PAPER *Special Section on Cryptography and Information Security*

Improvement of Anomaly Detection Performance Using Packet Flow Regularity in Industrial Control Networks

Kensuke TAMURA[†a)], ***Member and*** **Kanta MATSUURA**[††b)], ***Senior Member***

SUMMARY Since cyber attacks such as cyberterrorism against Industrial Control Systems (ICSs) and cyber espionage against companies managing them have increased, the techniques to detect anomalies in early stages are required. To achieve the purpose, several studies have developed anomaly detection methods for ICSs. In particular, some techniques using packet flow regularity in industrial control networks have achieved high-accuracy detection of attacks disrupting the regularity, i.e. normal behaviour, of ICSs. However, these methods cannot identify scanning attacks employed in cyber espionage because the probing packets assimilate into a number of normal ones. For example, the malware called Havex is customised to clandestinely acquire information from targeting ICSs using general request packets. The techniques to detect such scanning attacks using widespread packets await further investigation. Therefore, the goal of this study was to examine high performance methods to identify anomalies even if elaborate packets to avoid alert systems were employed for attacks against industrial control networks. In this paper, a novel detection model for anomalous packets concealing behind normal traffic in industrial control networks was proposed. For the proposal of the sophisticated detection method, we took particular note of packet flow regularity and employed the Markov-chain model to detect anomalies. Moreover, we regarded not only original packets but similar ones to them as normal packets to reduce false alerts because it was indicated that an anomaly detection model using the Markov-chain suffers from the ample false positives affected by a number of normal, irregular packets, namely noise. To calculate the similarity between packets based on the packet flow regularity, a vector representation tool called word2vec was employed. Whilst word2vec is utilised for the culculation of word similarity in natural language processing tasks, we applied the technique to packets in ICSs to calculate packet similarity. As a result, the Markov-chain with word2vec model identified scanning packets assimulating into normal packets in higher performance than the conventional Markov-chain model. In conclusion, employing both packet flow regularity and packet similarity in industrial control networks contributes to improving the performance of anomaly detection in ICSs.

key words: *industrial control systems, anomaly detection, natural language processing, machine learning, word2vec*

1. Introduction

ICSs control the devices managing critical infrastructure including electric power, gas and water. Whilst supplying core services to support the social lives of people, they are threatened with cyber attacks, i.e. cyberterrorism. Moreover, the companies managing ICSs face a threat of exfiltration for confidential information, i.e. cyber espionage. Cyberterrorism and cyber espionage have caused critical cyber incidents such as disruption of uranium enrichment facility in Iran and spying activity against US and European energy companies. It is said that origins of the incidents were the malware, called Stuxnet [1] and Havex [2], and they performed persistent reconnaissance against targeting industrial control networks [3].

Recently, there has been a great discussion about anomaly detection systems for ICSs. Some studies have claimed that the methods using packet flow regularity in industrial control networks achieved the high-accuracy detection of attacks disrupting normal behaviour of ICSs [4], [5]. In other words, it is shown that employing the packet flow regularity is effective to detect critical attacks against ICSs, i.e. cyberterrorism. The reason is that the regularity is caused from the normal behaviour of ICSs and that detecting abnormity of the regularity results in indentifying disruption of periodic processes of ICSs.

On the other hand, it is also important to detect preliminary scanning attacks because they are followed by cyber espionage and cyberterrorism. For example, Havex is the elaborate malware to gather the information on the targeting ICSs. It carries out cyber espionage through following four steps: infection, system information grabbing, network scanning and credential exfiltration [6]. In particular, the network scanning aims to make a thorough survey of components and software of ICSs such as Object Linking and Embedding for Process Control (OPC), Supervisory Control And Data Acquisition (SCADA) and Programmable Logic Controller (PLC). Based on the results of the investigation, further malware is employed to conduct more serious attacks against the ICSs such as cyberterrorism. Therefore, to prevent the following critical damages, it is essential to detect the preliminary scanning attacks against industrial control networks whilst such attacks are designed to be concealed behind general packets including TCP, UDP, ICMP and ARP packets, namely "noise."

However, the conventional alert systems using the packet flow regularity peculiar to ICSs have no capability to detect the probing attacks because the attacks do not constitute the regularity due to the fact that they are irregularly conducted to assimilate into noise. For example, excluding all the packets constituting TCP 3-way handshake, such as TCP SYN packets, the detection method proposed by Barbosa et al. cannot identify the scanning attacks with the general packets. The reason for the exclusion is that the general packets interrupt discovery of obvious cycles, i.e. the regularity, consisting of a series of regular packets peculiar

Manuscript received March 22, 2018.

Manuscript revised July 10, 2018.

[†] The author is with National Police Agency of Japan, Tokyo, 100-8974 Japan.

[††] The author is with Institute of Industrial Science, The University of Tokyo, Tokyo, 153-8505 Japan.

a) E-mail: research3073@gmail.com

b) E-mail: kanta@iis.u-tokyo.ac.jp

DOI: 10.1587/transfun.E102.A.65

to ICSs. Another method implemented by Maglaras et al. distinguishes general packets from anomalies by the regularity based on the time differences between two consecutive packets and on the size of each packet. Therefore, whilst the method can detect anomalies with extremely intensive packets or large-size ones such as Denial of Service (DoS) attacks, preliminary scanning attacks assimilating into the noise are not properly alerted.

The purpose of this study was to examine a novel detection method not only for anomalies against ICSs but for preliminary scanning attacks concealing behind the noise. To achieve the purpose, we built the Markov-chain model to represent the packet flow regularity originated with all the packets including the noise in industrial control networks. Also, we employed packet similarity based on the regularity to reduce false positives because it was indicated that the Markov-chain models for anomaly detection suffer from a number of false alerts [7]. Specifically, To calculate packet similarity, we adopted a vector representation technique, called word2vec, used in natural language processing tasks.

Finally, we demonstrated that our method can detect probing attacks with long intervals in higher performance than the Markov-chain model without applying packet similarity. The reason for adding the function of long intervals to the scanning attacks is because the attacks simulated slow scan [9]. The characteristic of the scan is that it is difficult for anomaly detection systems to detect the scanning with long intervals because it assimilates into a number of normal packets. In particular, anomaly detection systems using time differences between packets as proposed by [5] do not have the capability to detect slow scans. On the other hand, our proposal method employing the packet flow regularity can detect even slow scans.

2. Industrial Control Systems

ICSs are divided into two kinds: batch control systems and continuous control systems [8]. A fundamental characteristic of batch control systems is that a series of processes such as raw material inputs, heating, cooling and reactions is performed in one tank. For example, food and chemical plants are typical batch control systems. On the other hand, continuous control systems perform one process in a tank. After each process is completed, the product is transmitted to the next tank and is processed there. Oil refinery plants and steel mills are representative examples of continuous control systems.

2.1 Construction

In many cases, ICSs are installed with dedicated constructions, including software and network configurations, to each plant. To explain some features of ICSs, a sample construction of an ICS is shown in Fig. 1. The SCADA, OPC server and PLC in the Process Control Network of Fig. 1 are described as specific zones of SCADA Zone, DCS Zone and Control Zone, respectively, in a vendor's reference model [12]. The model was proposed based on the cyber security standard for ICSs of ISA99 [13]. The basic roles of the components are as follows:

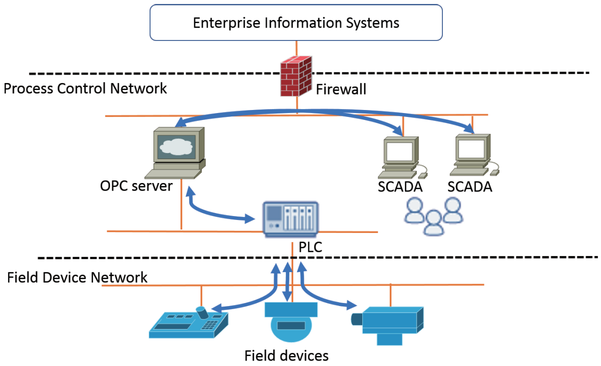

Fig. 1 Sample construction of ICS.

SCADA The computer transmitting commands by operators to the OPC server and displaying the status of field devices for the operators

OPC server The server computer mediating communication between SCADA and PLC

PLC The equipment controlling field devices by the commands transmitted from SCADA through OPC server

Field devices The equipment behaving in accordance with the commands from PLC and transmitting values measured by sensors to the PLC

2.2 Traffic in Industrial Control Networks

There are two kinds of network packets in industrial control networks. One is general packets irregularly communicated between Operating Systems (OSs) including Windows and Linux. The packets are necessary for operating basic services of OSs. For example, SCADAs and OPC servers constituting time critical ICSs are in need of communication based on Network Time Protocol (NTP). Also, Server Message Block (SMB) or Common Internet File System (CIFS) is employed to share files in ICSs. The other is ICSs-specific packets regularly communicated between components in ICSs. Compared with general enterprise information systems, the packets have two features: limited communication variety, and constant transmission rate and traffic volume.

2.2.1 Limited Communication Variety

The types of packets peculiar to ICSs are limited because it is only necessary to communicate between components, namely SCADA, OPC server, PLC and field devices, to operate ICSs. Furthermore, the order of communication between the components obeys a certain order to perform a series of processes properly. For example, two-headed arrows shown in Fig. 1 represent the ICSs-specific communication. A sequence of the communication depicted by downward arrows and upward ones is respectively as follows:

- Downward arrows
  1. SCADA transmits commands to OPC server.
  2. OPC server transfers the commands to PLC.
  3. PLC controls field devices based on the commands.
  4. Field devices behave in accordance with the commands.
- Upward arrows
  1. Field devices transmit measured values to PLC.
  2. PLC transfers the values to OPC server.
  3. OPC server transfers the values to SCADA.
  4. SCADA displays the status of each field device based on the values.

2.2.2 Constant Transmission Rate and Traffic Volume

The transmission rate in industrial control networks remains approximately constant. This is the reason that all the components regularly communicate with each other. For example, assuming that PLCs gather the values from field devices every second, the values are transmitted from PLCs to OPC servers with 1 second intervals. Similarly, OPC servers receiving the values from PLCs transmit them to SCADAs with the same intervals. Finally, SCADAs receiving the values update the status of field devices based on the values every second. In the case of command transmission from SCADAs to field devices, commands are transmitted in reverse order with the same intervals. In addition, the data size of the values and commands do not fluctuate dramatically. Therefore, since the traffic peculiar to ICSs regularly occurs and the size of data transmitted is approximately constant, the transmission rate and the traffic volume remain stable.

3. Related Work

Several studies have proposed anomaly detection techniques for ICSs using the packet flow regularity in industrial control networks [4], [5]. Specifically, these proposals take an approach to detect packets interrupting the regularity as anomalies. In this section, existing anomaly detection techniques based on the regularity are introduced and those problems are also clarified.

To develop ICSs-specific anomaly detection systems, Barbosa et al. [4] utilise the characteristic that a sequence of packets peculiar to ICSs is limited and periodic as described in 2.2.1. The key to detecting anomalies by focusing on the periodic communication based on packet flow is to determine packet cycles in the ample traffic. To be specific, all the irrelevant packets to obvious packet cycles are excluded because the packets interrupt detection of the periodic packets peculiar to ICSs. For example, all the general packets to establish 3-way handshake, e.g. TCP SYN and ACK packets, are eliminated from the training dataset. The method has a detectable capability for scanning attacks using the packets peculiar to ICSs including Modbus TCP. However, the scanning attacks using TCP SYN packets to investigate the services of OPC servers or SCADAs cannot be alerted by the method. The reason is that all the general packets interrupting a sequence of packets peculiar to ICSs are excluded from the training dataset to ascertain clear cycles of packets in industrial control networks.

Maglaras et al. [5] also proposed anomaly detection techniques using the regularity of packets in industrial control networks as described in 2.2.2. In particular, the methods utilise two kinds of regularities: the time differences between two consecutive packets and the size of each packet. The specific approach is that packets with outlier values were alerted as anomalies. The values are detected by the one-class support vector machine (OCSVM) using the time differences and the size normalised based on the data of normal packets. The anomalies using extremely intensive packets, e.g. DoS attacks, are detectable because the time differences between two consecutive packets would be incomparably smaller than those of normal ones. However, the detection method could be bypassed by scanning attacks with long intervals, called slow scan [9], because the time differences between the attack packets are not distinct. The disadvantage is critical especially because the cyber espionage and the cyber attacks against ICSs would be performed in obscurity [3].

4. Approach

The purpose of this study was to examine a novel detection technique not only for anomalies against ICSs but for preliminary scanning attacks using general packets. As even past studies have performed, the fundamental approach of detection methods for anomalies of ICSs is to employ the packet flow regularity in industrial control networks. Based on the approach of discovering the packet flow regularity, the framework of our method is as follows:

1. Learning the packet sequences based on the regularity in normal communication
2. Prediction of the following packet based on the packet sequences
3. Comparison between the predicted packet and the arrival packet
4. Evaluation of the differences between the packets

4.1 Learning and Prediction

Since existing studies have achieved high performance detection of critical attacks against ICSs, we took an approach along the lines of them. Specifically, we also focused on the packet flow regularity in industrial control networks. Moreover, to learn the regularity and to predict the following packet, we employed the Markov-chain model by regarding each packet as a state. The predicted packets are more likely to follow the latest packet in normal situation because the prediction is performed based on the normal packet sequences.

4.2 False Alerts Reduction

Reducing false alerts was important for this study because high error ratio based on a number of false positives was indicated as a disadvantage of the Markov-chain model in [7]. Therefore, we tried to add some flexibility to the results predicted by the Markov-chain model. To be more specific, we designed our method to accept not only the packets predicted by the model but their similar packets as normal ones. In this paper, the similar packets were defined as the packets having some similarity originated with the packet flow regularity in industrial control networks. As a result, since our approach accepted the similar packets as well as the predicted ones, the coverage of normal packets predicted by the Markov-chain model was expanded. Therefore, it is important to evaluate the effects of the expansion using the number of false alerts.

4.2.1 Packet Similarity

In this study, we employed packet similarity to reduce the number of false alerts. The technique to calculate the similarity between packets was inspired by a vector representation tool, called word2vec [10], utilised to calculate the similarity between words in natural language processing tasks. The main reason for the inspiration was that packet flow regularity, i.e. a sequence of packets, in industrial control networks would be regarded as a sequence of words in sentences. It was expected that prediction accuracy would be improved by incorporating packet similarity estimated by word2vec to the predicted packets.

4.2.2 word2vec

The word2vec builds the vector representation of words in natural language processing tasks. It has the advantage of calculating the similarity between words using the cosine distances between corresponding vectors. The basic hypothesis is that words used in contextually similar cases are semantically similar. The word2vec estimates the surrounding words, called contexts, from a given word or vice versa. In actual, the word2vec is designed to train a corpus to enable words with similar contexts to have higher similarity values.

5. Experiment

We evaluated the performance of our detection method by experiments. In this section, experimental environment, methods and evaluation are described.

5.1 Environment

Figure 2 illustrates the experimental environment. The components are deployed in the Process Control Network in Fig. 1. In particular, the environment shows two operations areas [14] where SCADA 1 and SCADA 2 observe PLC 1 and PLC 2 through OPC server 1 and OPC server 2, respectively. Each component, i.e. PLC, OPC server and SCADA, was installed a different software and played a role explained in 2.1.

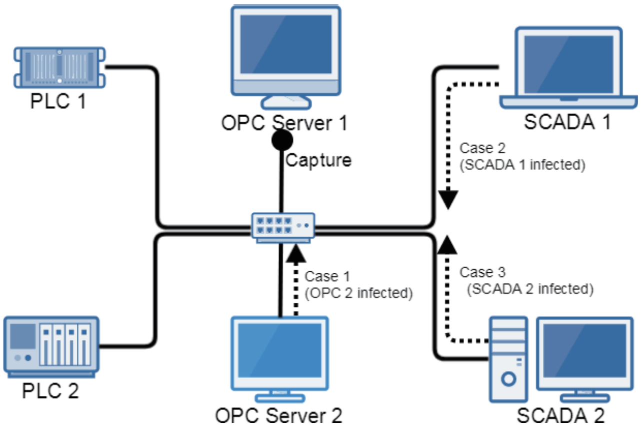

Fig. 2 Experimental environment.

The experiment simulated the processes to create a product in a chemical plant. A series of processes is as follows:

1. Input materials and water
2. Heat them to a set degree
3. Maintain them during a certain time period
4. Wait for reaction
5. Cool them to a set degree
6. Acquire the product

In general, these steps are performed in cooperation with field devices such as motors, physical tanks and sensors connected to PLCs. However, since the physical field devices were not deployed, the environment used the data virtually created in PLCs. In addition, a batch sequence (process 1 - 6) took approximately ten minutes.

Both training and test data were acquired at OPC server 1 (OPC 1) depicted in Fig. 2. Whilst training data consists of normal packets, test data includes attack ones. Also, the normal packets are composed not only of the regular packets peculiar to ICSs but of the irregular ones transmitted between OSs.

5.2 Attack Simulation

The experimental attacks simulated possible malware infection incidents. The purpose of the malware is to gather the information on ICSs-specific components, namely SCADAs, PLCs and OPC servers. To achieve the purpose, it performed scanning attacks to investigate opening ports of the components. Specifically, we evaluated the detection performance in each of three cases. In each case, one of OPC server 2 (OPC 2), SCADA 1 and SCADA 2 was infected and performed scanning attacks against OPC 1. Also, all the components including OPC 1 transmitted normal packets to the other components to operate the normal batch processes of the experimental chemical plant. As a result, OPC 1 received the following packets in each case:

Case 1 (OPC 2 infected) Normal packets and scanning packets from OPC 2
Case 2 (SCADA 1 infected) Normal packets and scanning packets from SCADA 1
Case 3 (SCADA 2 infected) Normal packets and scanning packets from SCADA 2

In this study, the data in each case is called test data. In addition, the scanning packets simulating investigative attacks to check opening ports of OPC 1 were generated by the following command:

```
nmap [OPC 1's IP address] --scan-delay 2000ms
```

The command means that the component transmits TCP SYN packets to OPC 1 every two seconds. According to the Nmap Reference Guide [15], the scanning technique using TCP SYN packets is the most popular one to recognise running services on a host. Every parameter except for interval time between scan packets is set as the default value. The interval of two seconds is longer than that of the default settings of "nmap." It means that this scanning attack is relatively slow. Each component such as OPC 2, SCADA 1 and SCADA 2 generates TCP SYN packets which consist of random source port numbers and destination port numbers originally defined in "nmap" command. In the experiment, the length of the TCP SYN packets in both the test data and the normal data is between 60 and 66 bytes. Therefore, TCP SYN packets generated by the command assimilate into normal packets.

5.3 Method

To detect the slow scanning attacks, we employed the Markov-chain model. The reason for our choice of the Markov-chain model is that it can represent the packet flow regularity, namely a sequence of packets, as the state transition. In addition, independent of time data, the Markov-chain model would have a capability to detect slow scan. Moreover, implementing the word2vec to calculate packet similarity, we tried to reduce the number of false alerts using the similarity.

Finally, to evaluate the detection performance of slow scan, we compared the following anomaly detection methods:

Method 1 Using only the Markov-chain model
Method 2 Using the Markov-chain with word2vec model

Figures 3 and 4 illustrate processing flows of each method. The elements depicted in Figs. 3 and 4 are explained in the following sections.

5.3.1 Data Acquisition

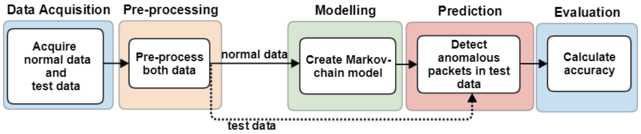

Fig. 3 Processing flow of Markov-chain model.

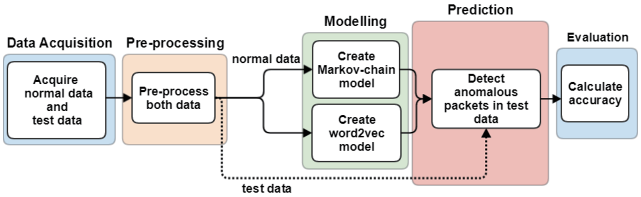

Fig. 4 Processing flow of Markov-chain with word2vec model.

Table 1 Characteristics of training and test data.
Numbers show the number of batch process cycles, that of all the packets and that of attack packets. The name of components, namely OPC 2, SCADA 1 and SCADA 2, shows the compromised component conducting scanning attacks against OPC 1.

Type	batch processes	all packets	attack packets
Training data	10	362,471	0
Case 1 (OPC 2)	1	24,440	336
Case 2 (SCADA 1)	1	23,623	344
Case 3 (SCADA 2)	1	24,186	344

In the data acquisition phases of Figs. 3 and 4, the normal data for training and test data including attack packets were collected. The normal one consists of the data gathered by operating the batch processes ten times. As mentioned above, the data includes not only regular packets peculiar to ICSs but irregular ones transmitted between OSs. On the other hand, the test data was collected by executing the command shown in 5.2 during the acquisition of the normal data generated by performing the normal batch processes once. Table 1 shows the characteristics of the training data and test data. Regarding the three kinds of test data, the number of normal packets is much more than that of scanning packets. It seems natural in actual cases because scanning attacks are performed behind a number of normal packets.

5.3.2 Pre-Processing

The pre-processing phases shown in Figs. 3 and 4 conducted replacement of source and destination port numbers in packets of all the dataset and transformation of the header fields in the packets.

The replacement of source and destination port numbers included in header fields of packets was performed under a rule to classify the port numbers into three groups: common server ports, random ports and ICSs-specific ports. First, the common server ports represent the well-known ports, namely numbers of 0–1023. Common services are operated on the port numbers. Therefore, these port numbers were used without any replacement.

Secondly, the port numbers which appeared once in the training data were regarded as random ones and substituted with the number of 100,000. The reason for the replacement

Table 2 Replacement and Transformation of packet header fields.

Packet header fields	Source IP address	Source port number	Destination IP address	Destination port number	Protocol
Original packet header	1.2.3.4	12345	10.20.30.40	80	HTTP
Replaced port numbers	1.2.3.4	100000	10.20.30.40	80	HTTP
One-dimensional text	1.2.3.4_100000_10.20.30.40_80_HTTP				

is that all the packets with random port numbers are treated as irrelevant packets each other even if only the port numbers are different. For example, in the communication between a client and a server, an approximately random number is used as a source port number of the client whilst the server uses the commonly pre-determined port number. Anomaly detection systems trained without replacement of port numbers would detect almost all the communication as anomalies in spite of the fact that the client used the same service on the same server again and again. This reason is that it is practically impossible that the training data, i.e. the actual communication data, includes all the random port numbers of the client. Therefore, the port numbers changed every connection need to be replaced with a certain number.

Finally, if a port number appeared more than once in the training data and it was not the well-known port, the number was replaced with the number of 200,000. It is assumed that the port is used to operate the service necessary for ICSs processes. As a result, the method to replace the port numbers based on the above policy is performed in the following order:

1. Well-known ports are not replaced to assume common server ports.
2. Ports which appeared only once are replaced with 100,000 to assume source ports of clients, i.e. random ports.
3. Ports which appeared more than once are replaced with 200,000 to assume ICSs-specific server ports.

In the case of actual industrial companies, the replacement should be performed by the following procedures:

1. Well-known ports should not be replaced.
2. Ports used to operate ICSs should be replaced with a certain number.
3. Other ports should be replaced with another certain number.

After the replacement of port numbers, the values of the header fields of packets needed to be transformed. In this experiment, the fields consisted of source IP address, destination IP address, source port number, destination port number and protocol name. These fields were treated not as multiple-dimensional data but as one-dimensional text by jointing all fields by the symbol of "_" to emphasise the relationships among the values of header fields in each packet. An example of data pre-processed by port number replacement and one-dimensional text transformation is shown in Table 2.

5.3.3 Modelling

We implemented two kinds of anomaly detection methods introduced as Method 1 and 2 in 5.3. Whilst Method 1 modelled the sequence of normal packets using the Markov-chain, Method 2 incorporated packet similarity using word2vec in the Markov-chain model based on the normal packet sequences. Specifically, building the Markov-chain models by both methods means the creation of the Markov dictionaries representing the packet sequences. In addition, word2vec in Method 2 means the acquisition of the vector representation of packets to calculate the similarity between packets.

The Markov dictionaries were created by training the normal data including both the periodic packets peculiar to ICSs and the packets irregularly transmitted between OSs. In particular, we modelled the second order Markov-chain because it has been claimed that high order Markov-chain would cause a decrease in the accuracy [11].

To calculate the packet similarity, the vector representation of packets was created by word2vec introduced in 4.2.2. We obtained the vectors representing all the normal packets by applying the same procedures as natural language processing tasks because of the transformation of packet header fields into one-dimensional text in pre-processing phase.

5.3.4 Detection

In Method 1 and 2, anomalies are alerted when the actual arrival packet is different from the predicted one. In this experiment, the prediction of the arrival packet is performed based on past two packets. Using past packets, the anomaly detection behaves every time a packet arrives.

Method 1 detects the packet as anomalies if the packet is not included in the list of the packets predicted using the Markov dictionaries. Figure 5 illustrates the flowchart to decide whether or not the arrival packet is normal. Let the p^{th} packet be the arrival packet to be identified whether or not it is normal. Define the i^{th} packet as the last one (i.e. $(p-1)^{th}$ packet) and the j^{th} packet as the one before the last (i.e. $(i-1)^{th}$ packet), respectively. If the p^{th} packet is predicted using the i^{th} packet and the j^{th} one, the p^{th} packet is normal. If not, j is traced back one by one to the $(i-5)^{th}$ packet. If the p^{th} packet is not predicted using the i^{th} packet and each packet until the $(i-5)^{th}$ one, both i and initialised j (i.e. $j = i - 1$) are traced back one by one similarly. Finally, if the p^{th} packet is not predicted using the $i = (p-4)^{th}$ packet and the $j = (p-9)^{th}$ one, the system detects the p^{th} packet as anomalous one. Although the boundary values are not optimised for the experiment, it is assumed that not all the four packets before the arrival packet (i.e. p^{th} packet) are attack packets. If those packets are not normal, the p^{th} packet is absolutely alerted because the combination of the i^{th} packet and the j^{th} one does not exist in the Markov dictionaries created based on normal data. Furthermore, to reduce the false alerts, the maximum numbers of tracing back for i and j are limited to four and five, respectively. This is the reason that the more i and j trace back, the higher the accidental hitting ratio of the prediction gets.

The flowchart of Method 2 is shown in Fig. 6. Let the s^{th} packet and the t^{th} one be the similar one to the i^{th} packet and the j^{th} one, respectively. The difference from Method 1 is that the prediction is performed not only using the i^{th} packet and the j^{th} one but using the similar ones, i.e. the s^{th} packet and the t^{th} one. The list of similar packets has n (n=10 in this experiment) packets in decreasing order of similarity values calculated by the cosine similarity based on the vector representation of packets obtained using word2vec.

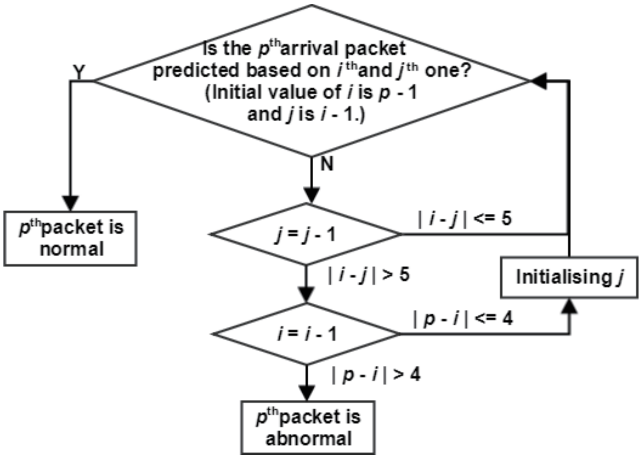

Fig. 5 Detection flow of Method 1.

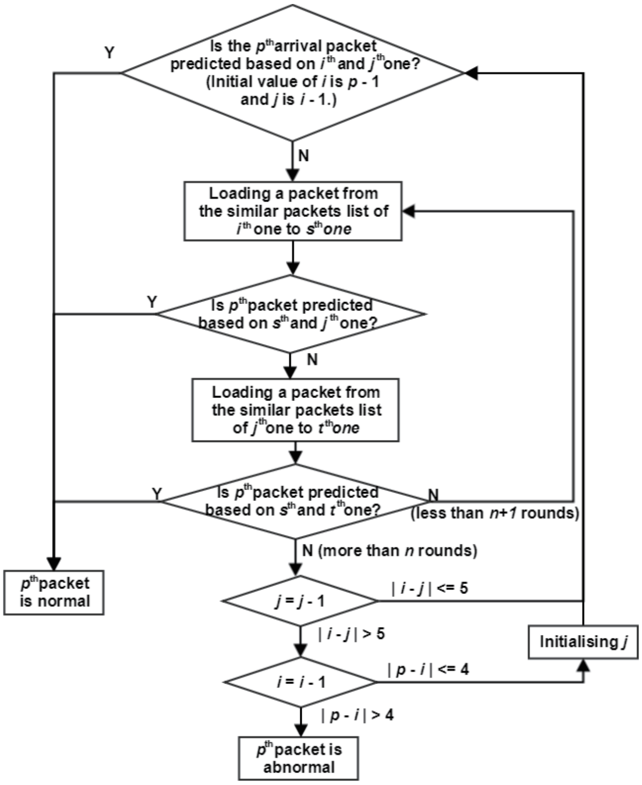
Is the pth arrival packet predicted based on ith and jth one? (Initial value of i is p - 1 and j is i - 1.)
Y
N
Loading a packet from the similar packets list of ith one to sth one
Is pth packet predicted based on sth and jth one?
Y
N
Loading a packet from the similar packets list of jth one to tth one
Is pth packet predicted based on sth and tth one?
Y
N (less than n+1 rounds)
N (more than n rounds)
pth packet is normal
j = j - 1
| i - j | <= 5
| i - j | > 5
Initialising j
i = i - 1
| p - i | <= 4
| p - i | > 4
pth packet is abnormal

Fig. 6 Detection flow of Method 2.

5.3.5 Evaluation

We evaluated the detection performance of Method 1 and 2 using the three types of test data, i.e. Case 1–3, including attack packets. The measures for evaluation are as follows:

- $Accuracy = (TP + TN)/(TP + FP + FN + TN)$
- $Precision = TP/(TP + FP)$
- $Recall = TP/(TP + FN)$
- $F1score = 2 * Precision * Recall/(Precision + Recall)$

Where, TP, TN, FP and FN stand for the number of True Positives, True Negatives, False Positives and False Negatives, respectively.

6. Result

The detection performance of Method 1 and 2 was evaluated using the four measures, namely accuracy, precision, recall and F1 score in the test cases shown in Table 1. Each result for Case 1–3 is illustrated in Figs. 7–9, respectively. In addition, Table 3 summarises the number of false positives and false negatives on each method. According to Figs. 7–9, the results of all the three cases showed almost the same tendency. To be more specific, the accuracy scores of both Method 1 and 2 recorded almost 100% in all the test cases. However, these accuracy scores would be influenced by the imbalanced number of normal packets and malicious ones as shown in 5.3.1. Regarding the precision score, the value of Method 2 indicated higher precision than Method 1 because Method 2 succeeded in reducing false positives using word2vec. According to Table 3, Method 2 achieved 60%, 32% and 59% decreases in the number of false positives of Case 1, 2 and 3, respectively. On the other hand, their recall values were approximately the same. It means that applying

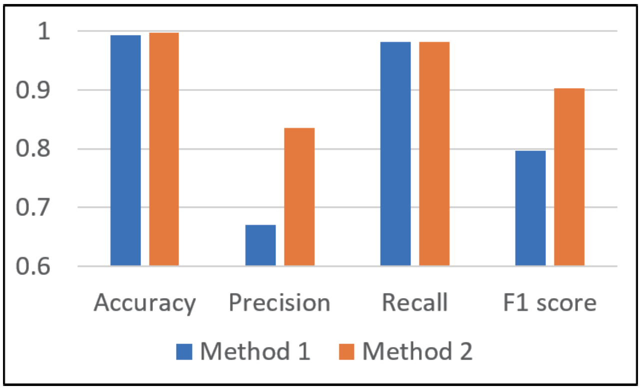

Fig. 7 Performance comparison in Case 1: Attacks from OPC2.

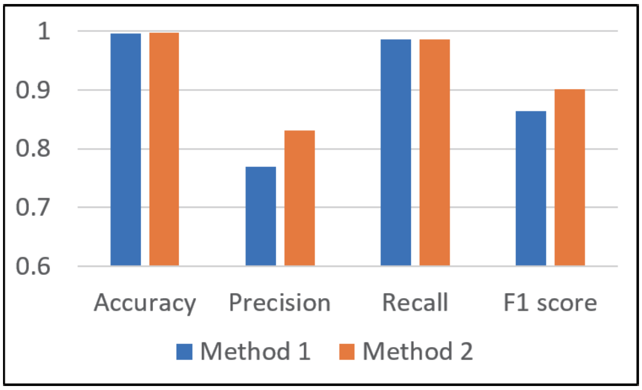

Fig. 8 Performance comparison in Case 2: Attacks from SCADA 1.

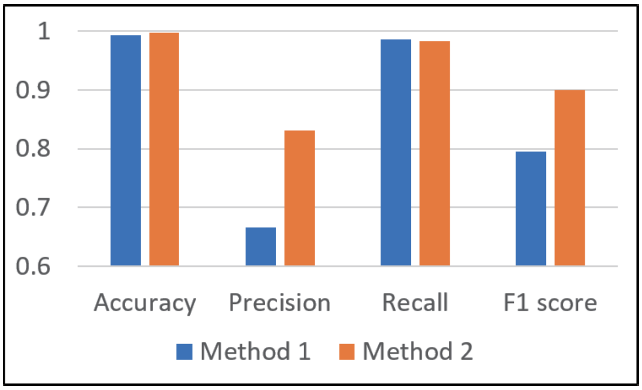

Fig. 9 Performance comparison in Case 3: Attacks from SCADA 2.

Table 3 Comparison results on the three types of attacks.

Attack	The number of FP		The number of FN	
	Method 1	Method 2	Method 1	Method 2
Case 1	162	65	6	6
Case 2	102	69	5	5
Case 3	170	69	5	6

packet similarity with word2vec to Method 2 had negligible effects on an increase in the number of false negatives.

As a result, Method 2 outweighed Method 1 in F1 score taking both false positives and false negatives into account. Therefore, it is evident from the experimental results that adopting packet similarity originated with packet flow regularity in industrial control networks contributed to the improvement of the detection performance of anomalies in ICSs.

7. Discussion

The purpose of our study was to propose the sophisticated detection technique for preliminary scanning attacks assimilating into noise as well as for specific attacks against ICSs. In this paper, a novel approach to detect investigative attacks against industrial control networks was evaluated. As a result, performance-improved detection model was obtained for scanning attacks with long intervals. The conventional ICSs-specific detection methods of Maglaras et al. and Barbosa et al. did not have a capability for identifying such slow scanning attacks. However, our method solved this problem using the detection technique based on packet flow regularity peculiar to ICSs. Moreover, in the literature [7], a number of false alerts based on the Markov-chain model have been reported. However, the drawback can be overcome by our approach employing packet similarity originated with the packet flow regularity in industrial control networks.

Our approach and the results are applicable to realistic ICSs. The experimental environment was based not only on the standard construction described in ISA99 but also on multiple operations constructed by different vendors. Also, although our approach conducted a specific way to replace port numbers and to transform packet data into one-dimensional text in pre-processing phase, we suggested the methods to apply them to actual environment in 5.3.2. Moreover, in the detection phase, the assumption that not all the several continuous packets before the arrival packet to be predicted are anomalous ones is not strict because the paper aims at the detection of slow scanning attacks with relatively long intervals to assimilate into normal ones.

In the future, from the standpoint of practicality, more elaborate anomaly detection systems using calculated values both state transition probabilities in the Markov-chain model and packet similarities in word2vec need to be developed. We conclude that ICSs-specific alert systems with improved detection performance using packet similarity originated with packet flow regularity in industrial control networks can be obtained.

8. Conclusion

We proposed a novel detection technique not only for anomalies against ICSs but for preliminary scanning attacks using general packets. The key to improving detection performance was to employ packet similarity originated with packet flow regularity in industrial control networks. To calculate the similarity, we utilised word2vec used in natural language processing tasks. The experimental results on detection performance of anomalies demonstrated that our model, i.e. the Markov-chain with word2vec model, had the capability to identify scanning attacks with long intervals in higher performance than the conventional Markov-chain model. In conclusion, what has been observed from the experiments indicates that investigative packets with long intervals, e.g. slow scans, can be alerted using packet similarity based on packet flow regularity in industrial control networks.

References

[1] R. Langner, "Stuxnet: Dissecting a cyberwarfare weapon," IEEE Security Privacy, vol.9, no.3, pp.49–51, 2011.

[2] S. Khandelwal, "New variant of Havex malware scans for OPC servers at SCADA systems," The hacker news. http://thehackernews.com/2014/07/new-variant-of-havex-malware-scans-for.html (accessed 2018-03-19).

[3] G. Wangen, “The role of malware in reported cyber espionage: A review of the impact and mechanism,” Information, vol.6, no.2, pp.183–211, 2015.

[4] R.R.R. Barbosa, R. Sadre, and A. Pras, “Exploiting traffic periodicity in industrial control networks,” International Journal of Critical Infrastructure Protection, vol.13, pp.52–62, 2016.

[5] L.A. Maglaras, J. Jiang, and T.J. Cruz, “Combining ensemble methods and social network metrics for improving accuracy of OCSVM on intrusion detection in SCADA systems,” Journal of Information Security and Applications, vol.30, pp.15–26, 2016.

[6] C. Bodungen, B. Singer, A. Shbeeb, K. Wilhoit, and S. Hilt, Hacking Exposed Industrial Control Systems: ICS and SCADA Security Secrets & Solutions, chap. 8, McGraw-Hill Education, 2016

[7] N. Ye, Y. Zhang, and C.M. Borror, “Robustness of the Markov-chain model for cyber-attack detection,” IEEE Trans. Rel., vol.53, no.1, pp.116–123, 2004.

[8] K. Stouffer, V. Pillitteri, S. Lightman, M. Abrams, and A. Hahn, “Archived NIST technical series publication superseding publication(s) guide to industrial control systems security,” NIST.SP.800-82r2.

[9] E Bou-Harb, M. Debbabi, and C. Assi, “Cyber scanning: A comprehensive survey,” IEEE Commun. Surveys Tuts., vol.16, no.3, pp.1496–1519, 2014.

[10] T. Mikolov, G. Corrado, K. Chen, and J. Dean, “Efficient estimation of word representations in vector space,” Proc. International Conference on Learning Representations, pp.1–12, 2013.

[11] H. Ren, Z. Ye, and Z. Li, “Anomaly detection based on a dynamic Markov model,” Information Sciences, vol.411, pp.52–65, 2017.

[12] M. Kawasumi, I. Ochiai, and K. Yokoyama, “Security engineering for control system,” Yokogawa Technical Report English Edition, vol.57, no.2, 2014.

[13] The International Society of Automation “Security for industrial automation and control systems Part 1-1: Terminology, concepts, and models,” Oct. 2007.

[14] E. Byres, “System integration: Revealing network threats, fears,” InTech Magazine of ISA Publications: https://www.isa.org/link/networkthreats/ (accessed 2018-06-01).

[15] G. Lyon, “Port scanning techniques,” Nmap Network Scanning: https://nmap.org/book/man-port-scanning-techniques.html (accessed 2018-06-01).

Kensuke Tamura received B.S. degree in Information Engineering from the University of Tsukuba in 1998, and M.S. degree in Computing and Security from King’s College London in 2017. He is currently Technical Official at National Police Agency of Japan and is also a collaborative researcher at Institute of Industrial Science, the University of Tokyo. He has been working as a digital forensic examiner since 1999. His research interests include network security and countermeasures against cyber terrorism. He is a member of IPSJ, IEICE and ACM.

Kanta Matsuura received his Ph.D. degree in electronics from the University of Tokyo in 1997. He is currently a Professor of Institute of Industrial Science at the University of Tokyo. From March 2000 to March 2001, he was a visiting scholar at University of Cambridge. His research interests include cryptography, computer/network security, and security management such as security economics. He was an Associated Editor of IPSJ Journal (2001–2005) and IEICE Transactions on Communications (2005–2008), and won Distinguished-Service Award from the IEICE Communications Society in 2008. He was Editor-in-Chief of Security Management (2008–2012), and is an Editorial-Board member of Design, Codes, and Cryptography (2010–present). He is a senior member of IEEE, ACM, IPSJ, and IEICE. He is a Vice President of JSSM (Japan Society of Security Management) (2016–present).