Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Detection and localization of cyber attacks on water treatment systems: an entropy-based approach*

Ke LIU[1], Mufeng WANG[2], Rongkuan MA[1], Zhenyong ZHANG[2], Qiang WEI[‡1]

[1]*State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou 450001, China*

[2]*College of Control Science and Engineering, Zhejiang University, Hangzhou 310027, China*

E-mail: bendawang@gmail.com; csewmf@zju.edu.cn; rongkuan233@gmail.com;
zhangzhenyong@zju.edu.cn; funnywei@163.com

Received Oct. 13, 2020; Revision accepted Jan. 21, 2021; Crosschecked Jan. 18, 2022; Published online Apr. 1, 2022

Abstract: With the advent of Industry 4.0, water treatment systems (WTSs) are recognized as typical industrial cyber-physical systems (iCPSs) that are connected to the open Internet. Advanced information technology (IT) benefits the WTS in the aspects of reliability, efficiency, and economy. However, the vulnerabilities exposed in the communication and control infrastructure on the cyber side make WTSs prone to cyber attacks. The traditional IT system oriented defense mechanisms cannot be directly applied in safety-critical WTSs because the availability and real-time requirements are of great importance. In this paper, we propose an entropy-based intrusion detection (EBID) method to thwart cyber attacks against widely used controllers (e.g., programmable logic controllers) in WTSs to address this issue. Because of the varied WTS operating conditions, there is a high false-positive rate with a static threshold for detection. Therefore, we propose a dynamic threshold adjustment mechanism to improve the performance of EBID. To validate the performance of the proposed approaches, we built a high-fidelity WTS testbed with more than 50 measurement points. We conducted experiments under two attack scenarios with a total of 36 attacks, showing that the proposed methods achieved a detection rate of 97.22% and a false alarm rate of 1.67%.

Key words: Industrial cyber-physical system; Water treatment system; Intrusion detection; Abnormal state; Detection and localization; Information theory

https://doi.org/10.1631/FITEE.2000546 **CLC number:** TP316.4

1 Introduction

Water treatment systems (WTSs) are critical infrastructures that are related to national economic and social stability (Fovino et al., 2012; Wikipedia, 2020a). With the development and popularization of information and communication technologies, traditionally closed WTSs have become accessible to the open Internet. Because WTS designs do not consider the security sector, cyber actors have a lot of entry points for launching attacks against WTS.

As a typical industrial cyber-physical system (iCPS), the security of WTSs has rarely been considered (Ponomarev and Atkison, 2016). Once attacked, the negative impact is incalculable. In the summer of 2013, a small New York dam suffered from an attack that resulted in a loss of $30 000 (The Wall Street Journal's San Francisco Bureau, 2015). In March 2016, a security incident report issued by Verizon stated that the WTS of Kemuri was attacked. The regular water supply has been damaged (SecurityWeek, 2016). In April 2020, Israel's water conservancy facilities, which include the Israeli supervisory control and data acquisition (SCADA) systems of waste water treatment facilities, water pumping stations, and sewerage networks, were

‡ Corresponding author

* Project supported by the National Natural Science Foundation of China (No. 61833015)

ORCID: Ke LIU, https://orcid.org/0000-0003-3386-7359; Qiang WEI, https://orcid.org/0000-0002-0288-0086 attacked (Kaspersky ICS CERT, 2020a). In addition, it is well recognized that there have been many undiscovered security events in WTSs (Walton, 2016).

According to historical security incidents, we find that industrial software, controllers, and communication protocols are surfaces that are usually exploited by attackers. To enhance the security of iCPSs, one of the best-known architectures is “defense-in-depth” (Stouffer et al., 2011). This architecture is established by deploying firewalls and intrusion detection systems (IDSs), along with effective security policies, etc. The IDS is a security technology that is widely used to protect the information technology (IT) system. However, most of IDSs are designed for IT systems and cannot be directly applied to a WTS (Ten et al., 2010) for the following reasons: First, WTSs follow the AIC (availability, integrity, confidentiality) principle, whereas the top priority of IT systems is confidentiality. Second, different from an IT system, there are many proprietary protocols in a WTS. Third, the stability requirements for a WTS are incredibly important, but its compatibility requirements are not.

In recent decades, researchers began to study IDSs for iCPSs. One of the most important problems is obtaining data (Khraisat et al., 2019). Generally, researchers build a testbed to obtain data and conduct experiments (Geng et al., 2019). However, the existing testbeds have poor flexibility and do not meet the research needs of multiple kinds of attacks and defenses (Mathur and Tippenhauer, 2016). Another important issue is the choice of metadata, which is critical for an efficient IDS. Most of the previous research used protocol data (Carcano et al., 2011), network traffic (Linda et al., 2011a), system configuration and log files (Hadeli et al., 2009; Feng et al., 2019), device response, and execution time as the metadata. However, in WTSs, fewer types of metadata are used. There are also other metadata, such as controller’s output states (COSs) that can better reflect the status of the WTS; however, it is difficult to obtain and process COSs. Therefore, new methodology should be developed to process COSs. In addition, most existing IDSs are used to locate intrusion points rather than abnormal points. However, abnormal point localization is more helpful in recovering the system than intrusion point localization. To the best of our knowledge, no one has located abnormal states in the physical plants. Availability is the top priority of a WTS, so if we can find the abnormal devices in time when the system is compromised, it can help system operators take measures to recover the system more quickly.

In this study, we focus on the COS and use it for intrusion detection. First of all, it is necessary to build a real-world testbed for experiments and data acquisition. Thus, we can conduct and test practical attacking and defensing measures. In primary data analysis, we denote the COS as a random variable. Then we find that when the system is in the normal status, COS entropy varies in certain ranges. However, when the system is compromised, the value of entropy increases. Based on this observation, we propose an entropy-based methodology to process the COS for intrusion detection. We also investigate the detection and localization of abnormal states, which can help system operators recover the system more quickly when the system suffers from a cyber-attack. In addition, we observe that this approach has a high false-positive rate with a static threshold for detection. Then we propose a dynamic threshold adjustment mechanism (DTAM).

The main contributions of this paper are as follows:

1. We build a high-fidelity WTS testbed for evaluating the intrusion detection method. The testbed includes more than 50 measurement points and is designed for practical attacking and defensing measures. Specifically, we design two attack scenarios, which include 36 attacks on the testbed.

2. We propose an entropy-based intrusion detection (EBID) method. EBID is established by collecting and processing the COSs in normal conditions. Then we calculate the joint entropy (JE) of the system determined by the multi-state condition and use it to evaluate the WTS security.

3. We propose a DTAM to improve the performance of EBID. We also propose an algorithm based on EBID to locate abnormal COSs, which helps system operators recover the WTS more quickly when the system is compromised.

4. We conduct a series of experiments on the testbed, the results of which show that the EBID method can achieve a detection rate of 97.22% and a false alarm rate of 1.67%, which validates the correctness of EBID.

2 Water treatment system

WTS is a system that improves water quality through several methods to make it suitable for a specific end use (Wikipedia, 2020b), such as drinking, industrial water supply, and irrigation. In this study, WTS refers to treatment of urban drinking water.

The architecture of WTS is shown in Fig. 1. WTS consists mainly of water treatment and sewage treatment. The main function of water treatment is to extract raw water from water sources and send it to the pump, and then mix, flocculate, sediment, and disinfect the raw water in the water treatment plant to turn it into tap water for residents. The sewage treatment collects the sewage generated by residents first, and then transports it to the sewage plant for grille treatment, precipitation, biochemical degradation, filtration, disinfection, and other treatments, and finally discharges the treated water into rivers, reservoirs, etc.

According to the annual assessment report of the US Department of Homeland Security (ICS-CERT, 2016) and the half-year threat landscape of Kaspersky ICS CERT (Kaspersky ICS CERT, 2019), WTSs have been the focus of cyber-attack and vulnerability research in recent years. Programmable logic controllers (PLCs), SCADA, and communication protocols in WTS are the attack surfaces that attackers can exploit to cause system damage, equipment damage, and casualties. The importance of research concerning WTS protection methods is self-evident. Because of important real-time availability requirements, security research cannot be carried out in the real WTS production environment. Therefore, it is necessary to build a testbed.

2.1 Water treatment system testbed

Most of the previous works built testbeds for attack and defense research. For example, SWaT (Mathur and Tippenhauer, 2016) is a well-known water treatment testbed and consists of a modern six-stage process. However, the data and models provided by these testbeds are insufficient to meet our research needs. Therefore, in this study, we design and build a real-world WTS testbed.

The main purpose of our testbed is to perform attack scenario analysis and verification. The testbed consists of two major parts: water treatment and sewage treatment. We simplify some of treatment details, but simulate the urban water treatment cycle, which is meaningful for the research on attack scenarios.

As shown in Fig. 2, the WTS testbed consists of two stages, which are controlled by two PLCs. In stage 1, pump P101 draws water from the reservoir and sends it to the water plant for purification; the water will flow through a distribution well, V-shaped filter tank, etc. After carbon treatment, mixing, flocculation, precipitation, and filtration, clean water is obtained for residents to use, and chemical treatment

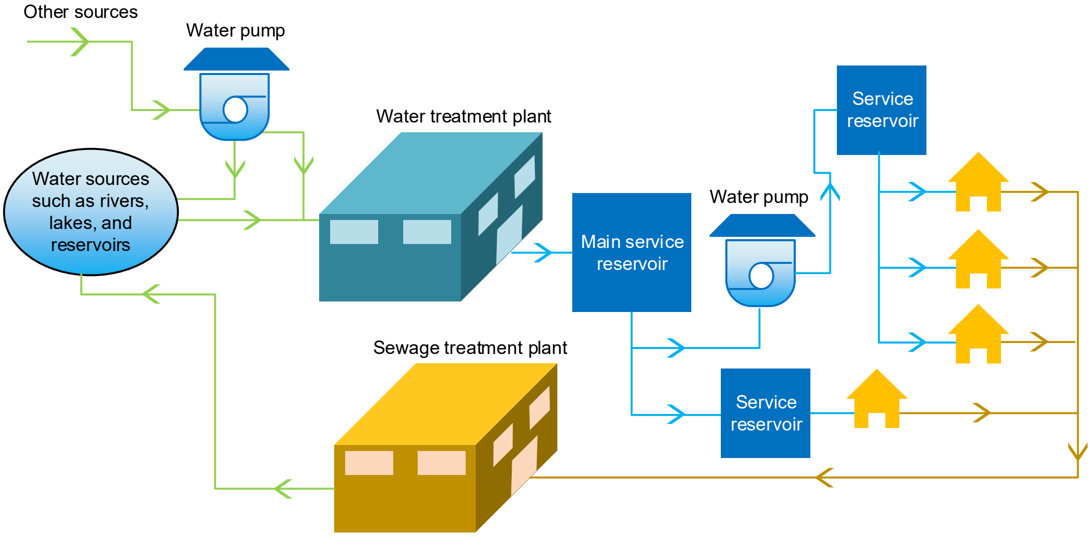

Fig. 1 Water treatment system (WTS) overview

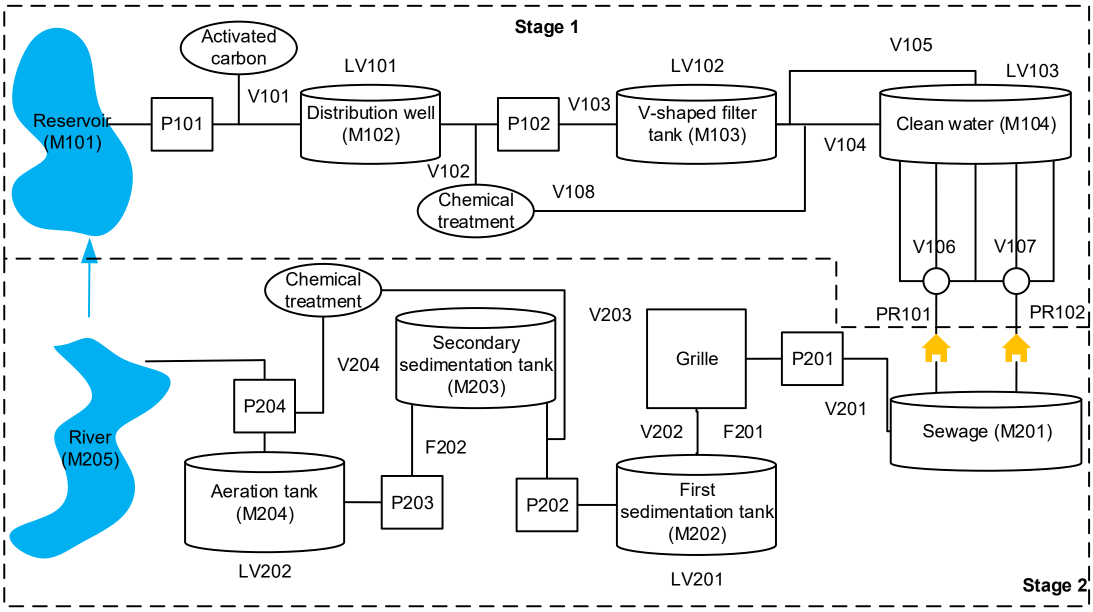

Fig. 2 Architecture of the water treatment system (WTS) testbed (LV: liquid level; P: pump; V: valve; PR: pressure; F: flow; M: well or tank)

is interspersed in the middle. In stage 2, the sewage generated by residents is discharged to the sewage treatment plant through the pipeline. The sewage will flow through the first sedimentation tank, secondary sedimentation tank, and aeration tank. After being treated by the sewage treatment plant, it will be sent to the river or reservoir. In Fig. 2, key pumps, valves, and some sensors are marked with tags, such as P101 and LV101. These tags are customized by the designers when the system is initialized and directly associated with a bank number and digital/analog pins of controllers' input/output (I/O).

The overall communication diagram of the WTS testbed is shown in Fig. 3. The PLCs obtain data from the sensors in the testbed. According to the real-time data, the PLCs control field devices, such as opening or closing the valves and pumps. The PLCs and upper computers, such as the engineer station, human–machine interface (HMI), and database, communicate with each other through a separate network that is based on Ethernet. The Modbus protocol is used between the PLCs and the upper computers for data and control command interaction and transmission. The PLCs and their directly connected sensors and actuators communicate with each other through the other network. S7-400 PLC and ABB AC500 PLC are used to control processes in the testbed. Moreover, TIA V14 SP1 software is used for the S7-400 PLC programming, and the Automation Builder v2.1 is used for the ABB AC500 PLC programming.

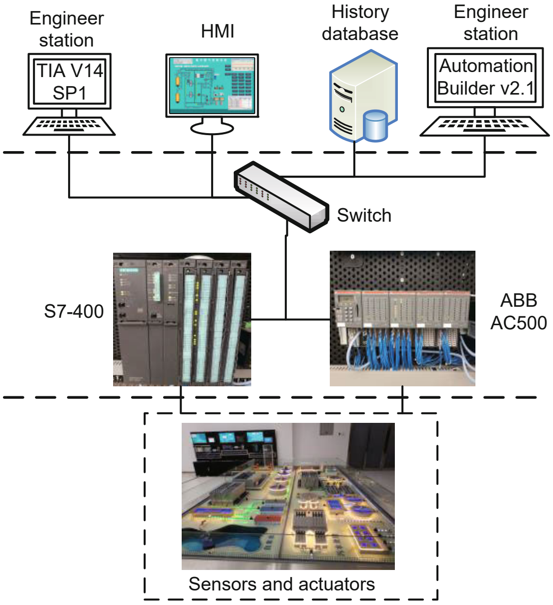

Fig. 3 Communication diagram of the water treatment system (WTS) testbed

In the testbed, more than 50 measurement points, which include nine analog outputs and 20 discrete outputs, are used to make the system run normally. The COSs are divided into two main types:

discrete output state (or digital output state) and analog output state (or continuous output state). The controller's discrete output state (CDOS) represents mainly discrete ON/OFF voltage signals, such as from a button, selector switch, travel switch, relay contact, photoelectric switch, and digital dialing. The controller's analog output state (CAOS) refers to a continuously changing signal, such as a potentiometer and various transmissions. Other special output states that stand for particular functions are not considered in this study.

2.2 Design for attack and defense

Next, we describe the special design of the testbed from two aspects: attack and defense.

1. Attack

First, all the water flowing in the testbed can be observed directly. Once the testbed is attacked and the status of WTS changes, we can observe the attack effects directly on the testbed. For example, when closing valves V106 and V107, the water is blocked and we can observe that the water level rises in the V-shaped filter tank until it overflows. In addition, the pipes in the testbed are not fixed when they are spliced. Therefore, if the water pressure in the pipe increases beyond the normal range, we can observe that the water overflows from the pipe joint. Also, the testbed has a physical reset function, which can drain all water and reset all the processes to the initial status we set. This reset function can reset the whole testbed within 3 min. It helps us conduct different attacks quickly. Based on these physical characteristics, we can design reconnaissance, network attacks, direct access attacks, etc.

2. Defense

How WTS responds to the attack is also an important problem. We design some basic rules to ensure safety. For example, when the water level in the V-shaped filter tank rises to the set maximum level, it will close the inlet valve to prevent overflow. When the pipe pressure increases, the testbed will take measures to reduce the amount of water entering the system, such as closing some valves and closing some pumps. These rules are also known as invariants. However, these invariants can be destroyed by cyber-attacks because they are designed for safety but not security.

In this study, we consider the industrial software, controllers, and communication protocols that are all attack targets of adversaries. We assume that attackers can access the control network and can compromise the upper computers or PLCs. Based on this assumption, we propose two attack scenarios including multiple attacks in the testbed.

3 Attack model

In 2019, more than 500 vulnerabilities were identified in different iCPS components and published on the US ICS-CERT website, covering dozens of vulnerability types (Kaspersky ICS CERT, 2020b). Among them, the iCPS software, SCADA, and PLC vulnerabilities represent more than 50% of the total. Using these vulnerabilities, we can design attack scenarios and experiment attacks.

After analyzing real iCPS security incidents, we find that once the status of the iCPS changes, the availability of the system will be affected. For example, "Stuxnet" (Farwell and Rohozinski, 2011) changes COSs to modify the rotor speeds of the centrifuges, causing damage to centrifuges. "Industroyer" (Lee et al., 2017) changes COSs and finally controls switches and circuit breakers, which cuts off one-fifth of Kiev's power for one hour. The status of an iCPS is determined by COSs, so we assume that no matter what methods the attackers use to invade WTS, the ultimate goal is to change the COSs, cause physical damage, and destroy the availability of WTS. The assumptions about attackers' capabilities can be listed as follows: (1) We assume that the system is secure initially when the system's JE is calculated for the first time. (2) We assume that the adversary has no knowledge of our intrusion detection method.

Based on these assumptions, two attack scenarios are proposed.

The first attack scenario, as indicated by the yellow path shown in Fig. 4, can be divided into three steps:

Step 1: The adversary first obtains privileges of the upper computer through the operating system and program software vulnerabilities (Nelson and Chaffin, 2011). In this step, the adversary can exploit vulnerabilities in the operating system and software or directly crack the upper computer password.

Step 2: After obtaining privileges of the upper computer, the adversary can use dynamic-link library (DLL) hijacking (Farwell and Rohozinski,

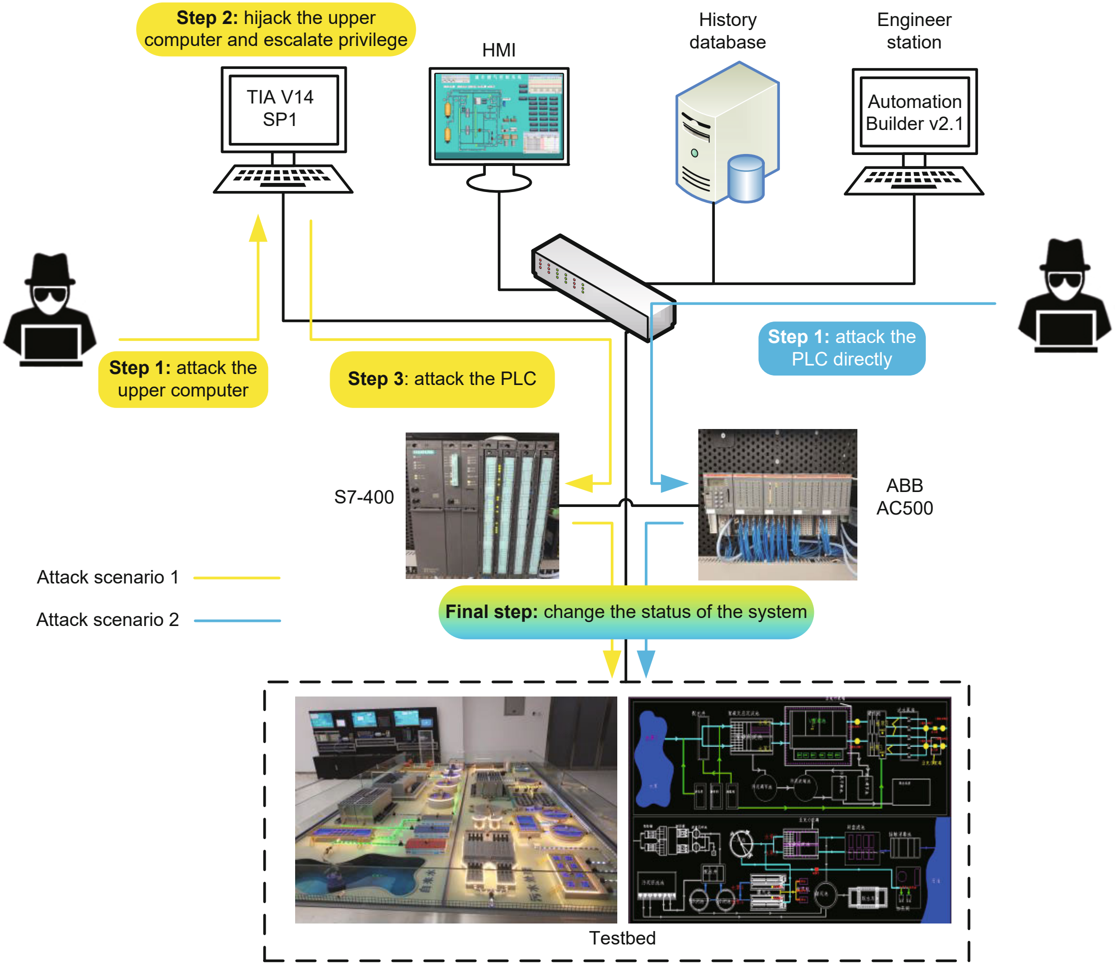
Step 2: hijack the upper computer and escalate privilege
TIA V14 SP1
HMI
History database
Engineer station
Automation Builder v2.1
Step 1: attack the upper computer
Step 3: attack the PLC
Step 1: attack the PLC directly
S7-400
ABB AC500
Attack scenario 1
Attack scenario 2
Final step: change the status of the system
Testbed

Fig. 4 Experimental environment

2011), program software password cracking, and other privilege escalation methods to secretly obtain elevated privileges stealthily. In this step, the adversary needs to obtain privileges (or privilege escalation) of the host computer and program software to remain hidden.

Step 3: Once the adversary obtains the proper privileges, the attacker can send counterfeit data, replay malicious packets, or directly manipulate the controller to change COSs to finally affect the regular operation of the WTS.

In the second attack scenario as indicated by the blue path shown in Fig. 4, the adversary can evade the upper computer and directly connect to the controller through the network. In this scenario, the adversary can directly attack the controller through multiple methods such as logic manipulation attack, system command injection attack, memory corruption attack, firmware modification attack, and malicious code within the firmware, and finally change COSs in the testbed (Ma et al., 2019).

The detailed steps of the above two attack scenarios are shown in Table A1 in Appendix.

According to the considered attack scenarios, we designed 36 attacks in our testbed for experiment. For example, one of the attacks can forcefully tamper with the PLC I/O to keep P101 and P102 open, and keep V106 and V107 closed. This will cause the pipeline pressure between M104 and V106/V107 to increase continuously, which results in pipeline crack. The attacker needs only to send two data packets to the field equipment. The impacts of the attacks include water level changes, water pipe bursts, pressure changes, valve status changes, and so on. We provide a list of attack details in Table A2 in Appendix.

4 Designing the entropy-based intrusion detection method

4.1 Overview

The main framework of the entropy-based intrusion detection method is shown in Fig. 5. EBID includes mainly the following four steps:

Step 1: preparing

EBID periodically obtains the COSs as the input of the model. We use two ways to obtain the input data in this study.

Step 2: preprocessing

Before building the model, we need to preprocess the data. Note that the entropy calculation methods of CDOS and CAOS are different. The correlation between two states also needs to be considered when calculating the JE of the system. In addition to classifying states during preprocessing, it is necessary to calculate the correlation between different states.

Step 3: modeling

After preprocessing, we can calculate the entropy of each state, and then the JE of the system. Before detection, EBID calculates a threshold. When the system is running, EBID compares the entropy value calculated in each cycle with the threshold to determine whether the system is secure or not.

Step 4: postprocessing

When we know the system is under attack or not, we can take actions appropriate to the situation. If the system's JE is smaller than the threshold, it means that the system is secure. Then EBID will record this value and dynamically adjust the current threshold using DTAM. If the system's JE is larger than the threshold, it means that the system is under attack. Then EBID will use the abnormal state localization algorithm to find the abnormal states caused by adversaries.

We deploy the system in the WTS control network. The data acquisition module transmits the data to the data analysis module. If the target is a multisite WTS, we can deploy the data acquisition module at each subsite to collect data. When obtaining input data, we assume that we know the communication protocol between the controllers and upper computers and the mapping between the variable values and actual COSs.

To evaluate the efficiency of the system, we consider two performance measures, i.e., the detection rate (DR, indicating the proportion of detected cases among all attacks) and the false positive rate (FPR, representing the proportion of normal cycles that are determined to be abnormal in all tests).

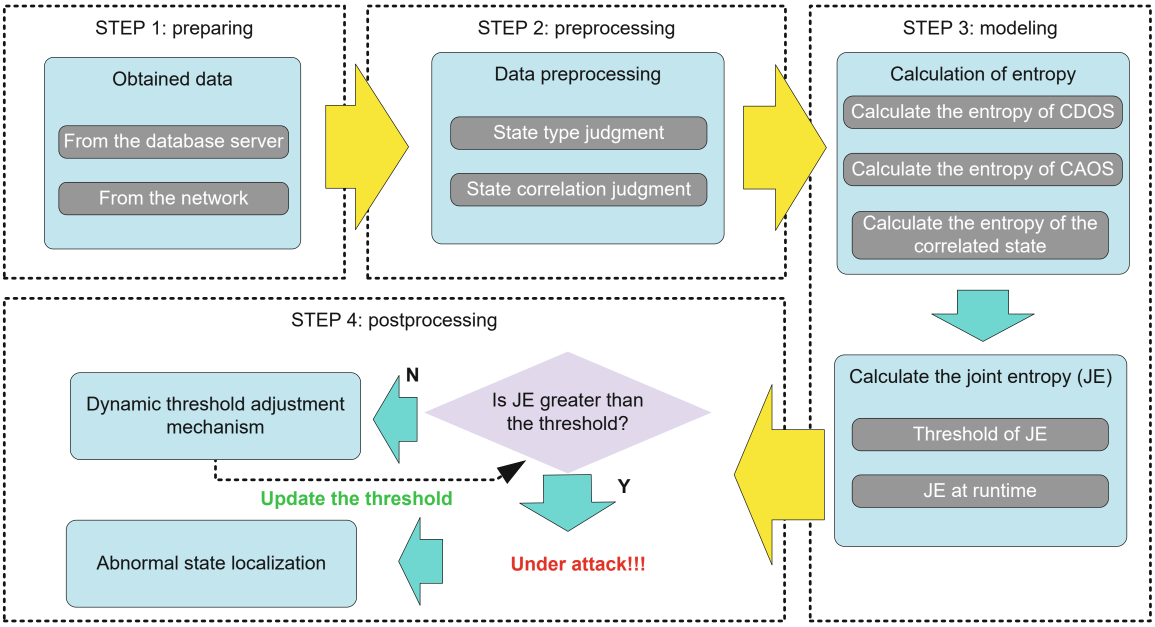
STEP 1: preparing
Obtained data
From the database server
From the network
STEP 2: preprocessing
Data preprocessing
State type judgment
State correlation judgment
STEP 3: modeling
Calculation of entropy
Calculate the entropy of CDOS
Calculate the entropy of CAOS
Calculate the entropy of the correlated state
Calculate the joint entropy (JE)
Threshold of JE
JE at runtime
STEP 4: postprocessing
Is JE greater than the threshold?
N
Y
Dynamic threshold adjustment mechanism
Update the threshold
Under attack!!!
Abnormal state localization

Fig. 5 Framework of entropy-based intrusion detection (EBID)

4.2 Preparing

In this subsection, we solve mainly the problem of obtaining model input data. There are two ways to obtain the state data (Fig. 6). The first way is to obtain data exported from software installed on the HMI or database. Taking the famous Siemens HMI product WINCC as an example, it supports the automatic export of state data using scripts so that the state data are periodically acquired for further analysis. The second way is to capture the traffic data packet directly from the network, and then parse the states according to the protocol specifications. By using these two methods at the same time and comparing their data, we can mitigate spoofing attacks from the network and ensure the reliability of the input data.

4.3 Preprocessing

Before calculating the system's JE, we need to calculate the entropy of a single state. Note that the CAOS and CDOS calculation methods are different. In addition, when calculating JE, we need to consider the correlation of different states; otherwise, the detection performance of EBID will be poor.

First, the definition of the quantity of security threat (QST) $T(\phi_i)$ is given. QST is used to indicate the influence of an output state on the security of iCPS. It is easy to understand that the security of the system depends on the frequency of occurrence of common values for COS. Taking a CDOS as an example, assume that the value of CDOS has 0.8 probability of being 1 (and 0.2 of being 0) in the normal status. The controller is more likely to be secure when we observe that the frequency of the CDOS value being 1 is closer to 0.8. Therefore, QST is a low value when COS reveals a high probability value, and the system is more likely to be secure. QST is high when COS reveals a low probability value, and the system is more likely to be unsecure. Then let $\gamma_{\mathrm{D}}(\phi)$ represent the set of all CDOSs and let $\gamma_{\mathrm{A}}(\phi)$ represent the set of all CAOSs. We have

$$\gamma(\phi) = \gamma_{\mathrm{D}}(\phi) + \gamma_{\mathrm{A}}(\phi), \tag{1}$$

where $\gamma(\phi)$ represents the set of all COSs in the WTS.

4.3.1 Entropy for CDOS and CAOS

Because CDOS and CAOS are different, their entropy calculation methods are different. Therefore, it is necessary to calculate their entropy separately.

For the CDOS $\phi_k \in \gamma_{\mathrm{D}}(\phi)$, the range of ϕ_k is $\mathit{\Gamma}(\phi_k)$. The probability mass function (PMF) $P(x)$ is obtained based on the experimental data. According to the previous analysis, it is inferred that QST is a function of PMF:

$$T(a_i) = f(P(a_i)),\ a_i \in \mathit{\Gamma}(\phi_k), \tag{2}$$

where $P(a_i)$ represents the probability when $\phi_k = a_i$, and $T(a_i)$ represents the QST of the a_i values.

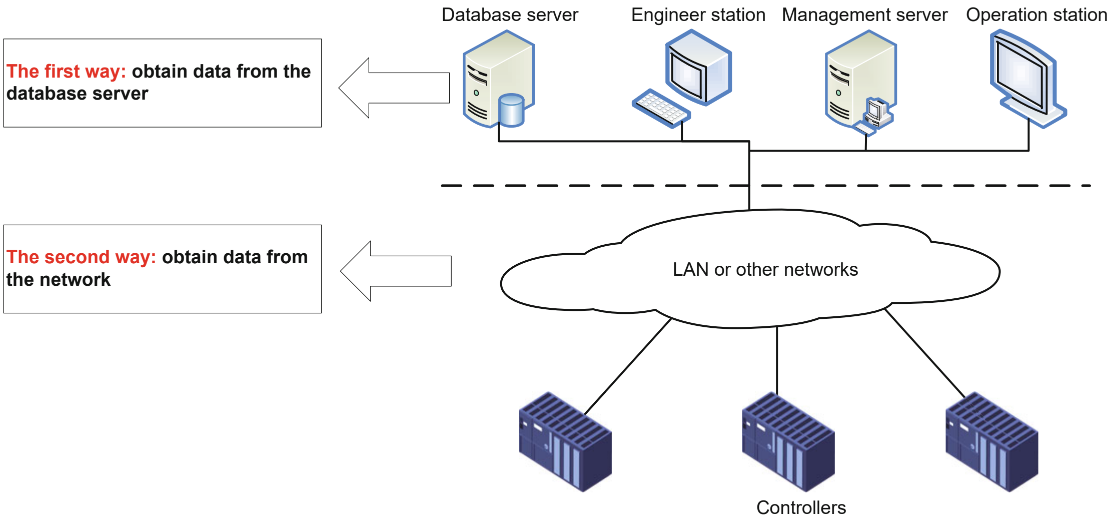

Fig. 6 Two methods of obtaining data in the testbed

Similar to the definition of self-information (Cover and Thomas, 2012), QST is defined as

$$T(a_i) = -\mathrm{lg}(P(a_i)). \tag{3}$$

QST refers to the impact of the value of a state on system security, so it is still a random variable. To measure the impact of a single state on system security, the entropy is calculated as

$$H_{\mathrm{D}}(X) = E[T(a_i)], \tag{4}$$

where $H_{\mathrm{D}}(X)$ represents the entropy of CDOS and is given by

$$H_{\mathrm{D}}(X) = -\sum_{i=1}^{N} P(a_i)\mathrm{lg}(P(a_i)), \tag{5}$$

where N represents the number of possible values of CDOS ϕ_k and satisfies

$$N = \mathrm{card}(\mathit{\Gamma}(\phi_k)), \tag{6}$$

where function "card" represents the number of elements of a finite set.

For the CAOS $\phi_k \in \gamma_{\mathrm{A}}(\phi)$, the cumulative distribution function (CDF) $F(x)$ is easily calculated according to the data, and then the probability density function (PDF) $f(x)$ is calculated. Similar to CDOS, the entropy of CAOS depends on the PDF. The formula for calculating entropy is extended to CAOS as

$$H_{\mathrm{A}}(\phi_k) = -\int_S f(x)\mathrm{lg}(f(x))\mathrm{d}x, \tag{7}$$

where $H_{\mathrm{A}}(\phi_k)$ represents the entropy of CAOS. S is the support set for the state ϕ_k and is expressed as

$$S = \{x|f(x) > 0\}. \tag{8}$$

4.3.2 Correlation between two states

The correlation between the two states also has a significant impact on the final result. In this study, we use an information gain ratio which depends on mutual information, to judge the correlation between two states. How to calculate the mutual information between different types should be determined first.

According to the previous analysis, if $\forall \phi_i, \phi_j \in \gamma(\phi)$, ϕ_i and ϕ_j are independent of each other, then the system's JE is described as

$$H_U(\phi_1, \phi_2, ..., \phi_M) = \sum_{k=1}^{M} H(\phi_k), \tag{9}$$

where $M = \mathrm{card}(\gamma(\phi))$.

However, in an actual iCPS, there is almost no situation in which any two states in a system are independent of each other. So, it is necessary to consider the existence of dependence between different states. Mutual information (Cover and Thomas, 2012) is introduced to determine whether two states are independent of each other. As for two states ϕ_i and ϕ_j, which depend on each other, letting $I(\phi_i, \phi_j)$ represent their quantity of mutual information to describe the relationship between the two states, we have

$$\begin{aligned} I(\phi_i, \phi_j) &= H(\phi_i) - H(\phi_i|\phi_j) \\ &= H(\phi_j) - H(\phi_j|\phi_i). \end{aligned} \tag{10}$$

Therefore, the value of mutual information depends on conditional entropy. Before talking about conditional entropy, the hypothesis is given that the conditional probability distribution of ϕ_i given ϕ_j obeys the Gaussian distribution (Tate, 1954). For example, if ϕ_j is a CDOS and ϕ_i is a CAOS, its conditional probability distribution is given as

$$P(x|\phi_j = y) = \frac{1}{\sigma\sqrt{2\pi}}\mathrm{e}^{-\frac{(x-\mu_y)^2}{2\sigma^2}}, \tag{11}$$

where $x, \mu_y \in \mathbb{R}$, $y \in \mathit{\Gamma}(\phi_k)$, $\sigma > 0$. We consider three different situations when calculating the conditional entropy of two states that depend on each other.

Case 1: The two states ϕ_i and ϕ_j depend on each other, and both of them are CDOSs. Their conditional entropy is calculated as

$$H(\phi_i|\phi_j) = \sum_{x\in\mathit{\Gamma}(\phi_j)} P(\phi_j = x)H(\phi_i|\phi_j = x). \tag{12}$$

Case 2: The two states ϕ_i and ϕ_j depend on each other, and both of them are CAOSs. Their conditional entropy is calculated as

$$H(\phi_i|\phi_j) = -\int f(x,y)\mathrm{lg}(f(x|y))\mathrm{d}x\mathrm{d}y. \tag{13}$$

Case 3: The two states ϕ_i and ϕ_j depend on each other, and ϕ_j is a CDOS while ϕ_i is a CAOS. In an iCPS, the value of a CDOS ϕ_j is 0 or 1; that is, $\mathit{\Gamma}(\phi_j) = \{0, 1\}$. Assuming that the probability of ϕ_j being 1 is p and being 0 is q, and according to Eq. (11), their joint PDF is given as

$$f_{\phi_i,\phi_j}(x,y) = \frac{p}{\sigma\sqrt{2\pi}}\mathrm{e}^{-\frac{(x-\mu_1)^2}{2\sigma^2}} + \frac{q}{\sigma\sqrt{2\pi}}\mathrm{e}^{-\frac{(x-\mu_0)^2}{2\sigma^2}}. \tag{14}$$

Then we can calculate the conditional entropy and the mutual information.

The next goal is to use mutual information to determine whether two states are independent of each other. Considering that the COS is divided into the CAOS and CDOS, these output states may not have a linear relationship. In this study, we use the information gain ratio to calculate the correlation between them. The definition of information gain ratio R, which represents the correlation between two states, is given as

$$\begin{aligned} R(\phi_i,\phi_j) &= \frac{I(\phi_i,\phi_j)}{\min(H(\phi_i),H(\phi_j))} \\ &= \frac{H(\phi_i)-H(\phi_i|\phi_j)}{\min(H(\phi_i),H(\phi_j))}. \end{aligned} \tag{15}$$

When the value of $R(\phi_i,\phi_j)$ is high, the correlation between the two states ϕ_i and ϕ_j is high, and the possibility that the two states are independent of each other is low. So, we have

$$0 \le I(\phi_i,\phi_j) \le \min(H(\phi_i),H(\phi_j)). \tag{16}$$

We can infer that the value range of R for the two states is $R\in[0,1]$. After calculating R, we use a threshold δ of R to determine whether two states are independent of each other. When $R(\phi_i,\phi_j)>\delta$, ϕ_i and ϕ_j are considered to be independent of each other. The mutual information of ϕ_i and ϕ_j needs to be removed from the system's JE.

4.4 Modeling

After entropy calculation and analysis of relevant states, all the preparatory work has been completed. The next problem is how to calculate the system's JE and use it for intrusion detection.

The mutual information of relevant state pairs needs to be subtracted when calculating JE. So, the system's JE is described as

$$H_U(\phi_1,\phi_2,\ldots,\phi_M)=\sum_{k=1}^{M}H(\phi_k)-\sum I(\phi_i,\phi_j), \tag{17}$$

where ϕ_i and ϕ_j depend on each other.

Because of the noise and other errors in the system, JE cannot be used as a threshold directly. Then a coefficient α is introduced to cover the impact of noise and error by amplifying JE. We can calculate the threshold $\bar{H}_U$ after calculating JE when the system is in the normal status. The threshold is

$$\bar{H}_U = H_U(1+\alpha),\ \alpha\in[0,1]. \tag{18}$$

Before normal operation of the system, EBID calculates a JE threshold. When the system is running normally, we set a calculation cycle T_u. In each cycle, we can obtain the amount of data that we have collected for a single COS as N_t, and we have

$$T_\mathrm{u}=\frac{N_t}{t}, \tag{19}$$

where t represents the data acquisition interval. JE will be calculated in each T_u. This means that the system has been attacked once JE exceeds the threshold $\bar{H}_U$ in a certain T_u.

4.5 Postprocessing

By comparing the JE that is calculated in each calculation cycle with the threshold, we can judge whether the system is under attack or not. The next problem is how we take action to address the issues faced by the system in different situations. When the system is not under attack, we try to dynamically adjust the threshold to improve the performance of EBID, or we try to locate the abnormal states.

4.5.1 Dynamic threshold adjustment mechanism

When analyzing the coefficient α according to the experimental data, we find that the determination of the JE threshold is a trade-off between DR and FPR. If the JE threshold is high, DR and FPR are low, and vice versa. FPR should also be considered while improving DR through the model (Yu et al., 2006). In addition, we find that when the initial α is fixed for continuous experiments, DR decreases and FPR increases with the increase of the number of types or number of attacks. Therefore, the value of the threshold should not remain static. To obtain a better detection efficiency, a DTAM that is based on variance is used to reduce FPR and improve DR.

First, the mean of the system $\hat{E}$ is given as

$$\hat{E}=\sum_{\phi_i\in\gamma(\phi)}\frac{H(\phi_i)}{M}. \tag{20}$$

Then the variance of the system is

$$\hat{\sigma}=\sqrt{\frac{\sum_{\phi_i\in\gamma(\phi)}(H(\phi_i)-\hat{E})^2}{M}}. \tag{21}$$

The data of one or more calculation cycles are collected in the normal status of the system to obtain an initial reliable JE $\bar{H}_{U_0}$, then dynamically adjusted according to the variance $\hat{\sigma}$ of the system. After n iterations, we can obtain a new threshold $\bar{H}_{U_n}$. Note that the premise of dynamic adjustment is that no intrusion behavior was detected in the previous calculation cycle. $\bar{H}_{U_n}$ is given as

$$\bar{H}_{U_n} = \begin{cases} \bar{H}_{U_0}, & \hat{\sigma} > V, \\ (1 - \varepsilon \frac{\hat{\sigma}}{V})\bar{H}_{U_0}, & \hat{\sigma} \leq V, \end{cases} \tag{22}$$

where V is a constant and $\varepsilon \in (0, 1)$ is a coefficient to determine the magnitude of DTAM. Considering that different systems have different V's, the JE threshold can be dynamically adjusted by the ratio of variance to V. A smaller value of ε and a larger value of DTAM's range will lead to a decrease in DR and FPR.

4.5.2 Locating abnormal states

In this subsection, we propose a method to locate the abnormal states when the system is attacked. The ultimate goal of attackers is to tamper with the COS and affect the availability of WTS, and locating abnormal states can help system operators take measures to recover WTS availability in time.

In the previous analysis, a method based on the information gain ratio is used as the basis for judging whether two states depend on each other. According to this, we can divide all mutually independent state pairs (represented by set U) and non-independent state pairs (represented by set I). Then U and I satisfy

$$I = \{\phi_i | \forall \phi_j \in \gamma(\phi), R(\phi_i, \phi_j) < \delta, \phi_i \in \gamma(\phi)\}, \tag{23}$$

$$U = \{(\phi_i, \phi_j) | R(\phi_i, \phi_j) > \delta, \phi_i \in \gamma(\phi), \phi_j \in \gamma(\phi)\}. \tag{24}$$

I and U are calculated at the beginning. After comparing the difference between I and U in the normal and abnormal statuses, we can locate the abnormal states. The detailed steps are shown in Algorithm 1. Then we can obtain the abnormal state set Ω.

Algorithm 1 first loops and explores the set $\gamma(\phi)$ and tries to find the independent states (line 1). It sets Flag as the loop control (line 2). It then calculates the information gain ratio to obtain all the independent states (lines 3–7). Once the independent state is not in I, it will add the state to Ω (lines 8–10). Then it calculates the non-independent state pairs and compares them with U (line 14). Once the calculated non-independent state pairs are not in set U, they must be abnormal states (lines 14 and 15).

Algorithm 1 Abnormal state localization method

```
Input: $R$, $\gamma(\phi)$, $I$, $U$
Output: Set of abnormal states $\Omega$
 1: for each state in $\gamma(\phi)$ do
 2:     Flag $\leftarrow$ 1
 3:     for each substate in $\gamma(\phi)$ do
 4:         if $R$(state, substate) $> \delta$ then
 5:             Flag $\leftarrow$ 0
 6:         end if
 7:     end for
 8:     if Flag == 1 and state not in $I$ then
 9:         Add state to $\Omega$
10:     end if
11: end for
12: for each state1 in $\gamma(\phi)$ do
13:     for each state2 in $\gamma(\phi)$ do
14:         if $R$(state1, state2) $> \delta$ and (state1, state2) not in
            $U$ then
15:             Add state1 and state2 to $\Omega$
16:         end if
17:     end for
18: end for
```

5 Evaluation

An experiment is conducted which includes 36 attacks in the WTS testbed; the ultimate goal of these attacks is to change the COS. For each case, take the first 10 critical calculation cycles that are intercepted for calculation. Make sure that the system is in the normal status in the first 10 calculation cycles, and all attack tests are performed beginning with the 11th calculation cycle. Note that the data acquisition interval in our testbed is 500 ms to simulate the SCADA data acquisition interval in a real water treatment system. In a real water plant, the data collection interval for most chemical links is 500 ms or even more, so we use 500 ms as the preset data acquisition interval.

DR is calculated as

$$\text{DR} = \frac{D_\text{a}}{36}, \tag{25}$$

where D_a is the number of cases correctly detected among all cases. FPR is calculated as

$$\text{FPR} = \frac{1}{36 \times 10} \sum_{i=1}^{36} \theta_i, \tag{26}$$

where θ_i is the number of calculation cycles that are marked as abnormal in the first 10 calculation cycles in each test case.

5.1 Impact of the information gain ratio and threshold coefficient

According to Section 4, there are two crucial coefficients of EBID (the threshold of information gain ratio δ and the threshold coefficient of JE α) which have direct impact on the overall system evaluation accuracy. A three-dimensional graph of DR, δ, and α is shown in Fig. 7. Note that DTAM is disabled in the evaluation of δ and α. From Fig. 7, it is easy to see that, with an increase of α, DR decreases significantly. Because α determines the JE threshold of the system, a larger α leads to a larger threshold and makes the system less sensitive to intrusion behaviors. With the value of δ increasing, the JE and the threshold also increase.

Fig. 8 shows the FPR change rule with different combinations of δ and α. A smaller α makes a higher FPR, which even exceeds 20%. The decrease in δ causes an increase in FPR. When considering the overall performance, it is necessary to comprehensively consider the trade-off between DR and FPR. A comparison of DR and FPR under different δ's and α's is shown in Fig. 9. The framework can achieve satisfiable performance when $\alpha = 0.3$ and $\delta = 0.85$. Although its DR is 94.44%, which is not as good as the highest (97.22%), it has achieved an FPR of 1.67%. Consider that a trade-off needs to be made according to the specific WTS, because different WTSs have different sensitivities to DR and FPR. So, $\alpha = 0.3$ and $\delta = 0.85$ are chosen in the experiments.

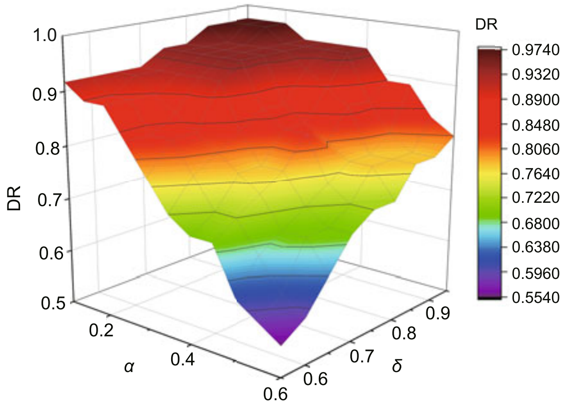

Fig. 7 Detection rate (DR) with different δ's and α's

5.2 Impact of the calculation cycle

The calculation cycle is also an important parameter affecting the detection performance. A longer calculation cycle creates a higher DR, and the COSs are periodic in a typical iCPS. Therefore, if T_{u} can better cover the COS cycle length, the comprehensive detection performance will be better. However, if the calculation cycle is too long, it will take more time to detect attacks. To find a satisfactory value of T_{u}, the varying DR and FPR are calculated with different T_{u} values, which are shown in Figs. 10 and 11, respectively. It is easy to see that DR is increasing, and FPR is decreasing when 110 s $\leq T_{\mathrm{u}} \leq$ 180 s, but DR remains unchanged and FPR increases when 180 s $< T_{\mathrm{u}} \leq$ 240 s. Also, a longer calculation period T_{u} takes a longer time to locate the abnormal states, which makes it difficult for the maintenance personnel to perform a system

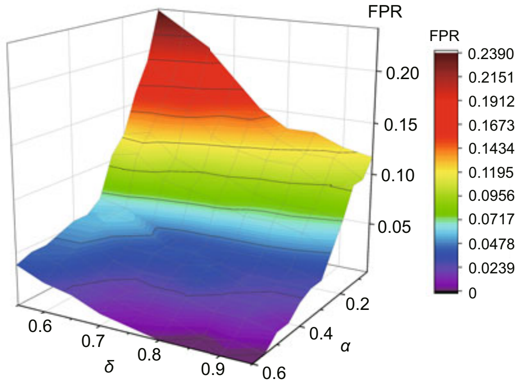

Fig. 8 False positive rate (FPR) with different δ's and α's

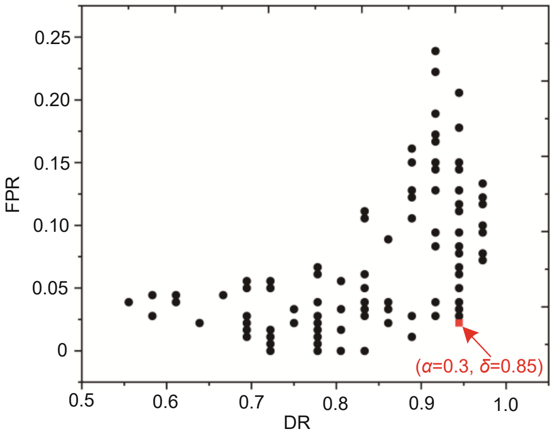

Fig. 9 False positive rate (FPR) and detection rate (DR) in different combinations of δ and α

recovery. Therefore, T_u should be as small as possible under the premise of ensuring that DR and FPR are acceptable.

Note that the calculation cycle in this study is more than 100 s, which seems to be too long. The main reason is that the preset data collection interval in our testbed is 500 ms. According to Eq. (19), under the premise of a certain number of values collected for a single COS, the larger the interval, the longer the detection time. In fact, we can set a minimum acquisition interval of 20 ms and obtain a detection time of less than 10 s, but this is not helpful in a real industrial environment.

5.3 Impact of the dynamic adjustment threshold

The last coefficient to be evaluated is the dynamic adjustment threshold of ε. For all cases, comparing DR and FPR with and without DTAM, the variations in DR and FPR for an increasing ε are shown in Tables 1 and 2, respectively. From Tables 1 and 2, we see that when ε increases, DR and FPR both increase. Then $\varepsilon = 0.08$ is chosen, which indicates that DR=97.22% and FPR=1.67%.

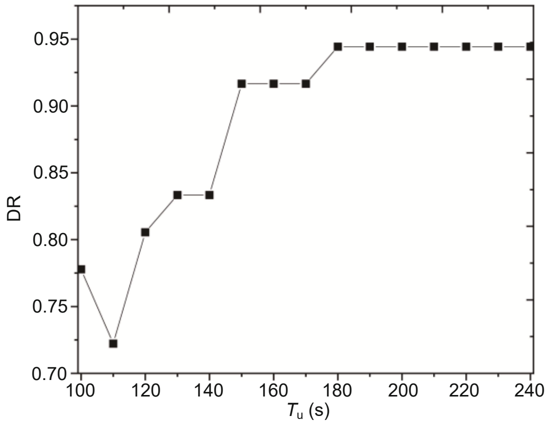

Fig. 10 The varying of the detection rate (DR) with different T_u's

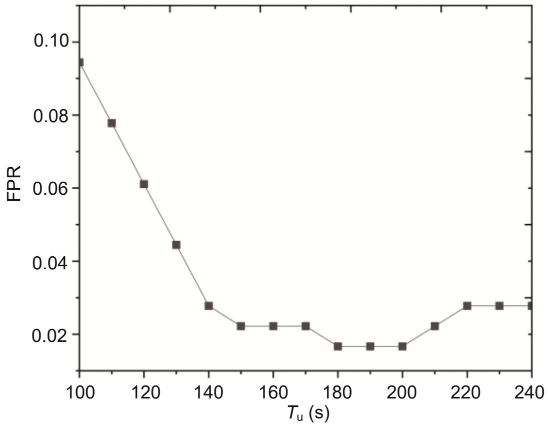

Fig. 11 The varying of the false positive rate (FPR) with different T_u's

From the results, we can see that the DR of the proposed model is obviously improved, and we can find a trade-off between DR and FPR. The experimental results show that the method has excellent performance in the system. The experimental results validate the rationality and validity of the model proposed in this study.

5.4 Impact of the abnormal state localization algorithm

To quantify the effectiveness of the algorithm in locating the abnormal state, we redefine false negative (FN) and false positive (FP) as follows:

FN: The results do not include the state attacked by the adversary.

FP: The results include states that are not attacked or indirectly affected by the adversary.

Fig. 12 shows that the FN of all cases is 0; that is, all the abnormal states have been found. However, in 66.7% of the cases, FP is greater than 0, and 25% of the cases have an FP higher than 8%. Algorithm 1 can help locate the abnormal state to a certain extent. Moreover, Algorithm 1 is used for only anomaly localization and cannot be used to measure system security. The security judgment is based only on the JE and the threshold.

Table 1 The varying of the detection rate (DR) with increasing ε

ε	DR without DTAM	DR with DTAM
0.04	94.44%	94.44%
0.08	94.44%	97.22%
0.12	94.44%	97.22%
0.16	94.44%	97.22%
0.20	94.44%	97.22%

DTAM: dynamic threshold adjustment mechanism

Table 2 The varying of the false positive rate (FPR) with increasing ε

ε	FPR without DTAM	FPR with DTAM
0.04	1.67%	1.67%
0.08	1.67%	1.67%
0.12	1.67%	2.22%
0.16	1.67%	3.33%
0.20	1.67%	6.11%

DTAM: dynamic threshold adjustment mechanism

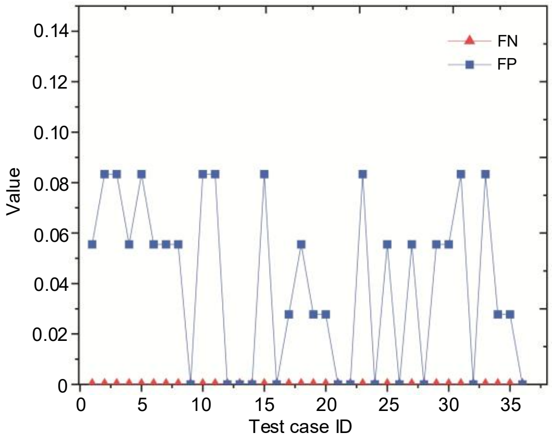

Fig. 12 The varying of FP and FN with different attacks

In a real WTS, the most important factor that affects the deployment of this method is the number of states. The method in this study can work normally in a real WTS with no more than 1000 states. Once the number of states exceeds 1000, the performance of our method decreases. In our previous field inspection of water plants, most independent water plants or sewage treatment plants have fewer than 800 states, and only a few large water plants have more than 1000 states. Therefore, our work can be applied in most independent water plants or sewage treatment plants.

6 Related works

Much research on IDS mechanisms in iCPSs has been carried out by researchers recently. IDS in iCPS can be divided into misuse-based intrusion detection and anomaly-based intrusion detection.

Misuse-based intrusion detection applies to some environments where strict false alarm rates are required. Specifically, many researchers have proposed targeted intrusion detection methods that are specific for different protocol types, such as the Modbus protocol (Vollmer et al., 2011; Barbosa et al., 2012; Morris et al., 2012; Wang et al., 2017), the distributed network protocol v3 (DNP3) (Lin et al., 2013), and the S7Comm protocol (Kleinmann and Wool, 2014). However, misuse-based intrusion detection technology is usually protocol-based, which makes the various proprietary protocols in the iCPS a problem that cannot be ignored. Moreover, misuse-based intrusion detection technology cannot resist the endless stream of new network attacks, and it is difficult to port between different iCPSs.

Anomaly-based intrusion detection can be considered as a classification problem (Sample and Schaffer, 2013). Intelligent computing technology has been introduced to design the IDS in iCPS, such as neural networks (Vollmer and Manic, 2009), fuzzy logic (Linda et al., 2011a, 2011b), and machine learning (Maglaras and Jiang, 2014; Terai et al., 2017). However, most of these intelligent computing technologies have significant computing power requirements, so these methods are very effective in certain resource-constrained iCPSs. For example, the fuzzy logic system needs many computing resources for validation, and membership function calculation may be difficult in some iCPSs. Although the intelligent computing technologies that are used in intrusion detection are versatile, they ignore many unique iCPS features that can be used for intrusion detection, such as the periodicity of iCPS traffic and behavior (Goldenberg and Wool, 2013; Song and Liu, 2019), configuration features (Zhang et al., 2019), and rules of message response and operation time (Formby et al., 2016).

In addition, the research of entropy-based intrusion detection technology in traditional information networks is more than that in industrial control systems (ICSs). In a traditional information network, entropy-based intrusion detection technologies are often used for distributed denial-of-service (DDoS) detection (Qian et al., 2009; Navaz et al., 2013), botnet detection (Bereziński et al., 2015), worm detection (Yu et al., 2006), etc., most of which are focused on traffic characteristics. In iCPSs, there is little research on entropy-based intrusion detection. Hu et al. (2020) proposed a permutation entropy based approach to detect stealthy attacks on ICS. They focused on residuals that are generated during stealthy attacks and used permutation entropy to characterize the non-randomness of residuals. Note that the research objects and evaluation methods of this paper are different from Hu et al. (2020)'s work. Hu and colleagues evaluated the residuals, and we focus on the COS, which has not been proposed in the literature. In calculating entropy, we consider the dependency between states and calculate the conditional entropy and joint entropy of multiple states. In addition, Hu et al. (2020)'s method can detect only stealthy attacks, whereas our method can detect more types of attacks; these attacks are listed in Appendix. For example, attack 19 in Table A2 is a stealthy attack in which the attackers use a DLL hijack and packet forgery to mislead operators.

7 Conclusions

In this paper, we first built a high-fidelity testbed to evaluate WTS vulnerabilities. Based on this testbed, we proposed EBID for intrusion detection by using COSs as input from the WTS. In addition, we improved the performance of EBID significantly by using a DTAM. To help system operators recover the system more quickly, we proposed an abnormal state detection and localization method. Finally, we conducted experiments over 36 attacks under two different attack scenarios. The results showed that EBID achieved a detection rate of 97.22% and a false alarm rate of 1.67%.

Contributors

Ke LIU and Mufeng WANG designed the research. Qiang WEI helped design the research. Ke LIU processed the data. Ke LIU and Mufeng WANG drafted the paper. Rongkuan MA, Zhenyong ZHANG, and Qiang WEI helped organize the paper. Ke LIU and Mufeng WANG revised and finalized the paper.

Compliance with ethics guidelines

Ke LIU, Mufeng WANG, Rongkuan MA, Zhenyong ZHANG, and Qiang WEI declare that they have no conflict of interest.

References

Barbosa RRR, Sadre R, Pras A, 2012. Towards periodicity based anomaly detection in SCADA networks. Proc 17$^{\text{th}}$ IEEE Int Conf on Emerging Technologies & Factory Automation, p.1-4. https://doi.org/10.1109/ETFA.2012.6489745

Bereziński P, Jasiul B, Szpyrka M, 2015. An entropy-based network anomaly detection method. *Entropy*, 17(4):2367-2408. https://doi.org/10.3390/e17042367

Carcano A, Coletta A, Guglielmi M, et al., 2011. A multidimensional critical state analysis for detecting intrusions in SCADA systems. *IEEE Trans Ind Inform*, 7(2):179-186. https://doi.org/10.1109/TII.2010.2099234

Cover TM, Thomas JA, 2012. Elements of Information Theory. John Wiley & Sons, New York, USA, p.250-252.

Farwell JP, Rohozinski R, 2011. Stuxnet and the future of cyber war. *Survival*, 53(1):23-40. https://doi.org/10.1080/00396338.2011.555586

Feng C, Reddy Palleti V, Mathur A, et al., 2019. A systematic framework to generate invariants for anomaly detection in industrial control systems. Proc Network and Distributed Systems Security Symp, p.1-22. https://doi.org/10.14722/ndss.2019.23265

Formby D, Srinivasan P, Leonard A, et al., 2016. Who's in control of your control system? Device fingerprinting for cyber-physical systems. Proc Network and Distributed Systems Security Symp, p.1-15. https://doi.org/10.14722/ndss.2016.23142

Fovino IN, Coletta A, Carcano A, et al., 2012. Critical state-based filtering system for securing SCADA network protocols. *IEEE Trans Ind Electron*, 59(10):3943-3950. https://doi.org/10.1109/TIE.2011.2181132

Geng YY, Wang Y, Liu WW, et al., 2019. A survey of industrial control system testbeds. *IOP Conf Ser Mater Sci Eng*, 569(4):042030. https://doi.org/10.1088/1757-899x/569/4/042030

Goldenberg N, Wool A, 2013. Accurate modeling of Modbus/TCP for intrusion detection in SCADA systems. *Int J Crit Infrastruct Protect*, 6(2):63-75. https://doi.org/10.1016/j.ijcip.2013.05.001

Hadeli H, Schierholz R, Braendle M, et al., 2009. Leveraging determinism in industrial control systems for advanced anomaly detection and reliable security configuration. Proc IEEE Conf on Emerging Technologies & Factory Automation, p.1-8. https://doi.org/10.1109/ETFA.2009.5347134

Hu Y, Li H, Luan TH, et al., 2020. Detecting stealthy attacks on industrial control systems using a permutation entropy-based method. *Fut Gener Comput Syst*, 108:1230-1240. https://doi.org/10.1016/j.future.2018.07.027

ICS-CERT, 2016. ICS-CERT Annual Assessment Report. Technical Report. NCCIC/ICS-CERT, Washington DC, USA.

Kaspersky ICS CERT, 2019. Threat Landscape for Industrial Automation Systems. H2 2018. Kaspersky. Available from https://ics-cert.kaspersky.com/publications/reports/2019/03/27/threat-landscape-for-industrial-automation-systems-h2-2018/ [Accessed on Jan. 1, 2021].

Kaspersky ICS CERT, 2020a. Targeted Attacks on Israeli Water Supply and Wastewater Treatment Facilities. Available from https://ics-cert.kaspersky.com/news/2020/04/29/israel-water-cyberattacks/ [Accessed on Jan. 1, 2021].

Kaspersky ICS CERT, 2020b. Threat Landscape for Industrial Automation Systems. Vulnerabilities Identified in 2019. Kaspersky. Available from https://ics-cert.kaspersky.com/reports/2020/04/24/threat-landscape-for-industrial-automation-systems-vulnerabilities-identified-in-2019/ [Accessed on Jan. 1, 2021].

Khraisat A, Gondal I, Vamplew P, et al., 2019. Survey of intrusion detection systems: techniques, datasets and challenges. *Cybersecurity*, 2:20. https://doi.org/10.1186/s42400-019-0038-7

Kleinmann A, Wool A, 2014. Accurate modeling of the Siemens S7 SCADA protocol for intrusion detection and digital forensics. *J Dig Forens Secur Law*, 9(2):37-50. https://doi.org/10.15394/jdfsl.2014.1169

Lee R, Slowik J, Miller B, et al., 2017. Industroyer/Crashoverride: Zero Things Cool about a Threat Group Targeting the Power Grid. Technical Report. Black Hat, USA.

Lin H, Slagell A, di Martino C, et al., 2013. Adapting Bro into SCADA: building a specification-based intrusion detection system for the DNP3 protocol. Proc 8th Annual Cyber Security and Information Intelligence Research Workshop, p.1-4. https://doi.org/10.1145/2459976.2459982

Linda O, Manic M, Vollmer T, et al., 2011a. Fuzzy logic based anomaly detection for embedded network security cyber sensor. Proc IEEE Symp on Computational Intelligence in Cyber Security, p.202-209. https://doi.org/10.1109/CICYBS.2011.5949392

Linda O, Manic M, Alves-Foss J, et al., 2011b. Towards resilient critical infrastructures: application of type-2 fuzzy logic in embedded network security cyber sensor. Proc 4th Int Symp on Resilient Control Systems, p.26-32. https://doi.org/10.1109/ISRCS.2011.6016083

Ma RK, Cheng P, Zhang ZY, et al., 2019. Stealthy attack against redundant controller architecture of industrial cyber-physical system. *IEEE Int Things J*, 6(6):9783-9793. https://doi.org/10.1109/JIOT.2019.2931349

Maglaras LA, Jiang JM, 2014. Intrusion detection in SCADA systems using machine learning techniques. Proc Science and Information Conf, p.626-631. https://doi.org/10.1109/SAI.2014.6918252

Mathur AP, Tippenhauer NO, 2016. SWaT: a water treatment testbed for research and training on ICS security. Proc Int Workshop on Cyber-Physical Systems for Smart Water Networks, p.31-36. https://doi.org/10.1109/CySWater.2016.7469060

Morris T, Vaughn R, Dandass Y, 2012. A retrofit network intrusion detection system for MODBUS RTU and ASCII industrial control systems. Proc 45th IEEE Hawaii Int Conf on System Sciences, p.2338-2345. https://doi.org/10.1109/HICSS.2012.78

Navaz ASS, Sangeetha V, Prabhadevi C, 2013. Entropy based anomaly detection system to prevent DDoS attacks in cloud. *Int J Comput Appl*, 62(15):42-47. https://doi.org/10.5120/10160-5084

Nelson T, Chaffin M, 2011. Common Cybersecurity Vulnerabilities in Industrial Control Systems. Technical Report. The U.S. Department of Homeland Security (DHS) National Cyber Security Division, Washington DC, USA.

Ponomarev S, Atkison T, 2016. Industrial control system network intrusion detection by telemetry analysis. *IEEE Trans Depend Sec Comput*, 13(2):252-260. https://doi.org/10.1109/TDSC.2015.2443793

Qian Q, Che HY, Zhang R, 2009. Entropy based method for network anomaly detection. Proc 15th IEEE Pacific Rim Int Symp on Dependable Computing, p.189-191. https://doi.org/10.1109/PRDC.2009.38

Sample C, Schaffer K, 2013. An overview of anomaly detection. *IT Prof*, 15(1):8-11. https://doi.org/10.1109/MITP.2013.7

SecurityWeek, 2016. Attackers Alter Water Treatment Systems in Utility Hack: Report. Available from https://www.securityweek.com/attackers-alter-water-treatment-systems-utility-hack-report [Accessed on Jan. 1, 2021].

Song ZW, Liu ZH, 2019. Abnormal detection method of industrial control system based on behavior model. *Comput Secur*, 84:166-178. https://doi.org/10.1016/j.cose.2019.03.009

Stouffer K, Pillitteri V, Lightman S, et al., 2011. Guide to Industrial Control Systems (ICSs) Security. NIST Special Publication 800-82. https://doi.org/10.6028/NIST.SP.800-82r2

Tate RF, 1954. Correlation between a discrete and a continuous variable. Point-biserial correlation. *Ann Math Stat*, 25(3):603-607. https://doi.org/10.1214/aoms/1177728730

Ten CW, Manimaran G, Liu CC, 2010. Cybersecurity for critical infrastructures: attack and defense modeling. *IEEE Trans Syst Man Cybern A*, 40(4):853-865. https://doi.org/10.1109/TSMCA.2010.2048028

Terai A, Abe S, Kojima S, et al., 2017. Cyber-attack detection for industrial control system monitoring with support vector machine based on communication profile. Proc IEEE European Symp on Security and Privacy Workshops, p.132-138. https://doi.org/10.1109/EuroSPW.2017.62

The Wall Street Journal's San Francisco Bureau, 2015. Iranian Hackers Infiltrated New York Dam in 2013. Available from https://www.wsj.com/articles/iranian-hackers-infiltrated-new-york-dam-in-2013-1450662559 [Accessed on Jan. 1, 2021].

Vollmer T, Manic M, 2009. Computationally efficient neural network intrusion security awareness. Proc 2nd Int Symp on Resilient Control Systems, p.25-30. https://doi.org/10.1109/ISRCS.2009.5251357

Vollmer T, Alves-Foss J, Manic M, 2011. Autonomous rule creation for intrusion detection. Proc IEEE Symp on Computational Intelligence in Cyber Security, p.1-8. https://doi.org/10.1109/CICYBS.2011.5949394

Walton B, 2016. Water Sector Prepares for Cyberattacks. Available from https://www.circleofblue.org/2016/world/water-sector-prepares-cyberattacks [Accessed on Jan. 1, 2021].

Wang YS, Fan KF, Lai YX, et al., 2017. Intrusion detection of industrial control system based on Modbus TCP protocol. Proc 13th IEEE Int Symp on Autonomous Decentralized System, p.156-162. https://doi.org/10.1109/ISADS.2017.29

Wikipedia, 2020a. Critical Infrastructure. Available from https://en.wikipedia.org/wiki/Critical_infrastructure [Accessed on Jan. 1, 2021].

Wikipedia, 2020b. Water Treatment. Available from https://en.wikipedia.org/wiki/Water_treatment [Accessed on Jan. 1, 2021].

Yu W, Wang X, Xuan D, et al., 2006. Effective detection of active worms with varying scan rate. Proc Securecomm and Workshops, p.1-10. https://doi.org/10.1109/SECCOMW.2006.359549

Zhang F, Kodituwakku HADE, Hines JW, et al., 2019. Multilayer data-driven cyber-attack detection system for industrial control systems based on network, system, and process data. *IEEE Trans Ind Inform*, 15(7):4362-4369. https://doi.org/10.1109/TII.2019.2891261

Appendix: Attack techniques and details

Table A1 Attack techniques

Attack scenario	Step	Techniques	Number
#1	Step 1	Program software vulnerabilities	(1)
		Operating system vulnerabilities	(2)
		Operating system password cracking	(3)
	Step 2	Dynamic-link library (DLL) hijacking	(4)
		Program software password cracking	(5)
		Privilege escalation	(6)
	Step 3	Protocol replay attack	(7)
		Packet forgery	(8)
		Direct manipulation	(9)
#2	Step 1	Logic manipulation attack	(10)
		System command injection attack	(11)
		Memory corruption vulnerability attack	(12)
		Firmware modification attack	(13)
		Malicious code within firmware	(14)
		Protocol replay attack	(15)
		Protocol counterfeit attack	(16)

Table A2 Attack details

Attack	Attack chain	Affected status*	Attack	Attack chain	Affected status*
#1	(2)+(6)+(8)[1]	P101[2]	#21	(3)+(5)+(8)	P204, V204
#2	(2)+(6)+(8)	P101[2]	#22	(2)+(4)+(9)	P203, P204
#3	(1)+(5)+(8)	P101	#23	(2)+(4)+(7)	V202, P204, P202
#4	(2)+(6)+(9)	P102	#24	(2)+(5)+(8)	P101, P204
#5	(3)+(6)+(9)	P202	#25	(12)	V103, V106, P201
#6	(11)	P204	#26	(3)+(4)+(9)	V104, V105, V203, P202
#7	(1)+(4)+(8)	V101, V102	#27	(1)+(6)+(8)	V101, V104, V107, PR102, V202, LV201
#8	(1)+(4)+(7)	P102, V106, V107			
#9	(15)	P101, V101	#28	(13)	P101, F201, F202
#10	(2)+(5)+(9)	V103, V108	#29	(1)+(6)+(7)	PR101, PR102
#11	(14)	V103, V104	#30	(2)+(6)+(7)	V101, V102, V201, V202, V204
#12	(10)	P102, V104, V105	#31	(1)+(6)+(9)	P102, V103, V201, P201, F202
#13	(1)+(5)+(7)	V106, V107	#32	(1)+(4)+(7)	P102, V103, LV102, V104
#14	(1)+(4)+(9)	V106, PR101	#33	(3)+(6)+(7)	P202, F202, P203, LV202, P204
#15	(3)+(6)+(8)	V107, PR102	#34	(3)+(5)+(9)	V106, V107, LV201, LV202, F201, F202
#16	(3)+(5)+(7)	V106, V107, V201			
#17	(1)+(5)+(9)	V201, P201	#35	(2)+(5)+(7)	P101, LV101, LV102, LV103
#18	(3)+(4)+(7)	V202, F201	#36	(16)	P101, P204, LV101, LV102, LV103, LV201, LV202
#19	(3)+(4)+(8)	P202, V203			
#20	(2)+(4)+(8)	P203, V202, V203, V204			

[1] The attack chain consists of the attacks in Table A1. [2] This table marks only the key device states drawn in Fig. 2, and some device states are not marked. The difference between the first two attacks is not marked. *P: pump; V: valve; PR: pressure; F: flow; LV: liquid level