Received August 7, 2020, accepted August 22, 2020, date of publication August 27, 2020, date of current version September 25, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3019973

Real-Time Intrusion Detection in Wireless Network: A Deep Learning-Based Intelligent Mechanism

LIQUN YANG[1], JIANQIANG LI[2], LIANG YIN[3], ZHONGHAO SUN[2], YUFEI ZHAO[1], AND ZHOUJUN LI[1,4], (Member, IEEE)

[1]State Key Laboratory of Software Development Environment, Beihang University, Beijing 100191, China
[2]National Computer Network Emergency Response Technical Team/Coordination Center of China, Beijing 100029, China
[3]Power Research Institute of State Grid Ningxia Electric Power Company, Ltd., Yinchuan 750001, China
[4]College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China

Corresponding authors: Jianqiang Li (jql@cert.org.cn), Liang Yin (realxyth@sina.com), and Zhoujun Li (lizj@buaa.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant U1636211, Grant 61672081, Grant 61370126, and Grant 61906075; in part by the Beijing Advanced Innovation Center for Imaging Technology under Grant BAICIT-2016001, in part by the Fund of the State Key Laboratory of Software Development Environment under Grant SKLSDE-2019ZX-17, in part by the Natural Science Foundation of Guangdong Province, China, under Grant 2019A1515011920, and in part by the Research on key technology of industrial control security protection for new energy grid-connection under Grant 5229DK18002V.

ABSTRACT With the development of the wireless network techniques, the number of cyber-attack increases significantly, which has seriously threat the security of Wireless Local Area Network (WLAN). The traditional intrusion detection technology is a prevalent area of study for numerous years, but it may not have a good detection performance in a real-time way. Therefore, it is urgent to design a detection mechanism to detect the attacks timely. In this paper, we exploit a CDBN (Conditional Deep Belief Network)-based intrusion detection mechanism to recognize the attack features and perform the wireless network intrusion detection in real time. To avoid the impact of the imbalanced dataset and the data redundancy on the detection accuracy, a window-based instance selection algorithm ‘‘SamSelect’’ is adopted to undersample the majority class data samples, and a Stacked Contractive Auto-Encoder (SCAE) algorithm is proposed to reduce the dimension of the data samples. By doing so, our proposed mechanism can effectively detect the potential attack and achieve high accuracy. The experiment results show that CDBN can be effectively combined with ‘‘SamSelect’’ and SCAE, and the proposed mechanism has a high detection speed and accuracy, with the average detection time 1.14 ms and the detection accuracy 0.974.

INDEX TERMS Intrusion detection, conditional deep belief network, Samselect algorithm, stacked contractive auto-encoder, real-time detection.

I. INTRODUCTION

With the extensive popularization of Wireless Local Area Networks (WLAN) technology used in hardware devices, the IEEE 802.11 protocol based short-distance transmission wireless network is facing great security challenges [1]. Any illegal users may have access to the local area network through the wireless access point [2], [3]. As the reasonable complement of the firewall, intrusion detection systems (IDS) can detect abnormal network behavior to the maximum extent [4]–[6]. It is necessary to design the effective intrusion detection mechanism for the wireless network.

The associate editor coordinating the review of this manuscript and approving it for publication was Firooz B. Saghezchi.

Various researches have been developed on proposing various IDSs in which the data-mining based methods have been proved to be very effective. Barbara *et al.* [7] proposed a wireless IDS, e.g., ADAM based on the association rule mining algorithm Apriori. In reference [8], the authors incorporated the cluster theory into Apriori algorithm to improve the efficiency of mining the association rules. To reduce the false positives, the authors in reference [9] proposed an unsupervised intrusion detection method based on sparse Auto-Encoder. To detect special attacks, Saxena *et al.* proposed an efficient IDS by analyzing the active entropy of the network abnormal traffic [10]. Other machine learning based methods, e.g., Peng proposed an IDS based on Mini Batch K-means combined with Principal Component

Analysis (PCA), and this method can be used over big data environment [11]. Farnaaz built a model for intrusion detection system using RF classifier, the model was efficient with low false alarm rate and high detection rate [12]. Besides, the deep learning based methods, e.g., Khan proposed a novel two-stage deep learning (TSDL) based approach to detect intrusion and the authors investigate the impacts on the performance of the proposed model [13]. Kim applied Long Short-Term Memory (LSTM) architecture, the experimental results show this deep learning approach was effective [14]. Kasongo proposed a Feed-Forward Deep Neural Network (FFDNN) based wireless intrusion detection method, and the experimental results show this method can achieve high detection accuracy [1]. At the same time, Al-Abassi proposed an ensemble method using Deep Neural Network (DNN) and Decision Tree (DT), and this method shows it can achieve high detection accuracy with low false-positive [12]. References [15], [16] used other deep learning algorithms to detect network intrusion manners, which can effectively identify various attacks. Although the above methods achieved good detection performance, the detection processes of these methods are offline, as a result, it is difficult to give a warning timely and to minimize the risk of network abnormal.

With the continuous expansion of network data, a large amount of non-linear network data brings new challenges to intrusion detection [17]. The performance of the existing methods heavily depends on the feature vectors in which some features may be redundant. In the wireless network environment, the commonly used experimental dataset is AWID which has the characteristics of high dimensionality and have some data redundancy [18].The high dimensional features may cause the "curse of dimensionality". To eliminate the impact of the data redundancy on intrusion detection performance, the dimension reduction methods such as principal component analysis (PCA), linear discriminant analysis (LDA), independent component analysis (ICA) and their variants have been adopted for dimensionality reduction in different research areas [19], [20]. However, these methods are linear and have good effects when the data is linear structure and Gaussian distribution. It is very difficult for these methods to find the nonlinear structure of the data when the data is with high distortion in the high dimensional space. The nonlinear dimension reduction methods such as isometric feature mapping (ISOMAP) and locally linear embedding (LLE), etc., have been adopted to improve the dimension reduction ability for the nonlinear data [21], [22]. However, these nonlinear dimension reduction methods are not effect to reduce the dimension for the new data. Interested readers can refer to [23], [24] for details on the nonlinear dimension reduction methods.

Moreover, the AWID dataset is a real-world dataset and it is imbalanced between the normal and attack samples. Using the imbalanced dataset to train the machine learning based detector will lead to the increase of the rate of false positives in minority samples [25]. To solve these issues, Reference [26] proposed a resampling strategy inside Oversampling based Online Bagging (OOB), and looked into their performance in both static and dynamic data streams. In reference [27], the authors proposed a new traffic classification method for imbalanced network data, by introducing and improving Synthetic Minority Oversampling Technique (SMOTE) algorithm, an improve SMOTE algorithm was proposed to realize the balance of traffic data [28]. However, these methods need to synthesize the minority-class samples, and this will lead to the minority-class samples tend to overlap with the majority-class samples [29].

II. MOTIVATION AND CONTRIBUTIONS

Based on the above analyses, we can see that there are some major challenges to develop an effective wireless intrusion detection mechanism: (1) When dealing with the AWID dataset, how to process the high-dimensional data samples in case of the "curse of dimensionality"? (2) Due to the AWID dataset is imbalanced, how to balance the dataset in case of leading to the over-fitting problem and improve the detection performance? (3) How to design a detect model to detect the abnormal in a real-time way with a satisfactory detection performance?

Motivated by the above challenges, in this paper, a wireless network intrusion detection mechanism based on Conditional Deep Belief Network (CDBN) that is composed of the Conditional Gaussian-Bernoulli RBM (CGBRBM) is proposed to detect the network abnormal in a real-time way. To overcome the amount imbalance between the normal data and the attack data in AWID training dataset, a window-based under-sampling selection algorithm "SamSelect" is adopted to balance the dataset. In addition, to overcome the disadvantages of the existing methods in data dimension reduction, a Stacked Contractive Auto-Encoder (SCAE) is proposed to reduce the dimension of the data sample. The main contributions of this paper can be summarized as:

1) This work is among the pioneer studies of using CDBN in wireless network intrusion detection research. In addition, the proposed mechanism is performed in a real-time way, which is novel and effective.

2) To avoid the impact of the excessive normal samples in AWID dataset on training the detection model. It is the first time that adopting the "SamSelect" algorithm to balance the dataset by under sampling the normal samples. The dimensionality of the feature vector is reduced by an improved Auto-Encoder (SDAE) method, which is novel.

3) We evaluate the performance of the proposed intrusion detection mechanism on two dataset (AWID, LITNET) processed by "SamSelect" and SCAE. The experimental results show the proposed mechanism has high detection accuracy.

4) The impact of the time observation window size of CDBN on the detection performance is carefully studied. Moreover, the robustness of our proposed mechanism to the noise is also investigated in this paper.

The remaining parts of this paper are organized as follows. Section 3 introduces the overview of our proposed IDS, and the data pre-processing methods including "SamSelect" algorithm and the proposed SCAE algorithm. Section 4 elaborates the implementation of the proposed real-time CDBN based IDS. Simulation results and conclusions are presented in Sections 5 and 6, respectively.

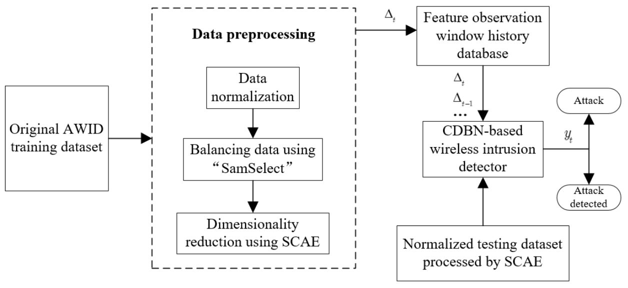
Original AWID training dataset
Data preprocessing
Data normalization
Balancing data using "SamSelect"
Dimensionality reduction using SCAE
Δ_t
Feature observation window history database
Δ_t
Δ_{t-1}
...
CDBN-based wireless intrusion detector
y_t
Attack
Attack detected
Normalized testing dataset processed by SCAE

FIGURE 1. Overview of the proposed detection mechanism.

III. OVERIVEW OF OUR PROPOSED MECHANISM AND DATE PRE-PROCESSING METHODS

A. OVERIVEW OF THE PROPOSED INTRUSION DETECTION MECHANISM

The wireless intrusion detection can be regarded as a multiple classification problem. As depicted in Fig. 1, to effectively train the CDBN detector based on the balanced training dataset, the AWID training dataset is firstly normalized, and then we adopt the "SamSelect" algorithm to select the normal samples. By doing so, the number of the normal samples will be basically equal to that of the attack samples. To eliminate the redundancy of the data in AWID dataset as well as improve the response speed of intrusion detection, the balanced training dataset and the testing dataset will be inputted into the SCAEs model to generate the low-dimensional datasets. At last, the CDBN model will be trained by the pre-processed dataset, by setting the observation window Δ and inputting the testing data, the attack will be identified by the output label y_t in a real-time way.

Different from the existing publications which adapt CDBN to model human motion, we would like to clarify that there are two main differences between our proposed mechanism and [30], [31]. The CDBN in our paper is designed as a classifier while that in the two references is a time-series generative model to predict the human motion. Moreover, the CDBN in our paper employs only one CGRBM as the first hidden layer and uses the conventional RBMs as the other hidden layers. Moreover, the CDBN adapt in the two references employ CGRBMs as all the hidden layers.

B. DATA NORMALIZATION AND DATA BALANCE PROCESSING

Due to the AWID dataset contains qualitative and quantitative data we should standardize it first. We use "factorzie" method of "pandas lab" in python, to map the symbol value attributes to integer values [32]. Some attributes with hexadecimal data type are also converted to integer values. Some attributes retain the continuous data type, and the data also contains unusable marks such as "?" which will be set to 0. After converting all attribute values to integer values, we use the following equation to normalize each attribute value with the range of [0, 1].

$$y = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{1}$$

where x is the standardized AWID data. The AWID training dataset contains 1795575 samples, of which 1633190 normal samples and 162385 attack samples [25]. The AWID training dataset is imbalanced which may cause the decrease of detection performance of the CDBN detector. It is necessary to select the instances from the AWID training dataset and consequently effectively suppress the dataset imbalanced problem. To ensure the number of the attack samples basically equals to that of the normal samples, the normal samples are under sampled. Due to the normal samples are distributed in all the sampling period, we use the window-based algorithm "SamSelect" [33] to collect the normal samples. Assuming that the sample sequence as $T = \{T_1, T_2, \ldots, T_n\}$, and the current sample as T_t and the "SamSelect" can be summarized as Algorithm 1.

Algorithm 1 Data Balance Using "SamSelect"

```
Input: AWID training dataset $D_A$, window size $\omega$, normal sample counter $c = 0$
Output: Under-sampling dataset $D_B$
for $t = 1$ to $|D_A|$ do:
  if $T_t$ is normal then:
      $c = c + 1$
    if $c \leq \omega$ then:
      put this sample into $D_B$
    end if
  end if
  if $T_t$ is not normal then:
    $c = 0$
    put this sample into $D_B$
  end if
end if
return $D_B$
```

C. DATA DIMENSIONALITY REDUCTION BASED ON SCAE

Auto-Encoder (AE) is an unsupervised learning neural network, which reconstructs the input data as much as possible. Two main processes involved in training AE, which are pre-training that initializes the network weights using the L-BFGS algorithm and fine-tuning that adjusts the network parameters using the BP (backward propagation algorithm) algorithm [34]. AE can achieve the dimensionality reduction function when the dimension of reconstructed data is smaller than that of the original data. By decreasing the unrelated and redundant features in the feature vector we can get more abstract high-level and low-dimension representations of the original feature data [35]. The structure of AE is composed of input layer, output layer, and hidden layer. The input layer and the hidden layer constitute the Encoder, and the input data is compressed in the encoder. The hidden layer and the output layer constitute the Decoder, and the input data is reconstructed in the Decoder. The encoding and decoding processes are shown in Fig. 2. f is a non-linear encoding function. After encoding, the n-dimensional input can be mapped into a m-dimensional hidden layer vector. Afterwards, the hidden layer vector is reversely reconstructed to n-dimensional vector z through the decoding function g. Define $\sum_{x\in X} L(x,z)$ as the loss function, where X is the dataset. The training process of AE is to minimize the loss function which can be solved by BP algorithm [34].

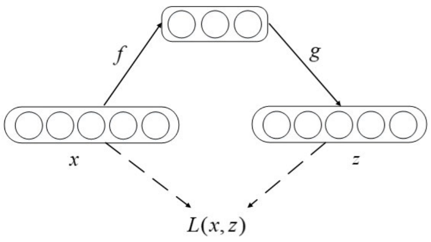

FIGURE 2. The processes of encoding and decoding.

After encoding, x is compressed into y, and after decoding, y is further reconstructed into z. Since the reconstruction process ensures that error is minimized, z not only retains the features of x, but also has lower dimension than x. The Contractive Auto-Encoder (CAE) [36] is proposed to ensure the reconstructed data contain the distribution characteristic of the input data and remove the noise. The loss function of CAE can be rewritten as follows:

$$\sum_{x\in X} L(x, g(f(x))) + \Omega(h,x) \tag{2}$$

where Ω is the square Frobenius norm which is defined as

$$\Omega(h,x) = \lambda \left\| \frac{\partial f(x)}{\partial x} \right\|_F^2 \tag{3}$$

To learn the deep-level representations of the original data, in this paper, we use the output of the previous CAE as the input of the next CAE to form the stacked CAE (SCAE), and achieve the purpose of layer-by-layer encoding. Define x_i as the encoded data of the ith CAE, and the encoded data of each CAE can be defined as follows:

$$x_i = f_\theta(x_{i-1}) \quad i = 1, 2, 3\ldots \tag{4}$$

After layer-by-layer greedy training and fine- the SCAE dimensionality reduction model is constructed. The difference between the original input data and the reconstruction training weights is minimized to achieve the initial parameters in the greedy training process. Fig. 3 is the SCAE dimensionality reduction model.

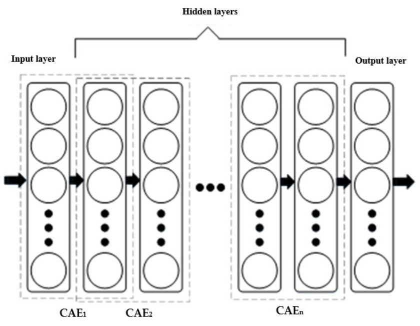

FIGURE 3. The dimensionality reduction model based on SCAE.

In the fine-tuning process, to ensure the reconstruction error is minimized, the BP algorithm is adopted to adjust the cross-entropy of the initial parameters. After the fine-tuning in the first CAE, the hidden layer output is calculated and inputted into the second CAE. Then the parameters of the second CAE can be obtained by training and fine-tuning. By doing like this, the SCAE model is trained by training and fine-tuning all the CAEs.

IV. REAL-TIME WIRELESS INTRUSION DETECTION MECHANISM

A. THE STRUCTURE OF CDBN DETECTOR

The CDBN structure is integrated by the standard deep belief network structure with one CGBRBM and stacking by few RBMs [36]. The CGBRBM can capture the correlations between the historical and the current input data. Based on CGBRBM, the proposed CDBN detector can effectively learn the temporal behavior features of the pre-processed dataset and detect the attacks in a real-time way. The input data of CGBRBM is a time-series data and the CGBRBM can capture the correlations between the historical and the current input data. After training the CDBN model, the testing data will be inputted to the CDBN model step by step by time instance instead inputting the whole testing dataset, so this is a real-time way to detect the newly inputted data. As illustrated in Fig. 4, our proposed CDBN-based detector employs the CGBRBM unit as the first layer and on the top of the CGBRBM there are N − 1 conventional RBMs, so there are N hidden layers in the whole CDBN architecture. We would like to clarify that a multiple classifier output unit is added on the top of the CDBN architecture, which can output the classification label and indicate whether the inputted data is an attack sample. In the next two subsections, we will introduce the pre-training and fine-tuning process of CDBN.

B. THE PRE-TRAINING PROCESS OF CDBN

CDBN uses the pre-training process to initialize the network parameters which are the connection weights between layers and the offset values of each layer neurons. Take a RBM as an example, which contains a visual layer with m visible units and a hidden layer with n hidden units. The energy function

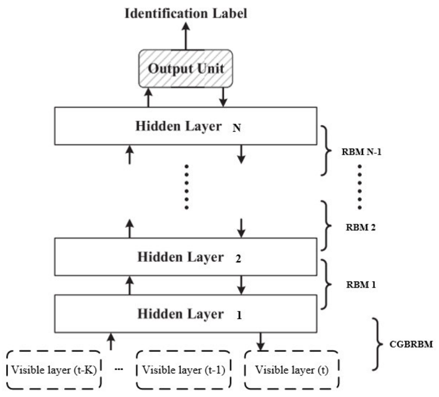

FIGURE 4. The structure of CDBN architecture.

of a conventional RBM can be defined as follows:

$$E(v,h) = -\sum_{i=1}^{n}\sum_{j=1}^{m} w_{ij}h_i v_j - \sum_{j=1}^{m} c_j v_j - \sum_{i=1}^{n} d_i h_i \tag{5}$$

where v_j is the *jth* element of the visible layer vector, and h_i is the *ith* element of the hidden layer vector, w_{ij} is the *ijth* element of the weight matrix between the visible and hidden units. We define d_i and c_j as the *ith* element of bias vector for the hidden layer and the *jth* element of the bias vector for the visible layer, respectively. Based on Eq. (5), given the adjacent layer unit values, the activation conditional probability distributions of hidden and visible units are calculated as follows:

$$\begin{cases} p(h_i = 1|v) = sigm(d_i + \sum_{j=1}^{m} w_{ij}v_j) \\ p(v_j = 1|h) = sigm(c_i + \sum_{i=1}^{n} w_{ij}h_i) \end{cases} \tag{6}$$

where $sigm(.)$ is sigmoid function. By using the CD (gradient-based Contrastive Divergence) method [37], the weights and biases of the conventional RBMs are updated as follows:

$$\begin{cases} w_{ij} = w_{ij} - \alpha(\langle v_j h_i \rangle_m - \langle v_j h_i \rangle_l) \\ d_i = d_i - \alpha(\langle h_i \rangle_m - \langle h_i \rangle_l) \\ c_j = c_j - \alpha(\langle v_j \rangle_m - \langle v_j \rangle_l) \end{cases} \tag{7}$$

where α is the learning rate, and $\langle . \rangle_m$ and $\langle . \rangle_l$ are the expectations computed over the data and model distributions.

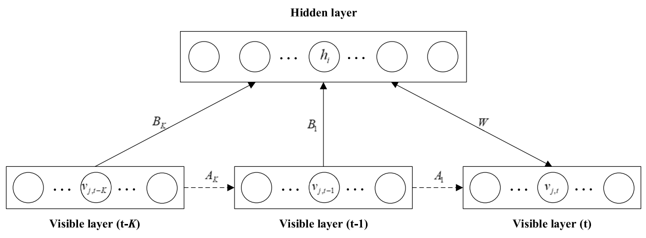

FIGURE 5. The structure of CGBRBM for CDBN.

Figure 5 illustrates the structure of the CGBRBM with one hidden layer and K+1 visible layers. Define K as the size of the time observation window. Same like Eq. (5), the energy function of CGBRBM is defined as follows:

$$E(v_t,\ldots,v_{t-K},h) = -\sum_{i=1}^{n}\sum_{j=1}^{m}\frac{v_j}{\widehat{\sigma}_j^2}h_i w_{ij} - \sum_{i}^{n} d_{i,t}h_i + \sum_{j}^{m}\frac{(v_{j,t}-c_{j,t})^2}{2\widehat{\sigma}_j^2} \tag{8}$$

where v_j and h_i are the *jth* element of the layer visible vector and the *ith* element of the hidden layer, respectively. w_{ij} is the *ijth* element of the weight matrix between the visible layer units and hidden layer units, $\widehat{\sigma}_j$ is the standard deviation of the *jth* element of the visible vector, n and m are the number of the hidden units and number of the visible units, respectively, Define b and c as the bias vector of hidden layer vector and the bias vector of the visible layer, d_t and c_t are calculated as follows:

$$\begin{cases} d_t = d + \sum_{k=1}^{K} v_{t-k}B_k \\ c_t = c + \sum_{k=1}^{K} v_{t-k}A_k \end{cases} \tag{9}$$

where v_{t-k} is the *kth* previous visible layer vector. Based on Eq. (8), the conditional probability distributions of the hidden and visible layer units can be calculated as follows:

$$\begin{cases} p(h_i = 1|v_t,\ldots,v_{t-N}) = sigm(d_{i,t} + \sum_{j=1}^{m}\frac{w_{ij}v_{j,t}}{\widehat{\sigma}_j^2}) \\ p(v_{j,t} = v|h) = N(c_{j,t} + \sum_{i=1}^{n} w_{ij}h_i, \widehat{\sigma}_j^2) \end{cases} \tag{10}$$

Applying the gradient-based CD technique, the structure of the CGBRBM can updated as follows:

$$\begin{cases} w_{ij} = w_{ij} - \alpha(\left\langle \frac{v_{j,t}}{\widehat{\sigma}_j^2}h_i \right\rangle_m - \left\langle \frac{v_{j,t}}{\widehat{\sigma}_j^2}h_i \right\rangle_l) \\ a_{ijk} = a_{ijk} - \alpha(\left\langle \frac{v_{j,t-k}}{\widehat{\sigma}_j^2}v_{i,t} \right\rangle_m - \left\langle \frac{v_{j,t-k}}{\widehat{\sigma}_j^2}v_{i,t} \right\rangle_l) \\ b_{ijk} = b_{ijk} - \alpha(\left\langle \frac{v_{j,t-k}}{\widehat{\sigma}_j^2}h_i \right\rangle_m - \left\langle \frac{v_{j,t-k}}{\widehat{\sigma}_j^2}h_i \right\rangle_l) \\ d_i = d_i - \alpha(\langle h_i \rangle_m - \langle h_i \rangle_l) \\ c_{j,t} = c_{j,t} - \alpha(\left\langle \frac{v_{j,t}}{\widehat{\sigma}_j^2} \right\rangle_m - \left\langle \frac{v_{j,t}}{\widehat{\sigma}_j^2} \right\rangle_l) \end{cases} \tag{11}$$

Define the weight matrices as W, A_k and B_k in which the elements are defined as w_{ij}, a_{ijk} and b_{ijk}, respectively. Define the $\langle . \rangle_l$ and $\langle . \rangle_m$ as the expectations calculated by the data and model distributions. After pre-training, we add a fully connected output node on the top of the model. To present the two labels indicating the attack and the normal samples, the output node is designed as a multiple node with sigmoid activation function defined in Eq. (6). After the above operations, the model will be fine-tuned using back-propagation supervised training with the available labeled data to fully achieve the trained structure of the neural network [38].

C. THE FINE-TUNING PROCESS OF CDBN

After pre-training process, the fine-tuning procedure is used to adjust the parameters such as the weights and biases. Take the *hth* hidden layer as an example, and define the learning rate as η, the weight matrix and the bias vector of it can be updated as follows:

$$\begin{cases} \Delta W_{h,i,j} = -\eta \delta_{h,i} p_{h-1,j} \\ \Delta d_{h,i} = -\eta \delta_{h,i} \end{cases} \tag{12}$$

where $\Delta W_{h,i,j}$ and $\Delta d_{h,j}$ are the updated value for the *ijth* element of the weight matrix and for the *jth* element of the bias vector, respectively. $p_{h-1,j}$ is the activation probability of the *jth* element of the $(h-1)$*th* hidden layer, and

$$\delta_{h,j} = p_{h,j}(1-p_{h,j}) \sum_{k}^{M} \delta_{h+1,k} W_{h+1,j,k} \tag{13}$$

where M is the number of elements in the $(h+1)$*th* hidden layer. $W_{h+1,j,k}$ and $p_{h,j}$ are the *jkth* element of the weight matrix of the $(h+1)$ hidden layer and the activation probability of the *jth* element of the *hth* hidden layer, respectively. Same like Eq. (12), the weight vector and the bias value of the output layer with single-unit are updated as follows:

$$\begin{cases} \Delta W_{o,j} = -\eta \delta_o p_{H,j} \\ \Delta d_o = -\eta \delta_o \end{cases} \tag{14}$$

where $\Delta W_{o,j}$ is the updated value for the *jth* element of the weight vector, Δd_o is the updated value for the bias, and Δd_o is the updated value for the bias, $p_{H,j}$ is the activation probability of the *jth* element of the last hidden layer whose index is $h = H$ and

$$\delta_o = p_o(1-p_o)(l_o - L) \tag{15}$$

where l_o and L are the predicted output label and the actual value of the output label, respectively. p_o is the activation probability of single output unit. By doing the processes in Section III-B and III-C, we can train the CDBN model which is subsequently used for detecting the attack in wireless network environment.

V. CASE STUDY

A. EXPERIMENT PREPARATION

In this paper, the AWID-CLSR-R-Trn and AWID-CLSR-tst datasets are adopted to train the CDBN detector and to test the detection performance, respectively [39]. The AWID-CLSR-R-Trn dataset contains 1795575 instances, including 1633190 normal samples and 162385 attack samples. The AWID-CLSR-tst dataset contains 575643 samples, including 530785 normal samples and 44858 attack samples. We would like to clarify that in this paper we use "SamSelect" algorithm to undersample the AWID-CLSR-R-Trn dataset regardless of the binary class and multiclass. Table 1 illustrates the distributions of different types of attacks in the training and test datasets.

TABLE 1. The attack distributions of training and testing dataset.

Attack type	AWID-CLSR-R-Trn	AWID-CLSR-tst
False attack	48522	20079
Flooding attack	48484	8097
Injection attack	65379	16682
Total	162385	44858

According to the above analysis, we can see that the AWID-CLSR-R-Trn dataset is imbalanced in which the number of the normal samples is larger than that of the attack samples, and the ratio between normal and attack samples is about 10:1. Using the imbalanced dataset to train the detector may cause the over-fitting problem and thereby reduce the detection accuracy [40]. After normalizing the AWID-CLSR-R-Trn dataset, the normal samples are undersampled by the window-based "SamSelect" algorithm. To balance the dataset as much as possible, the window size of "SamSelect" should be carefully investigated. As shown in Table 2, we can see that the dataset is almost balanced when the window size is set to 2.

TABLE 2. The relation between the window size and sample size.

Sample Size	Window Size			
	5	4	3	2
Normal sample	368038	304562	266524	201007
Attack sample	162385	162385	162385	162385

After balancing the training dataset, there are 201007 normal samples and 162385 attack samples in AWID-CLSR-R-Trn dataset and these data will be adapt to the following experiments. In this paper, the average results of 10-fold cross-validation of 10 times are used to evaluate the performance of our proposed mechanism.

B. THE FEASIBILITY ANALYSIS OF SCAE ON DIMENSIONALITY REDUCTION

In this subsection, we will investigate the feasibility of SCAE on dimensionality reduction. The essence of SCAE on dimensionality reduction is to verify whether the low-dimensional data outputted from SCAE can represent the original information contained in the input data, in other words, to verify the difference between the reconstructed data and original data. We use the Mean Squared Error (MSE) to evaluate the reconstruction performance during the training process [41]. The MSE is defined as follows:

$$MSE = \frac{1}{N} \sum_{i=1}^{N} (y_{input} - y_{recon})^2 \tag{16}$$

where y_{input} is the original input data and y_{recon} is the output data reconstructed by SCAE, respectively. N is the number of the samples. After normalizing the original AWID data, we can get 154 features for each AWID sample. In this experiment, the 154-dimensional sample is used as the input of SCAE, and we set 20 as the dimension of the output reconstructed data. In this paper, we design a 4-layer SCAE network structure, CAE1 154-120, CAE2 120-80, CAE3 80-50, CAE4 50-20, the numbers of the neurons in each hidden layer are: 120-80-50-20. The training times for each CAE and SCAE are set to 10.

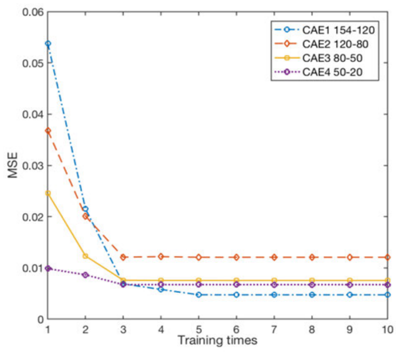

FIGURE 6. The reconstruction errors of CAEs.

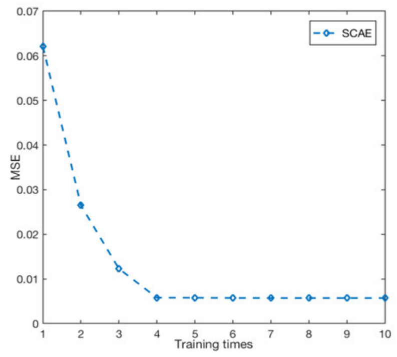

FIGURE 7. The reconstruction errors of SCAE.

From Fig. 6 and Fig. 7, we can see that that after 5 training times, all the reconstruction errors of the MSEs tend to be stable with the values below 0.015. Moreover, the MSE of SCAE also tends to be stable and less than 0.007 when the training time is greater than 4. Therefore, we can conclude that the 20-dimensional reconstructed data can basically represent the original data with a relative small reconstruction error, which proves that the feasibility of SCAE on dimensionality reduction.

C. THE IMPACT OF THE TIME OBSERVATION WINDOW SIZE ENVIRONMENT NOISE ON DETECTION PERFORMANCE

In this subsection, we continually investigate the impact of the time observation window size of the CDBN on the performance of our proposed detection mechanism. As shown in Table 3, we introduce the evaluation matrices such as Accuracy (Acc), Precision (Pre), Recall, F1, and MCC (Matthews correlation coefficient) [42].

TABLE 3. Evaluation metrics.

Precision	$TP/(TP+FP)$
Recall	$TP/(TP+FN)$
Mcc	$(TP*TNFP*FN)/((TP+FP)(TP+FN)(TN+FP)(TN+FN))^{0.5}$
Acc	$(TP+TN)/(TP+FN+TN+FP)$
F1	2*Precision*Recall/(Precision+Recall)

For the following simulations, we need to determine the best time observation window because larger window may learn more temporal information in the input sequence. Hence, it is important to investigate the impact of the time observation window size on the detection performance. Define the time observation window size as Δ which ranges from 2 to 5 with the increment $s = 1$. In this experiment, we set the number of hidden layers to 5. As shown in Fig. 8, the detection performance is the best when the time observation window size $\Delta = 4$. At the same time, The accuracy for Normal sample, Flooding attack, False attack and Injection attack is 0.989, 0.808, 0.727, 0.991.

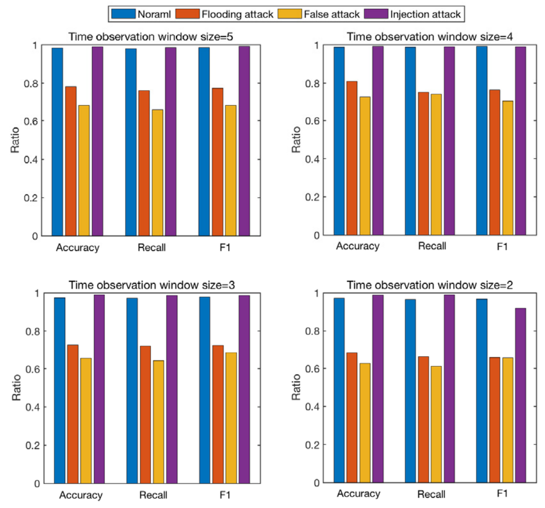

FIGURE 8. The detection performance with different time observation window sizes.

To study the robustness of the proposed mechanism to the environment noise, we firstly model the environment noise as the Additive White Gaussian Noise (AWGN) $\sim N(0, \sigma)$ with the standard deviation σ ranging from 0.5 to 3, and add the noise to the training data. Then the training data is preprocessed by "SamSelect" and SCAE. In this experiment, we use the preprocessed training data to train the CDBN-based, DBN-based [43] and RNN (Recurrent Neural Network)-based [44] detectors, by comparing the detection accuracy of the three detection models, from Fig. 9, we can see that our detection method can achieve the highest detection accuracy among the three different methods. Furthermore, as the standard deviation σ increases the detection accuracy decreases our proposed detection mechanism can achieve the accuracy of detection above 75%. This demonstrates that our proposed mechanism is robustness to the environment noise.

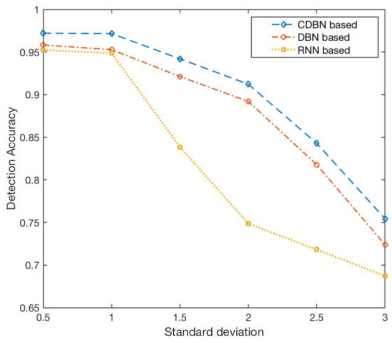

FIGURE 9. The detection accuracy with different noise level.

D. THE IMPACT OF THE CDBN STURCTURE AND DATA PREPROCESSING OPERATION ON DETECTION PERFORMANCE

We continue to numerically investigate the impact of the CDBN's structure on detection performance. In this experiment, the learning rates of pre-training and fine-tuning process of CDBN are both set to 0.5, and the size of mini-batch is 100, and the time observation window size is set to 5. We assume that each RBM has the same number of the hidden units 128, and the number of CDBN hidden layers ranging from 1 to 5 with an increment 1. The dimension of the sample in this experiment is reduced to 20. We will investigate the impact of the CDBN structure on the detection performance on different datasets, e.g., the datasets processed by "SamSelect" algorithm with different window size {5, 4, 3, 2}. The detection accuracy of the CDBN based architecture with different numbers of hidden layers is shown in Fig. 9 and Fig. 10. We can see that the detection accuracy goes up as the number of the hidden layers increases. When the number of the hidden layers is bigger than 5, the detection accuracy does not change much. However, the computational complexity will increase as the number of hidden layers increases. To achieve a good balance between detection accuracy and the computational complexity, we can say that the number of the hidden layers can be set to, and meanwhile the detection accuracy is 97.38%.

At this time, the learning rates for all the training procedures of RBMs and CGBRBM α are set to 0.1. The learning rate η for the fine-tuning procedure is set to 0.1. By doing the experiments, we set the number of hidden layers to 5. In the following simulation, we use the CDBN having 5 hidden layers and 50 units for each hidden layer.

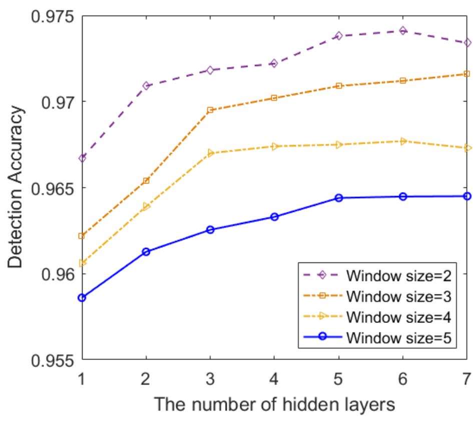

FIGURE 10. The impact of CDBN structure on detection performance.

The activation functions are chosen as the Sigmoid function. The momentum is set to 0.9, the minibatch is set to 100 and the number of initial iterations is set to 100. Moreover, it is easily found that when the window size of "SamSelect" is 2, the detection accuracy is the highest because the CDBN-based detector is trained by the most balanced dataset.

Furthermore, we do the 10-fold cross validation for 10 times to calculate the average detection accuracy and the detection error and then present the confidence limits for the detection error. The detection error is the percentage of the incorrect predictions from all predictions made and can be defined as follows: $Err = (FN + FP)/\ (TP + FN + TN + FP)$. The confidence interval can be calculated and presented as part of the model evaluation. A confidence interval is comprised of two things: Range is the lower and upper limit that can be expected on the model. Probability is the probability that the skill of the model will fall within the range. The confidence limits for detection error can be calculated as follows:

$$Err + / - const * \sqrt{(Err * (1 - Err))\ /n} \tag{17}$$

where *const* is a constant value that defines the chosen probability, and n is the number of the samples used to evaluate the model. The values for *const* are provided from statistics, and in our paper we choose 1.96 as the const which corresponds to the probability 95%. In this experiment, the window size of "SamSelect" is set to 2, and the hidden layer is set to 5, the time observation window size of CDBN is set to 4 and n is 36339. Under these conditions, the detection error is 7.147% and the confidence limits can be calculated as: 0. 07147+/−1.96*sqrt((0.07147*(1−0.07147))/36339), and the lower and upper limits are 0.07012 and 0.07282. For the sake of simplification, in this experiment we just investigate the confidence limits for the binary classification.

In addition, we evaluate the impact of the data preprocessing operations on detection performance. We investigate the detection performance of four detection mechanisms including CDBN, SamSelect+CDBN, SCAE+CDBN, SamSelect+SCAE+CDBN in which the

TABLE 4. Date pre-processing operations on detection performance.

Detection mechanism	Accuracy	Recall	F1	Mcc
CDBN	0.873	0.864	0.854	0.806
SamSelect+CDBN	0.965	0.967	0.962	0.887
SCAE+CDBN	0.906	0.891	0.894	0.845
SamSelect+SCAE+CDBN	0.974	0.976	0.971	0.914

order of the algorithms indicates the execution order in the detection mechanism. In this experiment, the number of hidden layers of CDBN is set to 5. The results of the data pre-processing operations on the detection performance are illustrated in Tab. 4. We would like to clarify that the metrics In Tab. 4 are calculated in a micro-average way, which are detailed explained and defined in [45]. From Tab. 4 we can conclude that the mechanism SamSelect+SCAE+CDBN can achieve the best detection performance. This is because the original high dimensional data contains more redundant features which can't be used to train the perfect detector.

We continually investigate the detection performance of our proposed mechanism using the newly developed dataset LITNET. There are 12 types of attack e.g., Smurf, ICMP flood, TCP flood, UDP flood, Http flood, Land attack, Blaster worm, Code Red worm, SPAM Reaper worm, Reaper worm, Scanning attack, Fragmentation attack. Each type of attack contains 85 network flow features [46]. The normal and under-attack flow datasets can be downloaded from https://dataset.litnet.lt, and the two datasets are reconstructed one dataset including the normal and under-attack data. After preprocessing these dataset, we use the average result of 10-fold cross validation of 10 times to evaluate the detection performance of our proposed mechanism. As shown in Tab. 5, we can see that our proposed mechanism can effectively detect the attack especially for the DoS attacks such as smurf, flooding and Land attack. This is because these attacks have more significant network flow features compared with other types of attacks.

TABLE 5. The detection accuracy of LITNET dataset.

Type of Attack	Detection Accuracy
Smurf	0.984
ICMP flood	0.972
UDP flood	0.989
TCP SYN flood	0.981
HTTP flood	0.967
LAND attack	0.991
Blaster Worm	0.967
Code Red Worm	0.974
Spam bot's detection	0.872
Reaper Worm	0.887
Scanning/Spread	0.914
Packet fragmentation attack	0.824
Normal	0.988

E. THE OVERALL DETECTION PERFORMANCE OF THE PROPOSED DETECTION MECHANISM

In this paper, we would like to calculate the confusion matrixes of the performance metrics based on testing the

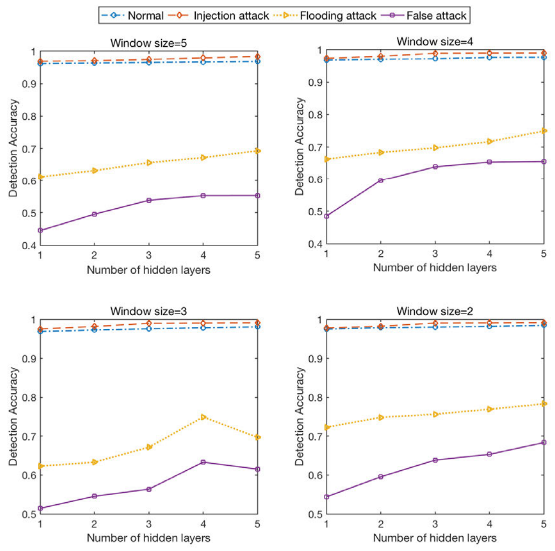

FIGURE 11. Detection accuracy of different attack types under different window size of "SamSelect".

AWID-CLSR-tst dataset. That is to say there are 530785 normal data and 44858 attack data including 20079 false attack data, 8097 Flooding attack data, 16682 Injection attack data in the testing dataset. The confusion matrix is shown as Tab. 6. From Tab. 6 we can see that the proposed mechanism can easily detect the normal data with a low false alarm rate. For different attacks, the mechanism is effective to detect inject attack but has a relatively low detection accuracy for the flooding attack and false attack. Moreover, we compare the proposed detection mechanism with the similar methods, e.g., the data balancing method SMOTE (Synthetic Minority Oversampling Technique) and the dimensionality reduction method PCA (Principal Component Analysis). From Fig. 12, we can see that the proposed mechanism is better than the combination of SMOTE, PCA and CDBN. It is clear that SCAE is better than PCA in reducing the dimension of the experimental dataset, and SCAE+CDBN is more effective than PCA+CDBN. Similarly, the dataset processed by "SamSelect" is more effective than that processed by SMOTE.

At last, we validate the advantages of our detection mechanism by comparing with RNN-based detection mechanism and DBN-based detection mechanism using the Receiver Operating Characteristic (ROC) [47] curve which is plotted in Fig. 13. True Positive Rate (TPR) is defined as the probability that the attack data is identified to be attack.

TABLE 6. The fusion matrix of the propose detection mechanism.

	Normal	Flooding Attack	Injection Attack	False Attack
Normal	522864	3047	1140	3824
Flooding Attack	1174	6341	119	463
Injection Attack	103	0	16542	37
False Attack	3574	476	2301	13728

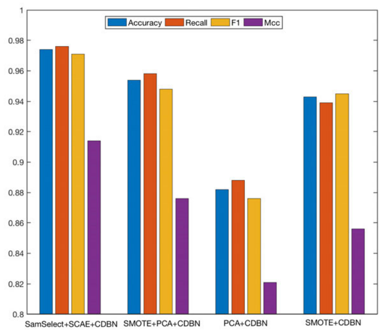

FIGURE 12. The detection performance with different methods.

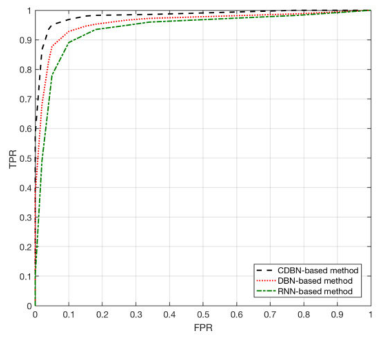

FIGURE 13. The detection accuracy with different noise level.

False Positive Rate (FPR) is defined as the probability that the normal data is identified to be attack [48]. From the result shown in Fig. 13, we can observe that the proposed mechanism can achieve the best performance, and the area under the curve which is called AUC equals to 0.978, the AUC of DBN-based method is bigger than that of the RNN-based method. We can conclude that our proposed detection mechanism is superior to RNN-based and DBN-based methods.

Moreover, the performance comparisons with other shallow learning methods are also investigated. We compare the proposed mechanism with the existing shallow methods such as SVM (Support Vector Machine) [49] and Logistic Regression (LR) [50] using the WEKA machine learning platform [51]. From Figure 14, it is clear that the proposed mechanism can remarkably outperform SVM- and LR-based detection mechanisms. According to the above analysis, we can also conclude that the detection performance of the deep learning based methods can get a better detection performance compared with the shallow learning based methods. This is because that the deep learning based methods can learn the essential features of the dataset. We would like to clarify that the ROC curves in Fig. 13 and 14 are for the binary classification which means that they are plotted by analyzing the results on detecting the attack and normal samples.

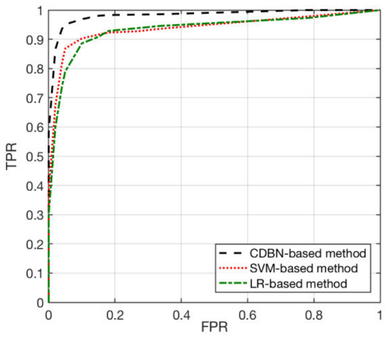

FIGURE 14. The ROC curves of three detection methods.

Our detection scheme employs the CDBN to recognize the high-dimension features of attacks. To achieve this goal, our proposed detection mechanism consists of two essential mechanisms: (1) Preprocess the original dataset and train the CDBN-based detector (2) Detect the potential FDI attacks by using the trained model. We would like to clarify that these two essential mechanisms are implement in parallel, and the dataset is ready-made and it is firstly processed and stored in the feature history database. Therefore, the computing time of the preprocessing and training procedures do not lead to any delay in the detecting procedure. The average time consumption on balancing and reducing dimensionality for each data sample is about 4.7ms. Moreover, like the reference [52], our proposed detection mechanism does not consider the time consumption of data preprocessing. The simulation time of our proposed scheme to detect the unobservable FDI attacks is 1.14ms, which can satisfy the requirement of the real-time detection.

TABLE 7. The results of different detection performance indicators.

Method	Precision	Accuracy	Recall	F1	Mcc
CDBN-based	0.966	0.974	0.976	0.971	0.914
LR-based	0.930	0.934	0.936	0.933	0.881
SVM- based	0.937	0.948	0.946	0.942	0.894

Moreover, we continually to evaluate the performance of our detection scheme by comparing with SVM- and LR-based methods. In this experiment, we use the performance indicators such as Precision, Recall, Mcc, Acc to illustrate the results. From Tab. 7 we can see that the proposed mechanism can achieve the best detection performance.

VI. CONCLUSION AND FUTURE WORK

In this paper, we propose an improved Deep Belief Network based scheme for detecting wireless network intrusion. Our proposed scheme employs Conditional Deep Belief Network (CDBN) to efficiently learn the temporal behavior features between the experimental data. We adopt a window-based under-sampling algorithm "SamSelect" to balance the numbers of the normal samples and that of the attack samples in the AWID training dataset. We use Stacked Contractive Auto-encoder (SCAE) algorithm to eliminate the redundancy of experimental data. In the simulations, we illustrate our work by four cases, and the first two cases show that SCAE is feasible to reduce the dimensionality with the average reconstruction error 0.058. The detection accuracy increases as the number of the hidden layers of CDBN increases, and the highest detection accuracy is 0.974 when the number of hidden layers is 6. In the third case, we carefully investigate the impact of time observation window size of CDBN on the detection performance, it shows that as this size goes up the detection performance get better, and when the observation window size is 4, the detection accuracy is the highest with the values for Normal sample, Flooding attack, False attack and Injection attack are 0.989, 0.808, 0.727, 0.991. In the last case, we validate the robustness of our proposed mechanism by changing the environment noise, and the detection accuracy is higher than 75% with different noise intensity. The experimental results show that our detection method can achieve a better detection performance compared with other deep learning and shallow learning methods. These experiments show that the proposed mechanism can be performed in a fast way with the average detection time 1.14 ms and CDBN can be effectively combined with "SamSelect" and SCAE. In the future, we will extend our work by studying an efficient detection method in big data environment and how to apply the proposed mechanism in detecting wider range of cyber-security attacks is worth investigating.

REFERENCES

[1] S. M. Kasongo and Y. Sun, "A deep learning method with wrapper based feature extraction for wireless intrusion detection system," *Comput. Secur.*, vol. 92, May 2020, Art. no. 101752.

[2] S. Agrawal and J. Agrawal, "Survey on anomaly detection using data mining techniques," *Procedia Comput. Sci.*, vol. 60, pp. 708–713, Jan. 2015.

[3] H. H. Pajouh, G. Dastghaibyfard, and S. Hashemi, "Two-tier network anomaly detection model: A machine learning approach," *J. Intell. Inf. Syst.*, vol. 48, no. 1, pp. 61–74, Feb. 2017.

[4] A. R. Syarif and W. Gata, "Intrusion detection system using hybrid binary PSO and K-nearest neighborhood algorithm," in *Proc. ICTS*, Surabaya, Indonesia, 2017, pp. 181–186.

[5] Z. Fan, P. Kulkarni, S. Gormus, C. Efthymiou, G. Kalogridis, M. Sooriyabandara, Z. Zhu, S. Lambotharan, and W. H. Chin, "Smart grid communications: Overview of research challenges, solutions, and standardization activities," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 1, pp. 21–38, 1st Quart., 2013.

[6] T. Ma, F. Wang, J. Cheng, Y. Yu, and X. Chen, "A hybrid spectral clustering and deep neural network ensemble algorithm for intrusion detection in sensor networks," *Sensors*, vol. 16, no. 10, p. 1701, Oct. 2016.

[7] D. Barbara, J. Couto, S. Jadodia, and N. Wu, "ADAM: A test bed for exploring the use of data mining in intrusion detection," *ACM SIGMOD Rec.*, vol. 30, no. 4, pp. 15–24, Dec. 2001.

[8] K. Nalavade and B. B. Meshram, "Evaluation of K-means clustering for effective intrusion detection and prevention in massive network traffic data," *Int. J. Comput. Appl.*, vol. 96, no. 7, pp. 9–14, Jun. 2014.

[9] S. Park, S. Seo, and J. Kim, "Network intrusion detection using stacked denoising autoencoder," *Adv. Sci. Lett.*, vol. 23, no. 10, pp. 9907–9911, Oct. 2017.

[10] H. Saxena and V. Richariya, "Intrusion detection in KDD99 dataset using SVM-PSO and feature reduction with information gain," *Int. J. Comput. Appl.*, vol. 98, no. 6, pp. 25–29, Jul. 2014.

[11] K. Peng, V. C. M. Leung, and Q. Huang, "Clustering approach based on mini batch Kmeans for intrusion detection system over big data," *IEEE Access*, vol. 6, pp. 11897–11906, Feb. 2018.

[12] A. Al-Abassi, H. Karimipour, A. Dehghantanha, and R. M. Parizi, "An ensemble deep learning-based cyber-attack detection in industrial control system," *IEEE Access*, vol. 8, pp. 83965–83973, May 2020.

[13] F. A. Khan, A. Gumaei, A. Derhab, and A. Hussain, "A novel two-stage deep learning model for efficient network intrusion detection," *IEEE Access*, vol. 7, pp. 30373–30385, Feb. 2019.

[14] A. Kim, M. Park, and D. H. Lee, "AI-IDS: Application of deep learning to real-time Web intrusion detection," *IEEE Access*, vol. 8, pp. 70245–70261, Apr. 2020.

[15] M. Alqahtani, A. Gumaei, H. Mathkour, and M. M. B. Ismail, "A genetic-based extreme gradient boosting model for detecting intrusions in wireless sensor networks," *Sensors*, vol. 19, no. 20, p. 4383, Oct. 2019.

[16] M. M. Hassan, A. Gumaei, A. Alsanad, M. Alrubaian, and G. Fortino, "A hybrid deep learning model for efficient intrusion detection in big data environment," *Inf. Sci.*, vol. 513, pp. 386–396, Mar. 2020.

[17] C. Hongsong, M. Caixia, F. Zhongchuan, and C.-H. Lee, "Novel LDoS attack detection by spark-assisted correlation analysis approach in wireless sensor network," *IET Inf. Secur.*, vol. 14, no. 4, pp. 452–458, Jul. 2020.

[18] G. Caminero, M. Lopez-Martin, and B. Carro, "Adversarial environment reinforcement learning algorithm for intrusion detection," *Comput. Netw.*, vol. 159, pp. 96–109, Aug. 2019.

[19] L. Xie and J. Li, "A novel feature extraction method assembled with PCA and ICA for network intrusion detection," in *Proc. Int. Forum Comput. Sci.-Technol. Appl.*, Chongqing, China, 2009, pp. 31–34.

[20] S. Lee, S. Kim, S. Lee, J. Choi, H. Yoon, D. Lee, and J.-R. Lee, "LARGen: Automatic signature generation for malwares using latent Dirichlet allocation," *IEEE Trans. Dependable Secure Comput.*, vol. 15, no. 5, pp. 771–783, Sep. 2018.

[21] J. Kortelainen, E. Väyrynen, and T. Seppänen, "Isomap approach to EEG-based assessment of neurophysiological changes during anesthesia," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 19, no. 2, pp. 113–120, Apr. 2011.

[22] M. Elhenawy, M. Masoud, S. Glaser, and A. Rakotonirainy, "A new approach to improve the topological stability in non-linear dimensionality reduction," *IEEE Access*, vol. 8, pp. 33898–33908, 2020.

[23] Y.-R. Yeh, S.-Y. Huang, and Y.-J. Lee, "Nonlinear dimension reduction with kernel sliced inverse regression," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 11, pp. 1590–1603, Nov. 2009.

[24] V. Laparra, J. Malo, and G. Camps-Valls, "Dimensionality reduction via regression in hyperspectral imagery," *IEEE J. Sel. Topics Signal Process.*, vol. 9, no. 6, pp. 1026–1036, Sep. 2015.

[25] S. J. Lee, P. D. Yoo, A. T. Asyhari, Y. Jhi, L. Chermak, C. Y. Yeun, and K. Taha, "IMPACT: Impersonation attack detection via edge computing using deep autoencoder and feature abstraction," *IEEE Access*, vol. 8, pp. 65520–65529, Apr. 2020.

[26] S. Barua, M. M. Islam, and K. Murase, "GOS-IL: A generalized over-sampling based online imbalanced learning framework," in *Proc. 22nd Int. Conf. Neural Inf. Process. (ICONIP)*, Istanbul, Turkey, Nov. 2015, pp. 680–687.

[27] X. Xiaolong, C. Wen, and S. Yanfei, "Over-sampling algorithm for imbalanced data classification," *J. Syst. Eng. Electron.*, vol. 30, no. 6, pp. 1182–1191, Dec. 2019.

[28] S. K. Guo, Y. Q. Liu, R. Chen, X. Sun, and X. X. Wang, "Improved SMOTE algorithm to deal with imbalanced activity classes in smart homes," *Neural Process. Lett.*, pp. 1–24, Oct. 2018.

[29] K. Napierala and J. Stefanowski, "Types of minority class examples and their influence on learning classifiers from imbalanced data," *J. Intell. Inf. Syst.*, vol. 46, no. 3, pp. 563–597, Jun. 2016.

[30] G. W. Taylor, G. E. Hinton, and S. T. Roweis, "Modeling human motion using binary latent variables," in *Proc. Adv. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, 2006, pp. 1345–1352.

[31] G. Taylor, "Composable, distributed-state models for high-dimensional time series," Ph.D. dissertation, Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, 2009.
[32] L. Thurner, A. Scheidler, F. Schäfer, J.-H. Menke, J. Dollichon, F. Meier, S. Meinecke, and M. Braun, "Pandapower—An open-source python tool for convenient modeling, analysis, and optimization of electric power systems," *IEEE Trans. Power Syst.*, vol. 33, no. 6, pp. 6510–6521, Nov. 2018.
[33] Q. Yu, D. Wei, and H. Huo, "SamSelect: A sample sequence selection algorithm for quorum planted motif search on large DNA datasets," *BMC Bioinf.*, vol. 19, no. 1, p. 228, Jun. 2018.
[34] Y. Su, J. Li, A. Plaza, A. Marinoni, P. Gamba, and S. Chakravortty, "DAEN: Deep autoencoder networks for hyperspectral unmixing," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 7, pp. 4309–4321, Jul. 2019.
[35] S. Ryu, H. Choi, H. Lee, and H. Kim, "Convolutional autoencoder based feature extraction and clustering for customer load analysis," *IEEE Trans. Power Syst.*, vol. 35, no. 2, pp. 1048–1060, Mar. 2020.
[36] H. A. Malki and A. Moghaddamjoo, "Using the Karhunen-Loe've transformation in the back-propagation training algorithm," *IEEE Trans. Neural Netw.*, vol. 2, no. 1, pp. 162–165, Jan. 1991.
[37] E. Q. Wu, G.-R. Zhou, L.-M. Zhu, C.-F. Wei, H. Ren, and R. S. F. Sheng, "Rotated sphere Haar wavelet and deep contractive auto-encoder network with fuzzy Gaussian SVM for Pilot's pupil center detection," *IEEE Trans. Cybern.*, early access, Jan. 11, 2019, doi: 10.1109/TCYB.2018.2886012.
[38] S. N. Tran and A. S. d'Avila Garcez, "Deep logic networks: Inserting and extracting knowledge from deep belief networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 2, pp. 246–258, Feb. 2018.
[39] S. Rezvy, Y. Luo, M. Petridis, A. Lasebae, and T. Zebin, "An efficient deep learning model for intrusion classification and prediction in 5G and IoT networks," in *Proc. 53rd Annu. Conf. Inf. Sci. Syst. (CISS)*, Baltimore, MD, USA, Mar. 2019, pp. 1–6.
[40] G. N. Karystinos and D. A. Pados, "On overfitting, generalization, and randomly expanded training sets," *IEEE Trans. Neural Netw.*, vol. 11, no. 5, pp. 1050–1057, Sep. 2000.
[41] B. Chen, L. Xing, J. Liang, N. Zheng, and J. C. Príncipe, "Steady-state mean-square error analysis for adaptive filtering under the maximum correntropy criterion," *IEEE Signal Process. Lett.*, vol. 21, no. 7, pp. 880–884, Apr. 2014.
[42] G. Zhang, M. Piccardi, and E. Z. Borzeshi, "Sequential labeling with structural SVM under nondecomposable losses," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 9, pp. 4177–4188, Sep. 2018.
[43] L. Zhao, Z. Wang, X. Wang, and Q. Liu, "Driver drowsiness detection using facial dynamic fusion information and a DBN," *IET Intell. Transp. Syst.*, vol. 12, no. 2, pp. 127–133, Mar. 2018.
[44] Q. Zou, H. Jiang, Q. Dai, Y. Yue, L. Chen, and Q. Wang, "Robust lane detection from continuous driving scenes using deep neural networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 1, pp. 41–54, Jan. 2020.
[45] (Dec. 2016). *Micro Average vs Macro Average Performance in a Multiclass Classification Setting*. [Online]. Available: https://datascience.stackexchange.com/questions/15989/micro-average-vs-macro-average-performance-in-a-multiclass-classification-settin
[46] R. Damasevicius, A. Venckauskas, S. Grigaliunas, J. Toldinas, N. Morkevicius, T. Aleliunas, P. Smuikys, "LITNET-2020: An annotated real-world network flow dataset for network intrusion detection," *Electronics*, vol. 9, no. 5, p. 800, 2020.
[47] X. Sun and W. Xu, "Fast implementation of DeLong's algorithm for comparing the areas under correlated receiver operating characteristic curves," *IEEE Signal Process. Lett.*, vol. 21, no. 11, pp. 1389–1393, Jul. 2014.
[48] A. Ashok, M. Govindarasu, and V. Ajjarapu, "Online detection of stealthy false data injection attacks in power system state estimation," *IEEE Trans. Smart Grid*, vol. 9, no. 3, pp. 1636–1646, Jul. 2016.
[49] B. Han and L. S. Davis, "Density-based multifeature background subtraction with support vector machine," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 5, pp. 1017–1023, May 2012.
[50] K. Kayabol, "Approximate sparse multinomial logistic regression for classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 490–493, Feb. 2020.
[51] J. M. Alonso and A. Bugarín, "ExpliClas: Automatic generation of explanations in natural language for WEKA classifiers," in *Proc. IEEE Int. Conf. Fuzzy Syst. (FUZZ-IEEE)*, New Orleans, LA, USA, Jun. 2019, pp. 1–6.
[52] J. J. Q. Yu, Y. Hou, and V. O. K. Li, "Online false data injection attack detection with wavelet transform and deep neural networks," *IEEE Trans. Ind. Informat.*, vol. 14, no. 7, pp. 3271–3280, Jul. 2018.

LIQUN YANG is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering, Beihang University, Beijing, China. He is also a Visiting Student with the Department of Electrical and Computer Engineering, Mcgill University, Canada. His research interests include industrial control systems security and cyber security.

JIANQIANG LI received the master's degree from the Beijing University of Posts and Telecommunications (BUPT), in 2016. He is currently a Senior Engineer with the National Computer Network Emergency Response Technical Team/Coordination Center of China (known as CNCERT or CNCERT/CC). His main research interests include Internet threat intelligence analysis and industrial Internet security monitoring. He is also an expert in design and development of large-scale distributed systems.

LIANG YIN was born in Zhongwei, Ningxia, China, in 1986. He received the master's degree, in 2018. He is currently a Senior Engineer with the State Grid Corporation of China. His research interest is dispatch automation of power grid.

ZHONGHAO SUN received the Ph.D. degree from Northwestern Polytechnical University, in 2017. He is currently an Engineer with the National Computer Network Emergency Response Technical Team/Coordination Center of China (known as CNCERT or CNCERT/CC). He has authored or coauthored over 20 papers in various journals and international conference. His main research interest is cyber security.

YUFEI ZHAO is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering, Beihang University. His current research interests include privacy-preserving data mining.

ZHOUJUN LI (Member, IEEE) received the Ph.D. degree from the National University of Defense Technology, Hunan, China, in 1999. He is currently a Professor with the School of Computer Science and Engineering, Beihang University, Beijing, China. His main research interests include concurrency theory and process algebra, formal analysis and verification of security protocols, information security, and data mining.