International Journal of Safety and Security Engineering
Vol. 10, No. 2, April, 2020, pp. 183-189
Journal homepage: http://iieta.org/journals/ijsse

Intrusion Detection Using Long Short-Term Memory Model for Industrial Control System

Asuka Terai[1*], Tatsuya Chiba[1], Hideyuki Shintani[2], Shoya Kojima[2], Shingo Abe[2], Ichiro Koshijima[2]

[1] Department of Complex and Intelligent Systems, Future University Hakodate, Hokkaido 041-8655, Japan
[2] Nagoya Institute of Technology, Aichi 466-8555, Japan

Corresponding Author Email: aterai@fun.ac.jp

https://doi.org/10.18280/ijsse.100204

Received: 13 February 2019
Accepted: 12 November 2019

Keywords:
intrusion detection system, packet pattern, singular spectrum analysis, long short-term memory model

ABSTRACT

Given the rapid progress of digital technology, systems are increasingly vulnerable to cyber-attacks. Intrusion detection systems (IDS), which monitor an industrial control system (ICS) network traffic and detect suspicious activities, are a necessity for the operation of ICSs. Previous studies argued that packet intervals could ideally be regarded as indicators of the cyber-attacks on ICSs and proposed an intrusion detection methodology relying on packet intervals using singular spectrum analysis (SSA). SSA is a nonparametric spectral estimation method, but it suffers from high computational cost. Thus, in this study, a long short-term memory (LSTM) model was developed based on the packet intervals during steady-state operation, and an intrusion detection method using the LSTM model was proposed. The LSTM model is a recurrent neural network model and can be used for time-series prediction problems. Furthermore, we evaluated the proposed method on a cybersecurity testbed using penetration tests. The results show that the LSTM model performs better than SSA and suggests the possibility of the application of the LSTM model to IDS for various types of plants by adjusting its complexity.

1. INTRODUCTION

Industrial control systems (ICSs) monitor and control complex industrial processes and critical infrastructure (such as power plants, manufacturing, and other social services). Any damage to an ICS entails severe impacts on a vast number of stakeholders. Hence, ICSs were isolated from the internet to remove security vulnerabilities. However, the rapid progress of digital technology demands that ICSs connect directly or indirectly to the Internet because of replacing specialized hardware and software with widely available and low-cost Internet Protocol (IP) devices. Given the network extendibility of such ICS components, the system becomes vulnerable to cyber-attacks [1]. In fact, cyber-attacks targeting ICSs have been known to interfere with plant/facility operations. A cyber-attack in 2010 ruined almost one-fifth of Iran's nuclear centrifuges and caused substantial damage to Iran's nuclear program [2]. Additionally, Ukraine's power grid was attacked in 2015 and 2016 [3]. Therefore, intrusion detection systems (IDSs), which can be used to monitor ICS network traffic, are a necessity to protect ICSs against cyber-attacks.

IDSs monitor a network traffic for suspicious activity and alert system or network administrators when such activity is detected. IDSs are classified into two types: anomaly-based and signature-based IDSs. Signature-based IDSs detect attacks by searching for specific patterns, such as byte sequence in networks created by malware. Although signature-based IDSs can protect ICSs from known attacks registered in the IDS database, they cannot screen out an unregistered attack. On the other hand, anomaly-based IDSs detect attacks using machine learning to create a model of steady-state activity. They monitor network traffic by conducting comparisons against the model. Thus, they can detect previously unknown attacks [4].

Many machine-learning techniques have been used to improve the accuracy of detection of anomaly activation for IDSs. They include k-nearest neighbor (kNN) methods (e.g., [5, 6]), neural networks (NNs) (e.g., [7, 8]), support vector machines (SVMs) (e.g., [7-10]), random forests [11], naive Bayes methods (e.g., [12, 13]), and time series association data mining [14]. There are two types of machine-learning techniques: supervised machine-learning and unsupervised machine-learning. Supervised machine-learning techniques generally require a labelled dataset to discriminate anomaly activation from normal one. A previous IDS for ICSs used a dataset obtained by penetration tests to create a discriminant model [10]. The discriminant criteria of the model reflected characteristics of the dataset, but there is a possibility that the model may fail to detect new cyber-attacks whose characteristics are not easily discernible despite penetration tests. Namely, the discriminant model cannot detect new cyber-attacks whose characteristics are completely different from attacks in the dataset obtained by penetration test. In contrast, unsupervised machine-learning techniques infer the function of a hidden structure. When an ICS network during steady-state exhibits a particular pattern, the techniques can infer the hidden structure of the pattern and detect anomalous behavior by comparing the pattern during steady-state. Matta et al. [15] demonstrated that packet intervals during steady-state have a particular pattern and cyber-attacks may disturb this pattern. Furthermore, a certain type of periodicity was observed in time-series packet intervals using a testbed, and an IDS was proposed using singular spectrum analysis (SSA) by searching for the disturbance [16].

SSA is a nonparametric spectral estimation method for

time-series analysis and decomposes time-series data into a sum of components to detect a change point [17]. However, SSA suffers from high computational cost because it has to decompose every time-series data. Similarly, some supervised machine-learning methods are used to forecast time-series data and applied to anomaly detection in time series. They include autoregressive models [18], recurrent neural networks (RNNs) [19], and the long short-term memory (LSTM) model [20]). The models are trained using a dataset during steady-state operation to infer the function of the hidden structure. Namely, they do not require a labeled dataset.

LSTM overcomes the vanishing gradient problem in parameter estimation experienced by RNN and appears to perform better than its counterparts [20]. In this study, a LSTM model was developed based on packet intervals during steady-state and an intrusion detection method using the LSTM model was proposed for ICSs. Furthermore, we evaluated the proposed method using penetration tests on a cybersecurity testbed [10, 15, 16] and compared the performance of the proposed method to that of a previous method using SSA.

2. INTRUSION DETECTION SYSTEMS

2.1 Packet intervals

Typical ICS networks use IP communications between the object linking and embedding (OLE) for process control (OPC) server and single loop controller (SLC) (programmable logic controller (PLC)). ICS communication transfers packets specified as industrial control protocols, such as Modbus/TCP [21], at specific time intervals. Modbus/TCP packets to the target machine are represented as $\{p_0, p_1, p_2, \ldots, p_n\}$ and the time stamp of the ith packet p_i is represented as t_i. The packet intervals d_i are defined as differences between the time stamps t_i and t_{i-1} as

$$d_i = t_i - t_{i-1}. \tag{1}$$

A previous work [15] suggested that packet intervals $\{d_1, d_2, \ldots, d_n\}$ reflect the characteristics of packets in a typical ICS network and exhibit a type of periodicity because they are forced by the activities of a plant to produce a specific type of periodicity.

2.2 Singular spectrum analysis

A previous work [16] used SSA to detect anomalies in time-series of packet intervals. The analysis constructs the corresponding subspaces for matrices (trajectory and test matrices) defined using lagged time-series packet intervals and computes the distances between these subspaces as change scores to detect the change-point (Figure 1).

2.3 Long short-term memory model for detecting structural change

The LSTM model is composed of LSTM units consisting of a cell, an input gate, an output gate, and a forget gate [22]. The cell in LSTM units remembers values over certain time intervals, and the three gates control the flow of information into and out of the cell. The models are adequate for predictions of time series data because they can learn long-term dependencies of the data. In this study, we proposed a model that predicts the ith time interval d_i from the immediately preceding sliding window ($d_{i-1}, d_{i-2}, \ldots, d_{i-M}$) (Figure 2).

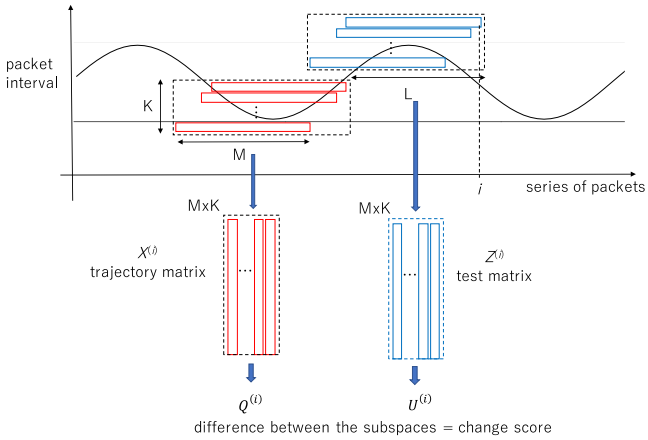

Figure 1. Estimation change scores using SSA (modified after [16])

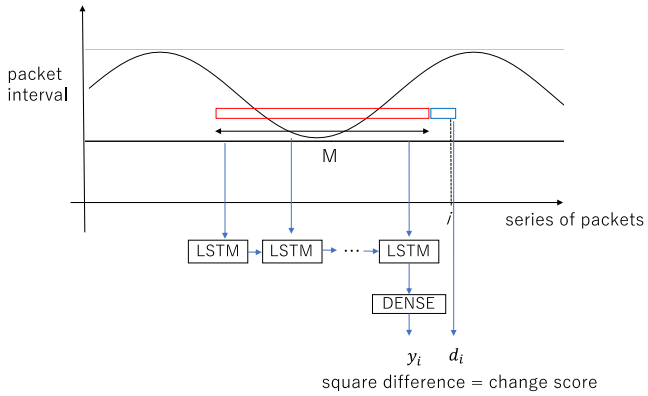

Figure 2. Estimation change scores using LSTM

There are many types of LSTM models: many-to-one, one-to-many, and many-to-many. To predict a time interval immediately after the time interval sequence, the many-to-one model was used. In this research, the LSTM model was composed of a sequential input layer followed by one LSTM layer and dense output layer with the hyperbolic tangent function.

The output in the LSTM layer is computed using the following equations:

$$a_t = \sigma(W_a d_t + R_a h_{t-1} + b_a), \tag{2}$$

$$z_t = tanh(W_z d_t + R_z h_{t-1} + b_z), \tag{3}$$

$$f_t = \sigma\left(W_f d_t + R_f h_{t-1} + b_f\right), \tag{4}$$

$$o_t = \sigma(W_o d_t + R_o h_{t-1} + b_o), \tag{5}$$

$$c_t = a_t \circ z_t + f_t \circ c_{t-1}, \text{ and} \tag{6}$$

$$h_t = o_t \circ \tanh(c_t). \tag{7}$$

where, $i - M \le t \le i - 1$. $tanh$ is the hyperbolic tangent function, σ refers to the sigmoid function, and $\circ$ indicates the element-wise product. $h_t \in \mathbb{R}^N$ ($h_{i-M-1} = 0$) denotes the

hidden state vector, which is also regarded as the output vector of the LSTM unit, and N indicates the number of hidden nodes. $W_* \in \mathbb{R}^{N\times 1}$, $R_* \in \mathbb{R}^{N\times N}$, $b_* \in \mathbb{R}^N$ are weight matrices and bias vector parameters.

In the dense output layer, the predicted ith time interval y_i is computed using the hyperbolic tangent function as follows:

$$y_i = tanh(W_d h_{i-1} + b_d) \tag{8}$$

where, $W_d \in \mathbb{R}^{1\times N}$, $b_d \in \mathbb{R}^1$ are the weight matrices and bias vector parameters in the dense output layer.

The parameters of the LSTM model (W_*, R_*, b_*) were estimated using a training dataset during steady-state operation to minimize the LSTM model's square error. Because the model was trained using the data set during steady-state operation, the model estimates time intervals similar to those during steady-state operation. Certain anomalous behaviors interfere with the interval pattern during steady-state operation and it results in significant differences between the estimated interval and real interval values. Thus, the squaring difference between the estimated and real interval values is defined as a change score at the ith packet:

$$s_{LSTM}(i) = (y_i - d_i)^2 \tag{9}$$

The change score threshold for intrusion detection was set based on the change scores for the dataset during steady-state operations. In a previous work [16] used SSA, it was hypothesized that the change scores during steady-state operations would obey a normal distribution, and the top 0.05% of change scores would indicate suspicious activities. The LSTM model is more sensible than SSA. Therefore, the maximum was used instead of the average as a reference, and the change score threshold was defined as follows:

$$\theta_{LSTM} = \max_{LSTM} + 3.29 * sd_{LSTM}, \tag{10}$$

where, max_{LSTM} and sd_{LSTM} indicate the maximum and standard deviation of the change scores for the dataset during steady-state operations for training. When the ith change score $s_{LSTM}(i)$ exceeds the threshold, the system estimates the ith packet p_i is an attack packet.

3. EVALUATION ENVIRONMENT

The proposed method was evaluated using datasets obtained from the testbed prepared for previous studies [10, 15, 16].

3.1 Security testbed

In the cybersecurity testbed, water is heated to be circulated between two tanks. Figure 3 shows the piping and instrumentation (P&I) diagram of the testbed. The testbed was equipped with actual control devices and controlled automatically. Yokogawa Digital Indicating Controllers (model number: UT35A and UT32A) were installed for proportional-integral-derivative (PID) control. The controllers are operated using an ICS network.

The ICS network contains three zones: one supervisory zone and two control zones (ICS1/ICS2) (Figure 4). The two control zones have the same structure, which consists of a gateway server, an OPC server, a supervisory control and data acquisition (SCADA) monitor, and SLCs. SLC1 controls the level of tank 1 and monitors the temperature of tank 2. SLC2 controls the inlet from tank 2 and the temperature of tank 1. To capture the OPC packets, a network tap was installed in the ICS-2 network. This configuration was designed by Hashimoto et al. [23].

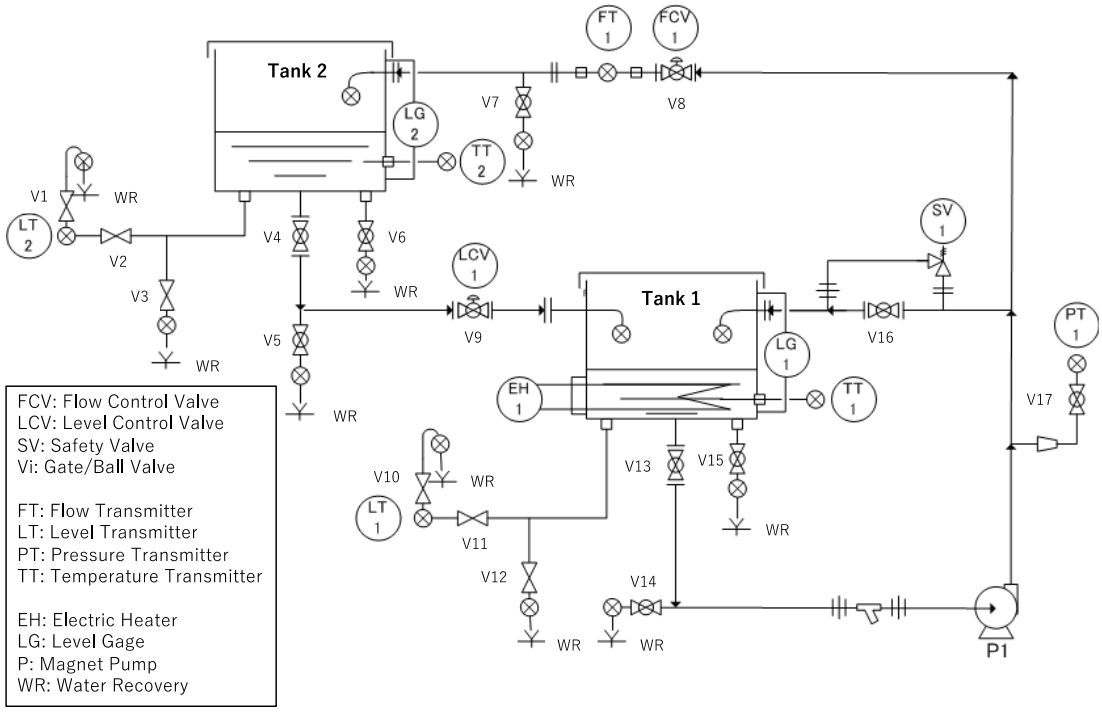

Figure 3. P&I diagram of the cybersecurity testbed used in this study [16]

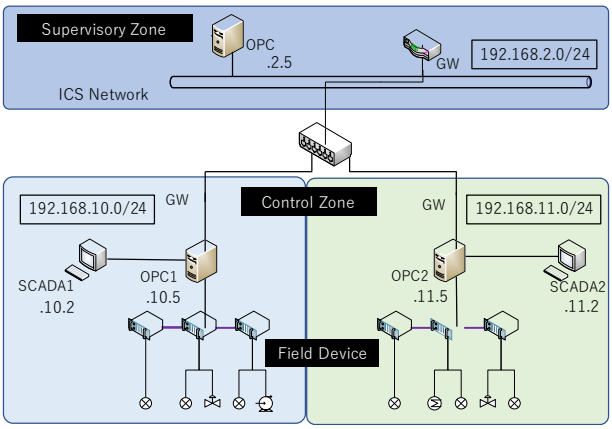

Figure 4. ICS network diagram of the cybersecurity testbed used in this study [16]

3.2 Evaluation packets

Table 1. Datasets used for steady-state operations and penetration test attacks

Dataset	No. of packets (Modbus/TCP)	Capture period (s)	No. of attack packets
Steady-state (for training LSTM)	1472	192.6	-
Steady-state (for valid LSTM)	543	70.9	-
Steady-state (for test)	628	82.22	-
Reading registers	708	92.26	3
Finding unit IDs	1812	834.9	254
Reading coils	502	65.28	2

We used datasets similar to those used in a previous study [16] for evaluation. At steady-state operation, the SCADA terminals monitor the OPC server, which collects and exchanges the process data to circulate the water in tanks 1 and 2 at constant levels. The packets to the controllers in the ICS-2 network were captured using the popular cross-platform packet-capturing program, Wireshark (https://www.wireshark.org). The purpose of the penetration test was to crack the target OPC2 and tamper with the configuration file using the Metasploit Framework (Rapid7) attack tool (https://www.metasploit.com). The penetration test was conducted for three types of cyber-attacks: reading registers, finding unit IDs, and reading coils. Table 1 presents the details of the datasets used for both steady-state operations and penetration test attacks.

4. EVALUATION RESULTS

4.1 Estimation of cycle period of packet intervals during steady-state operations

To estimate the periodicity of the packet intervals during steady-state operations, the autocorrelation coefficient with k gap (r_k) is computed as follows:

$$Cov_k = \frac{1}{n}\sum_{t=k+1}^{k+n}(d_t - \mu_d)(d_{t-k} - \mu_d) \tag{11}$$

$$r_k = \frac{Cov_k}{Cov_0} \tag{12}$$

where, μ_d indicates the average value of time intervals d_t $(k+1 \le t \le k+n)$. Autocorrelation coefficient is the correlation coefficient between a given time series and a lagged version of itself over successive time intervals and represents the degree of similarity between them. The autocorrelation coefficients for the steady-state operation dataset when the k gap is a multiple of 8 exceed 0.8 although the other values are less than 0.3. According to the results, the cycle period during the packet intervals during steady-state operation was estimated as 8 packets.

4.2 Evaluation criteria

We used three types of criteria to evaluate the system: time required to detect the first attack packet, maximum time difference based on real attack packets, and maximum time difference based on estimated attack packets. The two types of time differences based on real attack and estimated attack packets are shown in Figure 5.

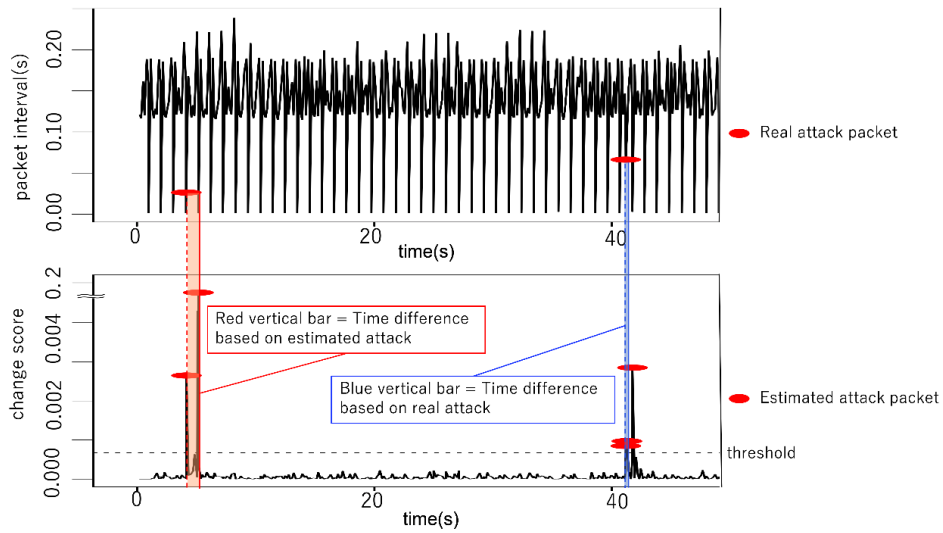

Figure 5. Two types of time differences between the real attack packet and estimated attack packet. The upper panel shows packet intervals, and the lower panel shows change scores

Table 2. Detection results for three evaluation criteria

Method	Dataset	Time required to detect first attack packet (s)	Maximum time difference: real attack (s)	Maximum time difference: estimated attack (s)
SSA	Reading registers	3.09	3.09	6.48
	Finding unit IDs	0.79	0.79	7.10
	Reading coils	0.85	0.85	6.80
LSTM (one hidden node)	Reading registers	0	0	1.20
	Finding unit IDs	0	0.08	1.60
	Reading coils	0.07	0.07	0.53
LSTM (two hidden nodes)	Reading registers	0	0.56	1.07
	Finding unit IDs	0	2.76	8.00
	Reading coils	0	0.53	0.53
LSTM (three hidden nodes)	Reading registers	0	0.07	1.07
	Finding unit IDs	0	2.04	1.60
	Reading coils	0.22	0.22	1.16

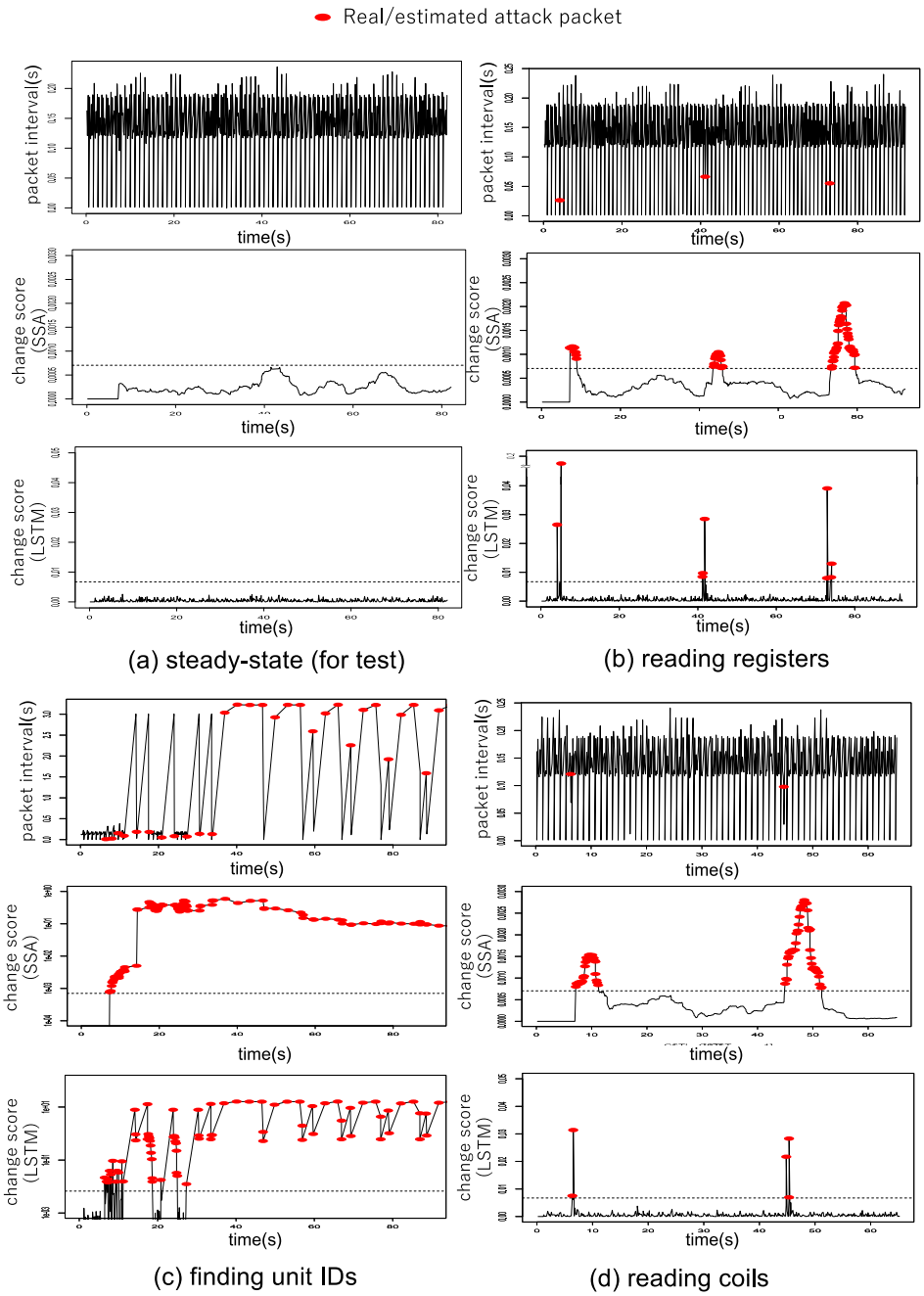

Figure 6. Packet intervals and change scores. The upper panels show packet intervals, and the central and lower panels show respectively change scores using SSA and LSTM. (a) steady-state operation, (b)-(d) three types of cyber-attacks: reading registers, finding unit IDs, and reading coils

The system needs to detect a series of anomaly activations as soon as possible. The time required to detect the first attack packet is computed as the time difference between the first attack packet and the first estimated attack packet by the system. Furthermore, the system must detect not only the first attack packet but also all attack packets. The maximum time difference based on the real attack packet refers to the maximum time differences between a real attack packet and the closest estimated attack. The criterion indicates the maximum time needed to detect every attack packet. In some cases, the system incorrectly estimates normal packets as attack packets, thus causing false alerts. Therefore, we used the maximum time differences between an estimated attack and the closest real attack packet (the maximum time difference based on the estimated attack). When the system estimates a cyber-attack during a period without a real attack and produces a false alert, the criterion takes a higher value.

4.3 Parameter settings

The LSTM model was trained using a training dataset during steady-state operation with the open-source neural network library Keras [24]. The number of hidden nodes N was optimized. A model with more hidden nodes has more parameters and is more complex. Thus, the number was changed from one to three. The sliding window size was fixed at $M = 8$ according to the estimated cycle period.

4.3.1 Detection results of the systems

The results of the evaluation criteria for the demonstration of the systems' detection ability are presented in Table 2. To compare the proposed system to the previous system using SSA [16], Table 2 also presents the detection results of the previous system.

The LSTM with one hidden node produces the best performance. The system requires less than 1 s to detect the first attack packet. Although the previous system using SSA requires more than 3 s to detect the first attack packets for reading registers, the proposed system using LSTM with one hidden node could detect it within less than 1 s. Furthermore, for all types of attacks, both the maximum time differences of the system using LSTM with one hidden node are shorter than 2 s, although the maximum time differences based on the estimated attack for the system using SSA are longer than 6 s.

Interestingly, the system using LSTM with two/three nodes could not show a better performance than the simpler system using LSTM with one node. The number of parameters in the LSTM models with one/two/three hidden nodes, which are estimated using the training dataset, are respectively 14, 35, and 64. The LSTM models with two/three hidden nodes are too sensitive because the models contain too many parameters, causing over-fitting.

Furthermore, both the time series for the packet intervals and the change scores of the systems using SSA and LSTM with one hidden node are shown in Figure 6. These figures show that a cyber-attack interferes with periodic patterns during steady-state operations and indicate that the systems can use the change score to correctly detect this interference.

Once the change scores estimated by the system using SSA exceed the threshold, a longer time period is required for the system to revert to below the threshold state in spite of the absence of attack packets. Similarly, the maximum time differences for the estimated attack packets of the previous system are longer than 6 s. On the other hand, the system using LSTM could estimate the attack packet more satisfactorily, that is, reflecting the timing of the real attack packets. These results suggest that there is less possibility that the proposed system using LSTM makes false alerts.

5. CONCLUSION

In this paper, we proposed an intrusion detection method using LSTM. Additionally, the proposed method was evaluated using pseudo-attacks on the cybersecurity testbed. The previous method using SSA required almost 3 s to detect the first attack packet. On the other hand, the proposed method using LSTM could detect the same in less than 1 s. The maximum time differences of the system using SSA were longer than those of the system using LSTM. In SSA, the trajectory and test matrices need to be defined using time-series packet intervals. Therefore, the method using SSA requires packet intervals during more than two cycles to provide adequate alerts. On the other hand, the method using LSTM estimates the next packet timing from the preceding packets and detects change points based on the difference between the estimated and real timings. Namely, time-series packet intervals during more than one cycle are necessary to provide adequate alerts. Therefore, the proposed method required a shorter time to detect the first attack packet than the previous approach.

According to the evaluation results, the simplest LSTM model with one hidden node showed the best performance. The cybersecurity testbed, which heats water to circulate it between two tanks, automates the simple process. Moreover, the packet intervals in the ICS network show a simple periodicity. The LSTM models with plural hidden nodes did not provide better performance for the datasets obtained from the ICS network of the testbed because of their over-fitting. It is possible that the proposed method can adapt to many types of industrial plants by adjusting the number of hidden nodes.

Typically, the system detects all behaviors that result in changes to the steady-state operation. When ICS operators make intentional changes to steady-state operations, the system flags such changes as anomalous. Therefore, an alert filtering system must be developed for the proposed IDS to ignore changes made by operators. We plan to assess and study both the above-mentioned aspects in a future study.

ACKNOWLEDGMENT

The research was partially supported by the Ministry of Education Science, Sports and Culture, Grant-in-Aid for Scientific Research (A), No. 16H01837.

REFERENCES

[1] Stouffer, K., Lightman, S., Pillitteri, V., Abrams, M., Hahn A. (2014). Guide to industrial control systems (ICS) security. NIST Special Publication 800-82 Revision 2 Initial Public Draft, http://www.gocs.com.de/pages/fachberichte/archiv/164-sp800_82_r2_draft.pdf.

[2] Kelley, M.B. (2013). The Stuxnet attack on Iran's nuclear plant was 'far more dangerous' than previously thought. Business Insider, Online,

http://www.businessinsider.com/stuxnet-was-far-more-dangerous-than-previous-thought-2013-11, accessed on 2019-01-03.
[3] Cherepanov, A., Lipovsky, R. (2017). Industroyer: Biggest threat to industrial control systems since Stuxnet, WeLiveSecurity, https://www.welivesecurity.com/2017/06/12/industroyer-biggest-threat-industrial-control-systems-since-stuxnet/, accessed on 2019-01-03.
[4] Axelsson, S. (2000). Intrusion detection systems: A survey and taxonomy. Chalmers University of Technology, Sweden, Technical Report 99-15, pp. 1-27.
[5] Liao, Y., Vemuri, V. (2002). Use of k-nearest neighbor classifier for intrusion detection. Computer and Security, 21(5): 439-448. https://doi.org/10.1016/S0167-4048(02)00514-X
[6] Wang, K., Stolfo, S.J. (2004). Anomalous payload-based network intrusion detection. Proceedings of Recent Advance in Intrusion Detection (RAID2004), pp. 203-222. https://doi.org/10.1007/978-3-540-30143-1_11
[7] Mukkamala, S., Janoski, G., Sung, A. (2002). Intrusion detection using neural networks and support vector machines. Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN02 (Cat. No.02CH37290), Honolulu, HI, USA, pp. 1702-1707. https://doi.org/10.1109/IJCNN.2002.1007774
[8] Chen, W.H., Hsu, S.H., Shen, H. P. (2005). Application of SVM and ANN for intrusion detection. Computer and Operations Research, 32(10): 2617-2634. https://doi.org/10.1016/j.cor.2004.03.019
[9] Mukkamala, S., Sung, A.H., Abraham, A. (2005). Intrusion detection using an ensemble of intelligent paradigms. Journal of Network and Computer Applications, 28(2): 167-182. https://doi.org/10.1016/j.jnca.2004.01.003
[10] Terai, A., Abe, S., Kojima, S., Takano, Y., Koshijima, I. (2017). Cyber-attack detection for industrial control system monitoring with support vector machine based on communication profile. 2017 IEEE European Symposium on Security and Privacy Workshop, Paris, pp. 132-138. https://doi.org/10.1109/EuroSPW.2017.62
[11] Zhang, J., Zulkern, M. (2006). Anomaly based network intrusion detection with unsupervised outlier detection. 2006 IEEE International Conference on Communications, Istanbul, pp. 2388-2393. https://doi.org/10.1109/ICC.2006.255127.
[12] Koc, L., Mazzuchi, T.A., Sarkani, S. (2012). A network intrusion detection system based on a Hidden Naive Bayes multiclass classifier. Expert Systems with Applications, 39(18): 13492-13500. https://doi.org/10.1016/j.eswa.2012.07.009
[13] Abd-Eldayem, M.M. (2014). A proposed HTTP service based IDS. Egyptian Informatics Journal, 15(1): 13-24. https://doi.org/10.1016/j.eij.2014.01.001
[14] He, W., Hu, G., Yao, X. (2009). Large-scale communication network behavior analysis and feature extraction using multiple motif pattern association rule mining. WSEAS Transactions on Communications, 5(8): 473-482.
[15] Matta, M., Koike, M., Machii, W., Aoyama, T., Naruoka, H., Koshijima, I., Hashimoto, Y. (2015). Industrial control system monitoring based on communication profile. Journal of Chemical Engineering of Japan, 48(8): 619-625. https://doi.org/10.1252/jcej.14we323
[16] Terai, A., Chiba, T., Shintani, H., Kojima, S., Abe, S., Koshijima, I. (2018). Intrusion detection method for industrial control systems using singular spectrum analysis. WIT Transaction Engineering Sciences, 121: 197-208. https://doi.org/10.2495/RISK180171
[17] Moskvina, V., Zhigljavsky, A. (2003). An algorithm based on singular spectrum analysis for change-point detection. Communications in Statistics Simulation and Computation, 32(2): 319-352. https://doi.org/10.1081/SAC-120017494
[18] Guralnik, V., Srivastava, J. (1999). Event detection from time series data. Proceedings of the Sixth ACM SIGKDD International Conference on Data Mining and Knowledge Discovery, pp. 32-42. https://doi.org/10.1145/312129.312190
[19] Kim, J., Kim, H. (2015). Applying recurrent neural network to intrusion detection with Hessian free optimization. WISA 2015 Revised Selected Papers of the 16th International Workshop on Information Security Applications, pp. 357-369. https://doi.org/10.1007/978-3-319-31875-2_30
[20] Malhotra, P., Vig, L., Shroff, G., Agarwal, P. (2015). Long short term memory networks for anomaly detection in time series. Proceedings of European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, pp. 89-94.
[21] Modicon Inc. (1996). Modicon Modbus Protocol Reference Guide. North Andover, USA.
[22] Hochreiter, S., Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8): 1735-1780. http://doi.org/10.1162/neco.1997.9.8.1735
[23] Hashimoto, Y., Toyoshima, T., Yogo, S., Koike, M., Hamaguchi, T., Jing, S., Koshijima, I. (2013). Safety securing approach against cyber-attacks for process control system. Computers and Chemical Engineering, 57: 181-186. https://doi.org/10.1016/j.compchemeng.2013.04.019
[24] Francois, C. (2015). Keras, https://keras.io, accessed on 2018-12-05.