of multiagent functionalities for effective intrusion detection system," *Security and Communication Networks*, vol. 7, pp. 1–15, 2017.

[6] J. M. Beaver, R. C. Borges-Hink, and M. A. Buckner, "An evaluation of machine learning methods to detect malicious SCADA communications," in *Proceedings of the 2013 12th International Conference on Machine Learning and Applications*, vol. 2, pp. 54–59, IEEE, Miami, FL, USA, December 2013.

[7] J. Li, H. Z. Wang, and D. Q. Chen, "A study on intrusion detection of industrial control system based on improved bat algorithm," *Journal of East China University of Science and Technology (Natural Science Edition)*, vol. 5, pp. 328–333, 2017.

[8] J. Ren, J. Guo, W. Qian, H. Yuan, X. Hao, and H. Jingjing, "Building an effective intrusion detection system by using hybrid data optimization based on machine learning algorithms," *Security and Communication Networks*, vol. 9, no. 2, p. 238, 2019.

[9] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[10] A. Y. Javaid, Q. Niyaz, W. Sun et al., "A deep learning approach for network intrusion detection system," in *Proceedings of the 9th EAI International Conference on Bio-Inspired Information and Communications Technologies*, pp. 21–26, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), New York, NY, USA, December 2015.

[11] T. A. Tang, L. Mhamdi, D. Mclernon et al., "Deep learning approach for network intrusion detection in software defined networking," in *Proceedings of the International Conference on Wireless Networks and Mobile Communications*, pp. 258–263, IEEE, Fez, Morocco, October 2016.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the 26th Conference on Neural Information Processing Systems (NIPS 2012)*, pp. 1097–1105, MIT Press, Lake Tahoe, CA, USA, December 2012.

[13] J. Liang, J. H. Chen, X. Q. Zhang, Y. Zhou, and J. J. Lin, "Anomaly detection based on one-hot encoding and convolutional neural network," *Journal of Tsinghua University*, vol. 25, pp. 1–7, 2019.

[14] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[15] F. Cheng, T. T. Li, and D. Chana, "Multi-level anomaly detection in industrial control systems via package signatures and LSTM networks," in *Proceedings of the 47th IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2017)*, pp. 261–272, IEEE, Denver, CO, USA, July 2017.

[16] Y. Lai, K. Yang, J. Liu, and Z. Liu, "An industrial control network protocol vulnerability mining method based on fuzzy testing," *Computer Integrated Manufacturing Systems*, vol. 25, no. 9, pp. 2265–2279, 2019.

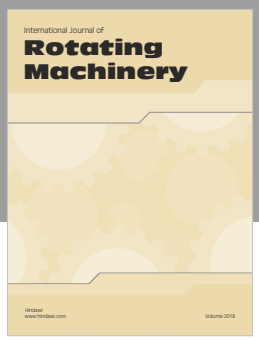
[17] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2818–2826, Las Vegas, NV, USA, June 2016.

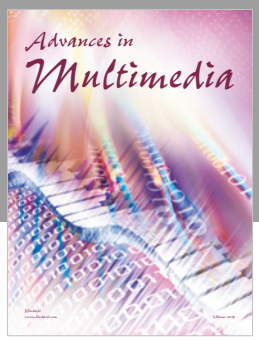
[18] A. Vaswani, N. Shazeer, N. Parmar et al., "Attention is all you need," in *Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017)*, pp. 5998–6008, MIT Press, Long Beach, CA, USA, December 2017.

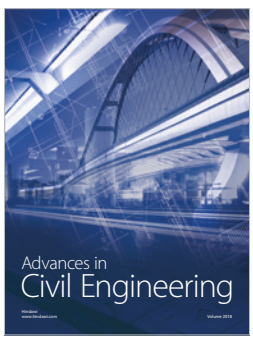
[19] Y. Lai, Z. Liu, Z. Song, Y. Wang, and Y. Gao, "Anomaly detection in industrial autonomous decentralized system based on time series," *Simulation Modelling Practice and Theory*, vol. 65, pp. 57–71, 2016.

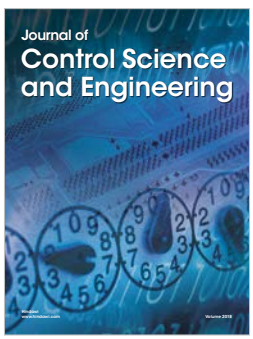
[20] I. P. Turnipseed, *A New SCADA Dataset for Intrusion Detection System Research*, Mississippi State University, Starkville, MS, USA, 2015.

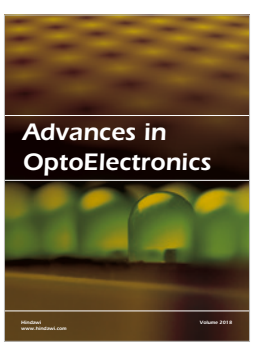
[21] A. Yang, L. M. Sun, X. S. Wang, and Z. Q. Shi, "Overview of intrusion detection techniques for industrial control systems," *Journal of Computer Research and Development*, vol. 53, no. 9, pp. 2039–2054, 2016.

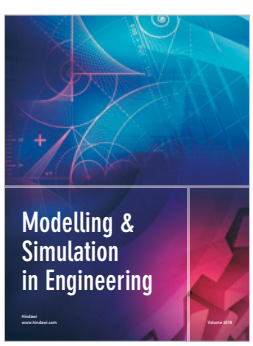


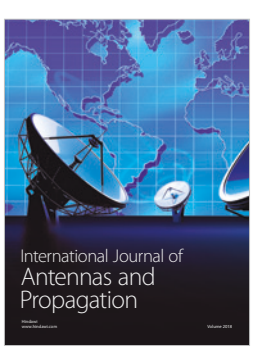


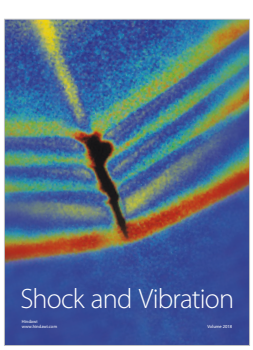


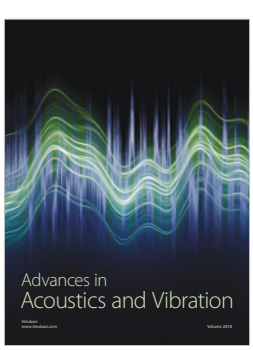












Hindawi

Submit your manuscripts at
www.hindawi.com