Computer Networks 185 (2021) 107677

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

A cyber-physical model for SCADA system and its intrusion detection

Chuan Sheng [a,b,*], Yu Yao [a,b], Qiang Fu [a,c], Wei Yang [c,d]

[a] College of Computer Science and Engineering, Northeastern University, Shenyang 110169, China
[b] Engineering Research Center of Security Technology of Complex Network System, Ministry of Education, Shenyang 110169, China
[c] Research Center of Safety Engineering Technology in Industrial Control, Neusoft Group Research, Liaoning Province, China
[d] Software College, Northeastern University, Shenyang 110169, China

ARTICLE INFO

Keywords:
Industrial control system
SCADA system
Model
Intrusion detection
Risk level

ABSTRACT

Supervisory Control and Data Acquisition (SCADA) systems are becoming increasingly susceptible to the sophisticated and targeted cyber attacks which are typically carried out by exploiting the vulnerabilities of industrial control devices or protocols. However, most of the existing network intrusion detection methods only focus on detecting and characterizing cyber attacks against the SCADA system, but cannot fully describe their real impact on the system. In this paper, we propose a cyber-physical model for the SCADA system to detect intrusions from the SCADA network and evaluate their risk levels against the industrial process. The model aims at characterizing the network structure and industrial process of the SCADA system through extracting and correlating the communication patterns and states of ICS devices. And any violation of the model is considered abnormal behavior, which can be caused by false operation or network attacks. Through associating network intrusions with the status of the SCADA system, a risk assessment method is proposed to estimate the potential damage degree of the attack on the system, which provides network administrators with richer information about network attacks. Moreover, the comprehensive performance evaluation conducted on public SCADA network data sets shows that the proposed method outperforms the existing methods in detecting and analyzing various cyber attacks against the SCADA system.

1. Introduction

Industrial Control Systems (ICSs) are widely used in industrial processes, such as power grids, water conservancy, natural gas, petrochemical and so on. From orchestrating complex maneuvers of the International Space Station to monitoring water levels on earth, ICSs can effectively process complex data and perform intended tasks they are designed for [1]. As a core component of ICSs, Supervisory Control and Data Acquisition (SCADA) systems are used to collect and process the data generated by Remote Terminal Units (RTUs) and Programmable Logical Units (PLCs), and allow engineers to monitor the status of ICSs and modify their parameters [2]. For example, in Smart Grid (SG), SCADA systems have been widely used to provide real-time controlling and monitoring of the electricity distribution network to ensure the reliability of the power supply and reduce the maintenance and operating costs of the network [3].

Because the installation and update of SCADA systems are costly and time-consuming, many such systems have remained in operation for many years, with more or less legacy products. In addition, many companies adopt commercial-off-the-shelf (COTS) SCADA products for their ICSs, which may come from different manufactures using various standards or proprietary communication protocols [4]. Furthermore, many ICS protocols, such as Modbus and Ethernet/IP, have some common vulnerabilities such as “plain-text”, “unauthorized” and “open” [5]. Although most of SCADA systems have been operating securely in isolation for many years, once they are connected to external networks, these vulnerabilities mentioned above will present them with unprecedented security risks [3,6].

Recently, more and more cyber attacks have manifested their severe damage to ICSs [7,8]. The sophistication of cyber attacks against ICSs mainly includes two aspects: attack approaches and attack tools. According to the ICS Cyber Kill Chain [9], cyber attackers target SCADA systems not in single incidents and breaches but, instead, through a campaign of efforts that enables access and provides sufficient information to devise an effect. Firstly, a fundamental and critical step is to discover exposed ICS devices in the Internet [10]. Then, the attackers will use compromised devices as a springboard to further probe the entire ICS network [11]. Finally, with such gained knowledge, the

* Corresponding author.
E-mail address: 1610538@stu.neu.edu.cn (C. Sheng).

https://doi.org/10.1016/j.comnet.2020.107677
Received 11 August 2020; Received in revised form 24 September 2020; Accepted 5 November 2020
Available online 10 November 2020

attackers can cause predictable effects on ICSs in a way that circumvents or impacts security mechanisms and achieves a true cyber-physical attack [12]. Therefore, network communication, especially that based on ICS protocols, plays an important role in the process of network attacks. As for attack tools, since the virus Stuxnet in 2010 [7], several kinds of viruses against ICSs have been detected, such as BlackEnergy, Night Dragon, Duqu, and so on [13]. Even the concept of viruses against ICS devices has been already discussed and proved [14,15].

Many methods have been proposed to mitigate the network threats faced by SCADA systems. The Intrusion Detection System (IDS) is a device or software application that monitors the system and/or network traffic for malicious activities or policy violations and responds to suspicious activities [16]. In terms of detection mechanism, the major types of IDS are signature-based, anomaly-based and specification-based [17,18]. In terms of analytic target, intrusion detection methods for SCADA systems can be roughly divided into network-level methods and physical-level methods [19]. Network-level methods focus on modeling network traffic and detecting abnormal traffic from normal traffic. But they rarely explore the potential impact of abnormal traffic on physical processes. Physical-level methods usually aim at modeling physical processes or parameters. Although they are sensitive to changes in physical parameters and able to estimate concrete physical impact, they cannot detect attack behaviors from network traffic in advance.

This paper presents a novel cyber-physical model for the SCADA system, whose purpose is to detect intrusions from normal SCADA network traffic and evaluate their risk levels against the industrial process. Firstly, the SCADA system is modeled from both network level and physical level. The network-level modeling aims at fingerprinting all normal communication patters of the SCADA network. The physical-level modeling is responsible for identifying the physical resources being used by ICS devices and associating them with normal communication patterns. By this way, any communication pattern out of the model is considered abnormal. According to the characteristics of intrusions, a risk assessment method is proposed to evaluate the risk levels of intrusions against the industrial process. In summary, the contributions of our method are as follows:

- We propose a novel cyber-physical model for the SCADA network in order to detect network intrusions against the SCADA system through extracting and correlating the communication patterns and states of ICS devices.
- Through associating network intrusions with the status of the SCADA system, the method attempts to map the intrusions to the potential damage degree on the system, which can provide network administrators with more useful information instead of “Yes or No”.
- The communication pattern sequence is proposed to characterize the SCADA network, which takes into account not only network characteristics but also the industrial process of the SCADA system.
- A flexible risk assessment method is proposed to evaluate the risk levels of intrusions against the industrial process, which reflects the potential impact of cyber attacks on the SCADA system.

The remainder of this paper is organized as follows. The related previous works are introduced in Section 2. Section 3 elaborates the cyber-physical model for the SCADA system, and describes the risk assessment method in detail. The implementation and evaluation of the proposed method are presented in Section 4. Discussion is addressed in Section 5. Finally, we conclude this work in Section 6.

2. Related work

In recent years, the research on intrusion detection methods for ICS network has received more and more attention. According to our best knowledge, existing ICS intrusion detection methods can be divided into three categories in terms of the analysis object: TCP/IP-level, ICS protocol level (ICS-level), and physical-level (or process-level). Though TCP/IP-level intrusion detection methods still play an important role in the traditional network defense, Barbosa, et al. [20] demonstrated that previous IP-level traffic models could not be easily applied to SCADA network traffic. To solve this problem, Garitano, et al. [21] attempted to generate the network traffic model starting from a real application description to distinguish different ICS traffic from different applications. Based on the stability of the ICS network, Genge, et al. [22] characterized ICS network traffic as a set of fixed connection patterns, and the traffic that violated the patterns was treated as anomalies. Gabriel, et al. [23] grouped SCADA network traffic into IP flows, and then studied the performance of nine different machine learning algorithms in classifying them. Shitharth, et al. [24] proposed a Probabilistic Relevance Classification (PRC) method to detect known and unknown attacks in the SCADA network through integrating Hidden Markov Model (HMM) and Relevance Vector Machine (RVM) algorithm. Xia, et al. [25] proposed an improved gravitational search algorithm (IGSA) combined with the wavelet neural network (WNN) to predict the network traffic of the smart substation with the aim of strengthening its system security protection. To ensure the security of multimicrogrid systems in smart grid, Hu, et al. [26] proposed a collaborative intrusion detection (CID) approach by using the consensus mechanism of blockchain, which was designed without the need of a trusted authority or center server while improving the accuracy of intrusion detection in a collaborative way. Niazi and Faheem [27] formulated the strategic interaction between a hypervisor monitoring its virtualized SDN (vSDN) controllers and the source of new flow requests potentially launching a DDoS attack to prevent the exploitation of a vSDN-based SG architecture. Although these methods demonstrated their effectiveness in detecting network intrusions, they usually had two inherent flaws because they did not consider the industrial process of the SCADA system. One is that they usually failed to estimate the risk levels of network intrusions against the SCADA system. The other is that they might miss more complicated and targeted attacks which looked legitimate in the IP level.

Different from TCP/IP-level methods, ICS-level methods usually extract special characteristics related to a specific ICS protocol from ICS network traffic as well as common network characteristics. Based on the high periodicity of the ICS network, Goldenberg and Wool [28] proposed modeling each HMI-PLC channel using the deterministic finite automation (DFA), which was very sensitive to anomalies. Shang, et al. [29] put forward a comprehensive Modbus protocol feature extraction method through depth analysis of operation modes of abnormal behaviors. Yusheng, et al. [30] designed a rule-based stereo depth IDS, which could not only analyze the characteristics of industrial traffic, but also explore the semantic relationships among key fields in the Modbus protocol. Marsden, et al. [31] presented a Probability Risk Identification based Intrusion Detection System (PRI-IDS) technique based on analyzing network traffic of Modbus TCP/IP for identifying replay attacks in SCADA systems. Khan, et al. [32] put forward a hybrid model to detect anomalies for the ICS, which took advantage of the anticipated and consistent nature of communication patterns that occur among ground devices in ICS setups. Guo, et al. [33] proposed a function-based model (FBM) according to IEC 61850, by which the substation automation system (SAS) could be decomposed into functions and described in mathematical language. Based on the model, a 9-level risk assessment criteria and a system risk integration method based on analytic hierarchy process were established. Jokar and Leung [34] presented a model-based intrusion detection mechanism for ZigBee-based home area networks in smart grid based on smart energy profile 2.0 (SEP 2.0) protocol specification as well as IEEE 802.15.4 standard, and proposed a Q-learning-based intrusion prevention system to counter various attacks. Compared with TCP/IP-level methods, ICS-level methods could further improve their accuracy through integrating with some specific ICS protocols. However, they were usually sensitive to the network fluctuations, which resulted in a higher false alarm rate. And few of them tended to map network attacks to the concrete impact on the SCADA

system.

Although the above network-based methods were good at detecting network intrusions, few of them were able to assess the impact of intrusions on the industrial process. In contrast to network-based methods, physical-level methods always tended to model ICS process parameters [35,36], or fingerprint physical devices in conjunction with their responding or processing characteristics [19]. Neha, et al. [37] presented a sine-cosine optimization based recurrent neural network (SCO-RNN) to detect the cyber physical attacks against SCADA systems. Selvarajan, et al. [38] used an enhanced model of machine learning technology to classify and predict attacks against SCADA systems based on the sensor data. Wang, et al. [39] proposed a Deep Learning based Locational Detection architecture (DLLD) to detect the exact locations of the false data injection attacks which could cause a severe threat to the state estimation of smart grid. Molzahn and Wang [40] proposed an algorithm for detecting and characterizing cyber attacks to network parameter data with specific application to optimal power flow problems through comparing the network parameters with historical operating point data. Efstathopoulos, et al. [41] proposed to introduce a complex feature representation of operational data from a real power plant into existing machine learning and deep learning models to detect possible cyber attacks and anomalies in smart grid. Although physical-level methods had achieved good effectiveness in detecting some complicated attacks, these methods were more dedicated to detecting attacks that had occurred rather than preventing attacks from occurring. In addition, when some network attacks do not change the physical parameters (such as DDoS [42]) or only utilize the unused physical resources (such as Botnet [43]), these methods would be no longer applicable.

Different from the above methods that were usually dedicated to modeling one specific analysis object, the proposed method attempts to build a cyber-physical model to associate network traffic with the industrial process based on the finite-state machine (FSM). Control theories based on FSM have been well developed as it addressed the fundamental issues in control of discrete event systems (DES), such as electric power systems [44]. The FSM, at its simplest, is the model of system' behaviors, with a limited number of defined conditions or modes, where mode transitions change according to circumstances. However, as far as we know, the FSM has not been used to combine and model the network process and industrial process of the SCADA system. The FSM is composed of four main elements: 1) states which define behavior and may produce actions, 2) state transitions that are movements from one state to another, 3) rules or conditions that must be met to allow for a state transition, and 4) trigger events that may trigger state transitions [45]. Accordingly, the proposed model takes the storage blocks used by the device as its state, and converts network traffic into a series of accesses to the storage blocks. These accesses, such as reading or writing operations, can trigger a series of events that may change the states of devices. More details are described in the next section.

3. Cyber-physical model

In this section, we elaborate the cyber-physical model for the SCADA system, and describe the risk assessment method in detail.

3.1. SCADA system model

As shown in Fig. 1 [46], a typical SCADA system consists of Master Terminal Units (MTUs) and RTUs, in which MTUs are responsible for sending commands to RTUs and monitoring their states, while RTUs are used to send commands to field devices and acquire measurement data from them. In general, the network structure of the SCADA system and the communication patterns among ICS devices is stable and constant, because the system is usually used to periodically carry out the defined fixed tasks. Therefore, any violation of existing communication patterns can be considered abnormal behavior which can be caused by false operation or network attacks. In addition, some of RTUs' storage blocks are used to receive commands from MTUs and store measurement data from field devices in the industrial processes. Obviously, these storage blocks record and determine the states of ICS devices and the SCADA system. Therefore, this paper attempts to model and correlate the communication patterns and states of ICS devices in order to detect and assess the risk level of network attacks against the SCADA system.

In order to construct a simple and effective model for the SCADA system, some significant modifications are made to the FSM, and the model is formulated as a 6-tuple:

$$G = (D, X, C, S, T, E) \tag{1}$$

where

- $D = \{d_1, d_2, \cdots, d_m\}$ is a set of devices (including MTUs and RTUs), which constitute the primary supervisory control and data

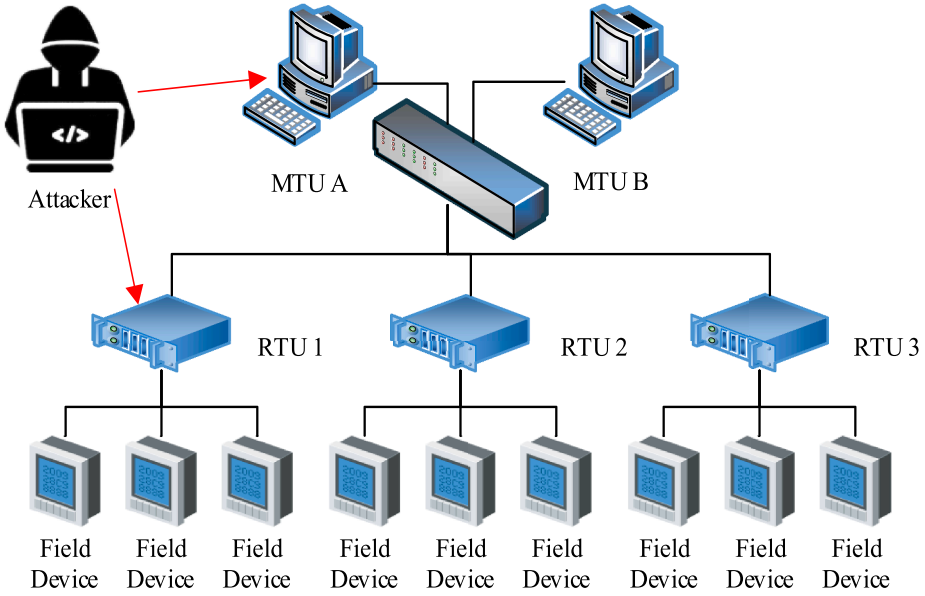


Fig. 1. Example of the SCADA system.

acquisition layer of a SCADA system. The $m \in N^*$ represents the number of devices in the system.

- $X = \{x_1, x_2, \cdots, x_r\}, r \leq m$, represents the state of a SCADA system, which consists of states of all RTUs in D. $x_i = \{v_{i,1}, v_{i,2}, \cdots, v_{i,p}\}(i \in \{1, 2, \cdots, r\})$ denotes the state of the deviced_i, where $v_{i,j}(j \in \{1, 2, \cdots, p\})$ represents the value of the jth storage block used by d_i. The $p \in N^*$ is the number of storage blocks used by d_i. Note that the change of $\forall v_{i,j} \in x_i$ will lead to the change of x_i. MTUs are excluded from X, because they are not directly involved in the physical process, and their changes can be reflected in the associated RTUs.
- $C = \{c_1, c_2, \cdots, c_y\}, y \leq m$ is a finite set of communication patterns, where $c_i = \{\zeta_{i,1}, \zeta_{i,2}, \cdots, \zeta_{i,q}\}$ denotes the set of communication patterns generated by the deviced_i and the q is the size of c_i. The definition of the communication pattern and its extraction method is described in detail later.
- $S = \{s_1, s_2, \cdots, s_y\}, y \leq m$ is a finite set of sequences of communication patterns, where $s_i = \{\varsigma_{i,1}, \varsigma_{i,2}, \cdots, \varsigma_{i,n}\}$ represents the set of sequences owned by the device d_i and the n is the size of s_i. Specifically, $\varsigma_{i,r} = \langle \zeta_{i,j}, \zeta_{i,k}, \zeta_{i,l}, \cdots \rangle (j, k, l \in \{1, 2, \cdots, q\})$ is an ordered sequence of communication patterns generated by d_i. Note that the length of $\varsigma_{i,r}$ is dynamic. More details are described later.
- $T = \{t_1^0, t_1^1, t_2^0, t_2^1, \cdots, t_r^0, t_r^1\}$ is a finite set of RTU transitions corresponding to X, where the t_i^0 means the device d_i is read and t_i^1 means d_i is written. In the model, all operations on RTUs are abstracted as reading or writing, where reading means acquiring data from a RTU, and writing means storing or modifying some data in storage blocks of a RTU. Furthermore, a RTU transition $t_i^a(a = 0 \text{or} 1)$ can be represented by a set of value transitions which correspond to the values in the state x_i, as $t_i^a = \{t_{i,1}^a, t_{i,2}^a, \cdots, t_{i,p}^a\}$. Obviously, the $t_{i,p}^1$ will change the state x_i. A transition $t_{i,p}^a$ can be caused by a communication pattern $\zeta_{j,k}$ $(i \neq j)$ in a communication pattern sequence $\varsigma_{j,r}$.
- $E = \{e_1, e_2, \cdots, e_n\}$ denotes the events that occur in the system. An event $e_i = (\kappa(x_1), \kappa(x_2), \cdots, \kappa(x_r))$ is triggered by a transition $t_{i,p}^a$, and the $\kappa(x_j)$ indicates a change in the state x_j of the RTU d_j, where

$$\kappa(x_j) = \begin{cases} 0, x_j \text{is unchanged.} \\ 1, x_j \text{is changed.} \end{cases} . \tag{2}$$

Note that because the state of the SCADA system is composed by the states of all RTUs in D and the SCADA system may include other devices, the size of X is not greater than the size of D. Because some devices may never actively tend to communicate with other devices, they may not have their own communication patterns, that is, the size of C is not greater than the size of D. The communication pattern sequence consists of communication patterns, so the size of C is consistent with the size of S. Take the SCADA system in Fig. 1 as an example, the state of the SCADA system consists of the states of three RTUs using three storage blocks to store the states of the monitored field devices respectively, as shown in Fig. 2. When the MTU queries field devices' measurement data through RTUs or write some data to unused storage blocks of RTUs, the system state will not be affected. However, when the MTU send a command to the RTU to change the state of a field device (such as the first device of RTU 1), the system state will be changed. In summary, changes in the system state can be caused by the administrator's manual operation or network attacks. Meanwhile, the system state will not be affected by network attacks, when they only tend to query the states of devices or take advantage of the unused storage blocks of RTUs for other aims (such as Botnet [43]).

3.2. Communication pattern and its sequence

The communication pattern is defined as a set of features that aims to characterize the SCADA network traffic. Instead of analyzing each network packet at a time, our method tends to extract a group of network packets at a time, called IP flow, to analyze the network behavior in the SCADA network. The main advantage of the IP flow is that it provides an overview of the network behavior, showing main statistics of interactions between network nodes, which can reveal deviations from the normal behavior caused by attacks. The IP Flow Information Export (IPFIX) Protocol [47] specifies that an IP flow consists of packets with the same source and destination IP addresses, source and destination ports, and transport protocol. Furthermore, IP flows are separated by a timeout threshold $T_{inactive}$ beyond which a flow is considered inactive and is cut off. To prevent an IP flow from spanning multiple polling cycles of a SCADA system, the $T_{inactive}$ has to be much less than the polling cycle of the system. In this paper, the $T_{inactive}$ is set to one twentieth of the polling cycle, which is considered able to satisfy the above constraint and provide an overview of the network behavior.

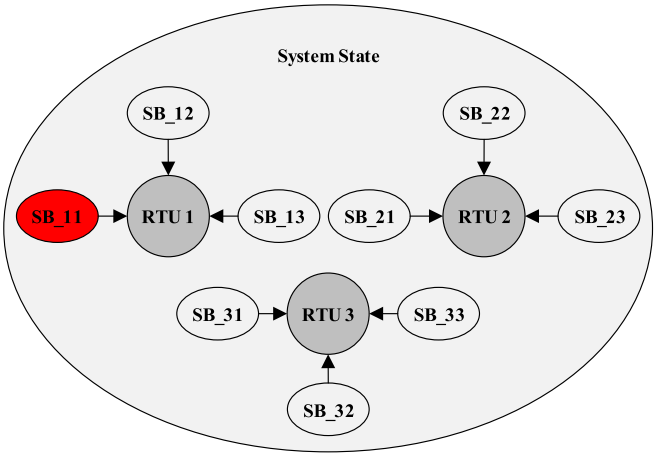


Fig. 2. State of the SCADA system.

Inspired by the features used by [23], the communication pattern is formulated as

$$\zeta_{j,k} = \left(d_{dst}, p_{dst}, p_{tr}, p_{num}, p_{size}, \tau, \Gamma\right) \tag{3}$$

where d_{dst} represents the destination device; p_{dst} represents the destination port; p_{tr} represents the transport protocol; p_{num} denotes the number of packets that compose the flow; p_{size} denotes the byte size of packets that compose the flow; τ denotes the duration of the flow; Γ represents the sequence of transitions caused by $\zeta_{j,k}$, which is formulated as

$$\Gamma = \left\langle t_{i,j}^{a_1}, t_{i,k}^{a_2}, \cdots, t_{i,l}^{a_n} \right\rangle, 1 \leq j, k, l \leq p \tag{4}$$

where i represents the number of the destination device d_i; p denotes the number of values in the state x_i; $a_i = 0 \text{or} 1, 1 \leq i \leq n$, and n denotes the number of transitions caused by $\zeta_{j,k}$. Obviously, the length of Γ is dynamic, which is determined by the concrete communication pattern. When the destination device is not a RTU, Γ will be set to empty.

By analyzing the industrial processes of SCADA systems, it is found that a device usually sends its commands (communication patterns) to one or more other devices in a continuous and ordered way, which is called "communication pattern sequence" in this paper. For example, a MTU periodically and continuously inquires the states of RTUs in a constant order. For convenience, the "communication pattern sequence" is later abbreviated as "CPSequence" in this paper. Obviously, the CPSequence is more representative than the communication pattern in characterizing the behavior of a device. Therefore, the CPSequence is used to represent a behavior of a device in this paper. Certainly, as the simplest mode, when the CPSequence covers only one communication pattern, the length of the CPSequence may be one. For separating

CPSequences of a device, a timeout threshold T_{seq} is adopted beyond which a CPSequence is considered inactive and is cut off. In this paper, the T_{seq} is equal to the $T_{inactive}$.

3.3. Intrusion detection

In this paper, we assume that the SCADA system is a "stable" system, which satisfies: 1) the network architecture and industrial process of the system remain constant; 2) the devices in the system and their configurations remain constant. Based on the above conditions, we define the network intrusion (or anomaly) as any CPSequence out of the set of CPSequences S in the model. According to the matching degree with S, network intrusions can be divided into three classes:

- A **sequence-class intrusion** is detected when the source IP of the abnormal CPSequence cannot be found in S.
- A **pattern-class intrusion** is triggered by a mismatch of communication patterns in the abnormal CPSequence and the CPSequences in S. Specifically, a mismatch can be caused by two cases: 1) the abnormal CPSequence shows a different order or combination of patterns; 2) abnormal patterns are detected by comparing their feature vectors with those of existing patterns in C.
- The **content-class intrusion** is an upgraded version of a pattern-class intrusion. In this case, patterns in a content-class intrusion has the same features as existing patterns except the sequence of transitions Γ. Specifically, the difference between Γs can be caused by two cases: 1) the abnormal Γ consists of transitions in a different order or combination; 2) the abnormal Γ contains unknown transitions.

In general, the set of CPSequences S in the model can be learned through analyzing the SCADA network traffic as the original training stage. Algorithm 1 summarizes the process of detecting abnormal CPSequences from the network traffic.

The proposed model can be used to detect various network intrusions with "licit" network traffic which may be ignored by traditional IDSs. For example, the sequence-class intrusion usually represents external attacks including reconnaissance attack, penetration attack and so on. The pattern-class intrusion is usually initiated by internal compromised or deliberate devices, such as spreading viruses, exploiting vulnerabilities, sending fake commands and so on. The content-class intrusion may be the most complex attack because the attack device tends to disguise itself as a legitimate device and just modify contents of network packets without any other suspicious behavior. The false data injection attack is a classical content-class intrusion [48].

3.4. Risk assessment

The risk level of a network intrusion on the SCADA system is estimated from three aspects:

- Sophistication: the complexity of an intrusion which corresponds to its intrusion class.
- Network impact: the network load caused by an intrusion on the system and devices.
- Physical impact: the number of states of devices manipulated by an intrusion.

For simplicity, we use $R_S = \{1, 2, 3\}$ to represent the sophistication of an intrusion, and R_P to denote the physical impact. Furthermore, the network impact is divided into the system impact R_{NS} and device impact R_{ND}. Specifically, the network impact on the system R_{NS} means the total network load generated by an intrusion, while the network impact on devices R_{ND} means the maximum network load on a single device. The CPSequence of an intrusion can be formulated as follows.

$$\varsigma = \langle \zeta_1, \zeta_2, \cdots, \zeta_n \rangle = \begin{bmatrix} d^1_{dst} & d^2_{dst} & \cdots & d^n_{dst} \\ p^1_{dst} & p^2_{dst} & \cdots & p^n_{dst} \\ p^1_{tr} & p^2_{tr} & \cdots & p^n_{tr} \\ p^1_{num} & p^2_{num} & \cdots & p^n_{num} \\ p^1_{size} & p^2_{size} & \cdots & p^n_{size} \\ \tau^1 & \tau^2 & \cdots & \tau^n \\ \Gamma^1 & \Gamma^2 & \cdots & \Gamma^n \end{bmatrix} \tag{5}$$

According to Eq. (5), R_P can be computed as:

$$R_P = \sum_{i=1}^{n} \sum_{j=1}^{|\Gamma^i|} \Phi(e_{i,j}), \Phi(e_{i,j}) = \sum_{l=1}^{r} \kappa(x_l) \tag{6}$$

where $|\Gamma^i|$ denotes the size of Γ^i; $e_{i,j}$ represents the event caused by the jth transition in the ith communication pattern. We assume that all storage blocks used by RTUs are covered in X of the model, which can be implemented by analyzing network traffic or referring to configuration specifications of devices. Therefore, other storage blocks out of X are not considered able to cause a direct impact on the system, and are not involved in R_P. R_{NS} can be calculated as:

$$R_{NS} = \sum_{i=1}^{n} p^i_{num} \tag{7}$$

Let R^i_{ND} denotes the number of packets associated with the device d_i, which can be formulated as:

$$R^i_{ND} = \sum_{j=1}^{n} I_i(N_j) * p^j_{num}, I_i(N_j) = \begin{cases} 1, N_j = i \\ 0, N_j \neq i \end{cases} \tag{8}$$

where N_j denotes the number of the device accessed by the jth communication pattern. Then R_{ND} can be calculated as $R_{ND} = \max\{R^i_{ND}\}$. Therefore, the risk level of a network intrusion R can be formulated as:

$$R = r_1 R_S + r_2 L_1(R_P) + r_3 L_2(R_{NS}) + r_4 L_3(R_{ND}) \tag{9}$$

where r_i, $1 \leq i \leq 4$, represents the corresponding weight of each element, and the function $L_i(x)$ denotes the ranking function. In the simplest case, the r_i can be set to 0.25, and $L_i(x)$ can be formulated as a piecewise function, which is determined by the importance of each element to the system and the device performance. For example, when all the system factors are considered equally important and devices have good performance, Eq. (9) can be formulated as:

$$R = 0.25 * R_S + 0.25 * L_1(R_P) + 0.25 * L_2(R_{NS}) + 0.25 * L_3(R_{ND}) \tag{10}$$

where

$$L_i(x) = \begin{cases} 0, x = 0 \\ 1, 0 < x < 2 * max_i \\ 2, 2 * max_i \leq x < 10 * max_i \\ 3, 10 * max_i \leq x \end{cases} \tag{11}$$

where max_i denotes the maximum value of the corresponding element observed in the model. Note that the model focuses on assessing the potential danger of a cyber attack rather than the system deviation caused by the changes in storage blocks of devices. In other words, the risk level R is used to estimate a network intrusion instead of the state of the system. Therefore, the study on specific physical parameters of the system is beyond the scope of this paper.

4. Experimental evaluation

In this section, the SCADA network data sets developed by Lemay

and Fernandez [46] are used to analyze the effectiveness of the proposed method in modeling SCADA systems, detecting intrusions, and assessing risk levels. The data sets contain packet captures including both malicious and non-malicious Modbus traffic, and corresponding label files that provide the ground truth for validating experimental results. Specifically, the data sets were generated in a SCADA sandbox, where electric network simulators were used to introduce realism in the physical component and real attack tools were used to generate the malicious traffic. Although the data sets provide different scenarios with different numbers of MTUs and RTUs, the data sets with 2 MTUs and 6 RTUs are chosen for our experiments. Because all attack date sets were generated based on this scenario.

As shown in Fig. 3, the two MTUs are responsible for monitoring the status of controllers and sending some commands to them, and each controller represents a small electric network that consists of one main supply branch and three sub-branches. The controllers provide measurements on the voltage of each branch to the MTUs. More details of the system implementation and process are described in [46]. Overall, the data sets generated in the SCADA network contain three types of data: regular polling data, manual operation data, and attack data.

4.1. SCADA system modeling

In this experiment, the system model is built automatically by analyzing the network traffic in the "Run1_6RTU" data set which contains 1 h of regular Modbus traffic including polling and manual operation. The network traffic analyzer is implemented in Java programming language based on Pcap4J [49] and TShark [50]. The system is modeled as:

$$G = (D, X, C, S, T, E),$$

where

- $D = \{d_1, d_2, \cdots, d_8\}$, where d_7 and d_8 denote MTU A and MTU B, d_1 through d_6 represent controllers 1 through 6.
- $X = \{x_1, x_2, \cdots, x_6\} = \{\{v_{1,1}, v_{1,2}, v_{1,3}\}, \{v_{2,1}, v_{2,2}, v_{2,3}\}, \cdots, \{v_{6,1}, v_{6,2}, v_{6,3}\}\}$, where the state number is consistent with the controller number; $v_{i,1}$ refers to the Discrete Inputs with a starting position of 4 and a length of 4; $v_{i,2}$ refers to the Coils with a starting position of 0 and a length of 4, and $v_{i,3}$ refers to the Holding Registers with a starting position of 8 and a length of 4.
- $C = \{c_7, c_8\} = \{\{\zeta_{7,1}, \cdots, \zeta_{7,19}, \cdots, \zeta_{7,31}, \cdots, \zeta_{7,49}\}, \{\zeta_{8,1}, \cdots, \zeta_{8,19}, \cdots, \zeta_{8,31}, \cdots, \zeta_{8,39}\}\}$, which means that MTU A contains 49 different communication patterns and MTU B contains 39 different communication patterns. These communication patterns can be divided into three types: regular patterns, manual patterns, and incomplete patterns. Regular patterns are generated by polling traffic, such as $\zeta_{7,1} = (d_1, 502, 1, 9, 398, 0.011, \langle t^0_{1,1} \rangle)$. Manual patterns are generated by manual operations, such as $\zeta_{7,19} = (d_3, 502, 1, 9, 400, 0.002, \langle t^1_{3,2} \rangle)$. Incomplete patterns are caused by dropping packets in polling traffic, such as $\zeta_{7,31} = (d_2, 502, 1, 4, 160, 0.001, \varnothing)$.
- $S = \{s_7, s_8\} = \{\{\varsigma_{7,1}, \varsigma_{7,2}, \cdots, \varsigma_{7,27}\}, \{\varsigma_{8,1}, \varsigma_{8,2}, \cdots, \varsigma_{8,25}\}\}$, which means that MTU A contains 27 different CPSequences and MTU B contains 25 different CPSequences. Theoretically, a MTU has only one regular CPSequence which consists of 18 communication patterns in order to periodically query the status of controllers. However, dropping packets and manual operations sometimes disturb the regular CPSequence in a random way. Therefore, the number of CPSequences is much greater than 1.
- $T = \{t^0_1, t^1_1, \cdots, t^0_6, t^1_6\} = \{\{t^0_{1,1}, t^0_{1,2}, t^0_{1,3}\}, \{t^1_{1,1}, t^1_{1,2}, t^1_{1,3}\}, \cdots, \{t^0_{6,1}, t^0_{6,2}, t^0_{6,3}\}, \{t^1_{6,1}, t^1_{6,2}, t^1_{6,3}\}\}$, where the transition number is consistent with the controller number; $t^0_{i,j}$ means that the jth storage block of the controller d_i is read; $t^1_{i,j}$ means that the jth storage block of the controller d_i is written. For example, the $t^1_{3,2}$ in $\zeta_{7,19}$ represents that the Coils with a starting position of 0 and a length of 4 of the controller d_3 is written.
- $E = \{e_1, e_2, \cdots, e_{64}\} = \{(0,0,0,0,0,0), (1,0,0,0,0,0), \cdots, (1,1,1,1,1,1)\}$, which represents all events that may occur in the system. The events caused by this data set can be formulated as $E^* = \{e_1, e_2, e_3, e_4\} = \{(0,0,0,0,0,0), (1,0,0,0,0,0), (0,1,0,0,0,0), (0,0,1,0,0,0)\}$. Note that most CPSequences from MTUs do not tend to change the status of controllers except for manual operations.

The effectiveness of the system model in detecting intrusions and estimating their risk levels is studied as follows.

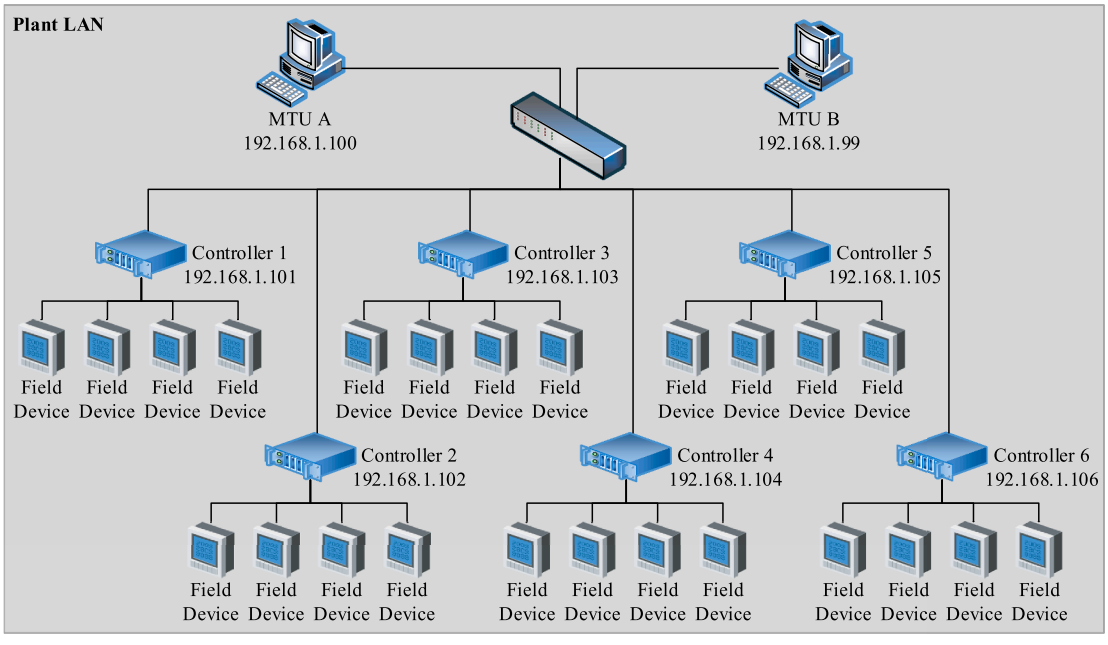


Fig. 3. The SCADA network used in data set generation with 2 MTUs and 6 Controllers.

4.2. Intrusion detection

All kinds of attack data sets in [46] are used to study the effectiveness of the model in detecting intrusions as well as another regular data set, whose description is presented in Table 1. These attacks simulate a series of real-world attacks on SCADA systems, such as the cyber attack against the Ukrainian power grid [8].

Next, the analysis is carried out by attack types. In this paper, the effectiveness of the model in detecting intrusions is evaluated by Accuracy, Detection Rate (DR), and F1-Score. The Accuracy refers to the proportion of correctly classified packets (TP + TN) to all packets (TP + FP + TN + FN). The DR is defined as the ratio of the number of correctly detected attacks (TP) to the total number of attacks present in the network (TP + FN). F1-Score measures the harmonic mean of Precision and Recall (Detection Rate). Precision stands for the ratio of the number of correctly detected attacks (TP) to the number of all detected attacks (TP + FP). They are formulated as follows:

Table 1
Description of SCADA network data sets.

Attack	Name	Description	Number of entries	Number of malicious entries
Attack 1	6RTU_with_operate	using an exploit (ms08_netapi) from a compromised RTU to compromise another RTU using Metasploit.	1856	1200
Attack 2	CnC_uploading_exe	sending an EXE file from a compromised RTU to another compromised RTU through a Metasploit meterpreter channel.	1426	121
Attack 3	Moving_two_files	sending two files from a compromised RTU to another compromised RTU.	3319	75
Attack 4	Characterization	sending a series of Modbus read commands to characterize available registers from a compromised RTU.	12296	6719
Attack 5	Send_a_fake_command	sending a Modbus write operation from a compromised RTU using Metasploit proxy functionality and the proxychains tool.	11166	10
Attack 6	Channel_4d_12s	Modbus covert channel using the four least significant digits of twelve storage registers.	44977	44977
Regular data	Modbus_polling_only	1 hour of regular Modbus traffic including polling only.	58325	0

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \tag{12}$$

$$DR = \frac{TP}{TP + FN} \tag{13}$$

$$F1 - Score = 2\frac{Precision * Recall}{Precision + Recall} = \frac{2TP}{2TP + FN + FP} \tag{14}$$

where TP, TN, FP, and FN stand for true positives, true negatives, false positives and false negatives respectively. The experimental results about various data sets are listed in Table 2.

4.2.1. Attack 1

Attack 1 is represented by the data set "6RTU_with_operate" in which a compromised machine launches a remote exploit to compromise the second controller. This simulates that an attacker expands his original foothold. The experimental result shows that the model can detect this attack accurately.

4.2.2. Attack 2 and Attack 3

Attack 2 and Attack 3 are generated by the data sets "CnC_uploading_exe" and "Moving_two_files", which aim at moving files through the Metasploit Meterpreter channel. This illustrates attackers that upload new tools or update their presence on the machines through sending a newer version of their malware. The experimental result shows that the model can detect such attacks with only a few false negatives. That is because the model ignores the broadcast packets which were labeled as malicious in label files.

4.2.3. Attack 4

Attack 4 is simulated in the data set "Characterization", which initiates a fingerprinting attack to characterize the content of the memory of a controller through sending a series of read packets. This represents attackers gathering information about the SCADA system. Although the model detects all packets labeled as malicious, it incorrectly identifies a non-malicious CPSequence with 192 packets as malicious. That is because a manual operation was inserted into the regular CPSequence, which resulted in an abnormal CPSequence.

4.2.4. Attack 5

Attack 5 is implemented by sending an unauthorized command to a controller in the data set "Send_a_fake_command". The command is a well formatted WRITE_COIL packet sent to another controller. The error in detecting Attack 5 is also caused by a manual operation in which MTU A accessed Controller 4 that had never been accessed by a manual operation.

4.2.5. Attack 6

Attack 6 represents a Modbus-based command and control channel in the data set "Channel_4d_12s" [43]. The command and control channel uses the least significant bits of Modbus data to transmit

Table 2
Results of the model in detecting attacks.

Attack	TP	TN	FP	FN	Accuracy	DR	F1-Score
Attack 1	1200	656	0	0	1.0	1	1
Attack 2	120	1305	0	1	0.9993	0.9917	0.9959
Attack 3	73	3244	0	2	0.9994	0.9733	0.9869
Attack 4	6719	5385	192	0	0.9844	1	0.9859
Attack 5	10	11122	34	0	0.997	1	0.3704
Attack 6	44977	0	0	0	1.0	1	1
Regular data	0	58142	183	0	0.9969	–	–
Total	53099	79854	409	3	0.9969	0.9999	0.9944

information. Obviously, this attack is more sophisticated and insidious. Because this data set is generated by other IP addresses out of the model, and has no the corresponding label file, we have to make some modifications to it. First, we change its source and destination addresses to the addresses of MTU B and Controller 1. Then, we label all packets in the data set as malicious according to its specification. The experimental result shows that Attack 6 can be detected by the model accurately.

4.2.6. Regular data

The data set "Modbus_polling_only" contains 1 hour of regular Modbus traffic including polling only, which is used to analyze the false alarm rate of the model. As shown in the experimental result, an abnormal CPSequence with 183 packets is detected by the model. That is because a manual operation is detected in the data set like in the data set "Characterization". However, this violates the description of this data set, that is, no manual operations are included. In addition, Detection Rate and F1-Score are not taken into account because there is no attack in this dataset.

In summary, the model can detect all kinds of attacks mentioned in [46] accurately. Although some manual operations, especially those mixed into the regular CPSequence, cause few misjudgments of the model, that is a rare occurrence. Overall, the model is effective in detecting network intrusions.

4.3. Comparison with state-of-the-art methods

To further verify the effectiveness of our method, the model is compared with some state-of-the-art machine learning-based approaches via the above data sets through 3 metrics: Accuracy, Detection Rate (DR), and F1-Score. As mentioned in [23], a total of nine supervised classification algorithms were tested to detect various attacks in data sets, including Support Vector Machine, Locally Deep Support Vector Machine, Averaged Perceptron, Neural Network, Logistic Regression, Bayes Point Machine, Boosted Decision Tree, Decision Forest, and Decision Jungle. In order to save space, three of the methods that performed best on each metric are selected to compare with our method.

The three competing methods include Decision Forest, Boosted Decision Tree, and Decision Jungle. Because Attack 1, Attack 2, Attack 3, and Attack 5 data sets were used to investigate the three competing methods, the 3 metrics of our model are also calculated based on these 4 datasets. The results of our model and the three best algorithms in [23] are shown in Fig. 4. Obviously, our model shows the highest score on each metric. In our opinion, two reasons account for the failure of the three methods. The first is that they only focused on IP-level features of the SCADA network traffic, which usually caused them to ignore some attacks based on protocol content. The other is that they failed to integrate with the industrial process of the SCADA system. In this case, a compromised device could send illegal or unauthorized commands. Furthermore, these drawbacks precluded the use of these methods for further assessment of the risk levels of attacks.

4.4. Risk assessment

The characteristics and risk levels of the attacks mentioned above are evaluated and discussed based on the risk assessment method proposed in Section 3. Because of the lack of the ground truth about the risk levels of attacks, the effectiveness of the risk assessment method is evaluated from two aspects: distinguishing different attacks and targeting different attacks. The risk indexes and levels of CPSequences of various attacks are listed in Table 3, which are calculated based on Eq. (10) with Eq. (11). Based on Table 3, some statistics about attacks can be obtained in Table 4.

4.4.1. Distinguishing different attacks

As shown in Figs. 5 and 6, different attacks display different combinations of statistical features that can characterize themselves. For example, Attack 6 has the maximum value in each statistical feature; all attacks show different average risk levels; the feature "Max risk level"

Table 3
Risk indexes and levels of attacks.

Attack	R_S	R_P	R_{NS}	R_{ND}	Risk level
Attack 1	2	0	1	2	1.25
	2	0	1	2	1.25
	2	0	2	3	1.75
Attack 2	2	0	1	1	1
	2	0	1	1	1
	2	0	1	1	1
Attack 3	2	0	1	0	0.75
	2	0	1	1	1
	2	0	1	1	1
	2	0	1	1	1
	2	0	1	1	1
	2	0	1	1	1
Attack 4	2	0	3	3	2
Attack 5	2	1	1	1	1.25
Attack 6	2	0	1	1	1
	2	0	1	1	1
	3	2	1	1	1.75
	2	3	3	3	2.75
	2	3	3	3	2.75
	2	3	3	3	2.75
	2	3	3	3	2.75
	2	3	3	3	2.75
	2	3	3	3	2.75
	2	3	3	3	2.75
	2	3	3	3	2.75
	2	3	3	3	2.75
	2	3	3	3	2.75

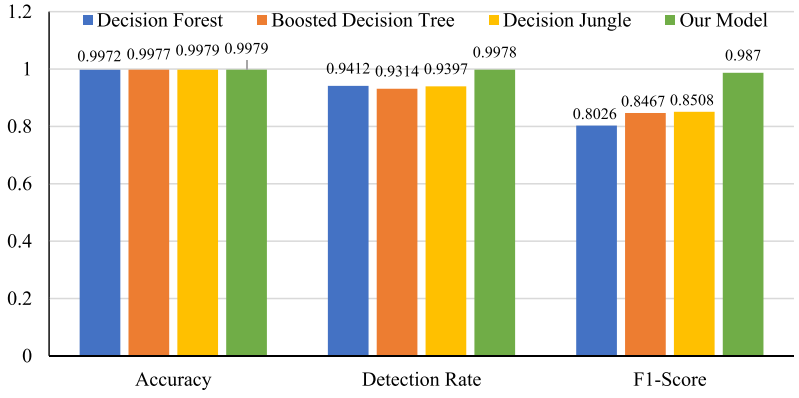


Fig. 4. Results of our model and the three best algorithms.

Table 4
Some statistics about attacks.

Attack	Max R_S	Max R_P	Max R_{NS}	Max R_{ND}	Avg. risk level	Max risk level	Number of CPSequences
Attack 1	2	0	2	3	1.42	1.75	3
Attack 2	2	0	1	1	1	1	3
Attack 3	2	0	1	1	0.96	1	6
Attack 4	2	0	3	3	2	2	1
Attack 5	2	1	1	1	1.25	1.25	1
Attack 6	3	3	3	3	2.4	2.75	13

Algorithm 1
Detecting abnormal CPSequences.

Input: A new CPSequence ς^*
Output: The type of ς^*
1. **if** source IP of ς^* cannot be found in S **then** label ς^* as "sequence-class intrusion"; //Assume that the number of the source IP of ς^* is i and, the size of ς^* is m.
2. **else**
3. $max\_level \leftarrow 0$;
4. **foreach** $s_i = \{\varsigma_{i,1}, \varsigma_{i,2}, \cdots, \varsigma_{i,n}\}$ **do**
5. **if** the size of ς^* != the size of $\varsigma_{i,j}$ and $max\_level < 1$ **then**
6. $max\_level \leftarrow 1$
7. **else**
8. **for** $k \leftarrow 1$ **to** m **do**
9. $flag \leftarrow false$
10. **for** $l \leftarrow 1$ **to** 6 **do**
//$\varsigma^*[k][l]$ represents the lth feature of the kth communication pattern of ς^*.
11. **if** $\varsigma^*[k][l]$!= $\varsigma_{i,j}[k][l]$ and $max\_level < 2$ **then**
12. $max\_level \leftarrow 2$, $flag \leftarrow true$;
13. **end**
14. **if** $flag == false$ and $\varsigma^*[k][7] == \varsigma_{i,j}[k][7]$ **then**
15. label ς^* as "normal";
16. **if** $flag == false$ and $\varsigma^*[k][7]$!= $\varsigma_{i,j}[k][7]$ and $max\_level < 3$ **then**
17. $max\_level = 3$;
18. **end**
19. **end**
20. **end**
21. **end**
//max_level: 1 denotes sequence-class, 2 denotes pattern-class, 3 denotes content-class.
22. **if** ς^* is not labeled **then** ς^* is labeled according to max_level;

can differentiate 5 types of attacks. Although some features cannot directly separate one attack from other attacks individually, their combinations are able to serve as good classifiers. For example, as shown in Fig. 7, the features, Max R_{ND} and Number of CPSequences, can absolutely differentiate all attacks. Although fingerprinting different types of attacks is beyond the scope of this paper, the risk assessment method can provide significant features to support it. This indicates that the method is effective in assessing attacks.

4.4.2. Targeting different attacks

In general, different SCADA systems may have different defense priorities, which are determined by many factors, such as industrial processes, device performance, network security situation, and so on. In this paper, the parameters and functions in Eq. (9) are used to adapt to different evaluation requirements. For example, when physical parameters stored in devices are very important and sensitive to changes, the parameter r_2 tends to be larger than other parameters. Then Eq. (9) can be formulated as:

$$R = 0.1 * R_S + 0.7 * L_1(R_P) + 0.1 * L_2(R_{NS}) + 0.1 * L_3(R_{ND}) \tag{15}$$

In addition, when the performance of devices is limited, the status of the system is sensitive to the network load. The ranking function can be formulated as:

$$L_i(x) = \begin{cases} 0, x = 0 \\ 1, 0 < x < max_i \\ 2, max_i \le x < 2 * max_i \\ 3, 2 * max_i \le x \end{cases} \tag{16}$$

Obviously, different parameter and function combinations have different levels of sensitivity to different attacks. Eq. (10) with Eq. (11) is less sensitive to attacks, so some risk levels of attacks in Fig. 5 are smaller than corresponding ones in Fig. 8 which are calculated based on Eq. (10) with Eq. (16). Eq. (10) with Eq. (16) is sensitive to the network load, like DDoS attacks. As shown in Fig. 8, Attack 4 and Attack 6 show higher risk levels than other attacks. In contrast, Eq. (15) with Eq. (11) is sensitive to the changes in physical parameters, like false data injection attacks. Accordingly, Attack 5 shows a higher risk level than Attack 4 in Fig. 9. Therefore, we think that the method is able to target different attacks with different parameter and function combinations.

In summary, the risk assessment method is effective in distinguishing and targeting different attacks. It is worth mentioning that the proposed assessment method does not tend to put forward a standard to estimate various attacks. Instead, we think a combination of different assessment equations, like Eqs. (15) with Eq. (11) and Eqs. (10) with Eq. (16), will provide a better way to estimate different attacks from different perspectives.

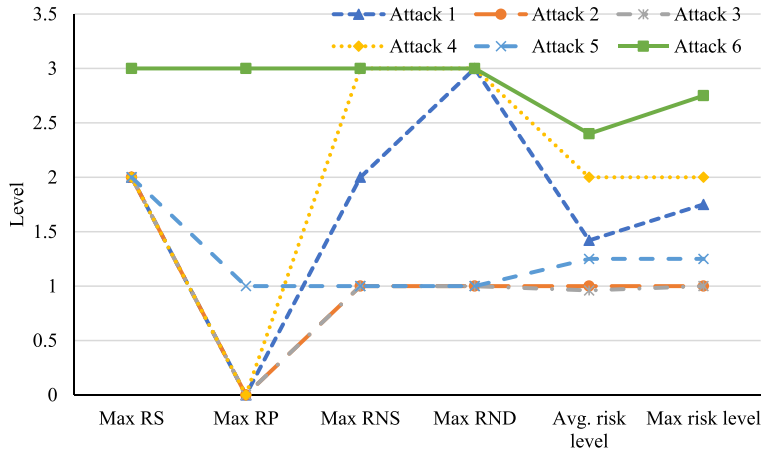


Fig. 5. Statistical features of attacks based on Eq. (10) with Eq. (11).

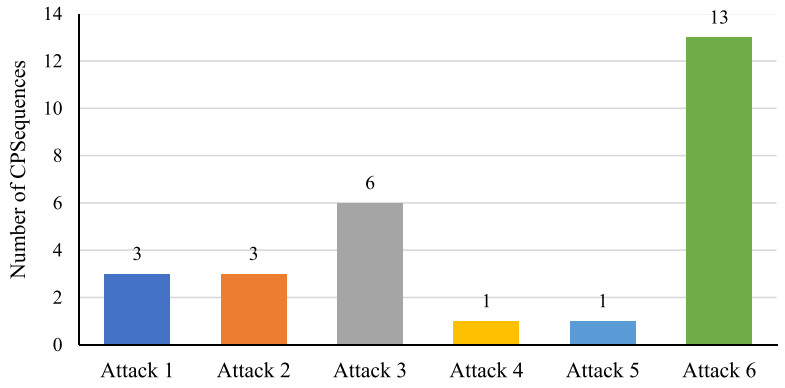


Fig. 6. The number of CPSequences of attacks

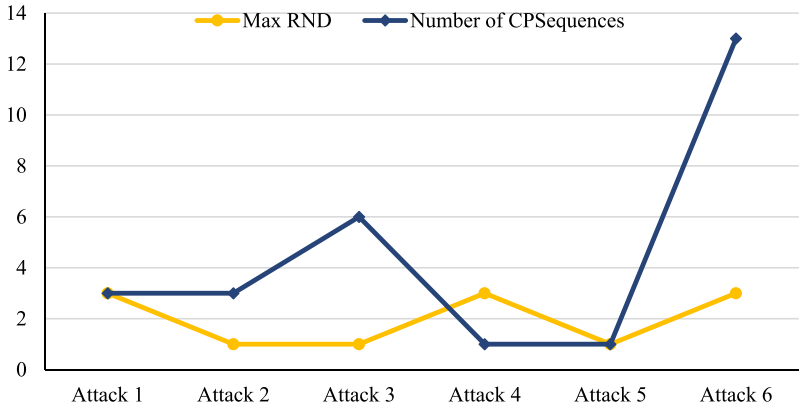


Fig. 7. The max R_{ND} and number of CPSequences of attacks

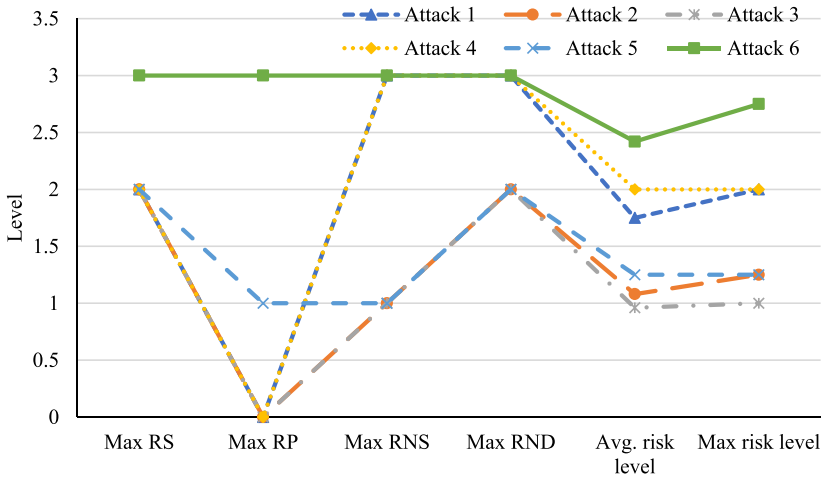


Fig. 8. Statistical features of attacks based on Eq. (10) with Eq. (16).

5. Discussion

Network attacks against the SCADA system can be roughly divided into three categories: network-level, device-level, and logic-level. Network-level attacks try to find interesting targets (like network scanning), or deplete the resources of the SCADA network like D/DoS attacks. TCP/IP-level methods are good at detecting them. Device-level attacks tend to compromise one ICS device as the foothold though exploiting its vulnerabilities. The compromised device can be directly used to affect the SCADA system or be used to attack other devices for a more complicated attack. ICS-level methods are always used to counter such attacks. Logic-level attacks usually refer to the false data injection attacks, which can be carried out by "legitimate network traffic". That is the reason why physical-level methods are proposed. Network-level and device-level methods can provide the necessary conditions for logic-level attacks, such as network information, device information, industrial process, and available resources of the target SCADA system. Obviously, there is no way to deal with all of these attacks. And these

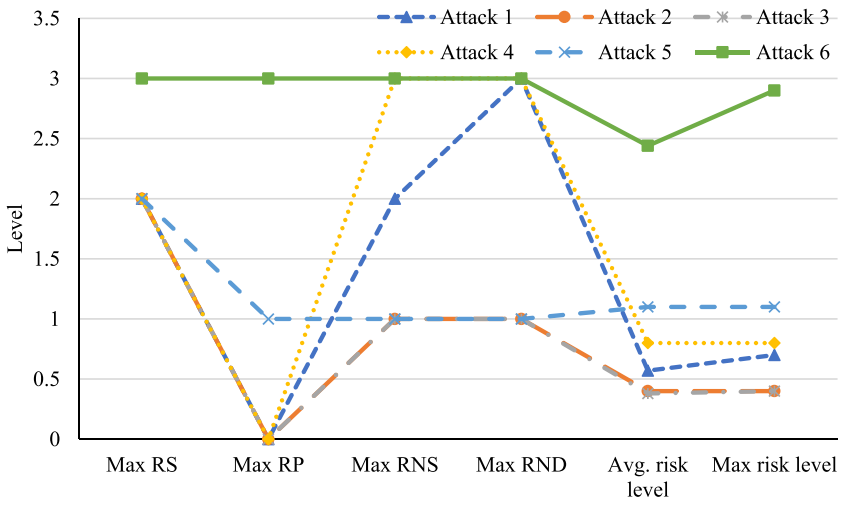


Fig. 9. Statistical features of attacks based on Eq. (15) with Eq. (11).

attacks can cause different impact on the SCADA system.

One main advantage of the method is that it associates network intrusions with the industrial process of the SCADA system. Although there were many ways to detect intrusions from SCADA network traffic, few of them were able to determine whether these intrusions were harmful to the system. Therefore, the results of these network intrusion detection methods could not be directly used to monitor the security status of the system. Some approaches attempted to model the industrial process of the SCADA system in order to detect stealthy and sophistic attacks and estimate their impact. However, these approaches were always passive, and could not detect attacks in advance. In order to alleviate this situation, this paper proposes a cyber-physical model for the SCADA system.

Because the proposed method focuses on analyzing ICS protocols, only the TCP/IP information about non-ICS application-layer protocols are considered in the model. If an attacker tries to compromise ICS devices through a non-ICS protocol, the proposed model is able to detect the related intrusion, but may underestimate its risk level. For example, an attack can use the protocol SMB to send an EXE file from a compromised RTU to another compromised RTU. Based on the risk assessment method in this paper, the attack will not be considered to have an impact on the industrial process.

In addition, an extreme content-class intrusion is beyond the scope of this paper. The abnormal CPSequence in the intrusion is identical to an existing CPSequence in terms of the order and content of communication pattern based on our definitions. However, the specific data content sent to RTUs has been manipulated. For example, a regular manual operation is used to send malicious data. Although this attack can effectively avoid the proposed method, the attacker has to encounter a series of challenges. First, s/he has to compromise an internal ICS device as the original foothold. Then, s/he has to monitor and analyze the network traffic to learn the communication patterns. In addition, s/he has to understand the industrial process in order to find the real target. Finally, s/he has to simulate the operator sending commands to the targets. Note that if the existing CPSequences in the model cannot meet the attacker's requirements, the attack will be detected due to its abnormal behavior.

6. Conclusion

This paper presents a novel cyber-physical model for the SCADA system. The model associates network attacks with the industrial process of the SCADA system through modeling the states of ICS devices and mapping attacks to operations on them. By this way, the model can not only detect intrusions against the SCADA network, but also assess their risk levels against the industrial process. One prototype of the model is implemented on the SCADA network data sets. The experimental results show that the prototype can detect various attacks in the data sets with accuracy of 0.9969, which is more effective than the existing approaches. In addition, the risk assessment method is effective in distinguishing different attacks and targeting various attacks through tuning the parameters in the risk level equation. Moreover, the model can be further extended by considering the following two aspects: mapping non-ICS network traffic to operations on ICS devices, and correlating the states of ICS devices to detect more stealthy network attacks.

CRediT authorship contribution statement

Chuan Sheng: Conceptualization, Formal analysis, Methodology, Software, Writing - original draft. **Yu Yao:** Funding acquisition, Project administration, Supervision. **Qiang Fu:** Data curation, Software, Validation. **Wei Yang:** Investigation, Visualization, Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This paper is supported by Key Research and Development Program of Liaoning Province under Grant No. 2019JH2/10100019, and "the Fundamental Research Funds for the Central Universities" under Grant Nos. N181606001, N2016011, and N2024005-1.

We declare that there is no conflict of interest regarding the publication of this paper.

References

[1] S. Ponomarev, T. Atkison, Industrial control system network intrusion detection by telemetry analysis, IEEE Trans. Dependable Secure Comput. 13 (2) (2016) 252–260.

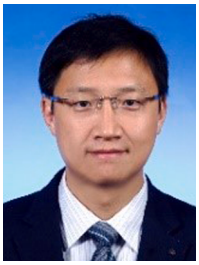
[2] H.R. Ghaeini, N.O. Tippenhauer, HAMIDS: hierarchical monitoring intrusion detection system for industrial control systems, in: Proceedings of the 2nd ACM Workshop on Cyber-Physical Systems Security and Privacy - CPS-SPC '16, 2016, pp. 103–111.

[3] M.Z. Gunduz, R. Das, Cyber-security on smart grid: threats and potential solutions, Comput. Netw. 169 (2020).

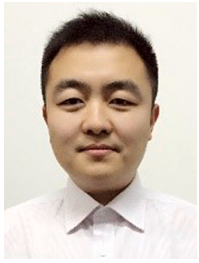
[4] S. Ghosh, S. Sampalli, A survey of security in SCADA networks: current issues and future challenges, IEEE Access 7 (2019) 135812–135831.

[5] D.S. Pidikiti, R. Kalluri, R.K.S. Kumar, B.S. Bindhumadhava, SCADA communication protocols: vulnerabilities, attacks and possible mitigations, CSI Trans. ICT 1 (2) (2013) 135–141.
[6] M.B. Mohamad Noor, W.H. Hassan, Current research on internet of things (IoT) security: a survey, Comput. Netw. 148 (2019) 283–294.
[7] N. Falliere, L.O. Murchu, and E. Chien, W32. Stuxnet Dossier. 2011 [Online]. Available: https://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers, [Accessed 15 December 2011].
[8] A. Hern, Ukrainian blackout caused by hackers that attacked media company. Researchers Say, The Guardian, 2016. Online]. Available, http://www.theguardian.com/technology/2016/jan/07/ukrainian-blackout-hackers-attacked-media-company [Accessed 7 January 2016]..
[9] M. J. Assante, and R. M. Lee, The industrial control system cyber kill chain. 2015 [Online]. Available: https://www.sans.org/reading-room/whitepapers/ICS/industrial-control-system-cyber-kill-chain-36297, [Accessed 19 May 2015].
[10] E. Bou-Harb, M. Debbabi, C. Assi, On fingerprinting probing activities, Comput. Secur. 43 (2014) 35–48.
[11] E.M. Hutchins, M.J. Cloppert, R.M. Amin, Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains, Lead. Issues Inf. Warfare Secur. Res. 1 (1) (2011) 80.
[12] E.-M. Kalogeraki, N. Polemi, S. Papastergiou, T. Panayiotopoulos, Modeling SCADA attacks. Smart Trends in Systems, Security Sustainability, 2018, pp. 47–55. Lecture Notes in Networks and Systems.
[13] Y. Yao, C. Sheng, Q. Fu, H.X. Liu, D.J. Wang, A propagation model with defensive measures for PLC-PC worms in industrial networks, Appl. Math. Modell. 69 (2019) 696–713. May.
[14] B. Merino, Modbus stager: using PLCs as a payload/shellcode distribution system. 2016 [Online]. Available: http://www.shelliscoming.com/2016/12/modbus-stager-using-plcs-as.html, [Accessed 1 May 2016].
[15] R. Spenneberg, M. Brüggemann, H. Schwartke, Plc-blaster: a worm living solely in the plc, in: Black Hat Asia, 16, 2016.
[16] Á. MacDermott, Q. Shi, M. Merabti, K. Kifayat, Intrusion detection for critical infrastructure protection, in: 13th Post Graduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting (PGNet2012), 2012.
[17] S. Sridhar, M. Govindarasu, Model-based attack detection and mitigation for automatic generation control, IEEE Trans. Smart Grid 5 (2) (2014) 580–591.
[18] S. Adepu, A. Mathur, Distributed attack detection in a water treatment, IEEE Trans. Dependable Secur. Comput. (2018) 1.
[19] Q. Gu, D. Formby, S. Ji, H. Cam, R. Beyah, Fingerprinting for cyber-physical system security device physics matters too. IEEE Secure Privacy, 2018, pp. 49–59.
[20] R.R.R. Barbosa, R. Sadre, A. Pras, Difficulties in modeling SCADA traffic: a comparative analysis, Passiv. Act. Measure. (2012) 126–135. Lecture Notes in Computer Science.
[21] I. Garitano, C. Siaterlis, B. Genge, R. Uribeetxeberria, U. Zurutuza, A method to construct network traffic models for process control systems, in: Proceedings of 2012 IEEE 17th International Conference on Emerging Technologies & Factory Automation (ETFA 2012), 2012, pp. 1–8.
[22] B. Genge, D.A. Rusu, P. Haller, A connection pattern-based approach to detect network traffic anomalies in critical infrastructures, in: Proceedings of the Seventh European Workshop on System Security - EuroSec '14, 2014, pp. 1–6.
[23] V. Gabriel, R.S. Miani, B.B. Zarpelao, Flow-Based Intrusion Detection for SCADA networks using Supervised Learning, in: XVII Simpósio Brasileiro em Segurança da Informação e de Sistemas, 2017, pp. 168–181.
[24] S. Shitharth, K. Sangeetha, B. Praveen Kumar, Integrated probabilistic relevancy classification (PRC) scheme for intrusion detection in SCADA network, Des. Framew. Wirel. Netw. (2020) 41–63. Lecture Notes in Networks and Systems.
[25] X. Xia, X. Liu, J. Lou, A network traffic prediction model of smart substation based on IGSA-WNN, ETRI J. 42 (3) (2020) 366–375.
[26] B. Hu, C. Zhou, Y.-C. Tian, Y. Qin, X. Junping, A Collaborative intrusion detection approach using blockchain for multimicrogrid systems, IEEE Trans. Syst., Man Cybern. 49 (8) (2019) 1720–1730.
[27] R.A. Niazi, Y. Faheem, A bayesian game-theoretic intrusion detection system for hypervisor-based software defined networks in smart grids, IEEE Access 7 (2019) 88656–88672.
[28] N. Goldenberg, A. Wool, Accurate modeling of Modbus/TCP for intrusion detection in SCADA systems, Int. J. Crit. Infrastruct. Prot. 6 (2) (2013) 63–75.
[29] W. Shang, J. Cui, M. Wan, P. An, P. Zeng, Modbus Communication behavior modeling and SVM intrusion detection method, in: Proceedings of the 6th International Conference on Communication and Network Security - ICCNS '16, 2016, pp. 80–85.
[30] W. Yusheng, F. Kefeng, L. Yingxu, L. Zenghui, Z. Ruikang, et al., Intrusion detection of industrial control system based on modbus TCP protocol, in: 2017 IEEE 13th International Symposium on Autonomous Decentralized System (ISADS), 2017, pp. 156–162.
[31] T. Marsden, N. Moustafa, E. Sitnikova, G. Creech, Probability risk identification based intrusion detection system for SCADA systems, Mobile Netw. Manage. (2018) 353–363. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering.
[32] I.A. Khan, D. Pi, Z.U. Khan, Y. Hussain, A. Nawaz, HML-IDS: a hybrid-multilevel anomaly prediction approach for intrusion detection in SCADA systems, IEEE Access 7 (2019) 89507–89521.
[33] C. Guo, B. Yu, J. Guo, B. Wen, J. Zhang, et al., Security risk assessment of the IEC61850-based substation automation system, Proc. CSEE 34 (4) (2014) 685–694.
[34] P. Jokar, V. Leung, Intrusion detection and prevention for ZigBee-based home area networks in smart grids, IEEE Trans. Smart Grid (2016) 1.
[35] D.I. Urbina, J.A. Giraldo, A.A. Cardenas, N.O. Tippenhauer, J. Valente, et al., Limiting the impact of stealthy attacks on industrial control systems, in: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security - CCS'16, 2016, pp. 1092–1105.
[36] J. Hao, E. Kang, J. Sun, Z. Wang, Z. Meng, et al., An adaptive markov strategy for defending smart grid false data injection from malicious attackers, IEEE Trans. Smart Grid 9 (4) (2018) 2398–2408.
[37] N. Neha, S. Priyanga, S. Seshan, R. Senthilnathan, V.S. Shankar Sriram, SCO-RNN: a behavioral-based intrusion detection approach for cyber physical attacks in SCADA systems, Invent. Commun. Comput. Technol. (2020) 911–919. Lecture Notes in Networks and Systems.
[38] S. Selvarajan, M. Shaik, S. Ameerjohn, S. Kannan, Mining of intrusion attack in SCADA network using clustering and genetically seeded flora-based optimal classification algorithm, IET Inf. Secur. 14 (1) (2020) 1–11.
[39] S. Wang, S. Bi, Y.-J.A. Zhang, Locational detection of false data injection attack in smart grid: a multi-label classification approach, IEEE Internet Things J. (2020) 1.
[40] D.K. Molzahn, J. Wang, Detection and characterization of intrusions to network parameter data in electric power systems, IEEE Trans. Smart Grid 10 (4) (2019) 3919–3928.
[41] G. Efstathopoulos, P.R. Grammatikis, P. Sarigiannidis, V. Argyriou, A. Sarigiannidis, et al., Operational data based intrusion detection system for smart grid, in: 2019 IEEE 24th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), 2019, pp. 1–6.
[42] C. Kolias, G. Kambourakis, A. Stavrou, J. Voas, DDoS in the IoT: mirai and other botnets, Computer 50 (7) (2017) 80–84.
[43] A. Lemay, J.M. Fernandez, S. Knight, A Modbus command and control channel, in: IEEE Systems Conference, 2016.
[44] J. Zhao, Y.-L. Chen, Z. Chen, F. Lin, C. Wang, et al., Modeling and control of discrete event systems using finite state machines with variables and their applications in power grids, Syst. Control Lett. 61 (1) (2012) 212–222.
[45] K. Jezernik, R. Horvat, J. Harnik, Finite-state machine motion controller: servo drives, IEEE Ind. Electron. Mag. 6 (3) (2012) 13–23.
[46] A. Lemay, J.M. Fernandez, Providing SCADA network data sets for intrusion detection research, in: 9th Workshop on Cyber Security Experimentation and Test (CSET 16), 2016.
[47] B. Claise, B. Trammell, and P. Aitken, "Specification of the IP flow information export (IPFIX) protocol for the exchange of flow information," 2013.
[48] G. Hug, J.A. Giampapa, Vulnerability assessment of AC state estimation with respect to false data injection cyber-attacks, IEEE Trans. Smart Grid 3 (3) (2012) 1362–1370.
[49] Kaitoy, Pcap4J. 2020 [Online]. Available: https://github.com/kaitoy/pcap4j, [Accessed 1 May 2020].
[50] TShark, TShark 2020 [Online]. Available: https://www.wireshark.org/docs/man-pages/tshark.html, [Accessed 1 May 2020].

Chuan Sheng received the B.Sc. and M.Sc. degrees in Computer Science from Northeastern University of China in 2014 and 2016, respectively, and he is currently working toward the Ph.D. degree at the College of Compute Science and Engineering, Northeastern University, Shenyang, China, under the supervision of Prof. Yu Yao. He joined the Engineering Research Center of Security Technology of Complex Network System, Shenyang, China, in 2018. His research topics of interest include network security situation awareness, network intrusion detection, network threat intelligence analysis and big data of cybersecurity. His email address is 1610538@stu.neu.edu.cn.



Yu Yao received the B.Sc., M.Sc. and Ph.D. degrees in computer science from the Northeastern University, Shenyang, China in 1998, 2001 and 2005, respectively. He is Professor and Ph.D. Tutor at the Northeastern University, Shenyang, China since 2011. And he was Deputy Director of Shenyang Big Data Administration Bureau from 2015 to 2018. His main research direction is network security, data analysis and modeling, data visualization, nonlinear dynamic system analysis. His email address is yaoyu@mail.neu.edu.cn.



Qiang Fu received the B.Sc. and M.Sc. degrees in physics from Nanjing University of Information Science and Technology, Nanjing, China in 2013 and 2016, respectively, and he is currently working toward the Ph.D. degree at the Northeastern University, Shenyang, China, under the supervision of Prof. Yu Yao. He joined the Engineering Research Center of Security Technology of Complex Network System, Shenyang, China, in 2018. His research interests include network security, nonlinear dynamic system analysis and malware propagation modeling. His email address is qiang.fu@outlook.com.

Wei Yang received the B.Sc., M.Sc. and Ph.D. degrees in computer science from the Northeastern University, Shenyang, China in 1998, 2001 and 2012, respectively. She is lecturer at the Northeastern University, Shenyang, China since 2004. Currently she is a visiting scholar at the University of British Columbia in 2019. Her main research direction is network security, malware propagation modeling, nonlinear dynamic system analysis. Her email address is yangwei@mail.neu.edu.cn.