Sensors and Materials, Vol. 36, No. 3 (2024) 1127–1134
MYU Tokyo

S & M 3586

Intrusion Detection in IoT Network Traffic Using Markov Model

I-Hsien Liu,[1,2] Hsiao-Ching Huang,[1,2] Meng-Huan Lee,[1,2] and Jung-Shian Li[1,2*]

[1]Department of Electrical Engineering, National Cheng Kung University,
No. 1, University Road, Tainan City 701401, Taiwan
[2]Institute of Computer and Communication Engineering, National Cheng Kung University,
No. 1, University Road, Tainan City 701401, Taiwan

(Received October 20, 2023; accepted March 19, 2024)

Keywords: IoT, intrusion detection, Markov model, empirical probability law, Hellinger distance

The rapid development of IoT-related technology accelerates the increase in network traffic volume. Hence, network traffic monitoring and analysis are more challenging than before in terms of possible malicious acts due to the immense traffic volume. Being a crucial measure to identify malicious network traffic that might enter a private network, an intrusion detection algorithm has always been an ongoing research topic, owing to its importance in cybersecurity. In this work, we aim to enhance cybersecurity in industrial IoT by performing intrusion detection on the generated network traffic. Therefore, we present a lightweight intrusion detection algorithm based on the Markov model, taking advantage of the source and destination payload lengths, and connection states defined in Zeek logs. We are able to detect intrusive network traffic with high accuracy, using the empirical probability law and Hellinger distance. The pattern similarities between the normal traffic and the cyberattack traffic are the key to our detection method. Lastly, the algorithm is evaluated with ToN_IoT public datasets, followed by an analysis of the experimental results.

1. Introduction

With the advent of communication technologies, IoT applications have gradually taken a crucial role in our daily life. From household to industrial environments, IoT devices are applied to realize the concept of a smart home as well as smart factories. Furthermore, IoT is also extended to other scenarios, for instance, healthcare centers, which significantly benefits patients located in remote areas to access medical assistance. Nevertheless, some concerns arise with regard to the cybersecurity of IoT. For typical households, hackers might steal confidential information from sensor information or surveillance cameras by infiltrating the network. Likewise, for industrial IoT, cyberattacks can cause termination in the manufacturing process and considerably impact profits. Hence, it is essential to implement cybersecurity measures to prevent violations of confidentiality, integrity, and availability in an IoT environment. An intrusion detection system (IDS) is one of the common cybersecurity measures, which introduces network traffic monitoring and analysis[(1)] into detecting cyberattack flows. An

*Corresponding author: e-mail: jsli@cans.ee.ncku.edu.tw
https://doi.org/10.18494/SAM4713

ISSN 0914-4935

anomaly-based IDS may characterize the past normal behavior of the IoT system. By extracting features in terms of this past normal traffic, IDS can detect attack flows by differentiating them with information from normal circumstances. These features are usually derived from attributes and statistical contexts from the network traffic, for instance, packet lengths, packet counts, interarrival time, and connection states. Moreover, we take into account that the Markov analysis is a well-developed theory, and with its simplicity and interpretability, it is applied to various fields of study, specifically network traffic analysis. Hence, the proposed intrusion detection method is based on the concept of a high-order Markov model and empirical probability law (PL).

The intrusion detection algorithm is a widely researched field. Recent work focuses on using machine learning and deep learning algorithms to achieve intrusion detection with high accuracy. Park *et al.*[2] not only adopted an autoencoder for intrusion detection but also utilized a generative adversarial network (GAN) to produce synthetic data to address data imbalance issues found in common AI-based network IDS system design. Nie *et al.*[3] proposed an intrusion detection algorithm to tackle specifically distributed denial of service (DDoS) with deep reinforcement learning to predict past network statistical features. Wu *et al.*[4] introduced big data mining into an intelligent intrusion detection algorithm, which was implemented first by feature selection using a fuzzy rough set, feature extraction using a deep convolutional neural network (DCNN), and also GAN.

The Markov model is a well-developed theory that has been applied to various fields of study, particularly network analysis, owing to its simplicity. Aceto *et al.*[5] predicted traffic from a mobile app using the hidden Markov model and high-order Markov chains. Sha *et al.*[6] introduced a multi-order Markov chain framework into anomaly detection on a cloud server system. Liu *et al.*[7] tackled multimodal prediction for network traffic with the adoption of tensor operations in multivariate multi-order Markov chains.

In this study, by considering the underlying correlation between different attributes through multivariate analysis, we propose an intrusion detection method based on the concept of a high-order Markov model and an empirical PL. The remainder of this paper is organized as follows. In Sect. 2, our proposed method will be thoroughly explained. Experimental results will be illustrated and discussed in Sect. 3 We will conclude in Sect. 4.

2. Proposed Method

Our proposed method is based on the concept of the high-order Markov chain. First, we define the set

$$X = \{X_1, X_2, X_3, \ldots, X_{t-1}, X_t, X_{t+1}, \ldots\}, \tag{1}$$

where the elements are consecutive and each one of them is a random variable that describes the state at time t for the traffic flow. The finite state set for the traffic flow is denoted as

$$S \equiv \{1,2,3,\ldots,I\}, \tag{2}$$

where I is the total number of states.

2.1 Preliminary

For a classical first-order Markov chain, the current state is determined by the preceding state as shown below.

$$\begin{aligned} &\mathrm{P}(X_t = j \mid X_{t-1} = i, X_{t-2} = i_{t-2}, \ldots, X_0 = i_0) \\ &= \mathrm{P}(X_t = j \mid X_{t-1} = i) \\ &= p_{i,j}, \end{aligned} \tag{3}$$

where state $j,\ i,\ i_0,\ldots i_{t-1} \in S$. The transition probability is expressed as

$$p_{i,j} = \mathrm{P}(X_{t+1} = j \mid X_t = i). \tag{4}$$

Note that the temporal homogeneity is assumed, meaning that it does not depend on time t. Hence, the transition probability matrix is represented as

$$P' = \left(p_{i,j}\right), \tag{5}$$

where $\sum_{i=0}^{I} p_{i,j} = 1, P \in \mathbb{R}^{I \times I}$.

In comparison with the classical first-order Markov chain, a k-order Markov chain not only depends on the previous state but also takes into consideration the k preceding states. For instance, a two-order Markov chain takes two preceding states into account:

$$\begin{aligned} &\mathrm{P}(X_t = j \mid X_{t-1} = i, X_{t-2} = i_{t-2}, \ldots, X_0 = i_0) \\ &= \mathrm{P}(X_t = j \mid X_{t-1} = i,\ X_{t-2} = h) \\ &= p_{h,i,j}. \end{aligned} \tag{6}$$

The transition probability matrix is converted into a three-dimensional tensor:

$$P' = \left(p_{h,i,j}\right), \tag{7}$$

where $P' \in \mathbb{R}^{I \times I \times I}$, $\sum_{h=0}^{I} \sum_{i=0}^{I} p_{h,i,j} = 1$.

2.2 Multivariate high-order Markov model with Hellinger distance (MHMMH)

First, we consider a multivariate scenario, where we selected source payload lengths, destination payload lengths, and connection states. Since the source and destination payload lengths of each flow range from 0 to more than 10000 bytes, it is not possible for us to process in such a large number of states by assigning each flow length to a state. Hence, we first apply binning by using K-Means on the concatenated attributes and source and destination payload lengths, to start with. After this, the empirical PL is computed to characterize traffic behavior.[8]

$$\mathbf{\Gamma} = \left(\Gamma_{1,1,1}, \Gamma_{1,1,2}, \ldots, \Gamma_{I,I,I}\right), \tag{8}$$

$$\mathbf{\Gamma}_{h,i,j} = \frac{1}{N_t - 1} \sum_{t=2}^{N_t} 1_{\{X_{t-2}=h\}} 1_{\{X_{t-1}=i\}} 1_{\{X_t=j\}}, \tag{9}$$

where N_t is the total time of the observed traffic and $1_{\{\cdot\}}$ is the indicator function. Note that for clear expressions, multivariate is omitted from the equation for the empirical PL. The idea is that by taking into account the distribution of the current state and the previous two states, we can obtain the overall empirical PL. Using the training data from which the normal traffic data are collected over a long period of time, we can estimate the actual PL $\mathbf{\Gamma}_0$ from the empirical PL. This long-term behavior of the system is also considered as the null hypothesis H_0. For any sample of the testing data $\mathbf{Y}$, the dissimilarity between the computed PL $\mathbf{\Gamma}_{test}$ and PL $\mathbf{\Gamma}_0$ can be determined by applying the Hellinger distance (HD),[5]

$$\boldsymbol{HD}\left(\mathbf{\Gamma}_{test}, \mathbf{\Gamma}_0\right) = \frac{1}{\sqrt{2}} \sqrt{\sum_{h=1}^{I} \sum_{i=1}^{I} \sum_{j=1}^{I} \left(\sqrt{\mathbf{\Gamma}_{h,i,j}^{test}} - \sqrt{\mathbf{\Gamma}_{h,i,j}^{0}}\right)^2}, \tag{10}$$

where $0 \leq HD \leq 1$, with the score 1 meaning the most dissimilar and 0 exactly the same. Given a threshold γ, the hypothesis H_{test} of the testing data rejects the null hypothesis H_0 if and only if $\boldsymbol{HD}(\mathbf{\Gamma}_{test}, \mathbf{\Gamma}_0)$, indicating that the testing samples are determined as cyberattack flows. The pseudocode can be found in Algorithm 1.

3. Experiments and Discussion

3.1 Dataset preparation

We utilize the ToN_IoT datasets[9–14] to evaluate the proposed intrusion detection algorithm, MHMMH, in IoT network traffic. The datasets include IoT/IIoT telemetry data from sensors, operating system data from Windows and Linux systems, and network traffic data collected during normal operations and under different cyberattacks. The network traffic datasets are collected using pcap tools and Zeek logs. Moreover, the testing data include various types of

Algorithm 1
MHMMH algorithm.

Given: Order of the model (m), class number (k), sampling window size (w), threshold (γ), training traffic data ($\boldsymbol{X_{train}}$), and testing traffic data ($\boldsymbol{X_{test}}$) with src_bytes, dest_byes, conn_state.
Initialization
1. Binning $\boldsymbol{k}$ classes using K-Means for the (src_bytes, dest_bytes) pairs in $\boldsymbol{X_{train}}$ and $\boldsymbol{X_{test}}$
2. Obtain $\boldsymbol{\Gamma}_0$ with (8) and (9) from $\boldsymbol{X_{train}}$
3. Select a testing sample of window size $\boldsymbol{w}$ from $\boldsymbol{X_{test}}$
4. Obtain $\boldsymbol{\Gamma_{test}}$ with (8) and (9) using the testing sample
5. Apply (10) to calculate the dissimilarity score (**HD**)
6. **if** $HD(\Gamma_{test}, \Gamma_0) > \gamma$
7. return **TRUE**
8. **else:**
9. return **FALSE**
10. **end**

attack interference, for instance, the DDoS attack, XSS attack, backdoor attack, and man-in-the-middle attack. The algorithm is evaluated with training data consisting of 234928 flows containing only normal traffic flows, and the testing dataset consists of 391043 flows mixed with different cyberattack traffic flows.

3.2 Performance metrics

We apply four common performance metrics[15] to evaluate the proposed intrusion detection algorithm: precision, recall, F1 score, and true negative rate (TNR) for evaluation.

$$Precision = \frac{TP}{TP + FP} \tag{11}$$

$$Recall = \frac{TP}{TP + FN} \tag{12}$$

$$F1 = 2\frac{(Precision)(Recall)}{Precision + Recall} \tag{13}$$

$$TNR = \frac{TN}{TN + FP} \tag{14}$$

True positive (TP) is the correct detection for cyberattack network traffic flows, whereas true negative (TN) is the accurate no detection for normal traffic flows. False positive (FP) shows a wrong detection that the normal traffic flows are classified as cyberattack ones. The opposite is false negative (FN), where cyberattack flows are classified as normal ones. Usually, a model with a higher TNR is seen as having a low false alarm rate. A higher F1 score signifies a better classification overall.

3.3 Results

The experimental results are shown in Table 1, where we compare our method with two other algorithms, multivariate multi-order Markov chain (MMMC)[6] and long short-term memory (LSTM) with an autoencoder. How we determine the final hyperparameters for the MHMMH algorithm is discussed later. The best result is found when we set the threshold ã to 0.80 accompanied by the sampling window size of 50. All experiments are conducted under the order of 3 for the high-order Markov chain. From the results in Table 1, it is clear that MHMMH achieves the best overall outcome, especially in precision, F1 score, and TNR.

To investigate how different hyperparameters affect the performance of the intrusion detection algorithm, we conduct an experiment, where we show how the window size affects the four metrics in Fig. 1. From our observations, we can see that the performance metrics of precision and TNR are better when the window size is larger, whereas recall slightly decreases with a larger window size. As for how different thresholds affect the performance, precision and TNR are also better with a larger threshold. However, recall performs better if the threshold is set to a smaller value. In Table 2, we show how a multivariate method can outperform a univariate choice in our algorithm. In the first row, we only adopt the connection states to perform the intrusion detection algorithm. In comparison with the second row where we apply source and destination payload lengths, we obtain better results in the multivariate case.

3.4 Discussion

Both being Markov chain-based methods, MMMC and our proposed MHMMH take multivariate features into consideration. However, MHMMH considers not only the previous state, but also several preceding states. It also uses a dissimilarity score designed on the basis of empirical PL and HD. These are the reasons why MHMMH performs better for differentiating attack traffic. Moreover, compared with LSTM-AD, MHMMH achieves better results with less computational resources. Moreover, a deep learning algorithm requires a larger dataset and an appropriate preprocessing method to improve its performance.

Table 1
Comparison between different peer methods.

Method	Precision	Recall	F1	TNR
MMMC	0.6343	0.6343	0.6343	0.6664
LSTM-AD	0.8104	1	0.8995	0.3
MHMMH	0.9560	0.9678	0.9619	0.9681

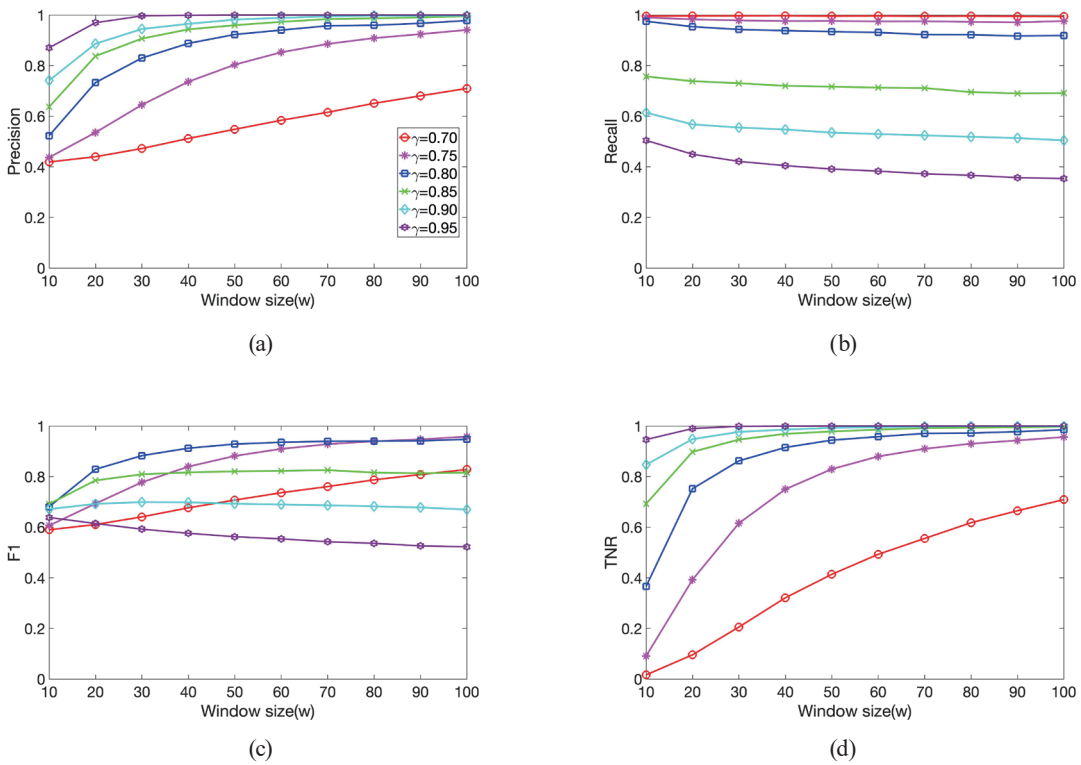

Fig. 1. (Color online) Four performance metrics with different sampling window sizes and thresholds: (a) precision, (b) recall, (c) F1 score, and (d) TNR.

Table 2
Comparison between the univariate and multivariate choices of attributes.

Number of variates	Precision	Recall	F1	TNR
1	0.8979	0.9349	0.9160	0.9239
3	0.9560	0.9678	0.9619	0.9681

4. Conclusions

In this paper, we aimed to enhance cybersecurity in industrial IoT through intrusion detection on the generated network traffic. We proposed a lightweight intrusion detection algorithm based on the Markov model utilizing multiple attributes including the source and destination payload lengths and connection states defined in Zeek logs. The algorithm can detect intrusive network traffic with high accuracy, which depends on obtaining the pattern similarities between the normal traffic and the cyberattack traffic using HD. In the experiments, we discussed how different hyperparameters affect the performance of the algorithm, from which we can conclude that the balance between recall and TNR is a key metric to differentiate between normal and cyberattack network traffic. In the future, we plan to extend the algorithm to tackle more attack scenarios and other environment settings.

Acknowledgments

This work was supported by the National Science and Technology Council (NSTC) of Taiwan under contract numbers NSTC 112-2634-F-006-001-MBK and NSTC 111-2218-E-006-079-.

References

1 I. Lohrasbinasab, A. Shahraki, A. Taherkordi, and A. Delia Jurcut: Trans. Emerging Telecommun. Technol. **33** (2021). https://doi.org/10.1002/ett.4394
2 C. Park, J. Lee, Y. Kim, J. -G. Park, H. Kim, and D. Hong: IEEE Internet Things J. **10** (2023) 2330. https://doi.org/10.1109/JIOT.2022.3211346
3 L. Nie, W. Sun, S. Wang, Z. Ning, J. J. P. C. Rodrigues, Y. Wu, and S. Li: IEEE Trans. Green Commun. Networking **5** (2021) 778. https://doi.org/10.1109/TGCN.2021.3073714
4 Y. Wu, L. Nie, S. Wang, Z. Ning and S. Li: IEEE Internet Things J. **10** (2023) 3094. https://doi.org/10.1109/JIOT.2021.3112159
5 G. Aceto, G. Bovenzi, D. Ciuonzo, A. Montieri, V. Persico, and A. Pescapé: IEEE Trans. Netw. Serv. Manage. **18** (2021) 907. https://doi.org/10.1109/TNSM.2021.3051381
6 W. Sha, Y. Zhu, M. Chen, and T. Huang: IEEE Trans. Cloud Comput. **6** (2018) 401. https://doi.org/10.1109/TCC.2015.2415813
7 H. Liu, L. T. Yang, J. Chen, M. Ye, J. Ding and L. Kuang: IEEE Trans. Netw. Serv. Manage. **16** (2019) 828. https://doi.org/10.1109/TNSM.2019.2934133
8 J. Zhang and I. C. Paschalidis: IEEE Trans. Signal Process. **66** (2018) 589. https://doi.org/10.1109/TSP.2017.2771722
9 T. M. Booij, I. Chiscop, E. Meeuwissen, N. Moustafa, and F. T. H. d. Hartog: IEEE Internet Things J. **9** (2022) 485. https://doi.org/10.1109/JIOT.2021.3085194
10 N. Moustafa: Sustainable Cities Soc. **72** (2021). https://doi.org/10.1016/j.scs.2021.102994
11 J. Ashraf, M. Keshk, N. Moustafa, M. Abdel-Basset, H. Khurshid, A. D. Bakhshi, and R. R. Mostafa: Sustainable Cities Soc. **72** (2021) 1. https://doi.org/10.1016/j.scs.2021.10304
12 A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, and A. Anwar: IEEE Access **8** (2020) 165130. https://doi.org/10.1109/ACCESS.2020.3022862
13 N. Moustafa, M. Keshky, E. Debiez, and H. Janicke: 2020 IEEE 19th Int. Conf. Trust, Security and Privacy in Computing and Communications (TrustCom). (IEEE, 2020) 848.
14 N. Moustafa, M. Ahmed, and S. Ahmed: 2020 IEEE 19th Int Conf. Trust, Security and Privacy in Computing and Communications (TrustCom). (IEEE, 2020) 727.
15 S. Fathi-Kazerooni and R. Rojas-Cessa: IEEE Trans. Network Sci. Eng. **8** (2021) 3392. https://doi.org/10.1109/TNSE.2021.3113656