2021 IEEE International Conference on Recent Advances in Systems Science and Engineering (RASSE) | 978-1-6654-3441-6/21/$31.00 ©2021 IEEE | DOI: 10.1109/RASSE53195.2021.9686839

False Data Injection Attack Testbed of Industrial Cyber-Physical Systems of Process Industry and A Detection Application

1st Yichi Zhang
School of Automation
Central South University
Changsha, China
Email: zndxzyc@csu.edu.cn
ORCID: 0000-0002-2267-5442

2nd Wenfeng Deng
School of Automation
Central South University
Changsha, China
Email: wfdeng@csu.edu.cn

3rd Keke Huang
School of Automation
Central South University
Changsha, China
Email: huangkeke@csu.edu.cn
ORCID: 0000-0003-3553-3424

4th Chunhua Yang
School of Automation
Central South University
Changsha, China

***Abstract*—False data injection (FDI) attack is a common and destructive attack method in Industrial Cyber-Physical Systems (ICPSs), which is mounted in the cyber layer, compromises the measurement data and interferes the physical system at last, especially in the process industry and smart grid. In response, researchers developed many detection method rely on simulation, but the real situations are not ideal simulation environment. This leads to situation in which the high-level methods cannot applied to industrial sites directly. In this paper, we design a testbed of process industry, which is a hardware-in-the-loop platform, to simulate the real industrial production and applied a FDI attack on the platform. The physical process is simulated by a host, and the cyber items are real industrial controller or engineer station. Next, we design an efficient FDI attack detection method, DRIF. Based on our proposed framework, the optimal potential features of high-dimensional industrial process data can be fully extracted, which is conducive to the stage of accurate detection. In addition, it makes our proposed method practicable in real-world scenarios where data instances in normal condition can be used for model training only. The proposed method is applied on the designed platform, and the promising case studies show our framework can achieve satisfactory detection performance, which sheds light on the industrial security to some extent.**

***Index Terms*—Industrial Cyber-Physical Systems, false data injection, attack detection, hardware-in-the-loop platform**

I. Introduction

Industrial Cyber-Physical Systems (ICPSs) are highly complex systems with comprehensive computing, cyberspace, and physical process [1]. Through computation, communication, and control (3C), the original isolated industrial manufacturing systems can be upgraded by powerful engineering techniques and tools such as pattern recognition [2], optimization [3], prediction control [4], machine learning [5], etc. However, to make production more efficient, these powerful techniques usually collect data from physical processes to train their high-level model. Therefore, the originally isolated plants are needed to connect to the external network, and due to the limited resources (energy, expenditure, and bandwidth) and remote unattended operation, these nearly semi-open systems are easily suffering widely cyber-attacks [6], [7], such as the "Sapphire Worm" virus in the United States in 2003 [8], the "Stuxnet" virus in Iran in 2010 [9] and the world-shaking blackout in Ukraine in 2015 [10].

False data injection (FDI) attacks are common in power networks [11], control systems [12], and wireless sensor networks [13], and researchers have focused on these problems extensively. For example, Liu et al. [14] proposed a sparse optimization method based on the slow change of power grid system state and FDI attack sparsity. Rigatos et al. [15] use the Kalman filter to estimate the state of a control system and use statistical decisions to identify the working condition. Besides, there are also some pieces of research, such as [16], [17], which use data-driven methods to determine whether the systems are in normal working condition by extracting the relationship between the system state and monitoring data. Very recently, with the dramatic development of artificial intelligence, deep learning methods are widely used in cyber-attack identification, which has greatly promoted the development of industrial security research. For example, He et al. [18] proposed Conditional Gaussian-Bernoulli Restricted Boltzmann Machine (CGBRBM) which utilizes the deep learning architecture to identify the FDI attack. However, most of the intrusion detection methods, including FDI attack, rely on the simulation environment, and it is hard to apply these methods to the industrial site directly. In summary, there are three difficulties:

1) Artificial intelligence is a powerful tool, but it usually requires high-quality data to train a high-precision model. However, in real conditions, the data distributions are very unbalanced, which means that positive samples are usually readily available while the negative ones are not, indicating the overall data acquired is not representative. This leads to difficulties in the model training phase.
2) Some methods rely on state estimation, but the state transition equations are difficult to obtain because most

Corresponding author: Keke Huang.

978-1-6654-3441-6/21/$31.00 ©2021 IEEE

systems in the real process industry are non-linear systems. This is a great obstacle to the application of these methods in manufacturing systems.

3) There are many units, including equipment and subsystems, in a process industry, and each unit may have some input, output, and control variable value. The sum of them may cause the high dimension of the data acquired by SCADA at the same time. It is a tough task to rapidly train a high-accuracy model with limited computing resources.

In order to evaluate the performance of the data mining and machine learning algorithms for ICPS systems, a verification platform with is required. Therefore, we build a hardware-in-the-loop ICPS platform with a simulation host and real network devices. At the same time, we propose our own FDI detection Framework, DRIF, and applied it on the platform we built.

Here are the contributions:

- A hard-in-the-loop platform, which can be adjusted to different real industrial scenarios, is designed and built to test and mine the scientific problems.
- Dimensionality Reduction and Identification Framework (DRIF) is proposed to detect a designed false data injection attacks, which can not be detected by general bad data χ^2 detector.
- DRIF is composed of hierarchical Self-Organizing Map (SOM) and Support Vector Data Description (SVDD), where SOM is used for data dimension reduction and SVDD is used to detect the anormal behavior of the system precisely. The combination of SOM and SVDD improves the calculating efficiency and detecting performance.

The rest of the paper is organized as follows: In Section II, the architecture of the hardware-in-the-loop platform and the theory of FDI attack is introduced. The proposed FDI attack detection method, DRIF, is illustrated in Section III. Experiments results of case study are shown in Section IV. And the conclusion of this paper is in Section V.

II. Testbed Description

A. Hardware-in-the-Loop Platform

To test the performance of high-level methods of FDI attack or other type of intrusion detection methods, we build a hardware-in-the-loop system. The architecture is shown in Fig. 1. There are two layers in this platform: Cyber layer and Physical layer.

In the physical layer, the physical process is simulated on a high-performance computer which is called the simulation host. And the physical process, including chemical reaction, used in this platform comes from a classical Tennessee Eastman (TE) industrial process described in [19]. The simulation host is connected to a switch to transfer sensor measurements and receive the control command from a high-level controller. The system noise is the same as [19].

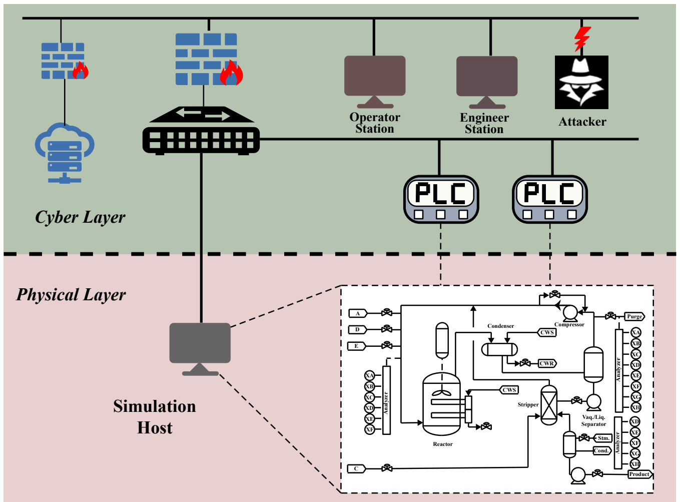

Fig. 1. The architecture of hardware-in-the-loop simulation platform

In the cyber layer, two PLCs are connected to a switch, which is linked to the simulation host, and they take on two functions: (1) PLCs obtain the sensor measurements from simulation and upload them to the operator station and engineer station. (2) PLCs calculate and issue the new control commands to control the simulation to operate normally. The PLCs use Modbus/TCP communication protocol to exchange data with the simulation host. In addition, go up through a firewall from a switch, there are one operator station and one engineer station connected. They can obtain the measurements for monitoring from PLCs and change the production state by changing the production parameter of PLCs. The data collection and command issuance system in the engineer station constitutes the SCADA system in actual production applications. At last, all the data will be sent to the cloud server from the engineer station and stored in a Transwrap Inceptor Database for analysis. The control variables and measurement variables are the data that is collected by our data acquiring system, and the description and measurement unit of them can be referred in [19].

B. False Data Injection Attack

Consider the following discrete nonlinear system where the state transition function and observation function are defined as:

$$\begin{aligned} x(k+1) &= f(x(k)) + w(k) \\ y(k) &= h(x(k)) + v(k) \end{aligned} \tag{1}$$

where $x(k+1) \in \mathbb{R}^{n_x}$ is the state system with $n_x \in \mathbb{Z}^+$, $y(k) \in \mathbb{R}^{n_y}$ is the sensor measurement with $n_y \in \mathbb{Z}^+$ at time point k, respectively. $w(k) \in \mathbb{R}^{n_x}$ is the process noise, $v(k) \in \mathbb{R}^{n_y}$ is the measurement noise, and $w(k)$ and $v(k)$ are uncorrelated zero mean Gaussian noises with covariance $\sum_w$ and $\sum_v$. $f(\cdot)$ represents the nonlinear state function and $h(\cdot)$ is the nonlinear observation function.

FDI attack aims at the cheap communication system in the production industry. It may hijack the communication line or tamper with the data block to achieve its purpose. Any successful attack will significantly hamper the economy, the environment or may even lead to loss of human life. In order to detect and reduce the catastrophic consequences of FDI attacks, relevant theory needs to be introduced first.

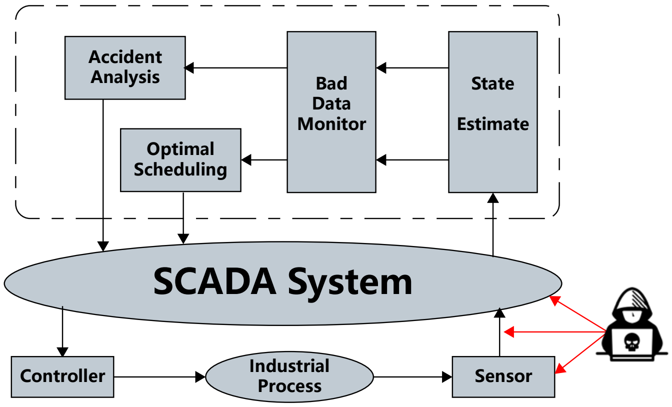

Fig. 2. The location of false data injection attack. Hackers can attack any of the three locations of sensor value, communication connection, and SCADA data block, to complete the FDI attack task.

The FDI attack may occur in three locations which are shown in Fig. 2, where the hacker attempts to modify the observation y ultimately. In a real industrial SCADA (Supervisory Control And Data Acquisition) system, a client usually deploys a bad data detector (BDD), which will estimate the system state $\hat{x}$ and then calculate the measurement residue r between actual observation y and estimation $h(\hat{x})$, the relationship is shown in Eq. (2).

$$r = y - h(\hat{x}) \tag{2}$$

In addition, most of the bad data detector in control systems are χ^2 detectors, they compute the following equation:

$$g = r^T P^{-1} r \tag{3}$$

where P is the covariance matrix of residue $y(k)$. Because of the existence of measure noise, residue r should follow the Gaussian distribution with zero mean. Therefore, g will be close to 0. Then the χ^2 detector can be used to compare with a certain threshold which is calculated from historical data for implementation. If g is larger than the pre-determined threshold, the system will trigger a bad data alarm.

$$\begin{aligned} r_a &= y_a - h(x_a) \\ &= y_a - h(x_a) + h(x) - h(x) \\ &= y + a - h(x_a) + h(x) - h(x) \\ &= r + a - h(x_a) + h(x) \end{aligned} \tag{4}$$

However, there are many pieces of research, such as [7], [12], proposed that if a hacker designs a certain attack, a χ^2 detector will be malfunctioned. For example, assuming that x is the true state vector and x_a represents the attacked state vector which is expected and injected by a hacker. $y_a = y + a$ means the injected malicious data y_a can be represented by the sum of true measure y and inject measure a. Therefore, the residue under attack is represented by Eq. (4).

From Eq. (4), if a hacker desires to bypass the residue test, the condition of $r_a = r$ should be satisfied. As a result, the injected measurement should be determined as:

$$a = h(x_a) - h(x) \tag{5}$$

The results show that the designed attack can pass the residue test and make an impact on the industrial systems.

In this platform, it is assumed that the attack launches from the cyberspace, and the hacker is powerful enough to bypass the BBD deployed in the cyber layer. Therefore, we set up an attack computer separately in the cyber layer, besides the operator station and engineer station. The place is shown in Fig. 1. The attack target is the data block in PLCs, and the attack action is to change the production sensor measurement and cheat the χ^2 bad data identifier. Then we use a self-developed attack script to steal the sensor measurements and inject the false data to the engineer station at the moment we set in advance.

III. Detection Method

FDI attack seriously affects the normal operation of an industrial system, but the detection of it is not a simple job. SCADA system acquires data periodically, and the time period of data polling time is not frequent, this consequence leads to that the data cannot capture the temporal pattern of an industrial system, but only state condition. Therefore, some methods, which rely on the state transaction equations, will not perform well. In addition, the data is very imbalanced because the negative samples are far less than positive ones, its difficult for methods which need a certain amount negative samples to train a good classifier. At the same time, the dimension of data is very high, which means it is a tough task to rapidly train a model or test. Hence, if we can find a method which not only can reduce the dimension of the data, but also maintain the state feature, besides that, the method can also deal with one-class sample training and save time. Fortunately, we can use a novel combination of two classic methods to satisfy our requirements. SOM [20] can be used for data feature extraction and SVDD [21] is a popular one-class classifier.

A. Related Work

a) SOM: Self-Organized-Map (SOM) is a typical artificial neural network that is trained by unsupervised learning which is called competitive learning rules, and we use it to produce a low dimension data here. A SOM training phase including competition, cooperation, and adaptation. The important variables used in SOM is shown in Tab. I.

And the training step is as follows:

b) SVDD: Support Vector Data Description (SVDD) was proposed by Tax and Duin [22] for getting a good description of training data. SVDD computes a spherical decision boundary for most training data. And samples outside are treated as outliers. Strong generalization ability is the

TABLE I
THE IMPORTANT SYMBOLS USED IN SOM

Symbols	Descriptions
k	The current iteration
λ	Iteration limit
t	Index of the target input data vector in input data set D
D	Input data vector
v	Index of the node in SOM
$\mathbf{W}_v$	The current weight vector of node v
u	Index of the best match unit (BMU) in SOM
$\theta(u, v, s)$	Restraint due to distance from BMU
α_s	Learning restraint due to iteration progress.

Algorithm 1 SOM Training

```
Require: Training data of a certain data group $D$.
Ensure: SOM net
  1. Create a square SOM net and the node number is related to the input dimension.
  2. Select one input vector and calculate the dot product with each node in the map and find the BMU $u$.
  3. Update the weight vectors of the nodes in the neighborhood of the BMU by:
     $W_v(s+1) = W_v(s) \dashv \theta(u,v,s) \cdot \alpha(s) \cdot (D(t) - W_v(s))$   (6)
  4. $k = k + 1$
  5. Repeat step 2,3,4 until $k > \lambda$.
```

characteristic of SVDD, but as the data dimension increases the training speed increases greatly.

Assuming that, there is a group of training data $\mathbf{x} \in R^{n \times d}$, where n is the sample number, and d is the dimension of training data. We use a transform function $\Phi : \mathbf{x} \to \mathbf{F}$ to map the data from the original space to the feature space, and most of the transform functions are Gaussian kernels with scale γ. Then the smallest size hypersphere will be calculated. But to find this sphere, the following optimization problem is needed to be solved:

$$\min_{\mathbf{a},\mathbf{R},\xi} \mathbf{R}^2 + C \sum_{i=1}^{n} \xi_i$$
$$s.t. \|\Phi(\mathbf{x}_i) - \mathbf{a}\|^2 \le \mathbf{R}^2 + \xi_i, \xi_i \ge 0, \forall i = 1, 2, \cdots, n \tag{7}$$

where $\mathbf{R}$ is the radius of the hypersphere, $\mathbf{a}$ is the center, and ξ is the relaxation factor, and C is the penalty parameter that weights the volume of the hypersphere and the misclassification rate. Combined with the Lagrange multiplier method, the dual problem of the original problem is:

$$\min_{\alpha_i} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) - \sum_{i=1}^{n} \alpha_i K(\mathbf{x}_i, \mathbf{x}_i)$$
$$s.t. 0 \le \alpha_i \le C, \sum_{i=1}^{n} \alpha_i = 1 \tag{8}$$

where α_i is the Lagrangian coefficient of the sample $\mathbf{x}_i$. $K(\cdot)$ is the kernel function that is equal to the inner product of the sample in the feature space, $K(\mathbf{x}_i, \mathbf{x}_j) = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle$. Therefore, the center and radius of the hypersphere are respectively by:

$$\mathbf{a} = \sum\nolimits_{i=1}^{n} \alpha_i \Phi(\mathbf{x}_i) \tag{9}$$

$$R = \sqrt{K(\mathbf{x}_v, \mathbf{x}_v) - 2\sum_{i=1}^{n} \alpha_i K(\mathbf{x}_v, \mathbf{x}_i) + \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j)} \tag{10}$$

At the same time, the distance between the test sample x_t and hypersphere center is:

$$d = \sqrt{K(\mathbf{x}_t, \mathbf{x}_t) - 2\sum_{i=1}^{n} \alpha_i K(\mathbf{x}_t, \mathbf{x}_i) + \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j)} \tag{11}$$

SVDD method can train a strong one-class classifier efficiently, but the speed will decrease dramatically with the dimension increases of training data. But here, data division rules in the proposed framework may help a lot to reduce the data dimension. At first, data will be separated and sent to their corresponding SOM, each SOM will output a value that is equal to the distance between new data and its champion node. Then, we stack these values into one vector as a piece of feature data. This feature data maintains the features of the original high dimensional data, and the dimension is much smaller. In this way, the speed of SVDD model training is improved.

B. DRIF Framework

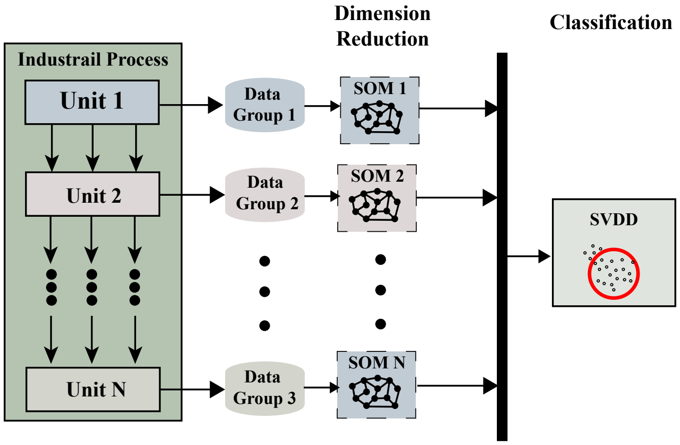

Fig. 3. The framework of DRIF

In this part, we will introduce the framework of Dimensionality Reduction and Identification Framework (DRIF). Specifically, it contains two processes: dimension reduction and classification, which is illustrated in Fig. 3. There are many subsystems and a lot of equipment in a process industry, and they are connected to each other through production processes and procedures. Here, we treat these subsystems or equipment as different production units. Industrial raw materials pass through each unit and are finally made into corresponding products. In a certain unit, control systems control the chemical reactions of input materials and produce new ones. Hence, the data of each unit include the input, output, and control variables. At one time point, a group data of sensors is collected by the SCADA system. We group the data for different units send them to their corresponding Self-Organized-Maps (SOM) network. Through the calculation of each corresponding SOM, the input data vector will be abstracted into one feature value. Then, these feature values are stacked into one feature vector and sent to the Support Vector Data Description (SVDD) classifier. The classifier will determine whether the system is suffering a false data injection attack at this time.

The whole detection process of the proposed framework is shown in Algorithm 2:

Algorithm 2 DRIF

```
Require: Real-time measurement data $D_t$.
Ensure: The label of the data, 1 or -1.
  1. Data division and send each group of data to their corresponding SOM.
  2. SOM calculates the distance between the input data and its BMU.
  3. Stack every value which is calculated by SOM into one vector $V_f$.
  4. Send the feature vector $V_f$ to SVDD and calculate the distance $d$ between this data instance and the center of the hypersphere.
  5. Compare the distance $d$ with the radius of the hypersphere $R$. If $d > R$, we determine this piece of data is under false data injection attack, and label it with -1. Otherwise, the instance is healthy operation data and labeled with 1.
```

C. Data Division Rules

In order to reduce the data dimension while retaining the data feature, we group the data and use different SOMs for data feature extraction. In a real industry scenario, different production units undertake different functions. For one unit, the amount of each kind of input material, the amount of each output material, and the control system variables are all influencing factors. Every factor is the manifestation of the state of this production unit. Hence, we put the input, output, and control variables that are related to a certain unit together, and this is the data group of this certain unit. It is worth noting that, the production units are connected with each other with the rule of the production process, and one unit's output will be another unit's input. Therefore, part of the data will be reused according to our data division rules.

IV. Experiments

A. Configuration of DRIF

Before DRIF is used for false data injection attack detection, we declare the parameters setting of DRIF, including neuron number of SOM, training iterations, the variance of Gaussian kernel in SVDD.

- The number of neurons of SOMs in the dimension reduction phase is equal to the square root of the size of the input data. This way of specifying the number of neurons proved that it can maintain a relatively high discriminatory capability, while reducing computational overhead [23].
- The number of iteration in SOM training is usually set as 500 times the number of neurons [24].
- The import hyperparameters of SVDD are scale γ and penalty parameter C. Here, we choose a group of relatively good hyperparameters $\gamma = 0.01, C = 0.9$.

B. Results Analysis

By comparing the standard, i.e., the true data label, with the predicted outcome, the result for each test instance can be classified as a true positive (TF), false positive (FP), true negative (TN), or false negative (FN). The confusion matrix is shown in Tab. II. In order to demonstrate the efficiency of the proposed method, we choose several evaluation indexes, false alarm rate (FAR), false detection rate (FDR) [25], and accuracy [26]. The definition is shown below:

TABLE II
Confusion Matrix for the Binary Classification of Predicted Outcomes

		Predicted Label	
		Positive Class	Negative Class
Actual Label	Positive Class	TP	FN
	Negative Class	FP	TN

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

$$FDR = \frac{TN}{FP + TN} \tag{13}$$

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{14}$$

a) Framework Validity: In this experiment, we collected data of 7200 time points of operation state in chronological order, and when the 5400th time point was reached, the hacker conducted a false data injection attack and tamper the data of reaction press to a relatively high value, this kind of attack can affect the industrial production even destroy the production equipment and cause great financial loss.

Here, we choose the first 4000 pieces of data, which are all positive samples, to train our framework. Other 3200 pieces of data are treated as the test dataset. In the test dataset, 1400 at the front are positive samples, and other samples are attacked. The results are shown in the Fig. 4. The gray line in Fig. 4 is the distance between the test data instance and the hypersphere center, and the red line represents the hypersphere radius. If one distance of test instance to the center is larger than the radius R, the instance will be treated as being attacked. The statistic results are shown at the top of Fig. 4, FAR is smaller than 1 percent and FDR is nearly

TABLE III
THE FEASIBILITY EXPERIMENT RESULTS OF DATA DIVISION

Training Sample Size	Method	FAR	FDR	Accuracy	Training Time (s)
2100	DRIF	0.0055	0.9900	0.9925	6.0084
	DRIF without data division	0.0014	0.9961	0.9975	10.7233
4000	DRIF	0.0050	0.9900	0.9929	34.8656
	DRIF without data division	0.0007	0.9961	0.9978	45.5481

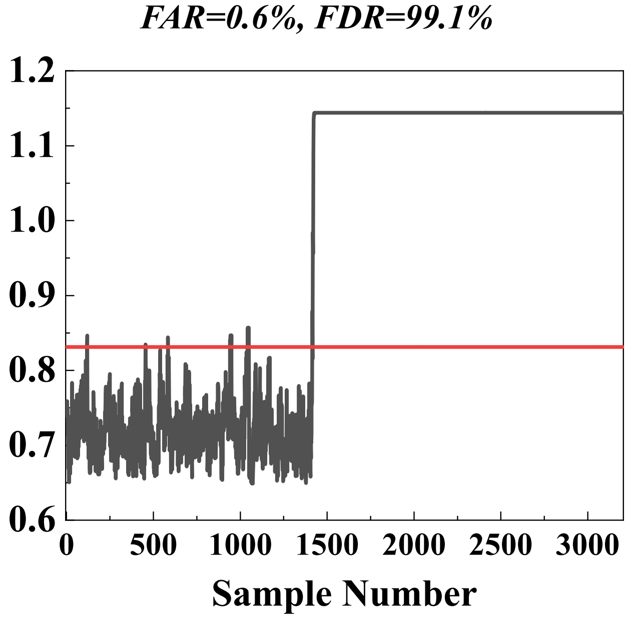

Fig. 4. The detection results of DRIF on test data set. The gray line represent the distance of the test data instance and hypersphere center of DRIF, and the red line represents the distinguish threshold of the normal data and attacked data.

100 percent. In summary, our proposed false data injection attack detection framework, DRIF, can accurately identify the normal data and attacked data.

b) Data Division Feasibility: DRIF will group the data for each equipment or subsystems, and train the corresponding SOM net parallelly, but the data that each SOM used is not global. Under the premise of loss of global information, is the DRIF framework worthwhile? To explore the efficiency about the data division rules, we make a comparison with a method which will not spilt the data. The results are shown in the Tab. III.

From Tab. III, we can find that the value of evaluation indexes of DRIF without data division are a little better than DRIF, because it use the data global information. However, the advance of is not obvious, and the training time of it is much longer than DRIF instead. In condition of the model accuracy loss can be accepted, DRIF can effectively improve the model training speed because of the parallel training strategy.

V. Conclusion

In this paper, we introduce the hardware-in-the-loop platform built to test the advanced algorithm on actual industrial sites. The platform is composed of two layers, Physical layer and Cyber layer. The physical layer is a simulation host which simulates the real industrial chemical reaction mechanism process to generate the production data measured by sensors. And the Cyber layer is composed of real networked devices, including PLCs, switches, and computers. These devices can obtain the production measurement data and issue the control command. All the data can be stored in the cloud server for analysis. At the same time, we propose our false data injection attack detection framework, called DRIF, and apply it on the platform for validation. In this framework, for data of different production units, we set corresponding SOM to reduce the data dimension and extract the optimal features, where different SOMs can be trained parallelly for efficiency improvement. Then the extracted features are stacked into one vector and sent to the SVDD classifier to determine whether the system is in normal operation mode. DRIF has demonstrated its ability to identify most of the limited and compromised measurements with low FAR, and high FDR and accuracy while only normal operation data can be offered during training. We hope our platform architecture and FDI identification framework can clarify the importance of industrial security detection to some extent.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China (Grant No. 62073340), in part by National key research and development program (Grant No. 2018YFB1701100), in part by the Fundamental Research Funds for the Central Universities of Central South University (Grant No. 2021zzts0200).

References

[1] B. Yu, J. Zhou, and S. Hu, *Cyber-Physical Systems: An Overview*. Cham: Springer International Publishing, 2020, pp. 1–11. [Online]. Available: https://doi.org/10.1007/978-3-030-43494-6_1

[2] R. J. Schalkoff, "Pattern recognition," *Wiley Encyclopedia of Computer Science and Engineering*, 2007.

[3] S. Kim and S. Park, "Cps(cyber physical system) based manufacturing system optimization," *5th International Conference on Information Technology and Quantitative Management, Itqm 2017*, vol. 122, pp. 518–524, 2017.

[4] M. Amir and T. Givargis, "Hybrid state machine model for fast model predictive control: Application to path tracking," *2017 IEEE/Acm International Conference on Computer-Aided Design (Iccad)*, pp. 185–192, 2017.

[5] M. Z. Chen, U. Challita, W. Saad, C. C. Yin, and M. Debbah, "Artificial neural networks-based machine learning for wireless networks: A tutorial," *IEEE Communications Surveys and Tutorials*, vol. 21, no. 4, pp. 3039–3071, 2019.

[6] Y. Zhou, Y. G. Fang, and Y. C. Zhang, "Securing wireless sensor networks: A survey," *IEEE Communications Surveys and Tutorials*, vol. 10, no. 3, pp. 6–28, 2008.

[7] M. A. Rahman and H. Mohsenian-Rad, “False data injection attacks against nonlinear state estimation in smart power grids,” *2013 IEEE Power and Energy Society General Meeting (Pes)*, 2013.

[8] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, N. Weaver *et al.*, “The spread of the sapphire/slammer worm,” CAIDA, ICSI, Silicon Defense, UC Berkeley EECS and UC San Diego CSE, Tech. Rep., 2003.

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

[10] G. Liang, S. R. Weller, J. Zhao, F. Luo, and Z. Y. Dong, “The 2015 ukraine blackout: Implications for false data injection attacks,” *IEEE Transactions on Power Systems*, vol. 32, no. 4, pp. 3317–3318, 2016.

[11] L. Xie, Y. L. Mo, and B. Sinopoli, “False data injection attacks in electricity markets,” *2010 IEEE 1st International Conference on Smart Grid Communications (Smartgridcomm)*, pp. 226–231, 2010.

[12] Y. Mo and B. Sinopoli, “False data injection attacks in control systems,” in *Preprints of the 1st workshop on Secure Control Systems*, Conference Proceedings, pp. 1–6.

[13] D.-W. Huang, W. Liu, and J. Bi, “Data tampering attacks diagnosis in dynamic wireless sensor networks,” *Computer Communications*, vol. 172, pp. 84–92, 2021.

[14] L. C. Liu, M. Esmalifalak, Q. F. Ding, V. A. Emesih, and Z. Han, “Detecting false data injection attacks on power grid by sparse optimization,” *IEEE Transactions on Smart Grid*, vol. 5, no. 2, pp. 612–621, 2014.

[15] G. Rigatos, D. Serpanos, and N. Zervos, “Detection of attacks against power grid sensors using kalman filter and statistical decision making,” *IEEE Sensors Journal*, vol. 17, no. 23, pp. 7641–7648, 2017.

[16] K. K. Huang, Y. M. Wu, C. Wang, Y. F. Xie, C. H. Yang, and W. H. Gui, “A projective and discriminative dictionary learning for high-dimensional process monitoring with industrial applications,” *IEEE Transactions on Industrial Informatics*, vol. 17, no. 1, pp. 558–568, 2021.

[17] K. K. Huang, Y. M. Wu, H. F. Wen, Y. S. Liu, C. H. Yang, and W. H. Gui, “Distributed dictionary learning for high-dimensional process monitoring,” *Control Engineering Practice*, vol. 98, 2020.

[18] Y. B. He, G. J. Mendis, and J. Wei, “Real-time detection of false data injection attacks in smart grid: A deep learning-based intelligent mechanism,” *IEEE Transactions on Smart Grid*, vol. 8, no. 5, pp. 2505–2516, 2017.

[19] N. L. Ricker and J. Lee, “Nonlinear model predictive control of the tennessee eastman challenge process,” *Computers & Chemical Engineering*, vol. 19, no. 9, pp. 961–981, 1995.

[20] T. Kohonen, “The self-organizing map,” *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1464–1480, 1990.

[21] Y. Zhao, S. W. Wang, and F. Xiao, “Pattern recognition-based chillers fault detection method using support vector data description (svdd),” *Applied Energy*, vol. 112, pp. 1041–1048, 2013.

[22] D. M. Tax and R. P. Duin, “Support vector data description,” *Machine learning*, vol. 54, no. 1, pp. 45–66, 2004.

[23] H. G. Kayacik, A. N. Zincir-Heywood, and M. I. Heywood, “A hierarchical som-based intrusion detection system,” *Engineering Applications of Artificial Intelligence*, vol. 20, no. 4, pp. 439–451, 2007.

[24] T. Kohonen, “Essentials of the self-organizing map,” *Neural Networks*, vol. 37, pp. 52–65, 2013.

[25] K. K. Huang, H. F. Wen, C. Zhou, C. H. Yang, and W. H. Gui, “Transfer dictionary learning method for cross-domain multimode process monitoring and fault isolation,” *Ieee Transactions on Instrumentation and Measurement*, vol. 69, no. 11, pp. 8713–8724, 2020.

[26] Y. Zhang, Y. Li, W. Deng, K. Huang, and C. Yang, “Complex networks identification using bayesian model with independent laplace prior,” *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 31, no. 1, p. 013107, 2021. [Online]. Available: https://aip.scitation.org/doi/abs/10.1063/5.0031134