Li *et al. EURASIP Journal on Wireless Communications and Networking* (2020) 2020:103
https://doi.org/10.1186/s13638-020-01718-0

EURASIP Journal on Wireless Communications and Networking

RESEARCH Open Access

An improved content-based outlier detection method for ICS intrusion detection

Huiping Li[1], Bin Wang[2*] and Xin Xie[2]

*Correspondence:
wangbin199702@163.com
[2]East China Jiaotong University, Nanchang, 330013, China
Full list of author information is available at the end of the article

Abstract

Due to the complexity of industrial control systems and the diversity of protocols in networks, it is difficult to build intrusion detection models based on network characteristics and physical modeling. In order to build a better flow model without additional knowledge, we propose an intrusion detection method based on the content of network packets. The construction of the model is based on the idea of ZOE method. The similarity between flows is calculated through the sequential coverage algorithm, the normal flow model is established by multi-layered clustering algorithm, and the Count-Mean-Min Sketch is used to store and count the flow model. By comparing the unknown flow with the constructed normal flow model, we achieve the intrusion detection of industrial control system (ICS). The overall experimental results on 4 ICS datasets show that the improved method can effectively improve the detection rate and reduce the false-positive rate. The detection rate reached 96.7% on average, and the false-positive rate reached 0.7% on average.

Keywords: Industrial control system, Intrusion detection, Sequence covering, Multi-layered cluster, Count-Mean-Min Sketch

1 Introduction

In recent years, the wave of informatization technology has swept the world, and information communication and manufacturing technology are integrated. At present, industrial control systems (ICSs) are used not only in the field of industrial production, but also in the field of infrastructure [1]. With the integration and complementarity of informatization and industrialization, ICS gradually adopts standardized network protocols and applications connected with the Internet. Emerging technologies, such as Ethernet, embedded system, and wireless network, replace the traditional bus technology and therefore expand the space for the development of ICS [2]; however, it also brings new challenges for its intrusion detection and fault diagnosis [3, 4]. In these years, some countries have suffered from damage to their ICS, including the far-reaching outbreak of the Stuxnet virus in Iran [5]; the attack on Ukraine's power grid in 2015 [6], which threw millions of people in total darkness during the Christmas season; and invasions of the power system in Israel since 2016. Therefore, intrusion detection technology of ICS has become the focus of researchers in the field [7].

Due to resource constraints and system isolation, network security was not fully considered in the initial design of ICSs [8]. With the development of modern information system and information technology, potential security problems are gradually exposed. To ensure the stability and reliability of industrial processes, integrated circuits need to be protected. Traditional information system solutions have been rapidly applied in the field of national defense. However, these methods cannot fully detect network physical attacks under real-time and resource-constrained conditions [9]. In recent years, ICS anomaly detection and security protection have been widely studied in the world. However, the widespread use of private binary protocol in industrial networks makes content-based intrusion detection methods very complex and often makes existing methods invalid. The focus of this study is to bridge this gap and improve the performance of industrial anomaly detection methods.

The contributions of this paper are as follows:

a) A content-based anomaly detection method for ICS that the performance of ICS is improved;

b) A method to construct prototype models for network messages of ICS which constructs different types of traffic models through the original content of network data, without sufficient protocol knowledge or deeper understanding of the physical system.

2 Related works

Intrusion detection [10] is an active protection technology that detects illegal operations by analyzing the network layer data of the system. Industrial control intrusion detection systems can be divided into three types: content-based attack detection, network feature-based detection, and physical process-based detection [11].

Through analyzing protocol and traffic content, content-based attack detection constructs normal protocol or traffic model to tell the existence of attack traffic in the system from comparison between the normal traffic model and the unknown traffic. Generally, intrusion detection [12, 13] is realized through reverse engineering, identification [14], analysis [15, 16], and shellcode [17, 18]. However, the above methods are usually based on some specific types of protocol traffic, and the detection accuracy of multi-type attacks is low. Because most network intrusions require specific code, such as buffer overflow attacks, it exhibits typical sequential patterns in network traffic. To this end, Rieck and Laskov [19] studied and discussed sequence characteristics and their efficient implementation, and used bag of token technology and q-gram method to deal with the application layer load in network traffic, thus achieving higher detection accuracy without the need to learn or mark the data in advance. Liao and Vemuri [20] first regarded symbol strings in network traffic as words in documents and applied the text classification method based on vector space model to network intrusion detection. Mahoney and Chan [21] and Ingham and Inoue [22] also used text classification method to study the intrusion detection problem. Kruegel and Vigna [23] introduced the q-gram model into intrusion detection to effectively express the closely related characters in the sequence. Lin et al. [24] proposed a novel character-level intrusion detection system based on the convolutional neural network, which treated network traffic records as character sequences and inputted each character into the convolutional neural network as a vector based on alphabetic encoding. This model has a good effect in binary classification and multi-classification. Recently, Wressnegger et al. [25] proposed a method called ZOE to detect

intrusion, which effectively utilized the content-based anomaly detection framework and improved the accuracy of anomaly detection.

The attack detection method based on network features is mainly to extract network traffic features through machine learning [26, 27], so as to build the characteristic model of normal traffic [28]. Vavra and Hromada [29] integrated a variety of classifiers to improve the feature learning and selection methods of classifiers, thus improving ICS intrusion detection capability. In order to filter redundant or useless data in network traffic packets, Su et al. [30] proposed a feature selection method based on automatic learning machine, which can select more important features for network traffic intrusion detection. Imtiaz Ullah et al. [31] proposed a hybrid model for intrusion detection applied to SCADA (Supervisory Control And Data Acquisition) networks based on feature selection and machine learning, which effectively improved the detection accuracy.

Based on modeling of the basic physical process, the attack detection method that is based on physical process is capable to predict the data and tell the intrusion after comparing the predicted data with the actual one [32]. Wang et al. [33] proposed a method based on extended distribution state estimation to detect new types of erroneous data injection attacks in smart grids. Kurt et al. [34] modeled the smart grid system as a discrete time linear dynamic system, estimated the system state by using Kalman's filter, and realized the rapid detection of network attacks through accumulation and algorithm.

However, if there is no effective protocol specification to assist network data preprocessing, it is difficult to deeply analyze in-depth network packet contents. On the other hand, it is difficult to build a physical process-based intrusion detection system, which requires a full understanding of the system flow. Moreover, it is also not ideal for detecting multiple types of attacks. Therefore, this paper proposes a content-based improved ZOE intrusion detection method, which constructs different types of traffic models through the original content of network data, without sufficient protocol knowledge or deeper understanding of the physical system. In addition, by integrating the sequence coverage similarity algorithm [35] with the multi-layered clustering algorithm [36], the accuracy of model construction and intrusion detection in ICS is improved. The experimental results show that the model can effectively improve the intrusion detection effect in ICS.

3 Sequence coverage similarity algorithm

Initializing a dictionary A, A^* is defined as a set of sequences of all elements on A. For any subset $B \subseteq A^*$, B_{sub} is extracted through $B \cup A$, and the multisets $M(B_{sub})$ of set B_{sub} can be obtained.

For an element e in $M(B_{sub})$ and a sequence s in A^*, e can be called partial coverage of s if and only if the following two conditions are met:

All subsequences in e are subsequences in s.

Indistinguishable copies of a particular element in e correspond to distinct occurrences of the same subsequence in s.

If e can completely cover s, it means that an optimal set $e_B^*(s)$ can be found from the permutation and combination of elements in e, which means that s can be covered by the least number of elements.

The coverage similarity can be defined through the above steps:

$$\varphi(s, B) = \frac{|s| - \left|e_B^*(s)\right| + 1}{|s|} \tag{1}$$

For any pair of sequences s_1 and s_2, the similarity between them is:

$$\varphi_{seq}(s_1, s_2) = \frac{1}{2}\left(\varphi(s_1, \{s_2\}) + \varphi(\{s_1\}, s_2)\right) \tag{2}$$

The greater the value of φ_{seq}, the more similar s_1 and s_2 are, and vice versa.

4 The ZOE method

The ZOE method is a content-based anomaly detection method, which uses the n-gram method in natural language processing [37] to map flows to the corresponding feature vector, which has a good interpretability for the content of flows. K-means clustering [38] is used to construct protocol models and flow models of different categories without supervision, and the comparison between the established model and the current input flow could be used to determine whether there is an anomaly.

A flow set D containing only normal flows is used to construct the normal flow model. For any flow d_i in D, use n-gram to extract all substrings $subs_l^i$ of length n in d_i. Do this for each flow in D to obtain the substring set S composed of all substrings in D. Then, d_i can be mapped to the corresponding eigenvector by S:

$$\phi : d_i \to \left(\phi_{subs_l^i}(d_i)\right)_{subs_l^i \in S} \tag{3}$$

$$\phi_{subs_l^i}(d_i) = occ\left(subs_l^i, d_i\right) \tag{4}$$

where the function of occ is to calculate the frequency of the lth substring $subs_l^i$ in d_i. The value of each dimension in the feature vector is the frequency of each substring in S. For $D = \{d_1, d_2, \cdots, d_n\}$ can use this way into the corresponding feature vector set $V = \{v_1, v_2, \cdots, v_n\}$, where $v_i = \phi(d_i)$.

The flow set D is transformed into the corresponding eigenvector set V, and then, different types of normal flow models are constructed by clustering the eigenvector in V. Input the number of cluster classes k and initialize k cluster classes $C_1, C_2, \cdots, C_k$. The eigenvector v_l is classified as the corresponding cluster class j by the construction function $j = \arg\max_{i \in [1,k]} prox(v_i, C_i)$.

$$prox : v, C \to \frac{1}{|C|}\sum\nolimits_{w \in C} sim(v, w) \tag{5}$$

$$sim : v, w \to \frac{v \cdot w}{\|v\|_2 \|w\|_2} = \cos(\theta) \tag{6}$$

where w is the eigenvector of cluster class C. Each cluster class can be regarded as the flow model of the corresponding type.

Since the dimension of eigenvectors obtained from n-gram processing is relatively large, the substrings in each cluster class are pruned after the completion of clustering. Set the threshold value t. If the frequency $P_{m,subs_l^i}$ of a substring $subs_l^i$ is less than t for all flows in cluster C_m, the substring can be eliminated to achieve the dimension reduction and denoising, so as to optimize the constructed flow model.

After completing the construction of the flow models, we can judge whether any flow is abnormal or not. Set the abnormal threshold T, and for any flow m with unknown properties, the outlier score of each flow model is calculated:

$$\text{score} : m, C \rightarrow \min_i d\,(m, C_i) \tag{7}$$

$$d : m, C \rightarrow 1 - \frac{1}{p} \sum\nolimits_{subs \in C} occ\,(subs, m) \tag{8}$$

where p is the number of substrings extracted by n-gram method for flow m. The above formula expressed the inconsistent degree of m with flow model C, also is the probability of $m \notin C$. If score $(m, C) \geq T$, flow m is abnormal.

5 ICS intrusion detection based on the improved ZOE method

The framework of the ZOE method makes use of content-based anomaly detection for proprietary binary protocols, because the content models are very well usable for environments that rely on undocumented protocols with high-entropy data. On the other hand, the framework of the ZOE method introduces the concept of the prototype models which characterize not only the structure of message types but also the data they typically contain. To this end, we introduce the framework of the ZOE method and the concept of prototype models in it.

The original ZOE method used n-gram method to extract substrings and took the frequency of each substring as the corresponding value of each dimension in the feature vector. However, n-gram essentially divides the flow into isolated units to be processed, and this processing mode corresponds to discrete one-hot vectors in mathematical form, which cannot consider the internal connection. Moreover, it is also very critical for the value of n. When n increases, more constraint information will appear on the next symbol, with greater discrimination. When n decreases, more substrings will appear, with more reliable statistical information. On the other hand, the original ZOE method adopted k-means to cluster the feature vectors. Since k-means is a supervised learning algorithm, some prior knowledge is required to set a value for k that can achieve better results. But in practical applications, it is usually impossible to directly determine the number of flow models to be built, and a proper k value can only be determined through continuous trials and experiments. Moreover, using the original k-means cannot make the classifier of the training diversified, thus leading to the data difficult to identify and classify.

The multi-layered cluster can build a machine learning model that learns from non-labeled or partially labeled data. So, it has the capability to learn from partially labeled data while achieving a detection performance comparable to that of supervised machine learning-based intrusion detection and prevention system. Therefore, this paper applies the sequential overlay similarity algorithm to the similarity calculation of flow. The similarity between flows is calculated based on the original flow, the contents in the original flow are fully considered, and the multi-layered clustering algorithm combined with the sequence coverage similarity algorithm is used to cluster the flow, so as to construct the flow model based on the original flow. Intrusion detection is carried out by the flow models.

A flow set D containing only normal flow is used to construct the normal flow model, and the flow in D is multi-layer clustered. The flows in D are divided into marked data and unmarked data. Each piece of marked data has a class label indicating what type of

flow it is (e.g., TCP, UDP, binary, text), while unmarked data does not have any class labels. Labeled data is denoted as $D_{labeled}$, unlabeled data is denoted as $D_{\text{unlabeled}}$, and flow set $D = \left\{D_{\text{labeled}}, D_{\text{unlabeled}}\right\}$. Specifically,

$D_{\text{labeled}} = \left\{(d_1, y_1), (d_2, y_2), \cdots, (d_n, y_n)\right\}$ where n is the number of labeled data D_{labeled} and y is the corresponding label

$D_{\text{unlabeled}} = \{(d_{n+1}), (d_{n+2}), \cdots, (d_N)\}$ where N is the amount of data in the dataset D. Clusters are generated at different k values on different layers using different sets of initialization parameters. If there is an L layer, the cluster generated on this L layer can be represented as:

$\left\{C_{1,1}, \cdots, C_{1,k_1}\right\}, \cdots, \left\{C_{L,1}, \cdots, C_{L,k_L}\right\}$

which contains three types of clusters, namely the fully labeled cluster, the partially labeled cluster, and the unlabeled cluster. The multi-layer cluster then identifies the three types of clusters and builds a learning model on each cluster. The learning model built on each layer can be regarded as a different basic classifier, which can be utilized to build an integrated model covering the whole decision space. The final label of each flow is determined by the corresponding classifier with the most votes on different layers.

Take the lth layer for example: First, a dictionary A is initialized with the flow set D, which contains all the numbers or letters that appeared in D. A^* is the set of substrings of length n composed of all elements on A. Any flow d_i in flow set D is divided into substrings $subd^i = \left\{subd_1^i, \cdots, subd_l^i\right\}$ of length n. The similarity degree $dist\,(d_i, d_n)$ between $subd^i$ and other flow substring set $subd^n$ is calculated by the sequence coverage similarity algorithm.

$$dist\,(d_i, d_n) = \theta\left(subd^i, subd^n\right) \tag{9}$$

$$\theta\left(subd^i, subd^n\right) = \frac{1}{l}\sum\nolimits_{k=1}^{l} \varphi_{seq}\left(subd_k^i, subd_k^n\right) \tag{10}$$

By constructing the function $j = \arg\max\limits_{i\in[1,k]}^{*} prox\,(d_i, C_i)$, the flow d_i is classified as the corresponding cluster class j in the lth layer.

$$prox^* : d_i, C \to \frac{1}{|C|}\sum\nolimits_{d_n \in C} dist\,(d_i, d_n) \tag{11}$$

where d_n is the flow in cluster class C. Thus, the clustering of the lth layer is completed. After the above operations, the clustering result:

$\left\{C_{1,1}, \cdots, C_{1,k_1}\right\}, \cdots, \left\{C_{L,1}, \cdots, C_{L,k_L}\right\}$

can be obtained from each layer (Fig. 1). By selecting category $\left\{C_{i,1}, \cdots, C_{i,k_L}\right\}$ which has the highest number of votes among all the decisions at all layers, as the final classification of the flow. In this way, the construction of the flow models is completed, and the nature of any unknown type of flow can be determined accordingly.

In order to facilitate the subsequent judgment of the property of any flows, a data structure is needed to store the flows in the models constructed above and count the flow in each model. Since the estimated result of Count-Min Sketch [39] is always not less than the actual value, and noise may be generated in the process of querying the flow, Count-Mean-Min Sketch [40] can be used for calculation. The use of Count-Mean-Min Sketch is more extensive. It reduces the collision probability during the flow storage and filters some noises. The diagram of Count-Mean-Min Sketch is shown in Fig. 2.

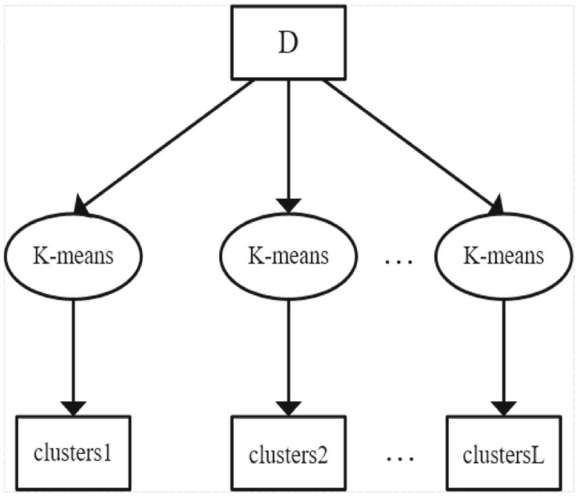

Fig. 1 Stepwise procedure of SMLC. A flow set *D* containing only normal flow is used to construct the normal flow model, and the flow in *D* is multi-layer clustered

value is the flow, d is the depth, and w is the width. Each flow is processed by d *hash* functions h_i to determine location $p_i = h_i\,(value)$, $i \in [1, d]$, $p_i \in [0, w-1]$. For any flow m, apply hash mapping and apply the result value to the position in the corresponding row. Using the hash function to determine the value associated with the flow m. The minimum of these values is approximate to the truth value, that is, the approximate value of $|C|$.

Set the abnormal threshold T, calculate the outliers of any flow m for each flow model, and judge whether the flow is abnormal by the outliers.

$$\text{score}^* : m, C \rightarrow \min_i d^*\,(m, C_i) \tag{12}$$

$$d^* : m, C \rightarrow 1 - prox^*\,(m, C_i) \tag{13}$$

where $prox^*\,(m, C_i)$ represents the similarity between flow m and flow model C_i. If the maximum value of the similarity between the flow and all models is less than T, it can be

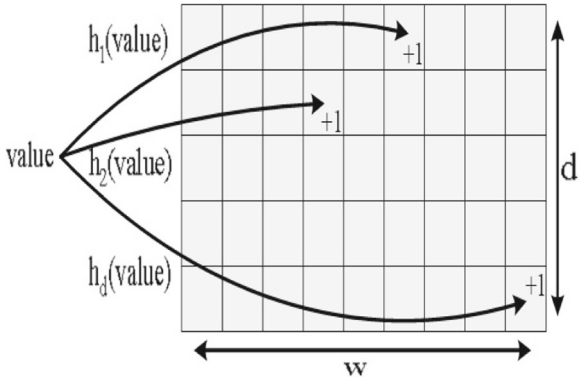

Fig. 2 Ideal step edge model. *value* is the flow, d is the depth, and w is the width. Each flow is processed by d *hash* functions h_i to determine location $p_i = h_i\,(value)$, $i \in [1, d]$, $p_i \in [0, w-1]$

Algorithm 1 Improved ZOE Method

Input: normal flows D; random set K;

Output: flows' models C; normal or abnormal;

Initialize parameters in SMLC and Count-Mean-Min Sketch;
for the layers of SMLC **do**
  for the flows in D **do**
    $j = \arg\max_{i \in [1,k]} prox^*(d_i, C_i)$;
    Add d_i to C_j;
  end for
end for
Choose the $\{C_{i,1}, \cdots, C_{i,k_L}\}$ with the highest number of votes among all decisions at all layers, $\{C_{i,1}, \cdots, C_{i,k_L}\}$ are the final flows' models C;
Use Count-Mean-Min Sketch to store C;
For the unknown flow m, calculate the $score^*(m, C)$;
if $score^* \geq T$ **then**
  return abnormal, C;
else
  return normal, C;
end if

judged as abnormal flow; otherwise, it is normal flow. When the opposite value is taken for $prox^*(m, C_i)$, it becomes score$^* \geq T$, and the flow is abnormal.

6 Analysis of experimental simulation results

6.1 The dataset

In order to verify the improved ZOE method, this paper uses the industrial control intrusion detection standard dataset [41] established by Mississippi State University (MSU) in 2014 and three public datasets [42, 43] to test and verify the method. These datasets contain the original record of the process parameter values and associated labels that indicate whether the flow is normal or abnormal.

The dataset established by MSU is derived from the network layer data of the natural gas pipeline control system. The researchers used 28 types of attacks to break into the industrial control system while using a network data recorder to monitor and store data collected the Modbus flow from RS-232. Each piece of data in the dataset is a sequence record of 27 dimensions, the first 26 dimensions represent 26 different eigenvalues, and the last one dimension represents 1 classification label, of which 26 characteristics are shown in Table 1. Classification labels represent different forms of attack, as shown in Table 2. All data has been numerically processed and can be divided into four categories of attack data: reconnaissance attack, command injection attack (MSCI, MPCI, and MFCI), denial of service attack, and response injection attack (NMRI and CMRI). Two of the three public datasets use single-hop and multi-hop topologies. Both were collected from an outdoor real wireless sensor network, for a duration of 6 h, and contain two process parameters (temperature and humidity). Each dataset has a tiny partition that marks the danger state. These datasets are called SORD (Single-hop Outdoor Real-time Data) and MORD (Multi-hop Outdoor Real-time Data). Each dataset is further divided

Table 1 The contents of the MSU dataset

Modbus	Payload	Other
Command address	Comm fun	Set point
Response address	Response fun	Control scheme
Command memory	Sub function	Solenoid state
Response memory	Measurement	Gain
Command memory count	Control mode	Reset
Response memory count	Pump State	Dead band
Command length	Manual pump setting	Rate
Response length	Label	Cycle time
Time	–	–
Crc rate	–	–

into two parts: the part containing abnormal flow and the part with normal flow. The fourth dataset is collected from the flow by the Urban Waste Water Treatment Plant sensor (DUWWTP, Data of Urban Waste Water Treatment Plant), which is composed of 38 process parameters.

6.2 The evaluation criteria

In this paper, the application effect of the method is evaluated by the detection rate and false-positive rate. The detection rate is the ratio of the number of abnormal flows correctly identified in the dataset to the total number of abnormal flows.

$$DR\text{ate} = \frac{TP}{TP + FN} \tag{14}$$

The false-positive rate represents the ratio of the number of normal flows marked as abnormal flows to all normal flows.

$$FR\text{ate} = \frac{FP}{FP + TN} \tag{15}$$

where TP (true positive) refers to the number of abnormal flows that have been correctly detected, FN (false negative) refers to the number of abnormal flows that have occurred but has not yet been detected, FP (false positive) refers to the number of normal flows that have been incorrectly marked as abnormal, and TN (true negative) refers to the number of normal flows that have been correctly identified.

In addition, in order to reflect the sensitivity of the method to abnormal flow, this paper also uses the hazard score [44] to compare the ZOE method with the improved ZOE method, which is defined as follows: the hazard score of a data point can be regarded as the distance from the hazard data point. The smaller the distance, the higher the hazard

Table 2 Meanings represented by classification labels in the MSU dataset

Label	Label value	Label description
Normal	0	Normal
NMRI	1	Simple malicious response injection attacks
CMRI	2	Sophisticated malicious response injection attacks
MSCI	3	Malicious state commands inject attacks
MPCI	4	Malicious parameter command injection attack
MFCI	5	Malicious function command injection attack
Dos	6	Denial of service attack
Reconnaissance	7	Reconnaissance attacks

score. Conversely, the greater the distance, the less dangerous it is, the less sensitive it is to the abnormal flow, and the slower it responds to the abnormal. It is calculated by the following formula:

$$\text{precision} = \frac{p}{n} \tag{16}$$

where n is the number of dangerous data points in the dataset, and p is the number of dangerous data points in the first n data points of the hazard score.

6.3 Experimental results and analysis

6.3.1 Experiment 1: Compare the counting effect with different data structures

The Count-Mean-Min Sketch data structure is adopted to store and count the flow, and $d = 7$ hash functions are used to map the flow. The same setting applies to the Count-Min Sketch data structure. The approximate count and the difference between the actual values obtained from the two data structures were statistically analyzed. The experimental results are shown in Figs. 3 and 4, respectively.

The horizontal axis represents the difference between the approximate count and the actual value, and the vertical axis represents the relative frequency of each difference. It can be seen that when the Count-Mean-Min Sketch is used, the relative frequency of the approximate value equal to the actual value increases, which shows that when Count-Mean-Min Sketch is used to count the flow, the probability of its value is equal to the actual value increases, thus improving the accuracy of similarity calculation. In addition, compared with the difference distribution range of Count-Min Sketch, the difference value of Count-Mean-Min Sketch is relatively concentrated in a smaller range, which indicates that in the case of Count-Mean-Min Sketch, even if errors still occur, the error range is narrowed and the overall accuracy is improved compared with Count-Min Sketch.

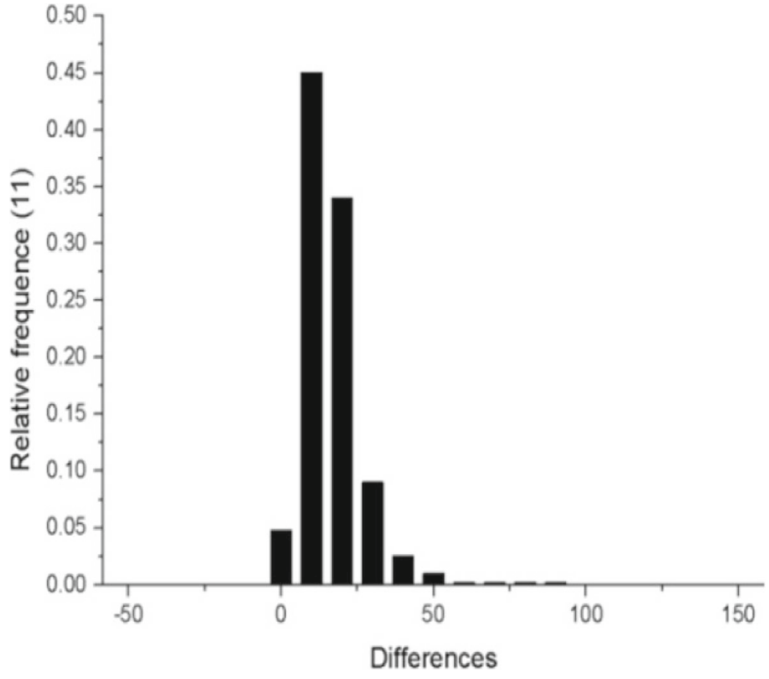

Fig. 3 The difference between exact counting and probabilistic counting of Count-Min Sketch

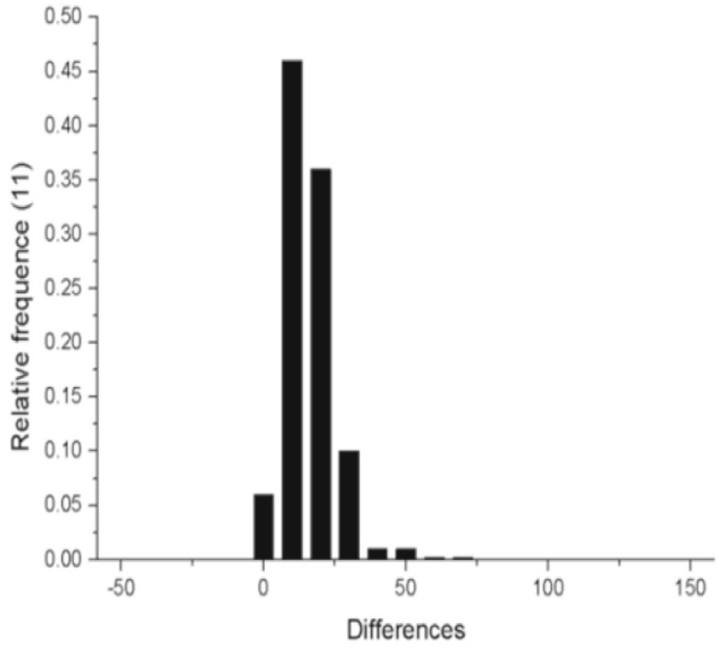

Fig. 4 The difference between exact counting and probabilistic counting of Count-Mean-Min Sketch

6.3.2 Experiment 2: Compare the detection effect of abnormal flow

The improved ZOE method and the original ZOE method in this paper are applied to the same four datasets. In addition, we compare our method with two related methods proposed in recent years. These two methods are data-driven clustering [44] and improved K-means clustering [45], respectively. The results shown in Table 3 are obtained in terms of detection rate and false-positive rate.

In MSU datasets with more diverse data types, compared with the original method, the improved ZOE method improved the detection rate of abnormal data without increasing the false-positive rate, in which the detection rate was close to 0.96. The detection rate on the MORD dataset increased by 0.02, with the most significant improvement. When applied to the SORD dataset and DUWWTP dataset, although the improvement effect of detection rate is less obvious, the false-positive rate of abnormal flow detection in the dataset is reduced.

Therefore, when the improved ZOE method is applied to the above four datasets, it not only improves the detection rate, but also reduces the false-positive rate. While in some datasets the detection rate does not increase significantly, the false-positive rate is reduced.

6.3.3 Experiment 3: Compare the hazard score

Calculate the hazard score of abnormal flow when the improved ZOE method and the original ZOE method are applied to the same four datasets above. The number of the first n hazard data points is calculated respectively, and the formula is applied to evaluate the results to obtain the comparison result of the hazard score, as shown in Table 4.

During the experiment, the ZOE method and improved ZOE method are used to calculate the hazard score for each data point in the dataset. The improved method increased

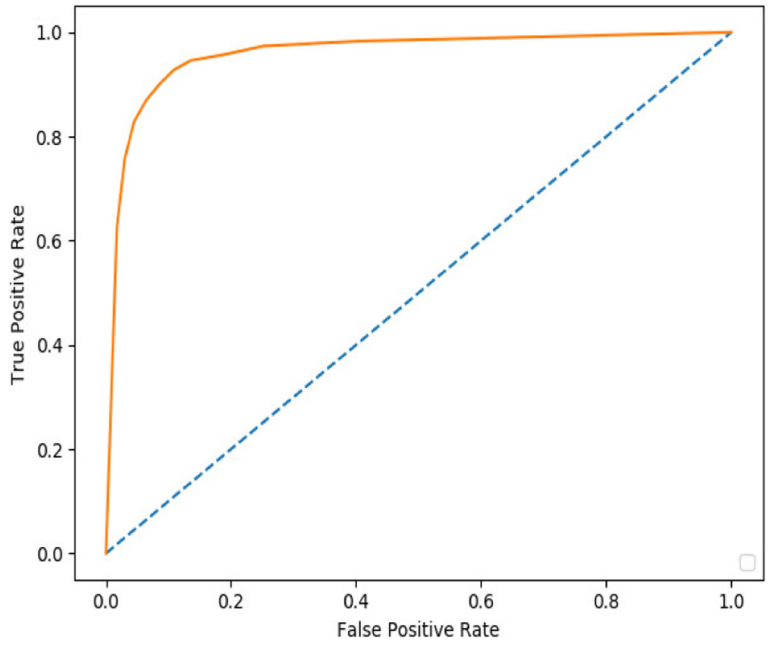

Fig. 5 The ROC curve of improved ZOE method

the hazard score on all four datasets. It reflects that the improved method is more sensitive to the abnormal flow and can respond to the abnormal more quickly. Among them, the sensitivity of DUWWTP dataset improved most obviously, while the sensitivity of other datasets also improved by 0.01 to 0.02.

Therefore, compared with the original ZOE method, the improved ZOE method improves the sensitivity to abnormal flow and enhances the response speed to abnormal flow when the exception occurs.

6.3.4 Experiment 4: The ROC curve of improved ZOE method

In the course of this evaluation, we describe detection performances with the aid of the receiver operator characteristics (ROC) and corresponding ROC curves. These curves plot the true-positive rate over the false-positive rate of a detector for different thresholds.

The Fig. 5 tells that the closer the ROC curve is to the upper left corner corresponds to fewer classification errors. In other words, the larger the area under ROC curve (AUC) equals a better classification effect. We can see that the method proposed in this paper quickly approaches the upper left corner. Additionally, we use the AUC as a single continuous measure for the detection performance that yields a minimal and maximal value of 0.0 and 1.0, respectively. And the AUC value of the improved ZOE method can reach 0.98.

Table 3 Comparison of detection results

Dataset	ZOE method		Improved ZOE method		Data-driven clustering		Improved *K*-means clustering	
	DRate	FRate	DRate	FRate	DRate	FRate	DRate	FRate
SORD	0.968	0.019	0.974	0.009	0.967	0.019	0.971	0.010
DUWWTP	0.985	0.020	0.988	0.008	0.980	0.203	0.982	0.109
MORD	0.929	0.006	0.948	0.003	0.928	0.007	0.932	0.004
MSU	0.944	0.010	0.959	0.009	0.938	0.013	0.943	0.012

Table 4 Comparison of detection results

Dataset	ZOE method precision	Improved ZOE method precision	Data-driven clustering precision	Improved *K*-means clustering precision
SORD	0.967	0.983	0.968	0.976
DUWWTP	0.923	0.945	0.921	0.941
MORD	0.952	0.973	0.935	0.949
MSU	0.962	0.977	0.934	0.951

7 Conclusion

This paper introduces the ZOE method and analyzes the defects and shortcomings of the method. For example, it is difficult to select parameters by using *n*-gram method and the accuracy of eigenvector clustering by using original *k*-means algorithm is insufficient. Then, sequential coverage similarity algorithm is proposed to calculate the similarity between any two flows in the industrial control system. The SMLC model is used to construct the normal flow model, and the sequence coverage similarity is used as the measure of the distance between the clusters. Based on the original ZOE method, an improved intrusion flow detection model of industrial control system is proposed by combining SMLC hierarchical clustering with sequence coverage similarity. The improved model is used to construct a higher quality normal flow model, which improves the accuracy of abnormal flow detection and reduces the false-positive rate. And the improved model also improves the sensitivity to the abnormal flow in the ICS and increases the response speed to the abnormal flow when it occurs. Moreover, the Count-Mean-Min Sketch is used to store the normal flow models. Compared with the difference distribution range of Count-Min Sketch, the difference of Count-Mean-Min Sketch is relatively concentrated in a smaller range. The probability of the value equal to the actual value increases, which improves the accuracy of the model for intrusion detection in industrial control systems.

Abbreviations

ICS: Industrial control system; SCADA: Supervisory Control and Data Acquisition; SMLC: Semi-supervised multi-layer cluster; MSU: Mississippi State University; MSCI: Malicious state command injection; MPCI: Malicious parameter command injection; MFCI: Malicious function code injection; NMRI: Naive malicious response injection; CMRI: Complex malicious response injection; SORD: Single-hop Outdoor Real-time Data; MORD: Multi-hop Outdoor Real-time Data; DUWWTP: Data of Urban Waste Water Treatment Plant; TP: True positive; FN: False negative; FP: False positive; TN: True negative; ROC: Receiver operating characteristics

Acknowledgements

The authors acknowledged the anonymous reviewers and editors for their efforts in valuable comments and suggestions.

Authors' contributions

H.L. and B.W. conceived and designed the experiments. H.L. and B.W. made the graphs and tables. H.L. and X.X. collected and analyzed the data. X.X. searched the related articles. H.L. and B.W. wrote the paper. The authors read and approved the final manuscript.

Funding

This work is supported by the National Natural Science Foundation of China, under grant no. 61762037, and Applied Innovation Program of Ministry of Public Security, under grant no. 2019YYCXHNST002.

Competing interests

The authors declare that they have no competing interests.

Author details

[1] School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu, 611731, China. [2] East China Jiaotong University, Nanchang, 330013, China.

Received: 7 February 2020 Accepted: 23 April 2020
Published online: 18 May 2020

References

1. D. Serpanos, Secure and resilient industrial control systems. IEEE Des. Test. **35**(1), 90–94 (2018)
2. J.-P. Auffret, J. L. Snowdon, A. Stavrou, J. S. Katz, D. Kelley, R. S. Rahman, F. Stein, L. Sokol, P. Allor, P. Warweg, Cybersecurity leadership: competencies, governance and technologies for industrial control systems. J. Interconnection Netw. **17**(1) (2017)
3. Y. Hu, H. Li, H. Yang, Y. Sun, L. Sun, Z. Wang, Detecting stealthy attacks against industrial control systems based on residual skewness analysis. EURASIP J. Wirel. Commun. Netw. **1**(1) (2019)
4. K. Paridari, N. O'Mahony, A. E.-D. Mady, R. Chabukswar, M. Boubekeur, H. Sandberg, A framework for at-tack-resilient industrial control systems: attack detection and controller reconfiguration. Proc. IEEE. **106**(1), 113–128 (2018)
5. M. Chung, W. Ahn, B. Min, J. Seo, J. Moon, An analytical method for developing appropriate protection profile of instrumentation & control system for nuclear power plants. J. Supercomput. **74**(3), 1378–1393 (2018)
6. M. Wu, Z. Song, Y. B. Moon, Detecting cyber-physical attacks in cyber manufacturing systems with machine learning methods. J. Intell. Manuf. **30**(3), 1–13 (2019)
7. G. S. Yilmaz E N, Attack detection prevention system against cyberattack in industrial control systems. Comput. Secur. **77**, 94–105 (2018)
8. R. Chabukswar, Y. Mo, B. Sinopoli, Detecting integrity attacks on scada systems. IEEE Trans. Control Syst. Technol. **22**, 1396–1407 (2014)
9. A. Almalawi, X. Yu, Z. Tari, A. Fahad, I. Khalil, An unsupervised anomaly-based detection approach for integrity attacks on scada systems. Comput. Secur. **46**, 94–110 (2014)
10. L. A. I. C. X.-t. Y. K.-x. Ying-xu, L. I. U. Zeng-hui, Research on intrusion detection of industrial control system. J. Commun. **38**(2), 143–156 (2017)
11. R. Mitchell, I. R. Chen, A survey of intrusion detection techniques for cyber physical systems. ACM Comput. Surv. **46**(4), 55–84 (2014)
12. P. M. Comparetti, G. Wondracek, C. Kruegel, E. Kirda, Prospex: protocol specification extraction (2009). https://doi.org/10.1109/sp.2009.14
13. Z. Lin, X. Jiang, D. Xu, X. Zhang, in *15Th Symposium on Network and Distributed System Security, 46*, Automatic Protocol Format Reverse Engineering through Context-Aware Monitored Execution, (2008). https://doi.org/10.1.1.120.2651
14. H. Dreger, A. Feldmann, M. Mai, V. Paxson, R. Sommer, in *15th conference on USENIX Security Symposium, 15*, Dynamic application-layer protocol analysis for network intrusion detection, (2008), pp. 257–272
15. R. Pang, V. Paxson, R. Sommer, L. Peterson, binpac: a yacc for writing application protocol parsers (2006). https://doi.org/10.1145/1177080.1177119
16. G. Wondracek, P. M. Comparetti, C. Kruegel, E. Kirda, S. S. S. Anna, Automatic Network Protocol Analysis. Network and Distributed System Security Symposium (NDSS), 1–18. https://doi.org/10.1.1.110.7553
17. M. Polychronakis, K. G. Anagnostakis, E. P. Markatos, Comprehensive shellcode detection using runtime heuristics (2010). https://doi.org/10.1145/1920261.1920305
18. K. Z. Snow, S. Krishnan, F. Monrose, N. Provos, in *Proceedings of the 20th USENIX Security Symposium*, SHELLOS: Enabling fast detection and forensic analysis of code injection attacks, (2011), pp. 123–138
19. L. P. Rieck K, Language models for detection of unknown attacks in network traffic. J. Comput. Virol. **2**(4), 243–256 (2007)
20. Y. Liao, V. Rao Vemuri, in *Proceedings of the 11th USENIX Security Symposium*, Using text categorization techniques for intrusion detection, (2002), pp. 51–59
21. M. V. Mahoney, P. K. Chan, Learning rules for anomaly detection of hostile network traffic (2003). https://doi.org/10.1109/icdm.2003.1250987
22. K. L. Ingham, H. Inoue, Comparing anomaly detection techniques for http. Recent Adv. Intrusion Detect., 42–62 (2007). https://doi.org/10.1007/978-3-540-74320-0_3
23. C. Kruegel, G. Vigna, Anomaly detection of web-based attacks, 251–61 (2003). https://doi.org/10.1145/948109.948144
24. S. Z. Lin, Y. Shi, Z. Xue, Character-level intrusion detection based on convolutional neural networks (2018). https://doi.org/10.1109/ijcnn.2018.8488987
25. C. Wressnegger, A. Kellner, K. Rieck, Zoe: content-based anomaly detection for industrial control systems (2018). https://doi.org/10.1109/dsn.2018.00025
26. N. Jiang, F. Tian, J. Li, X. Yuan, J. Zheng, Man: mutual attention neural networks model for aspect-level sentiment classification in siot. IEEE Internet Things J. https://doi.org/10.1109/jiot.2020.2963927
27. N. Jiang, D. Xu, J. Zhou, H. Yan, T. Wan, J. Zheng, Toward optimal participant decisions with voting-based incentive model for crowd sensing. Inf. Sci. **512**, 1–17 (2020). https://doi.org/10.1016/j.ins.2019.09.068
28. B. Y. SUN Ziwen, LIANG Guangwei, A hierarchical intrusion detection model in wireless sensor networks. Inf. Control. **42**(6), 670–676 (2013)
29. J. Vavra, M. Hromada, Anomaly detection system based on classifier fusion in ics environment (2018). https://doi.org/10.1109/icsiit.2017.35
30. Y. Su, K. Qi, C. Di, Y. Ma, S. Li, Learning automata based feature selection for network traffic intrusion detection (2018). https://doi.org/10.1109/dsc.2018.00099
31. L. Yingxu, J. Jiao, L. Jing, Analysis of industrial control systems traffic based on time series (2015). https://doi.org/10.1109/isads.2015.28
32. T. Morris, R. Vaughn, Y. Dandass, A retrofit network intrusion detection system for modbus rtu and ascii industrial control systems, 2338–2345 (2012). https://doi.org/10.1109/hicss.2012.78
33. Wang D.G.T., Guan X, Extended distributed state estimation: a detection method against tolerable false data injection attacks in smart grids. Energies. **7**(3), 1517–1538 (2014)
34. M. N. Kurt, Y. Yilmaz, X. Wang, Distributed quickest detection of cyber-attacks in smart grid. Trans. Inf. Forensic. Secur. IEEE, 1–1 (2018). https://doi.org/10.1109/tifs.2018.2800908
35. M. Pierre-Francois, Sequence covering for efficient host-based intrusion detection. IEEE Trans. Inf. Forensic. Secur. **14**(4), 994–1006 (2019)

36. O. Y. Al-Jarrah, Y. Al-Hammdi, P. D. Yoo, S. Muhaidat, M. Al-Qutayri, Semi-supervised multi-layered clustering model for intrusion detection. Digit. Commun. Netw. **4**(4), 277–286 (2018)
37. P. F. BROWN, Class-based n-gram models of natural language. Comput. Linguist. **18**(4), 467–479 (1992)
38. M. A. W. J.A. Hartigan, A k-means clustering algorithm. Appl. Stat. **28**(1), 100–108 (2013)
39. M. M. Cormode G, Approximating data with the count-min sketch. IEEE Softw. **29**(1), 64–69 (2012)
40. F. Deng, D. Rafiei, New estimation algorithms for streaming data: Count-min can do more. Webdocs.Cs.Ualberta.Ca (2007)
41. T. Morris, W. Gao, Industrial control system traffic data sets for intrusion detection research. Crit. Infrastruct. Protect. VIII. **441**. https://doi.org/10.1007/978-3-662-45355-1_5
42. A. Asuncion, D. J. Newman, UCI Machine Learning Repository (2007). http://www.ics.uci.edu/~mlearn/MLRepository.html%5Cnhttp://archive.ics.uci.edu/ml/datasets.html
43. S. Suthaharan, M. Alzahrani, S. Rajasegarar, C. Leckie, M. Palaniswami, Labelled data collection for anomaly detection in wireless sensor networks, 269–274 (2011). https://doi.org/10.1109/issnip.2010.5706782
44. A. Almalawi, A. Fahad, Z. Tari, A. Alamri, R. AlGhamdi, A. Y. Zomaya, An efficient data-driven clustering technique to detect attacks in scada systems. IEEE Trans. Inf. Forensic. Secur. **11**(5), 893–906 (2016)
45. Z. X. Weidong Cao, An efficient semi-supervised multi-level intrusion detection algorithm. J. Comput. Appl. **7**, 1979–1984 (2019)

Publisher's Note