A Lightweight Anomaly Detection Method for Industrial Processes Based on Event Correlation Behavior

Jianzhen Luo, Yan Cai(✉), Jun Cai, and Wanhan Fang

Guangdong Polytechnic Normal University, Guangzhou, Guangdong, China
luojz@gpnu.edu.cn, 1744409360@qq.com

Abstract. In recent years, the industrial Internet has faced severe threats of production process attacks. By injecting malicious commands or data into the application layer protocols, the attackers change the industrial control flow and disrupt the normal production process, leading to equipment failures and even production accidents. From a network perspective, the traffic of production process attacks does not violate the syntax of communication protocols. However, from the industrial system point of view, the production process attack violates some restrictive rules or physical laws of the industrial production process. This paper proposes a lightweight industrial process anomaly detection method based on event-associated behavior for the characteristics of industrial production process attacks, adopts HsMM with low model complexity to model the state data of field devices in the industrial production process, analyzes the temporal behavioral evolution law of the production process, constructs the temporal behavioral model of the production equipment, and then constructs a lightweight production process anomaly detection method based on behavioral offset.

Keywords: Anomaly detection · Behavioral profiling · Industrial security · Hidden semi-Markov model (HsMM)

1 Introduction

With the accelerated integration of new-generation information technologies such as cloud computing, artificial intelligence, and the Internet of Things(IoT) with manufacturing technologies, industrial control systems(ICS) have shifted from closed and independent to open and shared, from standalone to interconnected, and from automated to intelligent. Open and interactive industrial environments pose security risks to devices, networks, controls, and data. In industrial production, there are production process attacks that disrupt normal production processes by injecting malicious commands or data into application layer protocols, altering the industrial control flow, and causing production accidents. Therefore, there is an urgent need to apply anomaly detection(AD) methods to monitor the safe operation of systems [1].

J. Cai et al. (Eds.): SPNCE 2023, LNICST 525, pp. 175–187, 2025.
https://doi.org/10.1007/978-3-031-73699-5_12

Anomaly detection refers to the detection of patterns in data that do not conform to expected behavior. Exceptions themselves can have a positive or negative nature, depending on their context and interpretation. The importance of anomaly detection is due to the transformation of anomalies in data into important actionable information in various application fields. Correctly detecting such abnormal information enables decision makers to take action on the system in order to fully respond, avoid, or correct situations related to it [2].

During industrial anomaly detection, attackers can inject malicious instructions into the application layer to alter device configuration information, thereby disrupting normal industrial production processes. For example, "Stuxnet" uses highly complex malicious code and multiple zero day vulnerabilities as attack weapons, targeting uranium centrifuges, thereby changing the centrifuge speed and causing overpressure to cause batch damage to the centrifuges [3]. Due to the fact that malicious instructions do not violate the syntax of communication protocols and do not differ from normal communication modes, such attacks often cannot be judged based on industrial control protocol specifications or network traffic characteristics [4].

Therefore, this paper proposes a lightweight industrial process anomaly detection method based on event-related behavior for the characteristics of industrial production process attack. The HsMM model [5] with low model complexity is used to establish a normal sensor time series data model to describe the normal dynamic changes of field equipment state data in the industrial production process and analyze the law of state transfer in the production process. When detecting, the real-time collected sensor data are input into the model, and whether there is any abnormality is judged by calculating the absolute fluctuation derivative of the average entropy. At the same time, this paper also proposes a segmentation algorithm with the time series trend change point as the cut point to describe the equipment state in an intuitive and feasible way. Our major contributions in this paper include the following:

1. A production process event characterization and mining method is proposed to differentiate production event states based on the trends of time series influence factors, which can accurately detect production process attacks that violate the restriction rules or physical laws of industrial production processes.
2. An event-based data-driven industrial production process modeling approach is proposed to construct a time-series behavioral model of production equipment by simplifying the production process events into a hidden semi-Markov process.
3. A lightweight industrial process anomaly detection method based on event-related behaviors is proposed to detect anomalies in industrial processes efficiently, lightly and in real time.

The rest of the paper is organized as follows. Section 2 provides a brief overview of recent related work. Section 3 describes the details of the selected model and the anomaly detection method. Section 4 elucidates the implementation of the anomaly detection method and the results of the experiments with.

Finally, Sect. 5 summarizes the work of this paper and indicates future research directions.

2 Related Work

Industrial control system anomaly detection techniques are methods used to detect and identify anomalous behavior or events in an industrial control system [6]. Depending on the principle, they can be categorized as mechanism-driven and data-driven.

2.1 Mechanism-Driven Anomaly Detection

Mechanism-driven anomaly detection is an approach based on the working principle and physical model of a system, which describes the characteristics and behaviors of a system under normal operating conditions by building a mechanism model of the system based on its working principle, components and interrelationships. It can provide in-depth understanding and detailed analysis of the system state, and can respond quickly when abnormal events occur.

Astillo [7] proposed a distributed anomaly detection method based on the physical state of heterogeneous embedded IoT nodes, which can identify the misbehavior of heterogeneous embedded IoT nodes in a closed-loop smart greenhouse agricultural system. Yang [8] proposed an anomaly detection of an IoT system based on the fingerprinting of the physical state of the registers, which adopts a Boolean logic to represent the IoT system controller of each register physical state and generates a device fingerprint based on a deterministic finite automata model, thus realizing active detection and passive monitoring based on device fingerprints. He [9] modeled the supercharger efficiency as a key indicator using a thermodynamic mechanism approach, and proposed a hybrid driving model based on thermodynamic mechanism and data. Lv [10] analyzed the device physical state in real time based on the The physical state of the device analyzes the trustworthiness of the device in real time, and implements an industrial control system attack defense and monitoring system based on the trustworthiness of the device. Xie [11] propose a PLC malware detection method based on model checking, which utilizes Satisfaction Mode Theory(SMT) constraints to model the PLC system and generates detection rules based on the Invariant Extraction and Rule Design modes.

Mechanism-driven approaches typically identify anomalies based on known system workings and behavioral patterns. Emerging, unknown anomalies may not be accurately detected, thus requiring constant updating of the rule set to maintain detection capabilities for the latest attack methods.

2.2 Data-Driven Anomaly Detection

Data-driven anomaly detection is an approach to detecting and recognizing anomalies based on data analysis and statistical methods [12]. It does not rely on

a working or physical model of the system, but rather describes the distribution of normal data by modeling the normal data of the system. The data-driven anomaly detection approach does not rely on a deep understanding and modeling of the system and is suitable for systems that are complex and difficult to model accurately. It automatically learns the characteristics of normal data and performs anomaly detection when there are deviations from normal behavior.

Zolanvari [13] used a machine learning algorithm model to analyze network traffic features to detect attacks such as backdoor, command injection, and SQL injection. ANNIE [14] proposed an intelligent anomaly detection extensive deep clustering(ODC) algorithm for network traffic analysis and optimization by combining AutoEncoder and BIRCH clustering algorithms for detecting known and unknown malicious network traffic. Nguyen [15] use a federated learning approach to cluster unlabeled network traffic data to construct behavioral profiles for specific device types to detect anomalies. Matouek [16] use a probabilistic automaton to create normal profiles of the communication patterns of an industrial control system to identify the application layer of the industrial Internet of security attacks and unknown threats. Akpinar [17] propose an EtherCAT network anomaly detection technique to detect device-level periodicity and its offset stateby using protocol-specified operations and fields.

In recent years, anomaly detection in industrial control systems is often combined with machine learning algorithmic models. Compared with the lightweight production process anomaly detection method based on HsMM proposed in this paper, deep learning-based anomaly detection techniques are able to automatically learn feature representations and adapt to data in different domains, but they usually require a large amount of labeled data and a large amount of computational resources for training and inference, and they cannot be used to detect new types of attacks. Deep learning models are considered black-box models and it is difficult to explain their decision-making process and anomaly detection results.

3 Proposed Model

Production process attacks can often be manifested in abnormal fluctuations of data over time. HsMM is able to capture temporality and state transitions in time-series data to better reflect the dynamic changes of data.

Therefore, this paper constructs a transfer model of state based on HsMM. As shown in Fig. 1, the whole architecture is divided into four modules in total. In the preprocessing stage, the sensor temporal sequences are smoothed and cut, and then the feature extraction of sub-sequences is carried out, and then the sequences are subjected to DBSCAN clustering, so as to obtain the cluster corresponding to each sub-sequence. During training, based on the obtained state sequences as well as the set of sub-sequences, the subsequence state transfer process is mapped to the HsMM process with variable duration, and the parameters of the normal timing behavior model are updated by an unsupervised parameter estimation algorithm. Finally, the average logarithmic entropy is calculated based on the observed signals to do abnormal behavior detection.

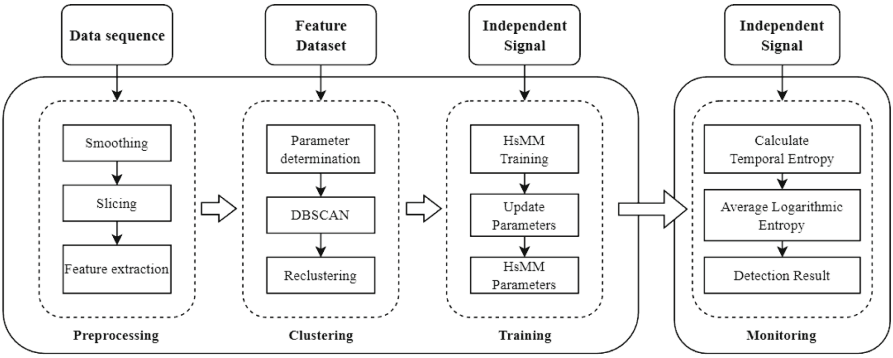

Fig. 1. Monitoring architecture: Abnormal detection of industrial control systems based on production behavior.

3.1 Event Characterization and Mining

Prior to modeling individual sensor data, preprocessing and clustering of the data is required in order to improve the quality and usability of the data to better suit the subsequent analysis and modeling tasks. Among the preprocessing there are three steps: data smoothing, data cutting and feature selection and transformation.

In order to make the data smoother and more reliable, the time series are smoothed using the Savitzky-Golay filter to obtain a data series that retains the trend of the data and has less noise, and then cut to obtain multiple subseries using the trend change point of the series as the cut point. The subseries obtained by cutting often have a single trend: increasing, decreasing or flat, for this characteristic, we choose to use the $< k, average, width >$ as a set of eigenvalues of the subsequence, where k represents the average slope of the subsequence, and *average* and *width* represent the mean and length of the subsequence, respectively, to obtain the feature matrix.

DBSCAN is a clustering algorithm based on densely connected regions, which is usually suitable for datasets where the number of clusters cannot be determined in advance and are of low dimensionality, so DBSCAN is chosen for clustering here. By DBSCAN clustering algorithm, the subsequences are divided into different classes, but such clustering algorithms are still not able to cluster the sequences according to the curvilinear trend very well and are unable to represent the continuity of the state.

Therefore, it is necessary to perform another clustering to classify the fuzzy classes into normal classes, and at the same time, the subsequences in each class are split according to the positive and negative slopes, and the neighboring similar classes are merged to obtain the final sequence class. The observation sequences collected by any sensor are segmented according to the trend of parameter changes, theoretically, the subsequences in each class after clustering have similar trends and change intervals, so in the training process, each class is

regarded as a state and the average eigenvalue of each class is taken as the state feature.

3.2 Behavioral Modeling of Production Processes

Throughout the production process, the variation of the observed parameters of the sensors can be considered as a Hidden Semi-Markov process, where the clusters and sequence lengths obtained in the previous stage are the implied states and state durations of the HsMM.

The HsMM allows the underlying process to be a Semi-Markov chain, where the duration or dwell time of each state is variable [18]. The number of observations generated in state j is determined by the state duration $d = \{0, 1, 2, \ldots, D\}$. Let the set of states be $S_j = \{1, 2, \ldots, n\}$, $p_j(d)$ represents the probability of the duration distribution of state j, and $o_{t_1:t_2}$ denotes the sequence of observations within the moments t_1 to moment t_2.

Given the HsMM model $\lambda \triangleq (\pi, A, B, P)$, where the initial state probability is π, A represents the state transfer probability matrix, B denotes the probability distribution matrix of the sequence emitted from the current state, and the probability matrix of the sequence duration distribution is denoted as P, as follows:

$$\begin{aligned} a_{(i,d')(j,d)} &= Pr[S_{[t+1:t+d]} = j | S_{[t-d'+1:t]} = i] \\ &= a_{ij} p_j(d) \end{aligned} \tag{1}$$

$$b_j(o_{t_1:t_2}) = Pr[o_{t_1:t_2} | S_{[t_1:t_2]} = j] \tag{2}$$

Since there are many unknowns and uncertainties in the actual production process, a limited test data set cannot accurately reflect the complexity and diversity of the real environment. Therefore, to avoid the situation of being too absolute, the following definitions are made.

Definition 1. *Events that do not occur in the test data are considered to be small probability events and are assigned a sufficiently small, but not zero, value at the corresponding position in the probability matrix.*

Definition 2. *The ratio of the distances between the observed sequence feature points and the state feature points is regarded as the similarity between the two, and $b_j(d)$ is obtained on this basis.*

Definition 3. *The state duration distribution is assumed to be an equal probability event, i.e. , $p_j(d) = \dfrac{Occurrence_{[d-step:d]}}{Occurrence_{[1:T]}}$, where step denotes the time unit, and T is the total duration.*

On the basis of (1) and (2), the parameter estimation of HsMM can be accomplished by Forward-Backward Algorithms. The forward and backward variables

are defined as follows.

$$\alpha_t(j) \triangleq Pr[S_{t]} = j, o_{1:t}|\lambda] = \sum_{\substack{i \in S \setminus \{j\} \\ d \in D}} \alpha_{t-d}(i) \cdot a_{ij} \cdot p_j(d) \cdot b_j(o_{t-d:t}) \tag{3}$$

$$\beta_t(j) \triangleq Pr[o_{t+1:T}|S_{t]} = j, \lambda] = \sum_{\substack{i \in S \setminus \{j\} \\ d' \in D}} a_{ji} \cdot p_i(d') \cdot b_i(o_{t:t+d}) \cdot \beta_{t+d'}(i, d') \tag{4}$$

In the initialization phase of the model, set $\pi = (\frac{1}{n}, \frac{1}{n}, \ldots, \frac{1}{n})$, and statistical methods and similarity comparisons can be obtained for A, P, B. To iterate over the model parameters, three probability functions are defined:

$$\eta_t(j, d) \triangleq Pr[S_{[t-d:t]} = j, o_{1:T}|\lambda] = \sum_{d \in \mathcal{D}} \sum_{i \in S \setminus \{j\}} \alpha_{t-d}(j) \cdot a_{ij} \cdot p_j(d) \cdot b_j(o_{t-d:t}) \cdot \beta_t(j) \tag{5}$$

$$\xi_t(i, j) \triangleq Pr[S_{t]} = i, S_{t+d]} = j, o_{1:T}|\lambda] = \sum_{d' \in \mathcal{D}} \sum_{d \in \mathcal{D}} \alpha_t(i) \cdot a_{ij} \cdot p_j(d) \cdot b_j(o_{t:t+d}) \cdot \beta_{t+d}(j) \tag{6}$$

$$\gamma_t(j) \triangleq Pr[o_1^T, q_t = j] = \gamma_{t+1}(j) + \sum_{i \in S \setminus \{j\}} [\xi_t(j, i) - \xi_t(i, j)] \tag{7}$$

Then, the model is iteratively optimized using the EM algorithm and the parameters are updated with the following equations:

$$\hat{\pi}_j = \gamma_0(j) / \sum\nolimits_j \gamma_0(j) \tag{8}$$

$$\hat{a}_{ij} = \sum\nolimits_t \xi_t(i, j) / \sum\nolimits_{j \neq i} \sum\nolimits_t \xi_t(i, j) \tag{9}$$

$$\hat{p}_j(d) = \sum\nolimits_t \eta_t(j, d) / \sum\nolimits_d \sum\nolimits_t \eta_t(j, d) \tag{10}$$

The EM algorithm estimates the values of the model parameters and the values of the missing data based on the training data, then re-estimates the parameter values based on the estimated missing data plus the previously observed data, and then iterates repeatedly until the final convergence, thus obtaining the trained HsMM model parameters.

3.3 Production Process Anomaly Detection

Entropy is a concept that measures the uncertainty of information and is often used to measure the degree of chaos or disorder in a data set [19]. Anomalous data

may cause the distribution of a dataset to become more uneven and disordered, thus giving it a relatively high entropy value. Although entropy can be used as a measure of data anomalies, there may be limitations in using entropy alone for anomaly detection.

Absolute Difference is a commonly used statistic to measure the degree of difference between each data point in a data set and its mean. By looking at the change in the derivative of the absolute difference, when anomalous line segments are usually characterized by sharp changes or unusual spikes in the derivative. Therefore, the absolute mean difference of average logarithmic entropy (ALE) is further obtained. Assuming that the current observed sequence is $o_{t-d:t}$, the Absolute difference in entropy(ADE) of the observed sequence is calculated according to the following equation:

$$\begin{aligned} ALE_t &= \ln(Pr[o_1^t|\lambda])/t \\ &= \ln(\sum_{j\in S} \alpha_t(j))/t \end{aligned} \tag{11}$$

$$ADE_t = \left| \frac{ALE_t}{\sum_{t\in T} ALE_t} \right| \tag{12}$$

In order to determine the abnormality more quickly and accurately, the middle interval of the normal data set is taken after modeling, and ADE_{normal} is obtained according to the formula (12), which is used to determine the threshold to determine whether the data is abnormal or not.

In real industrial production, when one device is attacked, it may have an impact on the operational status of the whole system. For example, if the attack results in the tampering or interruption of the control signal of one device, it may cause other devices to fail to operate normally. Therefore, indirect attacks need to be considered when detecting them. If the data of other devices is abnormal immediately after the attack, it may be a direct effect, and if there is a certain time delay or time interval, it may be an indirect effect.

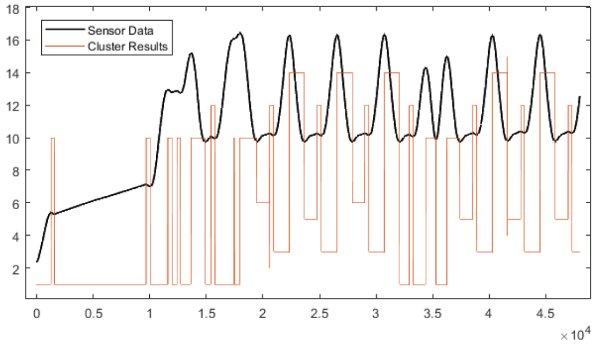

Fig. 2. Clustering effect: SWaT sensor LIT101.

4 Experiments

4.1 Implemention

The Safe Water Treatment (SWaT) system is a simulation and experimentation platform in the field of Industrial Internet. The test bed was monitored by 25 different types of sensors to monitor the operation of the equipment, and data sets were generated after seven days of operation under normal operating conditions and four days of operation under continuous or intermittent attack conditions. During this period, 36 different attacks were executed by outsiders. In many of the attacks, the attackers manipulated data received from other components, forcing the receiving component to misbehave.

This experiment was trained using normal data generated during the first 7 d of SWaT. In order for the sequence to contain as many attack signals as possible, while ensuring reliable detection accuracy. The time unit in the experiment is chosen to be 10 s, and different sensors correspond to different state maximum durations D. The length of the sequence was chosen to be 120 s. First, the observations of the time series are clustered to obtain the implied states under the HsMM model. The states clustered by different sensors are different, and the number of states of the sensors in the test is generally between 4 and 22. The clustering effect of the selected sensors LIT101 is shown in Fig. 2, and the number of clusters is 15.

Table 1. Detection Result: When the amplification factor is 1.65, the detection status of sensor LIT101 is shown, where the bold serial number represents the sensor that has been directly attacked.

Sensor	TP	FP	FN	Sensor	TP	FP	FN
1	0	0	0	14	138	29	0
2	12	0	0	**15**	17	2	0
3	0	7	0	16	0	1	0
4	2	0	0	**17**	22	1	0
5	0	0	0	18	51	0	0
6	0	0	0	**19**	20	0	0
7	5	0	0	20	14	0	0
8	0	0	0	21	10	0	0
9	9	0	0	22	15	0	0
10	0	0	0	23	12	1	0
11	24	5	0	24	10	1	0
12	14	0	0	25	0	0	0
13	14	1	0				

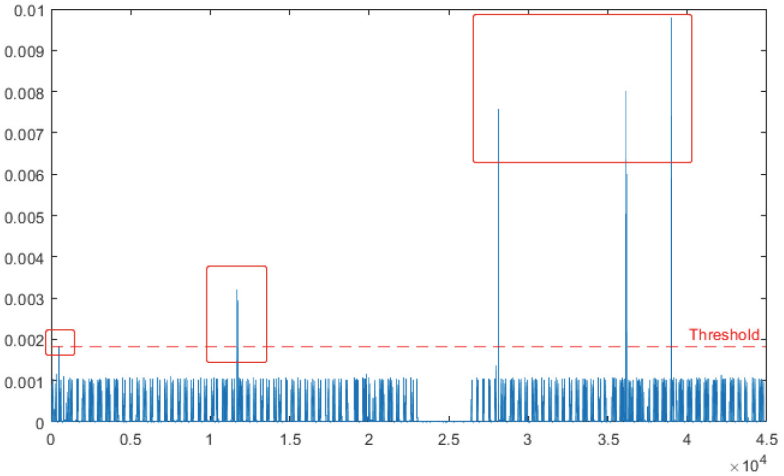

Fig. 3. ADE sequence: SWaT sensor LIT101 as an example.

Then, training takes the form of iterative training. The training set is cut into a number of subsets of length 3000, with every 50 subsets as a group. Each group of subsets is trained directly, and the corresponding η, ξ, γ are found by the formula (5)(6)(7), respectively. After in-group sequence training, the three parameters obtained are averaged, and then the HsMM model parameters are updated and inputted into the next training group. Finally, the anomaly detection model based on production behavioral features is obtained through iterative training.

4.2 Results and Analysis

Detecting attacks through the model is essentially a multiple classification problem, and the detection results can be categorized into 4 classes:False Positive(FN), False Negative(TN), True Negative(FP), TruePositive(TP). Based on the above four statistical results, the performance metrics *Precision*, *Recall*, and $F1$ are defined to evaluate the model. The above three metrics are in the range of (0,1), the closer to 1 means the better the detection performance of the model, and the closer to 0 means the worse the detection performance of the model. During detection, if there is an overlap between the detected attack time period and the real attack time period, it is considered correct, and when there is an attack time period that is not judged at all, the attack is considered to be ignored or unrecognized.

For anomaly detection, the average entropy derivative of three consecutive sequences is obtained according to the formula (12), where $t = 360$. In order to avoid that the threshold is set too low, which leads to the normal but slightly deviating from the average sequence being misjudged as abnormal, the threshold

Table 2. Comparison of Techniques Against SWaT Dataset [20]

Technique	Precision	Recall	F1 Score
CNN(8 layers)	N/A	N/A	0.861
DNN	0.98	0.68	0.8
One Class SVM	0.92	0.69	0.79
PCA	0.2492	0.2163	0.23
MADGAN	0.9897	0.6374	0.77
ABATegaussian(log-entry based)	0.95	0.63	0.76
ABATegaussian(attack-window based)	0.95	0.95	0.95
AE	0.7263	0.5263	0.61
TABOR	0.86	0.788	0.82
HsMM-ADE	0.8901	**1**	0.9419

needs to be appropriately enlarged. After several validation experiments, the best detection performance is achieved when the amplification factor is 1.65.

Table 1 describes the detection results of the 25 sensors. A sensor under direct attack will detect multiple small intervals within an attack interval at the same time. At this time, the ADE of LIT101 is shown in Fig. 3. A total of 5 direct attacks were made on LIT101, and it can be seen from the figure that a total of 5 time intervals exceeded the threshold, which roughly coincided with the time period when the device was attacked.

Table 2 reports the precision, recall, and $F1$ values for different techniques and the technique proposed in this paper. This method achieves 100% recall while also maintaining high precision and $F1$. Meanwhile, this part will discuss its performance in the following aspects.

1. **Adaptability and Interpretability:** In different scenarios, the HsMM model parameters can be adjusted so that the model has different sensitivities to different anomalies and adapts to more diverse environments. Meanwhile, HsMM can provide predictions and explanations for the moments when anomalies occur. By extrapolating the HsMM, the state and the duration of the state at each moment can be obtained, and the cause and duration of the anomaly can be explained.
2. **Computational Complexity:** In the training phase, $O((D+N)TNIKJ)$ of computation is required, where N is the number of states in HsMM, D is the maximum time the states last, T is the length of training samples, I is the number of sequences contained in each set of samples, K is the number of groups, and J is the number of iterations. In the experiment, the time unit is 10, $T = 300$, $I = 50$, $K = 7$, J is about 5. Due to the large cutting accuracy, the number of states is between 4 and 22, while D is kept around 160. With MATLAB, training is typically completed in about 2 to 8 hours per sensor. In the detection phase, the computation mainly focuses on the ADE of the observations with a computational complexity of about

$O(DNT)$. Experiments show that anomaly detection takes about 0.0028 s per 120 s sampling period.

5 Conclusion

In this paper, we propose a HsMM-based method for detecting behavioral anomalies in industrial equipment. It uses the historical data of industrial equipment to abstract the normal operating state transfer into a Hidden Semi-Markov process, and then uses the average entropy derivative of the observations to carry out the anomalies that exist in the time period. The test results of the SWaT dataset show that the detection recall for direct attacks reaches 100%, and the $F1$ scores reaches 94%, which indicates that this method is effective in detecting the attacks on the industrial production process. Where the decrease in precision mainly comes from the impact of indirect attacks, when a device is attacked, other devices in the same system may be affected, and this impact arrives at different times, making it difficult to limit it according to the same criteria. Therefore, in future plans, the addition of a neural network structure to the current architecture will be considered in order to analyze the interdependence of the devices from a systemic point of view, and thus to better distinguish indirect attacks.

Acknowledgement. Thanks for the support of the Key Project of Guangdong Provincial Department of Education (No. 2021ZDZX1031).

References

1. Chandola, V., et al.: Anomaly detection: a survey. ACM Compute. Surv. **41**, 15:1–15:58 (2009)
2. Martí, L., Sanchez-Pi, N., Molina, J.M., Garcia, A.C.B.: Anomaly detection based on sensor data in petroleum industry applications. Sensors **15**, 2774–2797 (2015). https://doi.org/10.3390/s150202774
3. Langner, R.: Stuxnet: dissecting a cyberwarfare weapon. IEEE Secur. Priv. **9**, 49–51 (2011)
4. Zhang, W., et al.: A survey of network intrusion detection methods for industrial control systems(2019)
5. Yu, S.-Z., Kobayashi, H.: An efficient forward-backward algorithm for an explicit duration hidden Markov model. IEEE Signal Process. Lett. **10**(1), 11–14 (2003)
6. Injadat, M., Salo, F., Nassif, A.B., Essex, A., Shami, A.: Bayesian optimization with machine learning algorithms towards anomaly detection. In: 2018 IEEE global communications conference (GLOBECOM), pp. 1–6 (Dec 2018)
7. Astillo, P., Kim, J., Sharma, V., et al.: SGF-MD: behaviorrulespecification-based distributed misbehavior detection of embedded iot devices in a closed-loop smart greenhouse farming system. IEEE Access **8**, 196235–196252 (2020)
8. Yang, K., Li, Q., Lin, X., Chen, X., Sun, L.: iFinger: intrusion detection in industrial control systems via register-based fingerprinting. IEEE J. Sel. Areas Commun. **38**(5), 955–967 (2020)

9. He, X.: Thermodynamic mechanism and data hybrid driven model based marine diesel engine turbocharger anomaly detection with performance analysis. In: CAA Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS) (2019), pp. 477–482 (2019)
10. Lv, Z., Han, Y., Singh, A., Manogaran, G., Lv. H.: Trustworthiness in industrial IoT systems based on artificial intelligence. IEEE Trans. Ind. Inf. **17**(2), 1496–1504 (2021)
11. Xie, Y., et al.: A malware detection method using satisfiability modulo theory model checking for the programmable logic controller system. Concurrency Comput. : Pract. Experience 34(16), e5724 (2020): n. pag
12. Hua, F., Peng, X., Ruoyan, X.: KLS-A: a full-life-time anomaly detection method. In: 2020 International Conference on Artificial Intelligence and Computer Engineering (ICAICE), Beijing, China, 2020, pp. 489–493 (2020). https://doi.org/10.1109/ICAICE51518.2020.00101
13. Zolanvari, M., Teixeira, M.A., Gupta, L., et al.: Machine learning based network vulnerability analysis of industrial internet of things. IEEE Internet Things J. **6**(4), 6822–6834 (2019)
14. Roselin, A.G., Nanda, P., Nepal, S., He, X.: Intelligent anomaly detection for large network traffic with optimized deep clustering (ODC) algorithm. IEEE Access **9**, 47243–47251 (2021). https://doi.org/10.1109/ACCESS.2021.3068172
15. Nguyen, T.D., Marchal, S., Miettinen, M., et al.: DIoT: a federated self-learning anomaly detection system for IoT. In: IEEE 39th International Conference on Distributed Computing Systems (ICDCS), pp. 756–767. IEEE (2019)
16. Matouek, P., Ryav, O., Grégr, M., et al.: Flow based monitoring of ICS communication in the smart grid. J. Inf. Secur. Appl. **54**, 102535 (2020)
17. Akpinar, K., Ozcelik, I.: Methodology to determine the device-level periodicity for anomaly detection in EtherCAT-based industrial control networks. IEEE Trans. Netw. Serv. Manag. **18**(2) 2308–2319 (2021)
18. Yu, S.: Hidden semi-Markov models. Artif. Intell. **174**, 215–243 (2010)
19. Xie, Y., Shunzheng, Yu.: Monitoring the application-layer DDoS attacks for popular websites. IEEE/ACM Trans. Network. **17**, 15–25 (2009)
20. Narayanan, S.N., Joshi, A., Bose, R.: ABATe: automatic behavioral abstraction technique to detect anomalies in smart cyber-physical systems. IEEE Trans. Dependable Secure Comput. 19(3), 1673–1686 (2022). https://doi.org/10.1109/TDSC.2020.3034331