Function-Aware Anomaly Detection Based on Wavelet Neural Network for Industrial Control Communication

Ming Wan, Yan Song, Yuan Jing, and Junlu Wang

1. Introduction

Nowadays, almost all CPSs (Cyber-Physical Systems) in critical infrastructures (such as electrical and petrochemical systems, sewage systems, and transportation systems) concerning the national economy and the people's livelihood have developed industrial control systems to realize significant automation of industrial processes [1, 2]. In particular, with the rise of Industry 4.0 and Internet of Things [3, 4], the flexible manufacturing and convenient interoperability has already been brought into schedule by academia and industry. Actually, smart CPSs can unleash strong driving forces for the innovation and integration of industrialization and informatization. As an applicable solution, information communication technologies have a positive influence on strengthening traditional industrial control systems [5]. However, the application of ICTs is gradually breaking the original "information island" status of industrial control systems, and the incoming cybersecurity can be dramatically impacted. In consequence, many experienced engineers shift their focus from the process safety to the information security [6]. Over the past several years, CPSs came under cyberattacks from all sides. According to the ICS-CERT (Industrial Control Systems Cyber Emergency Response Team) statistics [7], the ICS-CERT incident response team generalized and analyzed 290 industrial security incidents in 2016, and more and more sophisticated attacks against industrial control systems are developed by the adversaries. Actually, three comprehensible causes in such a situation can be recognized as follows: (1) multifarious vulnerabilities of industrial control systems have been exposed gradually in recent years, for example, system architecture vulnerability [8, 9], embedded control device vulnerability [9–11], and industrial communication vulnerability [11, 12]; (2) the types of cyberattacks are distinctive and diversified, and targeted attacks and APTs (Advanced Persistent Threats) have permeated to face reality [13]; (3) industrial-oriented defense technologies are in an underway and exploring stage, and the regular Internet security methods are unable to satisfy the special industrial control requirements [14].
Actually, the basic industrial control operations of existing DCS (Distributed Control System), SCADA (Supervisory Control And Data Acquisition), and PLC (Programmable Logic Controller) range over two aspects: function control and data acquisition [8, 15]. As an essential link in industrial process automation, function control always performs a series of well-organized and synergetic operations in industrial production and manufacture. For instance, in existing automobile manufacturing process, industrial robots under the master controls can complete the assembly of automobiles according to the predetermined procedures. Therefore, once some attacker deliberately destroys or disturbs the function control process by cyberattacks, huge losses may be caused.

In order to combat this tendency, the researchers make a preliminary probe into two types of defense methods: device-oriented and network-oriented cases. In the device-oriented cases, trusted computing for industrial embedded devices [16] is a burgeoning security technology to provide system integrity check and data confidentiality protection. In the network-oriented cases, industrial firewall [11, 17] and intrusion detection [15, 18, 19] are the typical applications in industrial control networks to improve the communication security. However, because we have not understood the boundary conditions between the availability and security of industrial control systems, the cases on trusted computing and industrial firewall may result in the processing delay or transmission delay in industrial process automation. In general, intrusion detection can discover potential intrusion behaviors in real time by collecting and analyzing various industrial network data, and it scarcely has an impact on industrial control communication activities due to the inconspicuous network sniffer. Moreover, intrusion detection in industrial control systems mainly includes signature-based approaches and anomaly-based approaches, and anomaly-based approaches can detect some unknown attacks by automatically classifying significant deviations from a learned normal behavior model [1]. However, two particular problems should deserve careful considerations: (1) extract the appropriate features according to industrial communication characteristics; (2) build an optimal detection model suited for the extracted features.

In this paper, we propose a function-aware anomaly detection approach based on WNN (Wavelet Neural Network) to identify industrial communication intrusions or anomalies. In particular, these intrusions or anomalies may cause the function control changes in industrial control communication. Furthermore, our approach extracts the time-related features from the communication packets to describe the function control characteristics and build an optimal behavior model based on WNN by using the normal function control samples. With the establishment of behavior model, the detection threshold can be calculated as a scale to differentiate the aberrant industrial control communication activities. Finally, a real-world control system is simulated to furnish the analyzed function control data, and the used industrial communication protocol in this system is Modbus/TCP. According to the experimental results, we fully demonstrate that this approach has the fine detection accuracy and adequate real-time capability.

The major contributions and advantages of this paper involve three aspects: Firstly, we propose a novel time-related feature calculation and construction algorithm to adequately describe the function control characteristics, and this algorithm can slickly extract function control behaviors from industrial control communication activities. Secondly, based on the time-related function control behaviors, we introduce the optimized wavelet neural network to realize the function-ware anomaly detection. Finally, a real-world control system is simulated to evaluate our approach, and the experimental results show that our approach is practicable and effective. Actually, the biggest difference of our approach focuses on the first aspect. That is, adequately modeling function control behaviors is one necessary prerequisite to further explore the real-time anomaly detection. In our approach, we design an original function control feature calculation and construction algorithm to overcome this difficulty.

2. Related Work

According to different detection techniques, the anomaly detection approaches in CPSs can involve three major aspects: rule matching, statistics analysis, and computational intelligence [18]. In the rule matching ones, the prior knowledge must be prepared to learn the general rules for intrusion detection and the rule match is executed to detect many kinds of attacks. Typically, Almalawi et al. [20] automatically extract the proximity detection rules from the consistent and inconsistent states of SCADA data to identify integrity attacks on SCADA systems. Genge et al. [21] propose a systematic and auto-configured anomaly detection approach, which includes modeling of ICS networks and generating anomaly detection rules, to identify the attacks violating ICS connection patterns. Due to the predefined rules, these approaches can improve the classification accuracy and have the practical detection efficiency. But the huge rule database is hard to build and update, because the extracted rules must cover all known attack instances. Besides, they also lack the ability to exploit the unknown attacks which frequently occur in today’s CPSs.

In the statistics analysis ones, the underlying distribution (such as the network traffic profile) can be learned to detect anomalies, and these techniques are better able to resist the incomplete and imprecise training data than the rule matching ones. For instance, Do [22] and Gawand et al. [23] introduce the CUSUM mechanism to detect the change point of industrial communication traffic. Different from the rule matching ones, these approaches can attempt to find the weaknesses of the unknown attacks, but this ability is very limited because the sophisticated and targeted attacks can easily bypass the distribution changes. Moreover, the high false positive and negative rate is another drawback because it is difficult to determine the traffic profile. In the computational intelligence-based ones, these techniques always have a strong correlation with data mining. Furthermore, the normal models or profiles are built from multivariate training data, and the corresponding anomaly detection is realized by using the mechanism of classification or optimization. Actually, the computational intelligence techniques have been attracting great interests of both industry and academia, and
many computational intelligence approaches have been researched, mainly including SVM (Support Vector Method) [15, 24, 25], neural network [26], decision trees [27], genetic algorithm [27, 28], and clustering technique [29]. Although the computational intelligence-based techniques have the relatively high computational overhead, they can achieve better performance in detection, tolerance, and generality [14]. Additionally, these approaches can not only detect known attacks with high detection efficiency, but also have a better function in identifying new intrusion modes [15]. It is worth mentioning that our approach belongs to the computational intelligence-based techniques [15, 24, 25], neural network [26], decision trees [27], genetic algorithm [27, 28], and clustering technique [29]. Although these approaches can not only detect known attacks with high detection efficiency, but also have a better function in identifying new intrusion modes [15]. It is worth mentioning that our approach belongs to the computational intelligence-based techniques

3. Function Control Feature Calculation and Construction

In industrial control communication, function codes, which represent control signals sent from the operator or engineer workstations, are distributed to the executive devices for the purpose of controlling industrial automation process. Therefore, the feature calculation and construction algorithm analyzes the time-related function codes to simulate the function control behavior. In particular, we cannot simply gather the function codes at regular time intervals as function control samples to train the behavior model, and the intrinsic reasons include the following: (1) the number of function codes at each regular time interval is distinct, and the prerequisite for the behavior model based on WNN is that the dimensions of input samples must be consistent with one another; (2) the number of function codes at each regular time interval may be very large, and it may waste computational resources and reduce detection efficiency. Figure 1 depicts the detailed feature calculation and construction process, and each step can be outlined below.

Step 1 (function code sequence preprocessing). Like our prior work in [15], in order to associate time characteristics with function control activities, we first parse the captured function control packets in depth and obtain the function code sequence \( F_t = f_1^t f_2^t \cdots f_m^t \) in every interval \( t \) (here, \( m_t \) is the serial number of every function code in \( F_t \)). The function code sequence set \( F_{set} = \{ F_1, F_2, \cdots, F_n \} \) in the interval \( T = T + \sum_{i=1}^{t} t \) can consist of all function code sequences \( F_i \) (\( i = 1, \cdots, n \)), and all sequence dimensions in \( F_{set} \) can separate from each other because of the different \( m_t \) in each sequence. After that, we recombine all \( F_i \) (\( i = 1, \cdots, n \)) to the one big sequence \( F \) according to the time order.

Step 2 (feature factor selection). Because the dimensions of obtained function control samples \( X_i \) (\( i = 1, \cdots, n \)) must be consistent, we first need to construct the feature base vector according to the selected feature factors. In particular, the selected feature factors consist of two main components: single function code and short sequence pattern. More specifically, all single function codes \( f_1, f_2, \cdots, f_\mathcal{V} \) are searched in sequential order from the large function code sequence \( F \), and each single function code is different from the others. According to each single function code \( f_j \) (\( j \in [1, \mathcal{V}] \)), we design the short sequence pattern \( M_{jk}^i \) (\( k \in [1, \mathcal{V}] \)). Furthermore, \( M_{jk}^i \) consists of \( f_j \) and \( f_k \), and we can get \( \mathcal{V} \) short sequence patterns \( M_{1j}^i, M_{2j}^i, \cdots, M_{\mathcal{V}j}^i \) for each single function code \( f_j \). Taken together, we can acquire the feature base vector \( f_1, M_{1j}^i, M_{2j}^i, \cdots, M_{\mathcal{V}j}^i, f_j, M_{1j}^i, M_{2j}^i, \cdots, M_{\mathcal{V}j}^i \). The intrinsic reasons for such feature factor selection include the following: (1) the single function code can represent its own role in each function code sequence; (2) the short sequence pattern can establish the relationship between two function codes and indirectly reflect the continuous control operations in industrial automation process.

Step 3 (function control sample calculation). According to the feature base vector, we further calculate the corresponding feature variable for each feature factor in the function code sequence \( F_i \). For the single function code \( f_j \), we regard its frequency \( x_{ji} \) in \( F_i \) as the corresponding feature variable, and the calculation formula is \( x_{ji} = g(f_j)/m_t \), here \( g(f_j) \) represents the number of \( f_j \) in \( F_i \). For the short sequence pattern \( M_{jk}^i \), we calculate its frequency \( x_{jk}^i \) in \( F_i \) as the corresponding feature variable by the formula \( x_{jk}^i = g(M_{jk}^i)/(m_t - 1) \). By calculating all feature variables in \( F_i \), we can complete the construction of the sample \( X_i = (x_{11}, x_{12}, \cdots, x_{1\mathcal{V}}, \cdots, x_{n1}, x_{n2}, \cdots, x_{n\mathcal{V}}) \). Additionally, there is a one-to-one correspondence between function code sequences and function control sample, and each function control sample contains \( \mathcal{V}(1 + \mathcal{V}) \) feature variables. To sum up, all calculated function control samples form the function control sample set \( X_{set} = \{ X_1, X_2, \cdots, X_n \} \).

4. Function-Aware Anomaly Detection Based on Wavelet Neural Network

After the feature construction, we can train a wavelet neural network to discover any functional change in industrial control communication. Moreover, we introduce the WNN’s prediction capability to realize that the function-aware anomaly detection, an optimized WNN, and the correlative detection threshold are achieved by the loop-based iterative train.

4.1. Architecture Design. Figure 2 shows the overall architectural design of function-aware anomaly detection based on WNN. As this figure shows, this detection approach is made up of two phases: model training and real-time detection. Actually, model training is an essential step or a prerequisite in order to improve the detection accuracy. In this phase, by using the training function control samples extracted from the normal industrial communication data, an optimized WNN-based behavior model is successfully built and an accompanying detection threshold (including an upper limit and a lower limit) is measured and recorded.
Figure 1: Detailed feature calculation and construction process according to function control characteristics.

Figure 2: Overall architectural design of function-ware anomaly detection approach based on WNN.
In the real-time detection phase, industrial communication data are captured and parsed in depth to form the test function control samples by means of the feature calculation and construction algorithm mentioned in Section 3, and the optimized wavelet neural network analyzes these input test samples to calculate the predicted results, which are further compared with the detection threshold. When the predicted results are not covered by the detection threshold, an alarm will be generated in real-time.

4.2. Wavelet Neural Network and Optimization. WNN has already been successfully applied to many practical areas [30], and in our approach it is introduced as the critical behavior model to identify function control misbehaviors. In practice, the topological structure of WNN evolves from BP neural network, and it regards the wavelet basis function as the activation function of hidden layer wavelets, which are referred to as the hidden units. In the hidden layer, the input variables are inserted and transformed to wavelets, and all wavelets are combined to estimate the approximation of the target values [31].

In the WNN’s structure depicted in Figure 2, \( x_1, x_2, \cdots, x_m \) are the input variables in the input layer, and \( y_1, y_2, \cdots, y_m \) are the predicted results in the output layer. Additionally, \( \omega_{ij} \) and \( \omega_{jk} \) stand for the network weights. If the input variables are \( x_i (i = 1, 2, \cdots, k) \), the corresponding outputs can be given by the expression:

\[
\begin{align*}
    h(j) &= h_j \left( \frac{g(x) - b_j}{a_j} \right) \\
    g(x) &= \sum_{i=1}^{k} \omega_{ij} x_i,
\end{align*}
\]

(1)

Here, \( h(j) \) is the output of the hidden unit \( j \) in the hidden layer; \( \omega_{ij} \) is the connection weight between the input layer and the hidden layer; \( b_j \) represents the translation parameter of the wavelet basis function \( h_j \); \( a_j \) represents the dilation parameter of the wavelet basis function \( h_j \).

In our wavelet neural network, the Morlet wavelet is selected as the wavelet basis function, given by

\[
y = \cos(1.75x) e^{-x^2/2}
\]

(2)

After the calculation of the hidden layer, we can further obtain the predicted results by the following expression:

\[
y(k) = \sum_{j=1}^{l} \omega_{jk} h(j), \quad k = 1, 2, \cdots, m
\]

(3)

Here, \( \omega_{jk} \) is the connection weight between the hidden layer and the output layer; \( h(i) \) is the output of the hidden unit \( i \) in the hidden layer; \( l \) is the number of the hidden units; \( m \) is the number of the output units.

It is worth mentioning that we use the loop-based iteration to train the optimized WNN, and its main purpose is to improve the network parameters, including the connection weights \( \omega_{ij} \) and \( \omega_{jk} \), the translation parameter \( b_j \), and the dilation parameter \( a_j \). Furthermore, the predicted error is introduced to shorten the distance between the predicted results and the expected outputs, and the predicted error can be computed by

\[
e_p = \sum_{k=1}^{m} \left[ y'(k) - y(k) \right]
\]

(4)

Here, \( y'(k) \) is the expected output, and \( y(k) \) is the predicted result.

Algorithm 1 shows the pseudocode of WNN’s optimization process. In this process, the parameter increments are introduced to update all network parameters, and the specific process can refer to the WNN’s training steps in Section 4.3. In practice, two different terminations of iteration for this process can be selected: one is the maximum number of iterations, and the other is the preconfigured error threshold which indicates the iteration is completed if the distance between the predicted results and the expected outputs is small enough. In our approach, we select the first one as the terminal condition.

4.3. Training and Detection. As mentioned in Section 4.1, the main steps of model training are outlined below.

**Step 1** (network parameter initialization). We first initialize the primary parameters of wavelet neural network, including the dilation parameter \( a_j \), the translation parameter \( b_j \), and the connection weights \( \omega_{ij} \) and \( \omega_{jk} \). Additionally, we also set the learning rate, which is used to improve the above parameters.

**Step 2** (predicted error calculation). According the training function control samples, we calculate the predicted error by (4).

**Step 3** (parameter modification). On the basis of the predicted error, we further improve the network parameters to shorten the distance between the predicted results and the expected outputs.

**Step 4** (detection threshold measurement). Finally, we repeat Steps 2 and 3 until the iteration ended and record the optimized detection threshold.

After the model training, we can perform the real-time detection to identify function control misbehaviors. Moreover, the basic prerequisite is that we must resolve the real-time function control samples from the observed industrial communication data by using our feature calculation and construction algorithm. As the input variables, these samples can further be analyzed by the optimized wavelet neural network to estimate the predicted results, which will be compared with the detection threshold. The judgment criterion to generate an alarm is that if the predicted results fall within the range from the lower limit to the upper limit, we can believe these function control activities are normal; if the
predicted results escape from these ranges, we may doubt the corresponding function control activities are abnormal.

5. Experimental Analysis and Discussion

5.1. Experimental Modbus/TCP Control System. In order to evaluate the detection performance, we use the simulation control system which is built in our earlier work [15] to furnish the analyzed function control data. Furthermore, the industrial control communication of this system is based on Modbus/TCP, in which various function codes are utilized to facilitate different control operations. Figure 3 shows the basic network architecture of this control system. Furthermore, the chief purpose of this system is to accomplish the material production by monitoring and controlling the valves and the liquid levels, and the detailed technological process has been presented in [15]. In particular, the whole technological process is repeated every 1 minute. Besides, in this control system we carry out some attack experiments to forge and replay some malicious Modbus/TCP packets, and our ultimate goal is to evaluate the detection accuracy and real-time capability by using these malicious function control data.

The normal communication packets are captured from the industrial switch to train the optimized wavelet neural network, and the capture time lasts 1h15m02s. After the preliminary statistics, the number of Modbus/TCP function codes in these packets reaches 11693. Additionally, we also use Matlab to analyze these packets in depth, and the hardware configurations are also the same with the ones in [15]. Per one minute, we compute the number of different function codes, and all statistical results are shown in Figure 4. As these results show, the simulation control system uses four categories of function codes to complete the whole technological process, and these function codes are 1, 3, 5, and 6, respectively. Besides, all five curves in this figure flatten out, and the number of every function code fluctuates smoothly. In brief, these results can also demonstrate that the simulation control system has the relatively steady communication patterns under normal circumstances, and its function control status appears on a relatively limited range.

5.2. Detection Performance Evaluation. Without loss of generality, we choose the detection accuracy and real-time capability as the main performance indicators to evaluate our approach. Before training the optimal behavior model, we first preprocess the captured Modbus/TCP packets. More specifically, we extract the function codes per 1 minute to form the function code sequences, and by using the feature calculation and construction algorithm we win a total of 75 function control samples. Because 4 function codes exist in the simulated technological process, each function control sample contains 20 feature variables. According to these normal function control samples, we further train and optimize the wavelet neural network. It is worth mentioning that we set the number of iterations to 200 in order to reduce the predicted error, and Figure 5 plots the change curve of predicted errors with the iteration times. From this figure we can see that, along with the increasing of iteration times, the curve of predicted errors changes from rapid reduction to gentle trend. In particular, the best detection accuracy in the 200th iteration can reach 98.67%; that is, the predicted accuracy of the optimized WNN to detect 75 normal function control samples can reach 98.67%, and only one normal function control sample is mistakenly regarded as the outlier.

By using the optimal behavior model, we further evaluate its detection performance, including detection accuracy and consuming time. In each experiment, we forge and replay some malicious Modbus/TCP packets to attack and destroy the normal technological process. Moreover, we suppose that these malicious Modbus/TCP packets cannot contain other function codes which are different with the four categories

```plaintext
Do initializing network parameters (ω𝑖,ω𝑗,α, b)
Do initializing parameter increments (Δω𝑖,Δω𝑗, Δα, Δb)
Do setting iteration number IntNum
for l to IntNum
   for all input variables
      for l to l
         Do computing h(j) and g(x) by Eq. (1)
         Do computing y(k) by Eq. (3)
      end
      Do computing and recording predicted error e by Eq. (4)
      for l to l
         Do correcting parameter increment (Δω𝑖, Δω𝑗, Δα, Δb)
      end
      Do updating connection weights ω𝑖, and ω𝑗 by Δω𝑖, and Δω𝑗
      Do updating dilation parameter α by Δα
      Do updating translation parameter b by Δb
   end
end

Algorithm 1: Pseudocode of WNN’s optimization process.
```
Figure 3: Basic network architecture of simulated Modbus/TCP control system [15].

Figure 4: Statistical results of different function code number.
of function codes used in the simulation control system, and these packets only change the function control process. The major reason of such assumption is that the malicious packets containing other function codes can be easily filtered by the applied industrial firewall [11, 32]. Besides, we generate 60 malicious function code sequences in each experiment. More specially, the percentage of the malicious function codes in each function code sequence is about 1/10, and the locations of the malicious function codes in each function code sequence can be considered random. Similarly, we can obtain 60 malicious function control samples in each experiment after the feature calculation and construction. By calculating the predicted result for each malicious function control sample, we compare it with the detection threshold to identify the corresponding abnormal function control behavior. Table 1 shows the experimental results of detection performance under 10 different experiments in detail. In this table, the average detection accuracy is 91.17%, and the average consuming time is 0.0104s. In the extreme case, the smallest detection accuracy is 88.33% in the 3rd, 7th, and 10th experiments, and the largest consuming time is only 0.0281s to detect 60 function control samples in the 7th experiment. In a word, we fully demonstrate the function-aware anomaly detection approach has the fine detection accuracy and adequate real-time capability; namely, they indirectly declare it has the remarkable capacity to differentiate the abnormal function control activities.

Actually, the adversary can change the attack intensity by adjusting the attack frequency; for example, the sending rate of malicious Modbus/TCP packets can be increased by the adversary to launch an attack with a higher probability of success. Therefore, the percentage of the malicious function codes in each function code sequence may also change accordingly. However, the different percentages of the malicious function codes in each function code sequence can have a marked impact on the detection accuracy of our approach. In order to define different influencing effects, we suppose the percentages of the malicious function codes are 1/5, 1/10, 1/15, 1/20, 1/25, and 1/30, and 5 distinct experiments are performed for each percentage. Similarly, we also generate 60 malicious function control samples in each experiment. Figure 6 plots the detection accuracy variation under different percentages of the malicious function codes in each function code sequence. In this figure, \( p_1, p_2, \cdots, p_6 \) represent the percentages whose values are 1/5, 1/10, 1/15, 1/20, 1/25, and 1/30, respectively, and the minimum detection accuracies, the average detection accuracies, and the maximum detection accuracies are plotted according to every 5 experiments. Viewed generally, the experimental results reflect the detection accuracy also decreases with the reduction of the percentage; that is, our approach can be more effective in detecting the function control misbehavior caused by the larger percentage of the malicious function codes. However, our approach still maintains a high detection accuracy; for instance, when the percentage is 1/30, the average detection accuracy can reach 76.67%. Additionally, Figure 7 shows the average consuming time under different percentages of the malicious function codes. From this figure, we can see that the consuming time fluctuates remarkably in a narrow range. In other words, the different percentages have almost no influence on the consuming time.

| Detection accuracy | Consuming time |
|--------------------|----------------|
| 1 90.00%          | 0.0066s        |
| 2 91.67%          | 0.0101s        |
| 3 88.33%          | 0.0072s        |
| 4 93.33%          | 0.0068s        |
| 5 90.00%          | 0.0066s        |
| 6 95.00%          | 0.0075s        |
| 7 88.33%          | 0.0281s        |
| 8 91.67%          | 0.0127s        |
| 9 95.00%          | 0.0099s        |
| 10 88.33%         | 0.0084s        |

Average value 91.17% 0.0104s

5.3. Compared Analysis. In practice, the innovations of our approach mainly include two aspects: (1) we propose a new feature calculation and construction algorithm to extract function control characteristics in industrial control communication; (2) according to the extracted function control samples, we introduce the optimal function-aware WNN model to differentiate the aberrant industrial control communication activities. Therefore, we also provide the compared analysis to explain its advantages from these two aspects.

For one thing, compared with the work in [15, 25], the feature calculation and construction algorithm in this paper can learn more information about function control characteristics from industrial communication packets. On the other hand, this algorithm selects the single function code...
as an independent feature factor to enhance its own role effect in each function code sequence. On the other hand, the short sequence patterns include all adjacent cases of two function codes, and they not only indirectly reflect the continuous control operations in the normal technological process, but also consider the impact of two nonadjacent control operations in the actual technological process. Therefore, more information about function control characteristics can be utilized to improve the detection efficiency.

Based on the same sample extraction by using the proposed feature calculation and construction algorithm, we compare our approach with BP neural network to evaluate the detection accuracy and explain that the proposed approach is more suitable and applicable to detect function control misbehaviors. Similarly, we also perform 10 experiments, and the function control samples in each experiment are the same with the ones whose percentage of the malicious function codes in each function code sequence is about $1/10$. Figure 8 plots the detection accuracy comparison between our approach and BP neural network under 10 experiments, and Table 2 shows the corresponding average detection accuracies of two approaches. From these results we can see that BP neural network has a relatively large fluctuation of the detection accuracy, and its average detection accuracy is only 83.50% which is lower than the one of our approach. Therefore, our approach has the ability to provide the better detection accuracy.

Table 2: Average detection accuracies of our approach and BP neural network.

|                  | Average detection accuracy |
|------------------|---------------------------|
| Our approach     | 91.17%                    |
| BP neural network| 83.50%                    |

Figure 6: Detection accuracies under different percentages of the malicious function codes in each function code sequence.

Figure 7: Average consuming time under different percentages of the malicious function codes in each function code sequence.

Figure 8: Detection accuracy comparison between our approach and BP neural network under 10 experiments.
6. Conclusion
Aiming at differentiating the aberrant industrial control communication activities, this paper proposes a function-aware anomaly detection approach based on WNN. Firstly, we design the feature calculation and construction algorithm to learn the function control characteristics and extract the time-related features. Secondly, a behavior model based on WNN is established and optimized to detect function control misbehaviors in industrial control communication. Finally, in order to evaluate our approach, we simulate a real-world control system based on Modbus/TCP to perform plenty of experiments, and the experimental results and the compared analysis are offered to express the advantages: our approach has the fine detection accuracy and adequate real-time capability.

Data Availability
In this manuscript, the analyzed function code data are captured and analyzed from our simulation control system, which is built to accomplish the material production according to one real-world control system. Actually, we have sketched the basic technological process in this manuscript, but some contents and specific parameters of this process are not completely open to the public due to the commercialized secrets. Therefore, the analyzed function code data used to support the findings of this study are currently under embargo. If other researchers want to verify the results, replicate the analysis, or conduct secondary analyses, please contact with the corresponding author or first author. The requests for the data will be considered by them after a confidentiality agreement.

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

Acknowledgments
This work is supported by the National Natural Science Foundation of China (Grant No. 61501447), Intelligent Manufacturing Project of the Ministry of Industry and Information Technology: Industrial Internet Data Mutual Recognition Research – Low-Power Message Distribution, and the General Project of Scientific Research of Liaoning Provincial Department of Education (LYB201616).

References
[1] C. Zhou, S. Huang, N. Xiong et al., “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.
[2] F. Kargl, R. W. Van Der Heijden, H. König, A. Valdes, and M. C. Dacier, “Insights on the security and dependability of industrial control systems,” IEEE Security & Privacy, vol. 12, no. 6, pp. 75–78, 2014.
[3] M. Wollschlaeger, T. Sauter, and J. Jasperneite, “The future of industrial communication: Automation networks in the era of the internet of things and industry 4.0,” IEEE Industrial Electronics Magazine, vol. 11, no. 1, pp. 17–27, 2017.
[4] Y. Huo, C. Hu, X. Qi, and T. Jing, “LoDPD: a location difference-based proximity detection protocol for fog computing,” IEEE Internet of Things Journal, vol. 4, no. 5, pp. 1117–1124, 2017.
[5] A. Carcano, A. Coletta, M. Guglielmi, M. Masera, I. Nai Fovino, and A. Trombetta, “A multidimensional critical state analysis for detecting intrusions in SCADA systems,” IEEE Transactions on Industrial Informatics, vol. 7, no. 2, pp. 179–186, 2011.
[6] A. A. Cárdenas, S. Amin, Z.-S. Lin, Y.-L. Huang, C.-Y. Huang, and S. Sastry, “Attacks against process control systems: risk assessment, detection, and response,” in Proceedings of the 6th International Symposium on Information, Computer and Communications Security (ASIACCS’11), pp. 355–366, Hong Kong, China, March 2011.
[7] ICS-CERT, “ICS-CERT year in review 2016,” https://ics-cert.us-cert.gov/sites/default/files/Annual_Reports-Year_in_Review-FY2016_Final508C.pdf, 2017.
[8] K. Stouffer, J. Falco, and K. Scarfone, “Guide to Industrial Control Systems (ICS) security,” National Institute of Standards and Technology (NIST) Special Publication 800-82, 2011, http://csrc.nist.gov/publications/nistpubs/800-82/SP800-82-final.pdf.
[9] K. Ly and Y. Jin, “Security challenges in CPS and IoT: from end-node to the system,” in Proceedings of the 2016 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), pp. 63–68, Pittsburgh, PA, USA, July 2016.
[10] Y. Huo, Y. Tian, L. Ma, X. Cheng, and T. Jing, “Jamming strategies for physical layer security,” IEEE Wireless Communications Magazine, vol. 25, no. 1, pp. 148–153, 2018.
[11] M. Wan, W. Shang, L. Kong, and P. Zeng, “Content-based deep communication control for networked control system,” Telecommunication Systems, vol. 65, no. 1, pp. 155–168, 2017.
[12] Z. Drias, A. Serhrrouchni, and O. Vogel, “Taxonomy of attacks on industrial control protocols,” in Proceedings of the 2015 International Conference on Protocol Engineering (ICPE) and International Conference on New Technologies of Distributed Systems (NTDS), pp. 1–6, Paris, France, July 2015.
[13] F. Skopik, I. Friedberg, and R. Fiedler, “Dealing with advanced persistent threats in smart grid ICT networks,” in Proceedings of the 2014 IEEE Power and Energy Society Innovative Smart Grid Technologies Conference, ISGT 2014, pp. 1–5, Washington DC, USA, May 2014.
[14] L. Neitzel and B. Huba, “Top ten differences between ICS and IT: Cybersecurity Understanding the different needs of ICS and IT system security leads to cooperation and collaboration between historically disconnected camps,” Intech, vol. 61, no. 5, pp. 1–6, 2014.
[15] M. Wan, W. Shang, and P. Zeng, “Double behavior characteristics for one-class classification anomaly detection in networked control systems,” IEEE Transactions on Information Forensics and Security, vol. 12, no. 12, pp. 3011–3023, 2017.
[16] P. Maene, J. Gotzfried, R. de Clercq, T. Müller, F. Freiling, and I. Verbauwhede, “Hardware-based trusted computing architectures for isolation and attestation,” IEEE Transactions on Computers, vol. 67, no. 3, pp. 361–374, 2018.
[17] M. Cheminod, L. Durante, L. Seno, and A. Valenzano, “Performance evaluation and modeling of an industrial application-layer firewall,” IEEE Transactions on Industrial Informatics, vol. 14, no. 5, pp. 2159–2170, 2018.
[18] S. Han, M. Xie, H. H. Chen, and Y. Ling, “Intrusion detection in cyber-physical systems: techniques and challenges,” IEEE Systems Journal, vol. 8, no. 4, pp. 1052–1062, 2014.

[19] T. Cruz, L. Rosa, J. Proenca et al., “A cybersecurity detection framework for supervisory control and data acquisition systems,” IEEE Transactions on Industrial Informatics, vol. 12, no. 6, pp. 2236–2246, 2016.

[20] A. Almalawi, X. 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.

[21] B. Genge, D. A. Rusu, and P. Haller, “A connection pattern-based approach to detect network traffic anomalies in critical infrastructures,” in Proceedings of the the Seventh European Workshop, pp. 1–6, Amsterdam, The Netherlands, April 2014.

[22] V. L. Do, “Statistical detection and isolation of cyber-physical attacks on SCADA systems,” in Proceedings of the IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society, pp. 3524–3529, Beijing, China, October 2017.

[23] H. Gawand, A. K. Bhattacharjee, and K. Roy, “Control aware techniques for protection of industrial control system,” in Proceedings of the 2014 Annual IEEE India Conference (INDICON), pp. 1–6, Pune, India, December 2014.

[24] A. Terai, S. Abe, S. Kojima, Y. Takano, and I. Koshiba, “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 (EuroSe-PW), pp. 132–138, Paris, France, April 2017.

[25] W. Shang, P. Zeng, M. Wan, L. Li, and P. An, “Intrusion detection algorithm based on OCSVM in industrial control system,” Security and Communication Networks, vol. 9, no. 10, pp. 1040–1049, 2016.

[26] A. F. Shosha, P. Gladyshev, Shinn-Shyan Wu, and Chen-Ching Liu, “Detecting cyber intrusions in SCADA networks using multi-agent collaboration,” in Proceedings of the 2011 16th International Conference on Intelligent System Applications to Power Systems (ISAP), pp. 1–7, Hersonissos, Greece, September 2011.

[27] J. Hosic, J. Lamps, and D. H. Hart, “Evolving decision trees to detect anomalies in recurrent ICS networks,” in Proceedings of the World Congress on Industrial Control Systems Security, WCICSS 2015, pp. 50–57, London, UK, December 2015.

[28] J. E. Rubio, C. Alcaraz, R. Roman, and J. Lopez, “Analysis of intrusion detection systems in industrial ecosystems,” in Proceedings of the 14th International Conference on Security and Cryptography, pp. 116–128, Madrid, Spain, July 2017.

[29] I. Kiss, B. Genge, and P. Haller, “A clustering-based approach to detect cyber attacks in process control systems,” in Proceedings of the IEEE 13th International Conference on Industrial Informatics, pp. 142–148, Cambridge, UK, July 2015.

[30] A. K. Alexandridis and A. D. Zapranis, “Wavelet neural networks: a practical guide,” Neural Networks, vol. 42, pp. 1–27, 2013.

[31] X. Yuan, G. Peng, F. Zhang et al., “Analysis of power grid harmonics with wavelet network,” in Proceedings of the 2017 29th Chinese Control And Decision Conference (CCDC), pp. 6052–6055, Chongqing, China, May 2017.

[32] M. Cheminod, L. Durante, A. Valenzano, and C. Zunino, “Performance impact of commercial industrial firewalls on networked control systems,” in Proceedings of the 2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA), pp. 1–8, Berlin, Germany, September 2016.
