Evolutionary Deep Belief Network for Cyber-Attack Detection in Industrial Automation and Control System

Kang-Di Lu, Guo-Qiang Zeng, *Member, IEEE*, Xizhao Luo, Jian Weng, *Member, IEEE*, Weiqi Luo, and Yongdong Wu

***Abstract*—Industrial automation and control systems (IACS) are tremendously employing supervisory control and data acquisition (SCADA) network. However, their integration into IACS is vulnerable to various cyber-attacks. In this article, we first present population extremal optimization (PEO)-based deep belief network detection method (PEO-DBN) to detect the cyber-attacks of SCADA-based IACS. In PEO-DBN method, PEO algorithm is employed to determine the DBN's parameters, including number of hidden units and the size of mini-batch and learning rate, as there is no clear knowledge to set these parameters. Then, to enhance the performance of single method for cyber-attacks detection, the ensemble learning scheme is introduced for aggregation of the proposed PEO-DBN method, called EnPEO-DBN. The proposed detection methods are evaluated on gas pipeline system dataset and water storage tank system dataset from SCADA network traffic by comparing with some existing methods. Through performance analysis, simulation results show the superiority of PEO-DBN and EnPEO-DBN.**

***Index Terms*—Cyber-attacks, deep belief network, ensemble learning, industrial automation and control system, population extremal optimization.**

Manuscript received September 28, 2020; revised November 28, 2020 and January 6, 2021; accepted January 17, 2021. Date of publication January 21, 2021; date of current version July 26, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 61972288, Grant 61972454, Grant 61932011, Grant 61872153, Grant 61825203, Grant U1736203, and Grant 61732021, in part by the National Key R&D Plan of China under Grant 2017YFB0802203, and in part by the Guangdong Key Laboratory of Data Security and Privacy Preserving, National Joint Engineering Research Center of Network Security Detection, and Protection Technology. Paper no. TII-20-4503. *(Corresponding author: Guo-Qiang Zeng.)*

Kang-Di Lu is with the National Laboratory of Industrial Control Technology, Institute of Cyber-Systems and Control, Zhejiang University, Hangzhou 310027, China (e-mail: kangdilu789@zju.edu.cn).

Guo-Qiang Zeng, Jian Weng, Weiqi Luo, and Yongdong Wu are with the College of Cyber Security and the National Joint Engineering Research Center of Network Security Detection and Protection Technology, Jinan University, Guangzhou 510632, China (e-mail: zeng.guoqiang5@gmail.com; cryptjweng@gmail.com; lwq@jnu.edu.cn; wuyd007@qq.com).

Xizhao Luo is with the School of Computer Science Technology, Soochow University, Suzhou 215006, China (e-mail: xzluo@suda.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TII.2021.3053304.

Digital Object Identifier 10.1109/TII.2021.3053304

I. Introduction

THE Internet of Things (IoT) can interconnect electrical equipment with servers and enable collection and aggregation of data [1]. IoT platform has experienced accelerated growth and increasingly been employed in the industrial automation and control systems (IACS). In industrial IoT (IIoT) networks [2], there are massive amount of supervisory control and data acquisition (SCADA)-based IACS. Fig. 1 shows a general structure of SCADA-based IACS. This structure is composed of physical layer, cyber layer, and operation/corporate layer. From Fig. 1, we can see that the physical layer contains programmable logic controllers (PLCs) and remote terminal units (RTUs), which can gather information from the sensors. The cyber layer is used to monitor and control the various devices in local control layer. Also, they include the cyber-attacks detection systems. Corporate layer is an IT system, which can support business processes and push management decisions. Massive devices are linked to SCADA-based IACS, which facilitates the less-expensive data communication and acquisition. However, these devices of SCADA network are vulnerable to various cyber-attacks [3].

Consequently, cyber-attacks detection methods have been introduced for handling cyber-security problems in IACS. Researchers have proposed many detection methods based on the shallow machine learning in the past two decades. However, with the development of novel technologies and the integration of enormous number of IoT devices in IACS, the internet traffic is increased sharply, producing large-scale and multi-dimensional data, which makes the cyber-attacks scenarios more sophisticated. Shallow machine learning-based detection methods may fail to exploit the deep relationship and implicit information from these datasets in handling the growing security challenges. Deep learning techniques have shown their advantages in various domains such as privacy preserving [4] and classification problems [5], compared to the shallow machine learning. Additionally, the cyber-attacks detection problem can be designed as binary-classification or multiclassification problem to detect the types of attacks [2]. Deep belief network (DBN), as one of popular deep learning approaches, can be used as classification model and has powerful ability in handling classification problems [6]. Therefore, DBN can be considered a good candidate to detect the cyber-attacks in IACS [7]. In DBN, the adjustable parameter settings largely affect its performance. As discussed

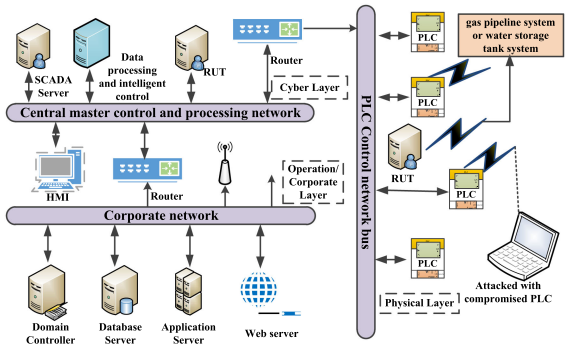

Fig. 1. General structure of SCADA-based IACS, modified from [2].

in [8], selecting appropriate parameters is a difficult task. It requires some practical experiences to determine the parameters, including learning rate, hidden units numbers, mini-batch size, etc. In practice, finding out appropriate parameters is often not available by trial-and-error method because manual testing requires considerable expertise in DBN and the considered problem domain. For example, a security engineer could find a DBN useful in detecting the cyber-attacks. While the security engineer has considerable expertise in the problem domain, they may lack the experience in DBN. This barrier may prevent the application of DBN in detection of cyber-attack domains. Although DBN and its variations have successfully solved many classification problems, no existing analytical approaches can obtain the most appropriate parameters in DBN. Thus, it is of great importance to design an algorithm to tune these parameters without requiring such expertise. As a metaheuristic search approach, evolutionary algorithm including genetic algorithm (GA) [9] and differential evolution (DE) [10] provides an effective way to determine the parameters of DBN. Population extremal optimization (PEO), as a kind of promising evolutionary algorithm, is extended from extremal optimization inspired by self-organized criticality [11]. In recent years, PEO and its variations have proven their performance by dealing with various optimization problems [12]. It is worth mentioning that no attention has been paid to the automatic parameter optimization of DBN based on PEO for cyber-attacks detection problem in IACS. Although the evolutionary DBN can automatically adjust the parameters, the single method can be further improved by ensemble learning method. Ensemble learning schemes [13] have shown the superiority over single-model-based method by employing a certain scheme for aggregation of the results obtained by single method. As a consequence, to further improve detection performance, ensemble learning can be viewed as a potential strategy.

Overall, the main differences between this article and existing studies can be summarized as follows: (a) For the optimization problem in determining the parameters of DBN, through introducing PEO, the DBN's parameters can be automatically determined to overcome the weakness in manual operation in Ref. [7]. Although other evolutionary algorithms (e.g., GA [9] and DE [10]) have been successfully used in tuning these parameters, no attention has been paid to dealing with the problem in parameter optimization of DBN based on PEO. (b) Because the single model has large influence on the final ensemble, this article employs different PEO-DBN methods as the single model, while Ref. [7] used the DBN by trial-and-error method, which may cause performance reduction in ensemble. Our research work is the first attempt to use PEO-DBN and EnPEO-DBN in cyber-attacks detection.

The main contributions are listed as following aspects.

1) PEO, an advanced optimizer, is employed to automatically tune the adjustable parameters of DBN to overcome the weakness in lacking considerable expertise in DBN or the problem domain.

2) Different PEO-DBN-based detection methods are employed to explore the implicit relation between network traffic data and various attack types. After obtaining the classification results, ensemble learning method is introduced to aggregate the results to further improve the detection performance of single method.

3) The comparison and evaluation of the detection performance is showed by comparing the proposed PEO-DBN and EnPEO-DBN with other methods. In the simulation, two real SCADA network datasets, i.e., gas pipeline system and water storage tank system, are considered, which include the new traffic behaviors and represent recent cyber-attacks scenarios.

II. Related Works and Related Methodologies

Nowadays, cyber-security problems have received much attention from academia and industrial domains. Due to large-scale traffic information, the traditional attack detection methods cannot efficiently meet the requirements of security offence and defence [14]. Great achievements have been made in cyber-attacks detection domain in the last few decades by various machine learning methods. For example, Esmalifalak *et al.* [15] proposed support vector machines (SVM) to detect stealthy false data injection attacks (FAIDs) in the smart grid, where the principal component analysis (PCA) is applied to reducing the dimensionality. Zheng *et al.* [16] put forward an extreme learning machine (ELM) classifier based on modified linear discriminant analysis to detect the attacks in IoT application. Due to the increased Internet traffic and large-scale data, shallow machine learning methods cannot effectively cope with these sophisticated security problems. Thus, designing deep learning-based cyber-attacks detection methods becomes a hot topic because of their ability in exploring and exploiting the complicated relationship from the collected dataset [6]. Xin *et al.* [17] discussed the definitions, similarities, and differences between shallow machine learning and deep learning, and summarized the methods in cyber-attacks detection topic. In addition, Aldweesh *et al.* [6] summarized the deep learning-based detection methods. The interested readers can refer to [6] and [17] for more details.

A. Deep Belief Network (DBN)

As one of deep learning techniques, DBN [18] is composed of several stacked restricted Boltzmann machines (RBMs) belonging to a generative model, which is trained by employing contrastive divergence procedure.

Fig. 2 gives a diagram for RBM. It learns the probability distribution of given input samples by using energy function;

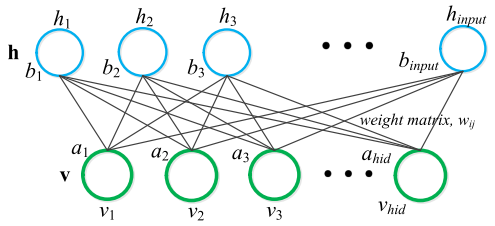

Fig. 2. Diagram for RBM.

this function of network can be described as follows:

$$E(\mathbf{v},\mathbf{h}) = -\sum_{i\in input} a_i v_i - \sum_{j\in hid} b_j h_j - \sum_{i,j} v_i h_j w_{ij} \quad (1)$$

where a_i and b_j mean the biases for input and hidden layer. v_i and h_i mean the state of the ith node. $\mathbf{v}$ and $\mathbf{h}$ mean the set of nodes in the input and hidden layers. w_{ij} is the weights linking two nodes.

Then, every possible pair of an input and hidden layers in RBM can be described by the following distribution:

$$p(\mathbf{v},\mathbf{h}) = \frac{e^{-E(\mathbf{v},\mathbf{h})}}{\sum_{v,h} e^{-E(\mathbf{v},\mathbf{h})}}. \quad (2)$$

Every node generates a random value, i.e., 0 or 1, which is obtained from marginal distribution of (2)

$$\begin{cases} p(v_i = 1|h) = \sigma(a_i + \sum_j w_{ij} h_j) \\ p(h_j = 1|v) = \sigma(b_j + \sum_i w_{ij} v_i) \\ \sigma(\varphi) = 1/(1+e^{-\varphi}) \end{cases}. \quad (3)$$

The training step is the stochastic steepest ascent method. The probability is updated as follows:

$$P(v) = \frac{\sum_h e^{-E(\mathbf{v},\mathbf{h})}}{\sum_{v,h} e^{-E(\mathbf{v},\mathbf{h})}}. \quad (4)$$

Then, the weights are updated as follows:

$$\Delta w_{ij} = \varepsilon\left(\langle v_i h_j\rangle_{\text{data}} - \langle v_i h_j\rangle_{\text{reconstruction}}\right) \quad (5)$$

where $\langle\rangle_{\text{data}}$ and $\langle\rangle_{\text{reconstruction}}$ represent the expectation regarding the distribution of the data and the reconstruction, respectively. ε is a positive value representing the learning rate. Δw_{ij} means the change of weight. After training of RBM, the hidden weights W are employed to initialize a neural network. Then, the output layer is added to the neural network. For the classification problem, the output nodes are the number of class values.

He *et al.* [19] exploited conditional DBN (CDBN) to recognize features of attacks for real-time detection attacks in smart grid based on the historical measurement data. Manimurugan *et al.* [20] have presented effective cyber-attacks detection method based on DBN for Internet of Medical Things. However, the parameters of these DBNs are still determined by trial-and-error method, which is tedious and requires considerable expertise. To alleviate this problem, GA is employed to adaptively generate the number of hidden layers and corresponding neuron numbers in each layer for intrusion detection for IoT [21]. Thus, to further improve the performance of DBN, a natural idea is applying a better evolutionary algorithm, such as PEO, to optimize the parameters in DBN. Additionally, PEO has not been employed to tune the parameters of DBN so far in designing cyber-attacks detection method for SCADA-based IACS. It should be remarked that the population-based training (PBT) technique for neural networks [22] combined the advantages of parallel search and sequential optimization for tuning parameters and it has be applied into various domains [23]–[25]. Considering the advantages of PBT, the performance of PBT-based DBN for cyber-attacks detection can be further investigated in future work.

B. Ensemble Learning of Classifiers

For classification problems, an ensemble method is composed of a cluster of classifiers. These classifiers obtain different results. The final results are determined by the average values or the voting technique. Here, suppose that we have obtained N network traffic samples $\mathbf{X}$ with m features and c classes. $x^j = \{x_1, x_2, x_3, \ldots, x_m\} \in \mathbf{X}$ is one of traffic samples. In addition, the class label of each sample is known as l^j, where $l^j \in \{l_1, l_2, \ldots, l_c\}$. Subspace-based features are used to construct M different training datasets, denoted as $\{\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3, \ldots, \mathbf{X}_M\}$, for training methods. Then, we use the obtained $\mathbf{X}_i$ to generate the relationship function between the feature space and the attack type space as follows:

$$l_i^j \in \{l_1, l_2, \ldots, l_c\} = f_i(\mathbf{X}_i). \quad (6)$$

According to the function f_i, we can detect the new traffic sample and categorize this sample.

Finally, the new traffic sample is categorized by a majority voting scheme

$$\bar{l}_i^j = \arg\max_i l_i^j. \quad (7)$$

Many researchers have illustrated that single method can seldom show consistently well in solving various domains, due to the inherent deficiency of single method. Ensemble learning takes advantage of base learners to improve the performance.

Gumaei *et al.* [26] proposed an efficient method to improve the detection precision and to reduce computational overload by feature reduction for cyber-attacks detection of SCADA power system in smart gird. Huda *et al.* [7] suggested ensemble of DBN (EnDBN) to enhance the securing operations in SCADA-IoT-based IACS. In this method, considering the difficulty in selecting appropriate parameters of DBN, ensemble learning is used for different DBN structures to improve the detection performance. By comparing with single DBN and SVM on a real SCADA dataset, the performance of EnDBN is illustrated. Besides, ensemble SVM shows better performance than the single method in detecting the attacks according to the four performance indices. As a potential scheme, ensemble learning of a cluster of PEO-DBNs is worth investigating in cyber-attacks detection for SCADA-based IACS.

III. Proposed Method

A. Proposed EnPEO-DBN Method

We give a novel ensemble method to enhance the detection performance of cyber-attacks in IACS based on PEO-DBN

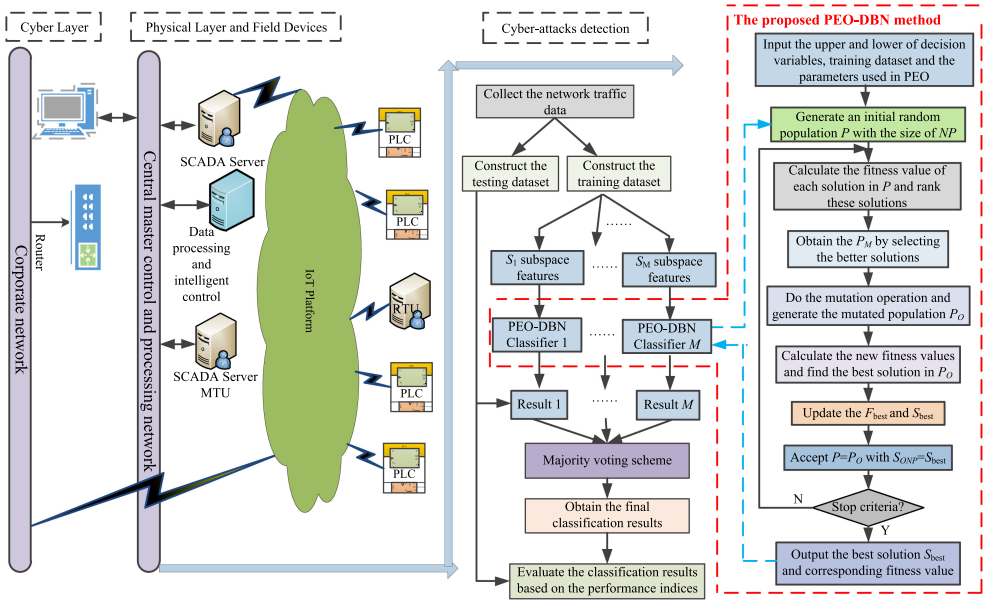

Fig. 3. Structure of the proposed EnPEO-DBN method for cyber-attacks in the IACS.

detection method in this section. The framework of the EnPEO-DBN is given in Algorithm 1. First, random subspace-based features section [2] is used to construct kth group features X_k from the SCADA network. Then, M base classifiers are obtained based on X_k. In other words, after training process accomplished, M different PEO-DBN-based classifiers are obtained. Finally, these classification results are integrated by the majority voting scheme mentioned in (7). The structure of the proposed EnPEO-DBN detection method for cyber-attacks in IACS is given in Fig. 3. Network traffic can be transmitted from cyber layer to physical layer or vice versa, which can be monitored by the secure structure. And the cyber-attacks detection method can receive any response from PLC/RTU to master terminal unit (MTU) or query from MTU to PLC/RTU. The main parts of method for detecting cyber-attacks are given as follows.

1) PEO algorithm, an advanced optimizer, is utilized to optimize the adjustable parameters, e.g., the learning rate, the size of mini-batch, and the number of hidden layers in DBN by minimizing the fitness function in (8). The details of proposed PEO-DBN are given in Section III-B.

2) PEO-DBN with different features are employed to learn the implicit relationship between network traffic data and attack types. Then, majority voting scheme is applied to aggregating the different classification results obtained by PEO-DBN to overcome the weakness of single method.

3) Several cyber-attacks detection methods are viewed as the competitors to demonstrate the detecting performance of PEO-DBN and EnPEO-DBN according to different evaluation metrics.

Algorithm 1: Framework of EnPEO-DBN.

Input: Training dataset, number of base classifiers M, testing dataset.
Output: Detection results.
1: **for** $k \leftarrow 1$ **to** M **do**
2: Construct kth group features $\mathbf{X}_k$;
3: Obtain the kth base classifier based on $\mathbf{X}_k$;
4: **end for**
5: **for** each sample i in testing dataset **do**
6: obtain the detection results based on each classifier;
7: obtain the final results by a majority voting scheme;
8: **end for**
9: **return** Detection results.

B. Proposed PEO-Based DBN Method

There is no existing clear knowledge to help users set the DBN's adjustable parameters. These parameters include the size of mini-batch, learning rate, and the number of hidden layers [8]. Many researchers determine these parameters by trial-and-error, which manually requires considerable expertise and cannot be used effectively. Additionally, PEO has been proven in handling a variety of complex optimization problems, but it has not been employed in optimizing DBN. Thus, we use the PEO as the search engine for automatic parameter optimization of DBN to enhance the ability of detecting cyber-attacks. The detailed fitness function is designed to tune the parameters of DBN as follows:

$$\begin{cases} F = 1 - g_{acc}(f\,(tr\_x, tr\_y, N_{hid1}, N_{hid2}, M_{bt}, \varepsilon_1, \varepsilon_2)\,, \\ \qquad\qquad tr\_x, tr\_y) \\ g_{acc} = \frac{TP+TN}{TP+TN+FP+FN} \end{cases} \tag{8}$$

where F is the fitness function value. tr_x and tr_y are the training samples and corresponding labels. f means the mapping function of tr_x and tr_y when the training process is finished. TP and FP mean true positive and false positive, respectively. TN and FN are the true negative and false negative, respectively.

Algorithm 2: Framework of PEO-DBN.

Input: The upper and lower of adjustable parameters X_L, X_U, training dataset tr_x, tr_y, population size NP, the maximum number of iterations $I_{\max}$, and the mutation parameter b_m.
Output: The best solution S_{best}.
1: $[P] \leftarrow$InitializeOperation(NP, X_L, X_U);
2: $I_t \leftarrow 0$
3: **while** $I_t < I_{\max}$ **do**
4: $[P_S, S_{\text{best}}] \leftarrow$ EvaluationAndSelection(P);
5: $[P, S_{\text{best}}] \leftarrow$MutationAndUpdate($P_S, b_m, S_{best}$);
6: $I_t \leftarrow I_t + 1$;
7: **end while**
8: **return** the best solution S_{best} employed in DBN.

TP, FP, TN, and FN can be obtained by tr_x, tr_y and the mapping function f. g_{acc} is the function to calculate the detection accuracy. N_{hid1} and N_{hid2} mean the number of hidden units of first layer and second layer, respectively. M_{bt} is the size of mini-batch. ε_1 and ε_2 are the learning rates. Here,

$$tr\_x = \begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_m^1 \\ x_1^2 & x_2^2 & \cdots & x_m^2 \\ \vdots & \vdots & \vdots & \vdots \\ x_1^{N_{tr\_x}} & x_2^{N_{tr\_x}} & \cdots & x_m^{N_{tr\_x}} \end{bmatrix}$$
$$tr\_y = \left[l_{i1},\ l_{i2}, \cdots, l_{ic} \right]^T \tag{9}$$

where $x_i^j, i = 1, 2, \ldots, m; j = 1, 2, \ldots, N_{tr\_x}$, represents the jth feature of ith sample. $\{i_1, i_2, \ldots, i_c\}$ represents the type of cyber-attacks. Algorithm 2 gives the framework of PEO-DBN, which consists of three parts including population initialization, evaluation, selection, mutation, and update operation. First, an initial population is randomly generated with the size NP (see line 1). Then, evaluate each solution and select the better solutions from all solutions based on the designed fitness function (see line 4). Next, the selected solutions participate in the mutation operation of PEO, and then, update the population for next iteration (see line 5). Finally, obtain the optimized parameters employed in DBN. The detailed three parts are given below.

1) Initialization Operation: Population initialization gives a population including multiple solutions for the evolutionary process. In general, the solutions are initialized in a random way with a uniform distribution. The adjustable parameters in DBN are encoded into PEO, where each individual represents a network structure. An initial population $P = \{S_1, S_2, \ldots, S_{NP}\}$ is randomly generated with the size NP, where $S_i = [N_{hid1\_i}, N_{hid2\_i}, M_{bt\_i}, \varepsilon_{1\_i}, \varepsilon_{2\_i}], i = 1, 2, \ldots, NP$. The formula of S_i is $S_i = X_L + (X_U - X_L)R_{mi}$, where R_{mi} is uniformly distributed random values in $[0, 1]$, X_L and X_U are the lower and upper values of $N_{hid1}, N_{hid2}, M_{bt}, \varepsilon_1$, and ε_2. Algorithm 3 presents the initialization operation (line 1 in Algorithm 2).

2) Evaluation and Selection: According to (8), we can obtain the fitness value F_i of each solution S_i in P. Then, based on the

Algorithm 3: Initialization operation (InitializeOperation).

Input: Population size NP, the lower and upper values of adjustable parameters X_L, X_U.
Output: Initialization P.
1: **for** $i \leftarrow 1$ to NP **do**
2: $P(i) \leftarrow X_L + (X_U - X_L) \times R_{mi}$;
3: **end for**
4: **return** P.

Algorithm 4: Evaluation and Selection (EvaluationAndSelection).

Input: The population P, training dataset tr_x, tr_y.
Output: The selected population P_S with S_{best}.
1: **for** $i \leftarrow 1$ to NP **do**
2: $F_i \leftarrow$ fitness function$(P(i), tr_x, tr_y)$;
3: **end**
4: Achieve a permutation $F_{\Psi(1)} \le F_{\Psi(2)} \le, \ldots, \le F_{\Psi(NP)}$ based on F_i;
5: Update the best solution $S_{best} \leftarrow S_{\Psi(1)}$;
6: Obtain the population P_S as $\{S_{\Psi(1)}, \ldots, S_{\Psi(NP/2)}, S_{\Psi(1)}, \ldots, S_{\Psi(NP/2)}\}$;
7: **return** P_S, S_{best}.

obtained NP fitness values, a permutation can be achieved. For convenience, symbol Ψ is used to represent the ranking result, i.e., $F_{\Psi(1)} \le F_{\Psi(2)} \le, \ldots, \le F_{\Psi(NP)}$. Afterwards, update F_{best} and S_{best} as $F_{\Psi(1)}$ and $S_{\Psi(1)}$. Obtain the population P_S by selecting solutions from P. To be a specific, we can generate the half population by selecting solutions, whose ranks are from $\Psi(1)$ to $\Psi(NP/2)$. Then, the population P_S can be obtained as $\{S_{\Psi(1)}, \ldots, S_{\Psi(NP/2)}, S_{\Psi(1)}, \ldots, S_{\Psi(NP/2)}\}$. In other words, those worse solutions are replaced with better solutions. Algorithm 4 presents the evaluation and selection (line 4 in Algorithm 2).

3) Mutation and Update Operation: Algorithm 5 presents the mutation and update operation (line 5 in Algorithm 2). Preform the mutation operation and new mutated population $P_M = \{S_{M1}, S_{M2}, \ldots, S_{MNP}\}$ can be obtained from the P_S. Here, the specific mutation operation is chosen as multi-nonuniform mutation (MNUM) given below

$$S_{Mi} = \begin{cases} S_{Si} + (X_U - S_{Si}) \times A(I_t), \text{if } r_1 < 0.5 \\ S_{Si} - (X_L - S_{Si}) \times A(I_t), \text{otherwise } r_1 \ge 0.5 \end{cases} \tag{10}$$

$$A(I_t) = \left[r_2 \left(1 - \frac{I_t}{I_{\max}} \right) \right]^{b_m} \tag{11}$$

where I_t is the current iteration number. Both r_1 and r_2 mean uniform random values between 0 and 1. S_{Si} is the solution in P_S. Besides, b_m is the mutation parameter adopted in MNUM.

Unconditionally, set $S_{NP} = S_{best}$ and $P = P_M$.

Remark 1: For the proposed algorithm, PEO-DBN includes two parts, i.e., PEO algorithm and DBN. PEO is used to determine the adjustable parameters in DBN based on the collected

Algorithm 5: Mutation and Update Operation (MutationAndUpdate).

Input: The selected population P_S, mutation parameter b_m and S_{best}.
Output: The updated population P and S_{best}.

```
P_M ← P_S;
for i ← 1 to NP do
    if r_1 < 0.5 then
        S_Mi ← S_Si + (X_U − S_Si) × A(I_t);
    else
        S_Mi ← S_Si − (X_L − S_Si) × A(I_t);
    end if
end for
P ← P_M;
S_NP ← S_best;
return P and S_best.
```

dataset from SCADA network. Compared to the updating time of dataset of SCADA system, the time costing of PEO is much larger. In practice, this process is implemented offline. After finishing the training process, the trained model can be used online to detect the cyber-attacks. First, we discuss the parameters of DBN determined offline. If the designer lacks the considerable expertise in DBN and problem domain, it is hard to tune the adjustable parameters to obtain the satisfied performance while PEO-DBN can adjust automatically these parameters without such expertise. Even if the designer has considerable expertise in DBN and problem domain, 100 times may be needed to determine the parameters of DBN by trial-and-error. For PEO, the number of fitness function evaluation is set as 100, which has the similar time cost. PEO can be adjusted automatically toward the optimal fitness function, which indicates that PEO is more likely to find a better solution under the similar time cost. Overall, PEO is effective to tune these parameters offline. Then, we discuss the time cost online for detecting the cyber-attacks. After determining the parameters of DBN, we use the model to find out whether there is an attack, which has the similar time cost (about 6 ms) for both PEO-DBN and trial-and-error-based models in this article. Overall, PEO-DBN is more likely to find a better solution with higher accuracy to detect the cyber-attacks. The simulation results also demonstrate the performance of PEO-DBN. Additionally, if updating time of dataset of SCADA system is smaller than the running time, the algorithm will affect the network traffic whether there is an attack or not, which will cause a delay to the system. If the running time is smaller than the updating time of SCADA, the proposed algorithm will not affect the network traffic under no attack scenario and no delay to the system. Under attack scenario, the decision-makers may make some measures or the system may start the defense strategy.

Remark 2: The proposed detection system detects the cyber-attacks in IACS by the characteristic of traffic dataset. If the attack changes the dataset obtained by the various devices, the proposed detection system can still work. In other words, if the detection system is placed within the SCADA system, the proposed algorithm can still work. Thus, if an attacker takes over the remote RTU, the proposed algorithm can be used to place between this RTU and the system to protect the system.

Remark 3: The time complexity of the proposed PEO-DBN includes three main parts: sorting operation of population fitness, population mutation, and evaluating population fitness. The first and second parts are the $\mathrm{O}(I_{\max} NP \lg NP)$ and $\mathrm{O}(I_{\max} NP(3n^2 + 6n))$. For the evaluating fitness, we need to analyze the training process of DBN. In DBN [18], the stacked restricted RBMs are the main step in DBN, whose time complexity largely depends the whole DBN. Here, suppose that there are three hidden layers with m_1, m_2, and m_3 numbers. The input size is N with n features and the iteration is K. For the first three hidden layers, the time complexity is $\mathrm{O}(KN(3nm_1 + 2m_1^2 + n^2 + 3m_1m_2 + 2m_2^2 + m_1^2 + 3m_2m_3 + 2m_3^2 + m_2^2))$. Then, time complexity for the backward tuning learning step is $\mathrm{O}(KN(nm_1 + m_1m_2 + m_2m_3))$.

IV. Simulations and Discussion

Section IV is devoted to demonstrating the performance of PEO-DBN and EnPEO-DBN on two real SCADA datasets. In the first dataset, the competitors include ELM [27], SVM [7], decision tree-based method (DT) [28], DBN [18], ensemble of SVM (EnSVM) [7], and ensemble of DBN (EnDBN) [7]. To further compare with other sophisticated deep learning approach, long short-term memory neural network (LSTM) [29] is used and another dataset from water storage tank system [28] is used to further validate the efficacy of the proposed methods under different data scales. The first and second methods belong to shallow machine learning and these two methods are successfully applied to detecting cyber-attacks. Thus, ELM [27] and SVM [7] are considered as the compared algorithms to illustrate that a deep learning method, i.e., PEO-DBN, can achieve better performance than two shallow machine learning-based methods. DBN [18] and LSTM [29] belong to deep learning, and thus, to further illustrate the performance of PEO-DBN, these two deep learning-based methods are used as the competitors. DT [28], EnSVM [7], and EnDBN [7], as the ensemble methods including the shallow and deep machine learning, are used as the competitors to show the performance of EnPEO-DBN. Three simulations are considered for different purposes. All detection methods are implemented by the aid of MATLAB 2016a software (except the LSTM implemented by MATLAB 2018a) on a computer with Windows 7 operating system, Inter Core I7 CPU @ 2.5 GHz and 8 GB RAM. To alleviate the influence caused by the randomness of different methods, we repeat ten times for each method on three simulations. As the suggestion in [2], [7], a 10-fold cross-validation scheme is considered to train and test the performance of each method on gas pipeline system dataset. For the water storage tank system dataset, a five-fold cross-validation scheme is considered. Besides, the parameters NP, $I_{\max}$, and b_m used in PEO are set as 4, 10, and 3, respectively. By employing PEO, the adjustable parameters are obtained for DBN. The hidden units of first layer and second layer are set as 56 and 94, respectively. The size of batch is set as 304, and the learning rates of first layer and second layer

are set as 0.6124 and 0.7962, respectively. The parameters of three different DBNs are as follows. For DBN1, we use 50 hidden units for both layers, 475 for the size of mini-batch, and 1, 0.5 for the learning rates of two layers. For DBN2, we employ 100 and 50 hidden units for first and second layers, respectively. The mini-batch size is set as 400, and the learning rates are set as 1 for both first and second layers. For DBN3, the number of hidden units in the first and second layers are set as 50 and 60, respectively. The mini-batch size is set as 380, and the learning rates are set as 0.5 for both first and second layers. For all detection methods, the epochs of ANN training and DBN training are set as 800 and 500, respectively.

Remark 4: For the parameters used in PEO, $I_{\max}$ and NP are selected based on trial-and-error. Before determining $I_{\max}$ and NP, we perform pre-experiments using different $I_{\max}$ and NP. Large $I_{\max}$ and NP will cost much time to finish the evolutionary process. Thus, $I_{\max}$ and NP are decided by considering the tradeoff between accuracy and computational efficiency and set as 4 and 10, respectively. The performance of PEO-DNB and EnPEO-DBN with five different values of b_m is robust for parameter b_m. The details can be seen in Section IV-E. Thus, b_m is chosen as 3. For the parameters of other competitors including DBN, we perform pre-experiments using different hyperparameters based on small-scale data. To make a tradeoff between learning performance and model complexity, the parameters used in the simulation are determined according to trial-and-error.

A. Dataset Description

The scenario of cyber-attacks in IACS is from the real scenario implemented in [28]. The real SCADA network data of gas pipeline system is provided and can be downloaded from [30]. Here, we use 10% random sample gas pipeline data for industrial cyber-attacks detection. Before using the proposed methods to detect the cyber-attacks, the data preprocessing has been done. The detailed data preprocessing is given as follows. To reduce the influence on the different data dimensions on experimental results, the data is normalized by min–max normalization. In addition, due to the large number of normal traffic, the class distribution will be imbalanced. Some samples of normal traffic are discarded and 8449 samples are used in the simulations, where 7600 samples are used as training samples and the others are testing samples. From the description of proposed methods, we can see that the proposed methods can be extended to different types of datasets. After the data preprocessing, the cyber-attacks detection problems of two considered systems can be viewed as the classification problems. After obtaining the training data of gas pipeline system, the optimized adjustable parameters can be obtained, and then PEO-DBN can be used to detect the cyber-attacks in gas pipeline system. Similarly, after obtaining the training data of water storage tank system, we can also achieve the optimized DBN, and then PEO-DBN can detect the cyber-attacks in water storage tank system.

In the considered traffic network, there are seven different attacks including naive malicious response injection (NMRI), complex malicious response injection (CMRI), malicious state command injection (MSCI), malicious parameter command injection (MPCI), malicious function code injection (MFCI), denial of service (DOS), and reconnaissance (Recon). Detailed description of these attack types is given as follows [28].

a) NMRI attack: Through the previously known information of network servers and devices, the attackers can inject random invalid information to the packet. But they cannot achieve the information of underlying process being monitored and controlled.

b) CMRI attack: Here, the attacks have full information of SCADA network and devices, so that they can mask actual state of the physical process and cause bad influence of the feedback control process.

c) MSCI attack: Through the actuators operation, the state of physical system (e.g., ON/OFF) can be controlled. MSCI attack can change the state of the register, which may cause the operation of actuators incorrectly.

d) MPCI attack: In this type of attack, the attack can alter the set points parameters in field devices, e.g., PLC. After changing these parameters (e.g., PID controller's parameters), the controller performance will be influenced or even does not work at all.

e) MFCI attack: If the attacks have the knowledge of the built-in protocol functions provided by the manufacturers to diagnostic purpose, they may abuse the functions.

f) DoS attack: DoS attack targets communications links and system programs to stop part of SCADA network. Sometime, the attackers can change a packet and send to the filed devices, which may result in crash for the operating system.

g) Recon attack: This attack collects SCADA system information, maps the network structure, as well as identifies device characteristics.

B. Evaluation Metrics

The performance of detection methods is justified according to two common evaluation metrics, i.e., accuracy (ACC) and false positive rate (FPR), whose definitions are given as follows:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{12}$$

$$FPR = \frac{FP}{FP + TN}. \tag{13}$$

For ACC, the closer ACC to 1, the better method performs. As for FPR, the smaller the values, the better the method performs.

Additionally, Friedman test (FT) and Quade test (QT) are used to systematically evaluate the statistical differences, which is performed on the KEEL software [31].

C. Simulation I: PEO-DBN Versus ELM, DT, SVM, and DBN1-DBN3

Simulation I is considered to testify the detecting performance of the PEO-DBN detection method by comparing with ELM [27], SVM [7], decision tree (DT)-based method [28], and three DBNs with different structures [18]. In this simulation, PEO-DBN with full features of collected dataset is considered. Fig. 4 shows the specific detection results via the average values

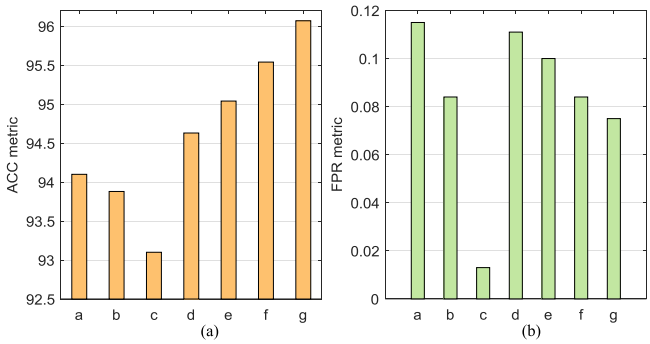

Fig. 4. ACC and FPR metric of simulation I. (a) ACC metric. (b) FPR metric.

TABLE I
COMPARISON OF EVALUATION METRICS OF SIMULATION I

Item	Statistical test	ELM	DBN1	DBN2	DBN3	PEO-DBN	Statistic	p-value
ACC	FT	4.70	3.90	3.40	2.00	**1.00**	35.44	0.0000
	QT	4.73	4.04	3.24	2.00	**1.00**	22.70	0.0000
FPR	FT	4.40	4.20	3.40	2.00	**1.00**	34.24	0.0000
	QT	4.35	4.24	3.42	2.00	**1.00**	18.18	0.0000

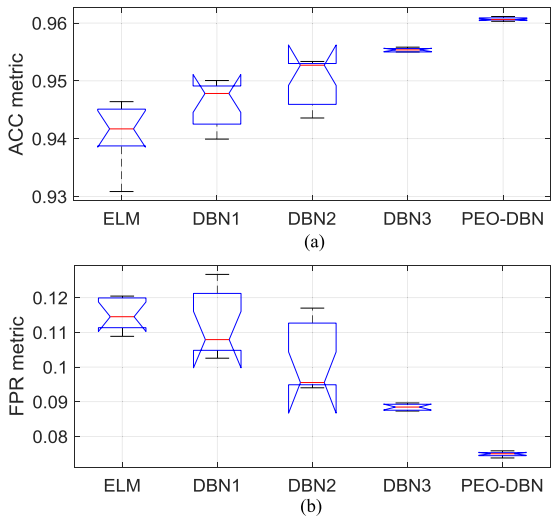

Fig. 5. Comparison of the ANOVA test results. (a) ACC metric. (b) FPR metric.

of ACC and FPR from ten runs. Besides, Table I presents the comparison results from the statistical perspective based on the FT and QT achieved by ELM, DBN1-DBN3, and PEO-DBN. Fig. 5 indicates the ANOVA comparison results of detecting performance obtained by four different methods. From the critical observation from Figs. 4 and 5 and Table I, we can see the following.

1) PEO-DBN performs better than ELM, SVM, DT, and DBN1-DBN3, with the maximum detecting accuracy, while DT method realizes the best performance in FPR. By comparing DBN1-DBN3 with ELM, SVM and DT, DBN achieves better performance in terms of ACC.

2) PEO-DBN ranks the first according to the FT and QT on ACC and FPR with p-values smaller than 0.001.

TABLE II
COMPARISON OF EVALUATION METRICS OF SIMULATION II

Item	Statistical test	PEO-DBN	PEO-DBN-SF1	PEO-DBN-SF2	EnPEO-DBN	Statistic	p-value
ACC	FT	2.45	3.90	1.95	**1.70**	17.43	0.0006
	QT	2.55	3.95	2.23	**1.27**	24.87	0.0000
FPR	FT	2.40	3.70	2.00	**1.90**	12.36	0.0062
	QT	2.47	3.87	2.27	**1.38**	17.31	0.0000

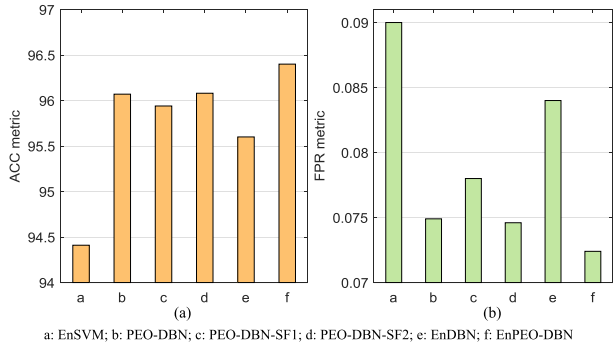

Fig. 6. ACC and FPR metric of simulation II. (a) ACC metric. (b) FPR metric.

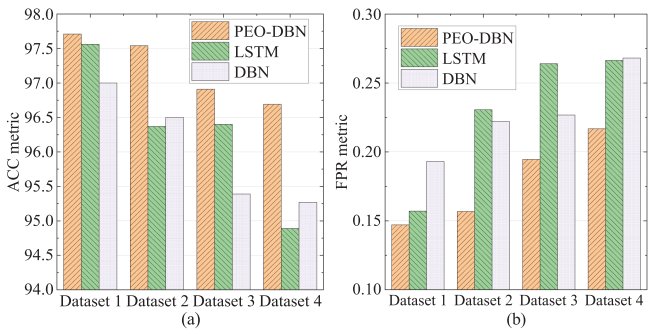

Fig. 7. Detection results of three different methods. (a) ACC metric. (b) FPR metric.

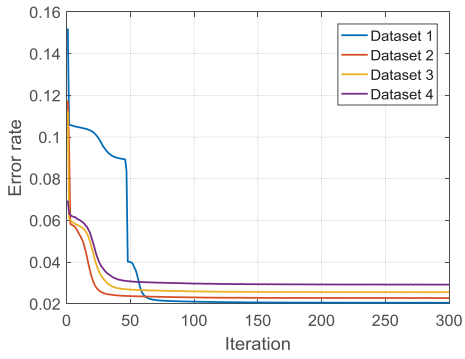

Fig. 8. Error rate of PEO-DBN under different datasets.

Moreover, DBN ranks second, followed by ELM. According to the ANOVA test results, the same conclusion can be drawn.

TABLE III
FPR METRIC OF DIFFERENT CYBER-ATTACK TYPES FOR DIFFERENT DETECTION METHODS ON SIMULATION II

Attack type	EnSVM* [7]	PEO-DBN	PEO-DBN-SF1	PEO-DBN-SF2	EnDBN* [7]	EnPEO-DBN
NMRI	0.000	6.017E-05	9.639E-05	1.209E-05	0.000	7.229E-05
CMRI	0.016	8.278E-03	8.219E-03	8.219E-03	0.013	**6.497E-03**
MSCI	0.000	1.5434E-04	3.209E-04	1.186E-04	0.000	**0**
MPCI	0.003	2.586E-03	3.244E-03	2.338E-03	0.002	**1.413E-03**
MFCI	0.000	4.728E-05	2.251E-04	2.364E-05	0.000	**0**
DoS	0.000	6.097E-04	1.743E-03	4.424E-04	0.000	**2.633E-04**
Recon	0.000	**0**	**0**	**0**	0.000	**0**

Note: "*" means the results of EnSVM and EnDBN are directly taken from [7].

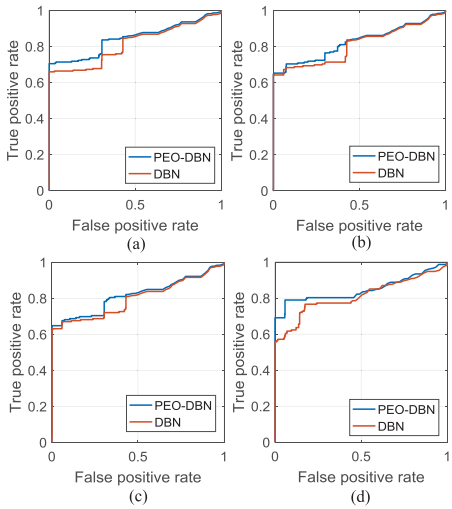

Fig. 9. ROC curve of PEO-DBN and DBN under different datasets. (a) Dataset 1. (b) Dataset 2. (c) Dataset 3. (d) Dataset 4.

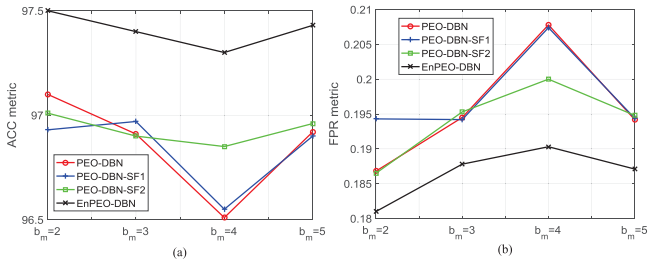

Fig. 10. Influence of mutation parameter b_m on the performance of PEO-DBN and EnPEO-DBN. (a) ACC metric. (b) FPR metric.

3) To summarize, PEO-DBN achieves better results of detecting cyber-attacks compared to SVM. And it achieves significantly better performance obtained by ELM and DBN1-DBN3 according to FT and QT for ACC and FPR.

D. Simulation II: EnPEO-DBN Versus Single Method in EnPEO-DBN, EnSVM, and EnDBN

In an effort to illustrate the superiority of the ensemble learning, we choose the three single methods in EnPEO-DBN, i.e., PEO-DBN with full features (termed as PEO-DBN), PEO-DBN with 19 features (termed as PEO-DBN-SF1), and PEO-DBN with 16 features (termed as PEO-DBN-SF2) and two existing ensemble methods, i.e., EnSVM [7] and EnDBN [7] as the competitors. Here, three different single methods can be used to prove the performance of ensemble learning. EnDBN and EnSVM are the counterpart to illustrate the superiority of EnPEO-DBN than existing ensemble methods. Fig. 6 gives the specific detecting results of average ACC and FPR metrics of EnPEO-DBN, PEO-DBN, PEO-DBN-SF1, and PEO-DBN-SF2 detection methods from ten runs as well as the performance of EnSVN and EnDBN from [7]. From Fig. 6, we can find that EnPEO-DBN achieves higher ACC and less FPR than the single method without ensemble, which indicates the ensemble strategy can further improve the detecting performance to some extent. Also, EnPEO-DBN realizes better performance than EnSVM and EnDBN. In addition, Table II shows the results of statistical tests obtained by EnPEO-DBN and three single methods. EnPEO-DBN significantly outperforms other three competitors as the related p-values are less than 0.001. To further compare, for different cyber-attacks types, Table III lists the FPR metric obtained by EnPEO-DBN, EnDBN, EnSVM, PEO-DBN, PEO-DBN-SF1, and PEO-DBN-SF2. From Table III, we can see that EnPEO-DBN performs better than other single PEO-DBN for almost all attack scenarios, although the EnPEO-DBN achieves larger FPR for NMRI than PEO-DBN-SF2. For CMRI and MPCI, EnPEO-DBN achieves better performance than EnSVM and EnDBN. For Recon, the FPR of EnPEO-DBN is the same with three different PEO-DBNs. For other attack types, the different methods obtain similar detection results in terms of FPR. Overall, the performance of EnPEO-DBN is better than single method without ensemble and two existing ensemble methods, i.e., EnSVM [7] and EnDBN [7].

E. Simulation III: PEO-DBN Versus LSTM and DBN on Water Storage Tank System

This simulation is used to show the effectiveness and applicability of the PEO-DBN by comparing with DBN and LSTM [29]. We use SCADA dataset from water storage tank system [28] and the data preprocessing is the same with gas pipeline system. Then, four different data scales are obtained, i.e., 7260 samples (termed as Dataset 1), 6534 samples (termed as Dataset 2), 5808 samples (termed as Dataset 3), and 5082 samples (termed as Dataset 4). Fig. 7 shows the performance of PEO-DBN compared with LSTM and DBN in terms of the ACC metric and FPR metric. Note that the false negative rate is obtained as 0 for all the considered methods; thus, we do not give related figure. From Fig. 7, we can see that PEO-DBN achieves better performance than LSTM and DBN under different datasets. To show how well does PEO-DBN performs, Fig. 8 gives the error rate of the proposed PEO-DBN algorithm under four datasets. To further compare PEO-DBN with DBN, four ROC curves of PEO-DBN and DBN for detection results are given in Fig. 9 under four datasets. From Fig. 9, it can be seen that PEO-DBN

achieves better performance than DBN. In addition, to show how PEO-DBN and EnPEO-DBN perform with different adjustable parameters, dataset 3 is considered to achieve this aim. In PEO-DBN, there are three adjustable parameters including $I_{\max}$, NP, and b_m in PEO, and adjustable parameters in DBN are tuned by PEO. In the real-word industrial problem, $I_{\max}$ and NP are generally used by a trial-and-error. Thus, the different mutation parameter b_m is discussed. Fig. 10 shows how the PEO-DBN and EnPEO-DBN perform with different b_m in terms of the ACC and FPR. Clearly, the performance of PEO-DBN and EnPEO-DBN is robust for parameter b_m, and they can obtain similar results with different b_m.

V. Conclusion

In this article, we proposed two novel methods (PEO-DBN and EnPEO-DBN) to detect cyber-attacks of SCADA network traffic in IACS by employing PEO algorithm and ensemble strategy. In PEO-DBN, PEO was employed for automatic parameter optimization of DBN rather other manual setting. In EnPEO-DBN, three PEO-DBNs with different features separately learned the mapping function between the traffic features and cyber-attacks types and then the majority voting scheme was used to obtain the final detection results. To verify the effectiveness of PEO-DBN and EnPEO-DBN, two real SCADA network datasets, i.e., gas pipeline system and water storage tank system, were chosen as the case studies by comparing with ELM [27], SVM [7], DT [28], DBN [18], EnSVM [7], EnDBN [7], and LSTM [29] in terms of different evaluation metrics. The simulation indicates that the proposed PEO-DBN and EnPEO-DBN can be considered as competitive cyber-attacks detection methods for SCADA-based IACS.

Although the proposed PEO-DBN can automatically tune the parameters in DBN, the fitness evaluation costs much time in the whole process of PEO. To address this problem, we can use the surrogate-assisted model, e.g., Gaussian process regression or Bayesian optimization to further accelerate the fitness evaluation in further work.

References

[1] S. M. Tahsien, H. Karimipour, and P. Spachos, "Machine learning based solutions for security of Internet of Things (IoT): A survey," *J. Netw. Comput. Appl.*, vol. 161, 2020, Art. no. 102630.

[2] M. Hassan, A. Gumaei, S. Huda, and A. Almogren, "Increasing the trustworthiness in the industrial IoT networks through a reliable cyber-attack detection model," *IEEE Trans. Ind. Informat.*, vol. 16, no. 9, pp. 6154–6162, Sep. 2020.

[3] R. Mitchell and R. Chen, "Behavior rule specification-based intrusion detection for safety critical medical cyber physical systems," *IEEE Trans. Dependable Secure Comput.*, vol. 12, no. 1, pp. 16–30, Jan./Feb. 2015.

[4] J. Weng, J. Weng, J. Zhang, M. Li, Y. Zhang, and W. Luo, "DeepChain: Auditable and privacy-preserving deep learning with blockchain-based incentive," *IEEE Trans. Dependable Secure Comput.*, to be published, doi:10.1109/TDSC.2019.2952332.

[5] M. S. Hossain, M. Al-Hammadi, and G. Muhammad, "Automatic fruit classification using deep learning for industrial applications," *IEEE Trans. Ind. Informat.*, vol. 15, no. 2, pp. 1027–1034, Feb. 2019.

[6] A. Aldweesh, A. Derhab, and A. Z. Emam, "Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues," *Knowl.-Based Syst.*, vol. 189, 2020, Art. no. 105124.

[7] S. Huda, J. Yearwood, M. M. Hassan, and A. Almogren, "Securing the operations in SCADA-IoT platform based industrial control system using ensemble of deep belief networks," *Appl. Soft Comput.*, vol. 71, pp. 66–77, 2018.

[8] G. E. Hinton, "A practical guide to training restricted Boltzmann machines," in *Neural Networks: Tricks of the Trade*. Berlin, Germany: Springer, 2012, pp. 599–619.

[9] K. Liu, L. M. Zhang, and Y. W. Sun, "Deep Boltzmann machines aided design based on genetic algorithms," *Appl. Mechanics Mater.*, vol. 568–570, pp. 848–851, 2014.

[10] W. Deng, H. Liu, J. Xu, H. Zhao, and Y. Song, "An improved quantum-inspired differential evolution algorithm for deep belief network," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 10, pp. 7319–7327, Oct. 2020.

[11] S. Boettcher and A. Percus, "Nature's way of optimizing," *Artif. Intell.*, vol. 119, no. 1/2, pp. 275–286, 2000.

[12] G. Q. Zeng, X. Q. Xie, M. R. Chen, and J. Weng, "Adaptive population extremal optimization-based PID neural network for multivariable nonlinear control systems," *Swarm Evol. Comput.*, vol. 44, pp. 320–334, 2019.

[13] C. Zhang, P. Lim, A. K. Qin, and K. C. Tan, "Multiobjective deep belief networks ensemble for remaining useful life estimation in prognostics," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2306–2318, Oct. 2017.

[14] X. Yan, Y. Xu, X. Xing, B. Cui, Z. Guo, and T. Guo, "Trustworthy network anomaly detection based on an adaptive learning rate and momentum in IIoT," *IEEE Trans. Ind. Informat.*, vol. 16, no. 9, pp. 6182–6192, Sep. 2020.

[15] M. Esmalifalak, L. Liu, N. Nguyen, R. Zheng, and Z. Han, "Detecting stealthy false data injection using machine learning in smart grid," *IEEE Syst. J.*, vol. 11, no. 3, pp. 1644–1652, Sep. 2017.

[16] D. Zheng, Z. Hong, N. Wang, and P. Chen, "An improved LDA-based ELM classification for intrusion detection algorithm in IoT application," *Sensors*, vol. 20, no. 6, p. 1706, 2020.

[17] Y. Xin *et al.*, "Machine learning and deep learning methods for cybersecurity," *IEEE Access*, vol. 6, pp. 35365–35381, 2018.

[18] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, 2006.

[19] Y. 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 Trans. Smart Grid*, vol. 8, no. 5, pp. 2505–2516, Sep. 2017.

[20] S. Manimurugan *et al.*, "Effective attack detection in Internet of medical things smart environment using a deep belief neural network," *IEEE Access*, vol. 8, pp. 77396–77404, 2020.

[21] Y. Zhang, P. Li, and X. Wang, "Intrusion detection for IoT based on improved genetic algorithm and deep belief network," *IEEE Access*, vol. 7, pp. 31711–31722, 2019.

[22] M. Jaderberg *et al.*, "Population based training of neural networks," 2017, *arXiv:1711.09846*.

[23] A. Li *et al.*, "A generalized framework for population based training," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 1791–1799.

[24] M. Jaderberg *et al.*, "Human-level performance in 3D multiplayer games with population-based reinforcement learning," *Science*, vol. 364, no. 6443, pp. 859–865, 2019.

[25] D. Ho, E. Liang, X. Chen, I. Stoica, and P. Abbeel, "Population based augmentation: Efficient learning of augmentation policy schedules," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 2731–2741.

[26] A. Gumaei *et al.*, "A robust cyberattack detection approach using optimal features of SCADA power systems in smart grids," *Appl. Soft Comput.*, 2020, Art. no. 106658, doi: 10.1016/j.asoc.2020.106658.

[27] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, no. 1–3, pp. 489–501, 2006.

[28] T. Morris and W. Gao, "Industrial control system traffic data sets for intrusion detection research," in *Proc. Int. Conf. Crit. Infrastructure Protection*, Berlin, Germany: Springer, 2014, pp. 65–78.

[29] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2015, pp. 4580–4584.

[30] U. Adhikari, S. Pan, T. Morris, R. Borges, and J. Beave, "Industrial control system (ICS) cyber attack datasets," Accessed: Apr. 2020. [Online]. Available: https://sites.google.com/a/uah.edu/tommy-morris-uah/ics-data-sets

[31] J. Alcalá-Fdez *et al.*, "KEEL: A software tool to assess evolutionary algorithms for data mining problems," *Soft Comput.*, vol. 13, no. 3, pp. 307–318, 2009.