Event-Based Anomaly Detection for Non-Public Industrial Communication Protocols in SDN-Based Control Systems

Ming Wan[1], Jiangyuan Yao[2, *], Yuan Jing[1] and Xi Jin[3, 4]

Abstract: As the main communication mediums in industrial control networks, industrial communication protocols are always vulnerable to extreme exploitations, and it is very difficult to take protective measures due to their serious privacy. Based on the SDN (Software Defined Network) technology, this paper proposes a novel event-based anomaly detection approach to identify misbehaviors using non-public industrial communication protocols, and this approach can be installed in SDN switches as a security software appliance in SDN-based control systems. Furthermore, aiming at the unknown protocol specification and message format, this approach first restructures the industrial communication sessions and merges the payloads from industrial communication packets. After that, the feature selection and event sequence extraction can be carried out by using the *N*-gram model and K-means algorithm. Based on the obtained event sequences, this approach finally trains an event-based HMM (Hidden Markov Model) to identify aberrant industrial communication behaviors. Experimental results clearly show that the proposed approach has obvious advantages of classification accuracy and detection efficiency.

Keywords: Event sequence, anomaly detection, non-public industrial communication protocols, SDN.

1 Introduction

In recent years, ICSs (Industrial Control Systems) are exposed to an increasing number of cyberattacks [NCCIC/ICS-CERT (2016); NCCIC/ICS-CERT (2017)]. Especially, with the development of network technology, the original closure of ICSs has been broken. Although industrial automation can benefit from the situation attaching ICSs to Internet, the corresponding cybersecurity can also be dramatically impacted. Actually, the original control systems are designed for the process safety [Knijff (2014)], and they are always used in the air-gapped security environments [Ponomarev and Atkison (2016)]. In other words, ICSs are very sensitive to various cyberattacks, because they contain few security features during the early design phases. As defined in SP800-82 [Padgette, Scarfone and

[1] School of Information, Liaoning University, Shenyang 110036, China.

[2] College of Information Science & Technology, University of Hainan, Haikou 570228, China.

[3] Department of Computer Science and Engineering, Washington University, St Louis MO 63130, USA.

[4] Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China.

[*] Corresponding Author: Jiangyuan Yao. Email: yaojy@hainu.edu.cn.

CMC. doi:10.3970/cmc.2018.02195

Chen (2013)], ICSs are a group of automation systems used in industrial production and manufacturing, and they are widely applied to many critical infrastructures, such as power stations and transportation systems. Therefore, if one destructive cyberattack is unavoidable, it may inflict enormous life and property losses. With the deepening of Industrial Internet, IoT (Internet of Things) and Big Data, some new ICTs (Information Communication Technologies) have been proposed and applied in ICSs, and the SDN technology is a typical one [Genge and Haller (2016)]. However, although SDN-based control systems improve the communication efficiency and reliability, they cannot avoid the destructive cyberattacks.

Through analysis, we find that one of the core reasons to result in these cyberattacks is the vulnerability of industrial communication protocols, which are not designed with security in mind. As the main communication mediums in industrial control networks, industrial communication protocols are always vulnerable to extreme exploitations. Although NIST (National Institute of Standards Technology) presents the “defense in depth” strategy which emphasizes the security defense by parsing industrial communication protocols in depth, there exist a multitude of different protocols and most industry organizations generally tend to use different protocols in their control networks. In particular, because ICSs include a wide range of industrial components which are not compatible enough with one another, many industry organizations have developed their own industrial communication protocols, whose essential property is non-public and private. That is, the protocol specification and message format are unknown for the cybersecurity workforce. In this case, it heightens the difficulty to carry out the appropriate security mechanisms in ICSs.

Compared with traditional IT systems, the availability and timeliness of ICSs is extremely critical [Zhou, Huang, Xiong et al. (2015)]. Although various flexible and appropriate IT security approaches have been developed by both academia and industry, these approaches cannot be directly applied due to industrial communication protocols and the special requirements [Han, Xie, Chen et al. (2014)]. At present, two categories of ICS security approaches have been concerned: device-based and network-based. In the device-based cases, the interesting approaches are vulnerability exploiting [Liu, Liu, Liu et al. (2013)] and trusted computing [Wang, Liu, Yang et al. (2015)] for industrial field devices. In the network-based cases, the researches on network defense [Wan, Shang, Kong et al. (2017)], penetration testing and intrusion detection [Han, Xie, Chen et al. (2014); Zhu and Sastry (2010)] have been regarded as the significant breakthroughs. In practice, intrusion detection is the first step to secure ICSs, and its results can provide some necessary preparations for the self-adaptive protection or real-time response [Zhou, Huang, Xiong et al. (2015); Ten, Manimaran and Liu (2010)]. Besides, as a bypass monitoring technology, intrusion detection can effectively identify abnormal communication behaviors without affecting the availability and timeliness of ICSs. In the existing researches, anomaly detection, which is a representative class of intrusion detections, has been attracting many attentions of researchers, because the attack behaviors in ICSs always incorporate the characteristics of concealment and unpredictability. Simply stated, anomaly detection need not predefine and understand each attack behavior, and can build a normal communication behavior model to identify misbehaviors. However, one of the important prerequisites is that we can parse enough communication contents from the captured packets. In ICSs, this prerequisite may get lost due to non-public industrial communication protocols. So, developing flexible and adaptive anomaly detection approach for non-public industrial communication protocols in ICSs becomes a severe challenge.

Differently, ICSs are equipped with the relatively stable communication patterns [Valdes and Cheung (2009)], that is, the communication behaviors and states of ICSs are limited and regular. Additionally, because the network architecture of logic control and data forwarding separation is presented in SDN-based control systems, the security functions can be designed as a series of software appliances, which can be installed in SDN switches to offer diversified network security measures for ICSs. According to these characteristics, this paper proposes a new event-based anomaly detection approach to identify misbehaviors which use non-public industrial communication protocols, and this approach can be installed in SDN switches as a security software appliance in SDN-based control systems. Furthermore, aiming at the unknown protocol specification and message format, this approach carries out the feature selection and event sequence extraction by using the *N*-gram model and K-means algorithm, and trains a normal event-based HMM model to identify the unexpected industrial communication behaviors. In order to evaluate our approach, we also build a simulated SDN-based control system, whose communication is based on Siemens Profinet protocol. The experimental results and analysis show that our approach has obvious advantages of classification accuracy and detection efficiency.

2 Related work

According to the proposed classification in traditional IT systems [Garcia-Teodoro, Diaz-Verdejo, Macia-Fernandez et al. (2009)], the anomaly detection approaches in ICSs can also fall into three categories: statistics-based, knowledge-based and machine learning-based. In the statistics-based approaches, Wei et al. [Wei and Kim (2012)] propose an intrusion detection system for wireless industrial networks, and this system introduce a data traffic prediction model based on autoregressive moving average (ARMA) using the time series data. Ozcelik et al. [Ozcelik and Brooks (2016)] use the CUSUM method to present a statistics description of the communication traffic in ICSs, and identify its abnormal change point. By using both statistical analysis of traditional network features and specification-based metrics, Kwon et al. [Kwon, Kim, Lim et al. (2015)] present a novel behavior-based IDS for the smart grid infrastructure. In the knowledge-based approaches, Khalili et al. [Khalili and Sami (2015)] propose SysDetect (a Systematic approach to Critical State Determination), which is an iterative algorithm to systematically determine the critical states of industrial processes based on Apriori algorithm. A novel multimodel-based anomaly intrusion detection system for industrial process automation is designed in Zhou et al. [Zhou, Huang, Xiong et al. (2015)], and this system uses a classifier based on an intelligent hidden Markov model to differentiate the actual attacks from faults. Goldenberg et al. [Goldenberg and Wool (2013)] design an algorithm to automatically construct the DFA (Deterministic Finite Automaton) of each HMI-PLC channel, and deploys a model-based intrusion detection system on Modbus/TCP networks. In the machine learning-based approaches, Almalawi et al.

[Almalawi, Fahad, Tari et al. (2016)] propose an innovative intrusion detection approach to detect SCADA tailored attacks, and it is based on a data-driven clustering technique of process parameters, which automatically identifies the normal and critical states of a given system. Linda et al. [Linda, Vollmer and Manic (2009)] present an intrusion detection system using neural network-based modeling to tailor the specifics of critical infrastructures. Anoop et al. [Anoop and Sreeja (2013)] put forward a new genetic algorithm-based approach for calculating filtering parameters for DDOS, R2L, U2R attacks to make SCADA systems more secure. Schuster et al. [Schuster, Paul, Rietz et al. (2015)] and Wan et al. [Wan, Shang and Zeng (2017)] suggest using one class SVM (Support Vector Machine) to identify protocol-specific misbehaviors in industrial control networks. From the above three categories of approaches, we can see that the anomaly detection in ICSs has been widespread concerned and acknowledged by both academia and industry. Furthermore, one common characteristic of anomaly detection in ICSs is to build a normal model (such as industrial communication behavior model or state model), and identify abnormal activities by comparing the observed industrial data with this model. In this paper, our approach also stems from this characteristic, because it do not require the specific attack samples, which are scarce and often not publicly disclosed by utility companies [Silva, Silva, Wickboldt et al. (2016)].

3 Software defined security function model

Based on the SDN technology [Gelberger, Yemini and Giladi (2013)], the network architecture can be decoupled into two parts: control plane and data plane. Furthermore, SDN controllers in the control plane give the network control decisions, and SDN switches in the data plane only accomplish the data forwarding function according to these decisions. From this point, some security mechanisms based on the SDN architecture have been announced by many researchers [Silva, Silva, Wickboldt et al. (2016); Kalman (2015); Sainz, Iturbe, Garitano et al. (2017)]. Similarly, a novel security viewpoint is that security defense functions can be dynamically configured in accordance with different network security requirements. Fig. 1 shows the dynamic configuration model of security defense functions based on the SDN architecture. More specifically, SDN controllers can gather diversified security defense functions in the form of software appliances, and install these softwares into SDN switches on the basis of the specific network resources and different security requirements. After the deep packet parsing, SDN switches can perform different network security defenses by using the downloaded functions. As an application example, we propose an event-based anomaly detection approach for non-public industrial communication protocols, and this approach can be applied in SDN switches to identify misbehaviors. Due to the similar work in Silva et al. [Silva, Silva, Wickboldt et al. (2016); Kalman (2015); Sainz, Iturbe, Garitano et al. (2017)], it is worth mentioning that we do not emphasize the specific implementation process on this model, and we only focus on the detailed design and evaluation on our event-based anomaly detection approach. Besides, to detect the encrypted communication data is beyond the applied scope of our approach, and one of the main reasons is that the encryption-based methods have not been properly used in today’s ICSs because of the highly-available and real-time requirements.

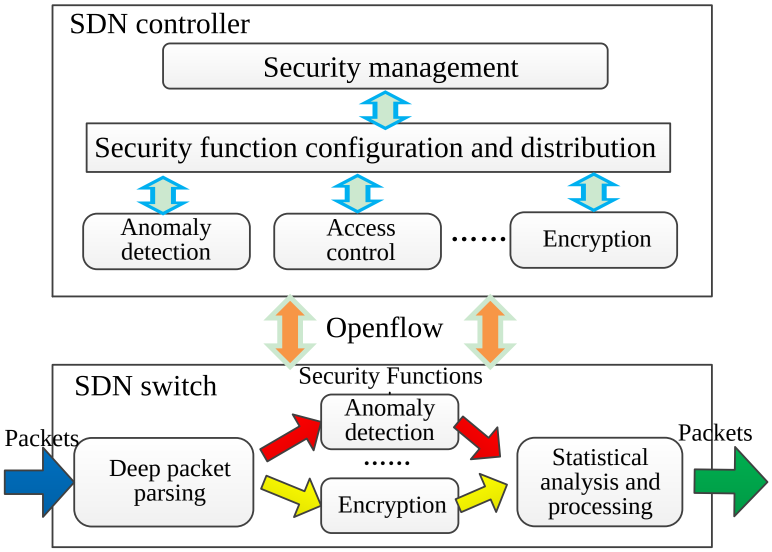

Figure 1: Dynamic configuration model of security defense functions based on the SDN architecture

4 Event-based anomaly detection for non-public industrial communication protocols

By capturing and analyzing industrial communication packets, this approach can differentiate misbehaviors (such as intrusion behaviors, unauthorized behaviors or misoperations) from the normal technological operations. Fig. 2 presents the basic model of the proposed approach, and this model is a two-stage process, including online self-learning stage and real-time detection stage. Furthermore, by using the extracted event sequences, the online self-learning stage mainly trains an effective event-based hidden Markov model, and achieves the behavior probability threshold by the iterative computing. The real-time detection stage calculates the responding behavior probability of the observed industrial communication data according to the trained hidden Markov model, and compares with the behavior probability threshold to realize anomaly detection.

4.1 Data preprocessing

A detailed implementation of data preprocessing in both online self-learning stage and real-time detection stage can be listed as follows:

Session reconstruction. By capturing industrial communication packets in real time, we recombine these packets in chronological order. In general, the non-public industrial communication protocols may involve two types: one is based on TCP/IP, and the other is directly layered on typical Ethernet. Furthermore, the biggest difference between these two types is to meet the special response time requirement. In the reconstruction process, if these packets are based on TCP/IP, we reconstruct each session according to the four tuples, including source IP address, destination IP address, source port and destination port; if these packet belongs to the second type, we reconstruct each session in accordance with the three tuples, including source MAC address, destination MAC address and protocol type. Besides, we also rearrange the out-of-order packets according to the sequence number and drop the duplicate packets.

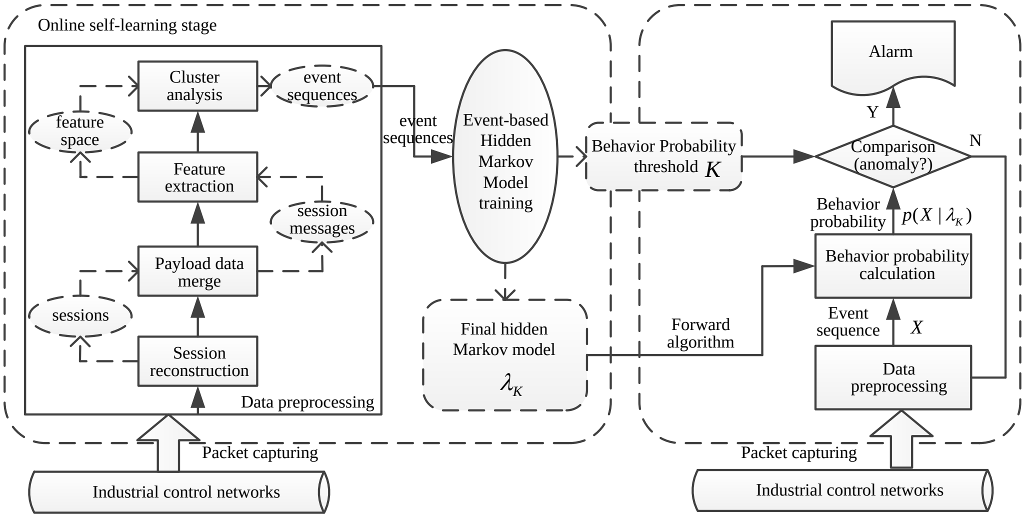

Figure 2: Basic model of event-based anomaly detection approach for non-public industrial communication protocols

Payload data merging. For each session, we get the payload data from the application layer of each packet, and form the session messages by merging all payload data belonging to the same session. In this case, the merged payload data in one session may either be too large or be not suitable for the further analysis. In order to resolve this problem, we can break these data into several small session messages in order.

Feature extraction. Although the protocol specifications of non-public industrial communication protocols are unknown, ICSs always implement the periodic technological process. So, the session messages belonging to different sessions have strong similarities. By establishing the *N*-gram model, we map the byte sequences in the session messages to a limited feature space, and all features in this feature space can represent the specific nature of session messages. Because ICSs have the relatively limited network scale and communication states, the "address field" or "function field" defined in industrial communication protocols generally do not exceed 2 bytes. So, we adopt *N=2* when establishing the *N*-gram model.

Cluster analysis. Because the initial dimension of feature space is very big, we use the cluster analysis to reduce the dimension of feature space, and improve the efficiency and accuracy of HMM model to some extent. In our approach, we introduce *K*-means algorithm to aggregate the features, and divide the whole feature space into several clusters. Specifically, the features in the same cluster have a remarkably general similarity, and the features in different clusters vary significantly. Besides, we regard each cluster as one kind of event. Above all, the interactive sessions belonging to non-public industrial communication protocols can be described as a series of event sequences, and each event sequence can be considered as one industrial communication behavior.

4.2 Event-based hidden Markov model

The hidden Markov model is a tool which can effectively represent the probability distributions of observed sequences [Ghahramani (2001)], and it is widely used to resolve three problems: decoding, evaluation and learning [Kohlschein (2013)]. In particular, our approach successfully utilizes its evaluation ability. The basic definition of a hidden Markov model can be briefly depicted by the following parameters:

1) N : the number of the states in the model. N states can be marked as $s_1, s_2, \ldots, s_N$, and if some state at time t is q_t , q_t must meet $q_t \in (s_1, s_2, \ldots, s_N)$.

2) M : the number of the observations in the model. M observations can be marked as $v_1, v_2, \ldots, v_M$, and if some observation at time t is o_t , o_t must meet $o_t \in (v_1, v_2, \ldots, v_M)$.

3) ω : the probability distribution of initial states. Where $\omega = (p_1, p_2, \ldots, p_N)$, and $p_i (i = 1, 2, \ldots, N)$ denotes the probability of the HMM to start in state s_i .

4) A : the matrix of state transition matrix. Where $A = (a_{ij})_{M \times N}$, and a_{ij} denotes the probability of a transition from state s_i to s_j .

5) B : the matrix of emission matrix. Where $B = (b_{ij})_{M \times N}$, and b_{ij} denotes the probability of emitting observation v_j in state s_i .

Based on the above parameters, the hidden Markov model can be further defined as $\lambda = (N, M, \omega, A, B)$ or $\lambda = (\omega, A, B)$.

According to the definition of the hidden Markov model, we use the event sequences obtained from the normal industrial communication data to train the event-based hidden Markov model through iterations, and build a normal behavior model for non-public industrial communication protocols. The detailed steps are the followings:

Step 1: Establish the initial hidden Markov model λ_0 according to the selected initial parameters.

Step 2: Based on the initial model λ_0 and the input event sequence O , introduce the Baum-Welch algorithm [Kohlschein (2013)] to train new hidden Markov model λ .

Step 3: By using the Forward algorithm [Kohlschein (2013)], calculate the behavior probabilities $p(O \mid \lambda)$ and $p(O \mid \lambda_0)$ of the event sequence O in the models λ and λ_0 , respectively. Moreover, the behavior probability can be computed by $p(O \mid \lambda) = {}^{\left|\log P_o\right|}\!\big/_{n_o}$. Here, $\log P_o$ is the log probability of the event sequence O , and n_o is the number of the events in the event sequence O .

Step 4: If the m consecutive comparisons are $\left| p(O \mid \lambda) - p(O \mid \lambda_0) \right| < \delta$ (Here, δ is a default value.), the trained process is over, and the final hidden Markov model λ_K and the behavior probability threshold K are obtained. Here, the behavior probability threshold is the smallest one of *m* training behavior probabilities.

Step 5: Conversely, set λ_0 equal to λ, and go to Step 2.

4.3 Real-time detection

In our real-time detection stage, we use the detection mechanism of HMM. That is, we calculate the behavior probability $p(X \mid \lambda_K)$ by using the observed event sequence X and the hidden Markov model λ_K, and compare $p(X \mid \lambda_K)$ with the behavior probability threshold K to identify misbehaviors. The real-time detection process can be stated as follows:

Step 1: Capture industrial communication data in real time, and carry out the data preprocessing, including session reconstruction, payload data merging, feature extraction and cluster analysis. After that, generate the observed event sequence X.

Step 2: Input the observed event sequence X into the hidden Markov model λ_K, and use the Forward algorithm to calculate the corresponding behavior probability $p(X \mid \lambda_K)$.

Step 3: Compare $p(X \mid \lambda_K)$ with the behavior probability threshold K. If $p(X \mid \lambda_K) > K$, an alarm will be generated; conversely, go to Step 1.

5 Experimental results and analysis

5.1 Experimental environment and preprocessing

In order to evaluate our approach, we build a simulated SDN-based control system in which some attack and detection experiments are performed. As shown in Fig. 3, this system simply simulates the automobile assembly line, and its control segment mainly consists of one master PLC and three slave controllers. Furthermore, the communication between master PLC and slave controllers conforms to Siemens Profinet protocol, which can be approximately considered as a non-public industrial communication protocol in our experiments. The technological process can be outlined as follows: Firstly, robot controller A receives the commands from master PLC to actuate industrial robot A, and industrial robot A catches the model car to the numerical control system; Secondly, master PLC sends the assembly commands to slave PLC, and salve PLC drives the numerical control system to complete the simulation assembly; Finally, based on the commands of master PLC, robot controller B controls industrial robot B to unload the assembled model car. In addition, we run this system and capture the Profinet packets from the SDN switch, which analyze the experimental data by using our anomaly detection approach to identify misbehaviors.

According to the above technological process, we capture the normal communication packets at three intervals, and obtain 3 Profinet data samples by removing all insignificant packets, such as broadcast packets. Additionally, these data samples represent the normal communication activities of this simulated control system, and mainly involve the control commands and running states of field execution devices. After the session reconstruction, we totally get 233 normal communication sessions, and Fig. 4 plots the number variation of packets in each session, which depicts the session comparison in different data samples. From this figure we can see that, the packet numbers of all sessions are roughly distributed into three areas, which are [0, 12], [56, 60] and [92, 126], respectively. Moreover, most of packets belong to the first area, that is, the majority of sessions are composed of a small quantity of packets. By extracting features from the session messages, we win a total of 101854 feature values from all sessions, and the number of features after the removal of duplicate data is 5670. Fig. 5(a) shows the number variation of features under different sessions. Similarly, the number of features in each session also presents three statuses: small, medium and large numbers. Furthermore, most of sessions are assigned to the small and medium statuses, and each status changes steadily and with little volatility. Fig. 5(b) plots the accumulated percentage variation for all features. According to the stepped curve in this figure, there are plenty of duplicative feature values in the extracted features, and it also means that the session has the relatively high similarity with each other. In other words, we can exploit the similarity to accomplish the anomaly detection for non-public industrial communication protocols.

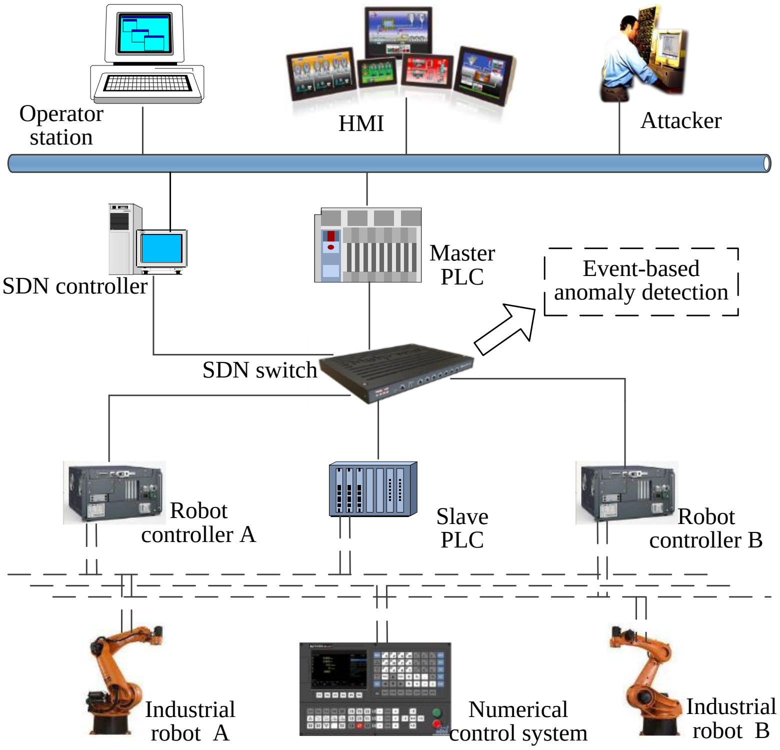

Figure 3: Simulated SDN-based control system based on Siemens Profinet protocol

Before evaluating the detection performance of our approach, we first use K-means algorithm to preprocess all features. In this experiment, we divide the whole feature space into 15 clusters, and Fig. 6 shows the distribution of the corresponding 15 cluster centers. On this basis, the event sequence description of each session can be established by using these 15 clusters. According to the 233 normal event sequences, we further introduce the Baum-Welch algorithm to train the event-based hidden Markov model. In order to build an optimal HMM model, it is worth mentioning that we select m=100 and $\delta = 0.01$ in the training process. Furthermore, the whole training time is about 2749.63 seconds, and the behavior probability threshold of the optimal HMM model λ_K is K=1.1333.

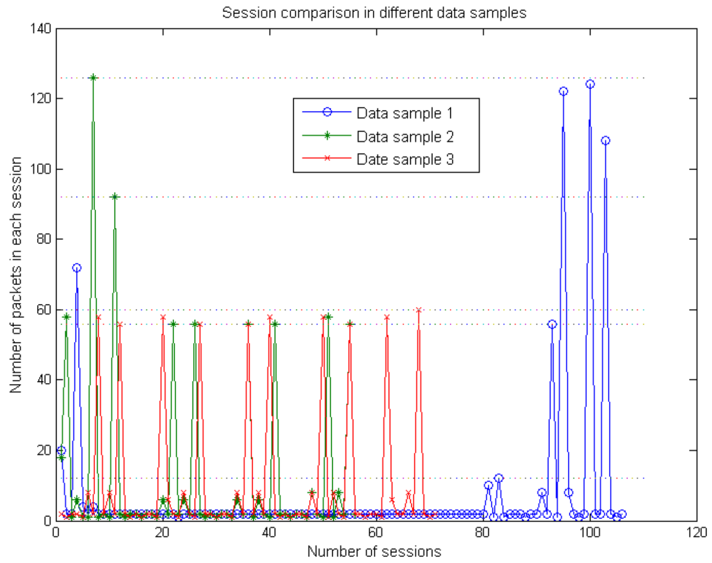

Figure 4: Number variation of packets in different data samples

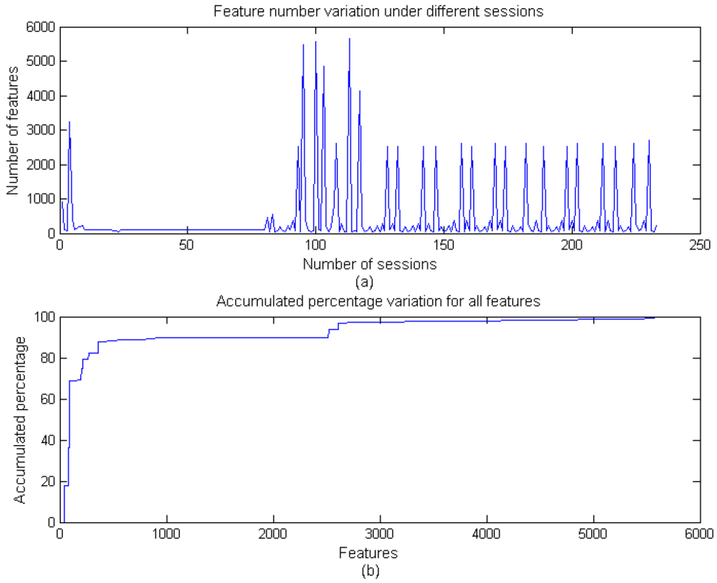

Figure 5: Number variation of features under different sessions and accumulated percentage variation for all features

5.2 Detection performance evaluation

In order to evaluate the practical detection performance, we perform 10 experiments to analyze the predicted accuracy and consuming time. In each experiment, we forge and replay the false Profinet packets to simulate the attack to the robot controllers and slave PLC. More specifically, the percentage of the forged data in these packets is about $1/25$, and these forged data basically change Profinet protocol parameters, such as register values. Finally, we generate 120 malicious sessions in each experiment according to the data preprocessing. Additionally, we calculate the behavior probability for each malicious session, which is compared with the above behavior probability threshold to identify the corresponding abnormal communication behavior. Tab. 1 shows the experimental results on the predicted accuracy and consuming time in detail. From this table we can see that the average predicted accuracy and average consuming time are 91.08% and 2.11 s, respectively. Especially, the maximum predicted accuracy can reach 94.17% in the 5th and 9th experiments, and the minimum consuming time is only 1.49 s to detect 120 malicious sessions. To some degrees, these results indicate that our approach has distinct advantages of classification accuracy and detection efficiency, and also indirectly verify it has the remarkable capacity to detect the abnormal communication behaviors under non-public industrial communication protocols.

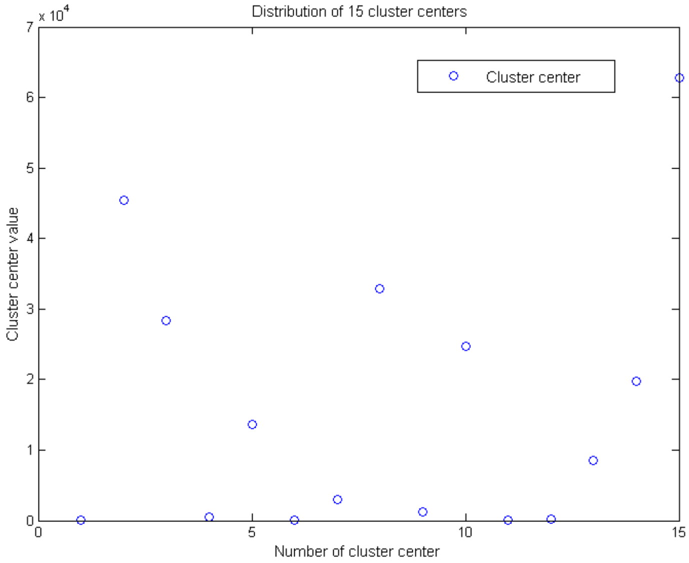

Figure 6: Distribution of 15 cluster centers

In practice, the different percentages of the forged data in the packets can exert a noticeable influence on the predicted accuracy. Fig. 7 draws the predicted accuracy variation when detecting the malicious sessions which using various percentages of forged data in the packets. In this figure, $p1 \ldots p9$ represent the different percentages of forged data, whose values are $1/45$, $1/40$, $1/35$, $1/30$, $1/25$, $1/20$, $1/15$, $1/10$ and $1/5$, respectively. One step further, for each percentage we perform 3 experiments, and also generate 120 malicious sessions in each experiment. As shown in Fig. 7, we plot the minimum predicted accuracies, the maximum predicted accuracies and the average predicted accuracies for every 3 experiments. The results indicate that the predicted accuracy markedly increases with the increment of the percentage of forged data in the packets. Especially, the predicted accuracy may reach 100%, when the percentage of forged data in the packets is $1/15$. In other words, our approach can be more effective to identify the abnormal communication behaviors caused by the large percentage of forged data.

Table 1: Predicted accuracy and consuming time under 15 clusters

	Predicted accuracy	Consuming time
1	90.83%	2.37 s
2	89.17%	1.69 s
3	90.00%	2.24 s
4	92.50%	2.23 s
5	94.17%	2.27 s
6	90.83%	1.49 s
7	84.17%	1.70 s
8	91.67%	2.64 s
9	94.17%	2.21 s
10	93.33%	2.29 s
Average value	91.08%	2.11 s

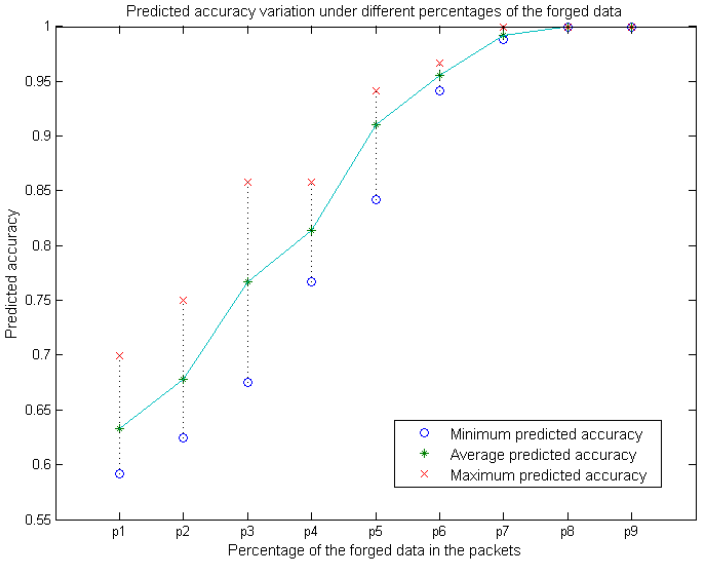

Figure 7: Predicted accuracy variation under different percentages of forged data in the packets

In particular, cluster analysis is an important link in our approach, because the event sequences which are used to build the HMM model are generated by this step. However, different numbers of clusters can have some performance impacts on the predicted accuracy. Fig. 8 shows the predicted accuracy comparison under different numbers of clusters. Furthermore, we perform 10 experiments for each number of clusters, and 120 malicious sessions in each experiment are generated. Additionally, the percentage of the forged data in the malicious packets is also about $1/25$. From this figure we can see that, the predicted accuracy for each number of clusters holds the characteristic of fluctuation, and the average predicted accuracies under 15 clusters, 20 clusters and 25 clusters are 91.08%, 92.42% and 95.08%, respectively. That is, the larger the number of cluster is, the higher the predicted accuracy is. However, it is infeasible to enlarge the number of clusters without constraint, because the large number of clusters can consume excessive time for the data preprocessing, and increase the computational complexity of the event-based HMM model. In short, we should carefully consider setting the number of clusters in accordance with actual experiences and network circumstances.

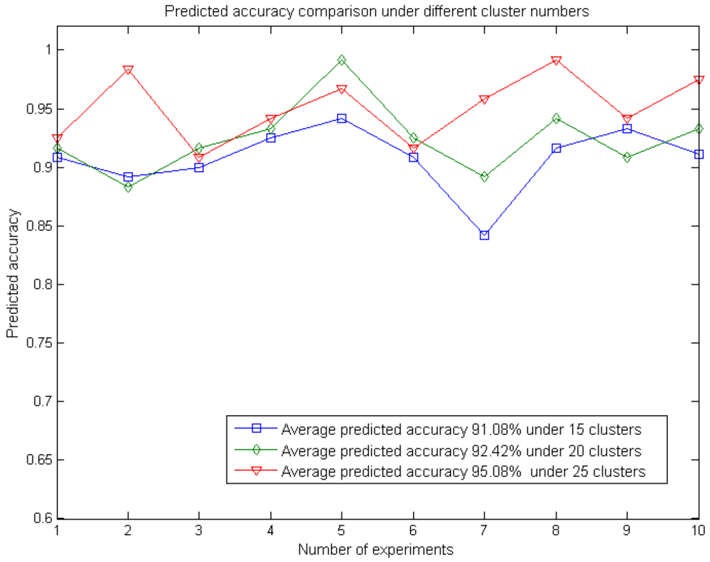

Figure 8: Predicted accuracy comparison under different cluster numbers

5.3 Performance comparison

Based on the general idea of data preprocessing, we introduce two different anomaly detection approaches to perform the predicted accuracy comparison, and analyze the reasons for the adopted HMM model in this paper. Moreover, two other detection approaches are BP neural network and NB (Naïve Bayes) detection algorithms, respectively. In these experiments, we still generate 120 malicious sessions in each experiment, and the percentage of the forged data in the malicious packets is also about $1/25$. Differently, we carry out the cluster analysis for each malicious session to generate 15 cluster centers, because these two detection algorithms need the test samples which have the same dimension. Fig. 9 plots the predicted accuracies of 6 experiments, and Tab.

2 depicts the comparison results of average predicted accuracies of these three anomaly detection approaches. From this table we can conclude that, the average predicted accuracies of BP and NB detection algorithms are 74.58% and 80.83% respectively, and the corresponding predicted accuracy of our approach is well above the ones of BP and NB detection algorithms. In conclusion, the proposed approach in this paper is more specifically suited to detecting abnormal behaviors for non-public industrial communication protocols, and two causes are related to this situation: one is that our approach can provide considerably better detection performance, and the other is that the event-based HMM model does not require the event sequences in all sessions to have the same dimension.

Table 2: Comparison of different anomaly detection approaches

Average predicted accuracy		
Our approach	BP neural network	NB (Naïve Bayes)
91.08%	74.58%	80.83%

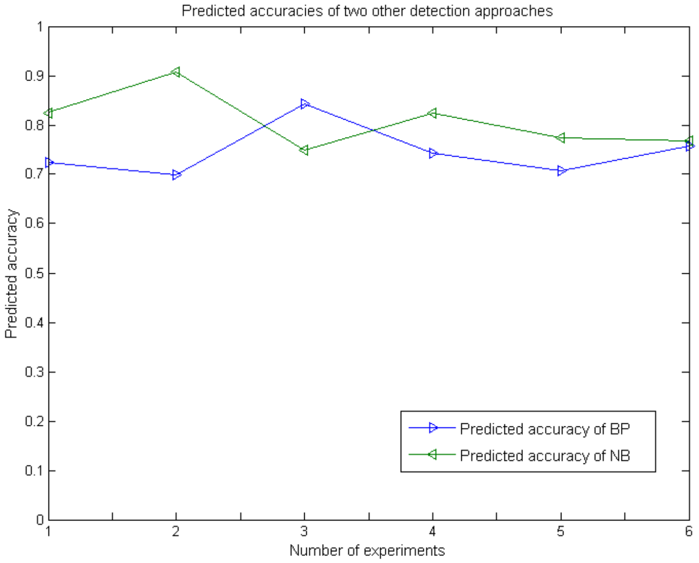

Figure 9: Predicted accuracies of BP neural network and NB (Naïve Bayes) detection algorithms

6 Conclusion

Based on the SDN technology, this paper first introduces the software defined security function model by using the architecture of logic control and data forwarding separation. After that, an event-based anomaly detection approach for non-public industrial communication protocols is proposed in SDN-based control systems, and the basic idea behind it is highly accessible. That is, identifying and diagnosing the anomalous communication behaviors for non-public industrial communication protocols by establishing the event-based HMM model. Actually, in order to overcome the difficulties of the unknown protocol specification and message format, we first design the practicable data preprocessing, including session reconstruction, payload data merging, feature extraction and cluster analysis. By obtaining the event sequences of all sessions, the event-based hidden Markov model is built to identify the corresponding misbehaviors. At last, many experiments are completed to evaluate our approach. We show that, the proposed approach has obvious advantages of classification accuracy and detection efficiency. Due to its salient characteristics, we believe that our approach is serviceable.

Acknowledgement: This work is supported by the Hainan Provincial Natural Science Foundation of China (618QN219), the National Natural Science Foundation of China (Grant No. 61501447) and the General Project of Scientific Research of Liaoning Provincial Department of Education (LYB201616). The authors are grateful to the anonymous referees for their insightful comments and suggestions.

References

Almalawi A.; Fahad, A.; Tari, Z.; Alamri, A.; AlGhamdi, R. et al. (2016): An efficient data-driven clustering technique to detect attacks in SCADA systems. *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 5, pp. 893-906.

Anoop A.; Sreeja M. S. (2013): New genetic algorithm based intrusion detection system for SCADA. *International Journal of Electronics Communication and Computer Engineering*, vol. 2, no. 2, pp. 171-175.

Garcia-Teodoro, P.; Diaz-Verdejo, J.; Macia-Fernandez, G.; Vazquez, E. (2009): Anomaly-based network intrusion detection: Techniques, systems and challenges. *Computer & Security*, vol. 28, no. 1-2, pp. 18-28.

Gelberger, A.; Yemini, N.; Giladi, R. (2013): Performance analysis of Software-Defined Networking (SDN). *21st International Symposium on Modeling, Analysis & Simulation of Computer and Telecommunication Systems*, pp. 1526-1539.

Genge, B.; Haller, P. (2016): A hierarchical control plane for software-defined networks-based industrial control systems. *IFIP Networking Conference and Workshops*, pp. 73-81.

Ghahramani, Z. (2001): An introduction to hidden Markov models and Bayesian networks. *International Journal of Pattern Recognition & Artificial Intelligence*, vol. 15, no. 1, pp. 9-42.

Goldenberg, N.; Wool, A. (2013): Accurate modeling of Modbus/TCP for intrusion detection in SCADA systems. *International Journal of Critical Infrastructure Protection*, vol. 6, no.2, pp. 63-75.

Han, S.; Xie, M.; Chen, H. H.; Ling, Y. (2014): Intrusion detection in cyber-physical systems: techniques and challenges. *IEEE Systems Journal*, vol. 8, no. 4, pp. 1049-1059.

Kalman, G. (2015): Security implications of software defined networking in industrial control systems. *10th International Multi-conference on Computing in the Global Information Technology*, pp. 17-22.

Khalili, A.; Sami, A. (2015): SysDetect: A systematic approach to critical state determination for industrial intrusion detection systems using apriori algorithm. *Journal of Process Control*, vol. 32, no. 11, pp. 154-160.

Knijff, R. M. (2014): Control systems/SCADA forensic, what's the different? *Digital Investigation*, vol. 11, no. 3, pp. 160-174.

Kohlschein, C. (2013): An introduction to hidden Markov model. http://www-lti.informatik.rwth-aachen.de/lehre/PRICS/WS2006/kohlschein.pdf.

Kwon, Y. J.; Kim, H. K.; Lim, Y. H.; Lim, J. I. (2015): A behavior-based intrusion detection technique for smart grid infrastructure. *IEEE Eindhoven PowerTech*, pp. 1-6.

Linda, O.; Vollmer, T.; Manic, M. (2009): Neural network based intrusion detection system for critical infrastructures. *International Joint Conference on Neural Networks*, pp. 1827-1834.

Liu, Y.; Liu, J.; Liu, T.; Guan, X.; Sun, Y. (2013): Security risks evaluation toolbox for smart grid devices. *ACM SIGCOMM Conference*, pp. 479-480.

NCCIC/ICS-CERT (2016): NCCIC/ICS-CERT year in review (2015). https://ics-cert.us-cert.gov/Year-Review-2015.

NCCIC/ICS-CERT (2017): NCCIC/ICS-CERT year in review (2016). https://ics-cert.us-cert.gov/Year-Review-2016.

Ozcelik, I.; Brooks, R. R. (2016): Cusum-entroy: An efficient method for DDoS attack detection. *4th International Istanbul Smart Grid Congress and Fair*, pp. 1-5.

Padgette, J.; Scarfone, K.; Chen, L. (2013): Guide to industrial control systems (ICS) security. http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-82.pdf.

Ponomarev, S.; Atkison, T. (2016): Industrial control system network intrusion detection by telemetry analysis. *IEEE Transactions on Dependable and Secure Computing*, vol. 13, no. 12, pp. 252-260.

Sainz, M.; Iturbe, M.; Garitano, I.; Zurutuza, U. (2017): Software defined networking opportunities for intelligent security enhancement of industrial control systems. *International Joint Conference SOCO'17-CISIS'17-ICEUTE'17*, pp. 577-586.

Schuster, F.; Paul, A.; Rietz, R.; Koenig, H. (2015): Potentials of using one-class SVM for detecting protocol-specific anomalies in industrial networks. *IEEE Symposium Series on Computational Intelligence*, pp. 83-90.

Silva, E. G. D.; Silva, A. S. D.; Wickboldt, J. A.; Smith, P.; Granville, L. Z. et al. (2016): A one-class NIDS for SDN-based SCADA systems. *IEEE 40th Annual Computer Software and Applications Conference*, pp. 303-312.

Ten, C. W.; Manimaran, G.; Liu, C. C. (2010): Cybersecurity for critical infrastructures: attack and defense modeling. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 40, no. 4, pp. 853-865.

Valdes, A.; Cheung, S. (2009): Communication pattern anomaly detection in process control systems. *IEEE Conference on Technologies for Homeland Security*, pp. 22-29.

Wan, M.; Shang, W.; Kong, L.; Zeng, P. (2017): Content-based deep communication control for networked control system. *Telecommunications Systems*, vol. 65, no. 1, pp. 155-168.

Wan, M.; Shang, W.; Zeng, P. (2017): Double behavior characteristics for one-class classification anomaly detection in networked control systems. *IEEE Transactions on Information Forensics and Security*, vol. 12, no.12, pp. 3011-3023.

Wang, J.; Liu, J.; Yang, S.; Li, D. (2015): Integrated trusted protection technologies for industrial control systems. *IEEE Seventh International Conference on Intelligent Computing and Information Systems*, pp. 418-423.

Wei, M.; Kim, K. (2012): Intrusion detection scheme using traffic prediction for wireless industrial networks. *Journal of Communications and Networks*, vol. 14, no. 3, pp. 310-318.

Zhou, C.; Huang, S.; Xiong, N.; Yang, S.; Li, H. et al. (2015): Design and analysis of multimodel-based anomaly intrusion detection systems in industrial process automation. *IEEE Transactions on Systems Man & Cybernetics Systems*, vol. 45, no. 10, pp. 1345-1360.

Zhu, B.; Sastry, S. (2010): SCADA-specific intrusion detection/prevention systems: A survey and taxonomy. *First Workshop on Secure Control Systems*, pp. 1-16.