IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 20, 2020, accepted March 5, 2020, date of publication March 16, 2020, date of current version March 30, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2981162

Cyber Intrusion Prevention for Large-Scale Semi-Supervised Deep Learning Based on Local and Non-Local Regularization

GUANGMING XIAN, (Member, IEEE)
School of Software, South China Normal University, Foshan 528225, China
e-mail: xgm20011@sina.com

This work was supported in part by the National Natural Science Foundation of China, a Computing Model Based on Formal Domain Fusion, under Grant 61070015, in part by the South China Normal University, and in part by the South China University of Technology.

ABSTRACT The cyber intrusion prevention model represents a new means of cyber protection with intelligent defense capability. It can not only detect intrusion behavior but also respond to such behavior in a timely manner. This study applies deep learning theory and semi-supervised clustering to cyber intrusion prevention technology. Deep learning based on deep structures represents the current development trend of neural networks. Semi-supervised learning uses a large amount of unlabeled cyber traffic data and a small amount of labeled cyber traffic data to achieve cyber intrusion prevention with a low recognition error rate. Discriminative deep belief network (DDBN)-based cyber defense technology has emerged as a research hotspot in the field of cyber intrusion prevention owing to its low error rate. This paper proposes a cyber intrusion prevention technology using DDBN for large-scale semi-supervised deep learning based on local and non-local regularization to overcome the problem of high classification error rates of the cyber intrusion prevention model. Through comparisons with the cyber intrusion prevention results of the Hopfield, support vector machine (SVM), generative adversarial network (GAN), and deep belief network-random forest (DBN-RFS) classifiers, the proposed DDBN model is shown to have the lowest error rate. Thus, the proposed approach can improve the performance of the cyber intrusion prevention system. The training and testing error rates of the exponent loss function with local and non-local regularization (exponent with LNR) are lower than those of the exponent, square, and hinge loss functions. The experimental results show that the running time decreases as the number of hidden layers increases, especially with 6144 and 4096 hidden layer nodes.

INDEX TERMS Cyber security, discriminative deep belief networks, intrusion prevention, local and non-local regularization, semi-supervised deep learning.

I. INTRODUCTION

Cyber security [1], [2] has become increasingly important in recent years [3]. Owing to the complexity of the current cyber environment, conventional protection technologies [4] cannot meet the requirements of cyber security. Intrusion detection is the core technology of intrusion prevention systems. Such systems combine the advantages of intrusion detection technology and firewall technology. Thus, they can not only detect intrusion but also adopt timely protective measures. In addition, they have an active defense function that effectively improves cyber security [5].

The associate editor coordinating the review of this manuscript and approving it for publication was Shadi Alawneh.

Intrusion prevention systems have been widely used to prevent information from being compromised, and various machine learning methods have been proposed to enhance the performance of such systems [6]. The intrusion prevention model (IPM) provides cyber security by implementing intelligent detection [7] and active response. It consists of several modules. Intrusion prevention systems provide cyber security through the following process [8]–[10]. They check the external data packets; normal data can enter the internal cyber space after checking, while suitable measures are adopted against abnormal data [11], [12]. The intrusion prevention model [13] is shown in Fig.1.

Machine learning technology [14]–[17] has been successfully applied to various fields [18]–[20]. As the detection

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

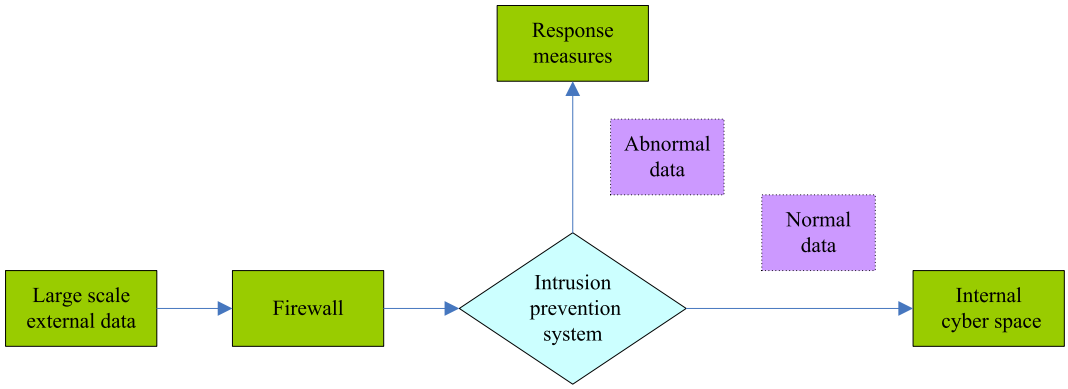

FIGURE 1. Cyber intrusion prevention model.

algorithm has a direct impact on the defense performance of the entire defense system, intelligent detection based on machine learning has emerged as a research hotspot in the field of intrusion detection. It is difficult for supervised-clustering-based intrusion detection algorithms used in traditional machine learning to detect unknown attacks. Further, it is difficult to predict the intrusion methods in actual cyber situations. Although methods based on unsupervised learning can detect unknown attacks, it is difficult for them to classify the attacks properly. In other words, such methods provide only general protection; they do not offer significant performance advantages in terms of cyber security. As methods based on supervised and unsupervised learning cannot achieve ideal classification results, methods based on semi-supervised learning have been considered. Numerous performance evaluations and comparative experiments have shown that such methods can extract more discriminant features and offer significant advantages in cyber intrusion prevention tasks. By combining the characteristics of supervised and unsupervised learning, semi-supervised learning offers the advantages of both approaches. Deep learning [21], [22] is a more accurate approach that can be combined with semi-supervised learning to fully exploit [23], [24] its own advantages as well as those of semi-supervised learning [18]. Many practical applications involve abundant unlabeled data and sparse labeled data. Semi-supervised learning relies on a large amount of unlabeled data and a small amount of labeled data for training; thus, it can overcome the shortage of labeled samples. In recent years, semi-supervised feature extraction has attracted increasing research attention, and many semi-supervised learning methods have been proposed [25]. Semi-supervised [25]–[27] feature selection methods [28]–[30] use the label information of labeled data from cyber traffic and data distribution or local structures of both labeled and unlabeled data from cyber traffic to evaluate feature relevance [25]–[31].

In this study, semi-supervised learning [32] is applied to cyber intrusion prevention, and a cyber intrusion prevention algorithm based on deep learning and semi-supervised clustering is proposed. The proposed algorithm can efficiently train the algorithm parameters using a certain amount of labeled data. In particular, it can improve the classification error rate using a small amount of labeled data.

The discriminative deep belief network (DDBN) is an effective semi-supervised approach for cyber intrusion prevention based on deep belief networks (DBNs) [33], [34]. It has the following three characteristics. First, DDBN uses a new deep architecture to integrate the abstraction capabilities of DBN and the discriminative capabilities of the loss function. The deep architecture is constructed layer by layer using greedy unsupervised methods, and the parameter space is further optimized using the gradient descent supervision method. Second, for unsupervised learning, DDBN inherits the advantages of DBN [35], [36], which preserves the information effectively from the high-dimensional feature space to the low-dimensional embedding. Thus, DDBN can use large-scale unlabeled data to improve the generalization ability of the system. Third, for supervised learning, through a well-designed objective function [37], the backpropagation strategy directly optimizes the classification results for the training dataset by refining the parameter space [38], [39]. Thus, DDBN can use a small amount of unlabeled data to achieve good classification results for cyber intrusion prevention. Specifically, when the labeled data are not sufficient, DDBN can improve the learning ability using a large amount of unlabeled data. The deep architecture is efficient in representing most common functions, and it can effectively solve difficult learning problems.

DDBN simulates the thinking process of the human brain, extracts abstract cyber traffic features step by step, and uses the abstract features for classification. The key aspect of the deep learning algorithm is multi-level learning. Through multi-level feature extraction, the intrinsic relationship between the cyber traffic data can be determined. DDBN can prevent intrusion by discovering hidden attacks and improving the classification error rate.

The successful application of the deep learning [40] model to various fields has led to the rapid development of deep neural networks based on regularization, i.e., regularization items are added to the objective function (loss function) of

the final deep learning optimization model, which makes the final trained features more beneficial for the subsequent application. However, with regard to classification tasks in cyber protection, current regularization methods are faced with major challenge. For example, deep learning algorithms based on local regularization, such as EmbedDBN, have improved feature discrimination; however, the non-local properties between samples, which also play an active role in improving feature discrimination, have been ignored. In view of the above-mentioned problems, this study designs a novel regularization method for classification tasks in cyber protection by integrating the local and non-local constraints of labeled and unlabeled samples, and extracting abstract features that can effectively preserve the separability of classes in the original sample space by using such information. By integrating local and non-local topological regularization terms of labeled and unlabeled samples, the proposed discriminant regularization terms can extract features that are more suitable for classification.

Local and non-local topological structures based on manifold learning are used to regularize the deep learning model. This model can obtain the semantic and geometric structure of the data, which is beneficial for the final classification of cyber traffic data. The discriminative ability of the model mainly depends on the design of the local and non-local weighting matrices between the data samples. As the design of the weight matrix is directly related to the local and non-local topological information of the encoded data space, it is also related to the validity and discrimination of the final feature [32].

Semi-supervised learning [41]–[43] is a natural choice in the case of insufficient labeled data. At the same time, a deep model is a natural choice for difficult AI tasks. Thus, many current applications can benefit from a combination of semi-supervised learning and a deep model. Therefore, deep models based on semi-supervised learning are attracting increasing attention. Some deep learning algorithms use semi-supervised regularization to extract more expressive and discriminant feature representations from the original data space to obtain superior application results. In this paper, we mainly focus on the application of deep learning based on semi-supervised regularization to cyber intrusion protection. In particular, we introduce the concept of ''non-locality''. For classification tasks, the introduction of non-locality is necessary because it involves discrimination between different manifold structures (here, we quantify non-locality by the distance between different manifolds). We propose local and non-local regularization for a semi-supervised deep learning algorithm that considers both the locality and the non-locality of samples in the process of deep learning model training and maps the topological structure between samples to the final feature space to achieve better classification performance. More specifically, we construct local and non-local constraints for each sample to form a topological structure constraint matrix between samples as a regularization term.

Although deep learning is widely used in the fields of pattern recognition, speech recognition, and natural language processing, it has few promising applications in the field of cyber intrusion prevention. The application of DDBN to cyber intrusion protection is the novelty of this study [44]–[46].

Currently, many researches have applied semi-supervised algorithms for training deep models, and the results show that a combined method is more effective. Ito *et al.* [47] proposed a semi-supervised learning framework to train a DNN. Compared with existing DNN-based methods, they obtained better and more stable results. Tang *et al.* [48] proposed a semi-supervised algorithm based on a CNN. Extensive evaluations showed that their algorithm outperforms many state-of-the-art algorithms in completing tasks. Chen *et al.* [49] revealed that deep learning-embedded semi-supervised learning outperforms the deep learning-embedded semi-supervised learning when labeled data are limited. However, these studies [47]–[49] did not consider the local and non-local topological information of unlabeled samples, which is more intuitive and authoritative for classification. This paper proposes a cyber intrusion prevention technology using DDBN for large-scale semi-supervised deep learning based on local and non-local regularization. The proposed cyber intrusion prevention technology has a lower error rate and wider applicability than other cyber intrusion prevention methods.

The main contributions of this paper are as follows:

(1) A semi-supervised discriminant regularization method for cyber intrusion protection is proposed to train a deep neural network, i.e., some topological regularization items are added to the objective function (loss function) of the final optimization of the deep model. This method integrates local and non-local constraints in labeled and unlabeled samples, and extracts abstract features that can effectively preserve the separability of categories in the original sample space. For labeled samples, we use class labels to define local and non-local information and then obtain topological regularization terms by minimizing the intra-class compactness (locality) and maximizing the inter-class separability (non-locality). For unlabeled samples, we use the average distance between one sample and the other samples as a threshold to determine its neighbor and non-neighbor samples; then, the topological regularization term maximizes the non-local divergence and minimizes the local divergence simultaneously. By integrating local and non-local topological regularization terms of labeled and unlabeled samples, our discriminant regularization terms can extract those features that are more suitable for classification in cyber intrusion protection.

(2) Semi-supervised learning combines the characteristics and advantages of unsupervised learning and supervised learning. Deep learning has a more accurate detection effect, and a combination of semi-supervised learning and deep learning can fully exploit the advantages of semi-supervised learning and semi-supervised learning. In this study, deep learning theory and semi-supervised classification

are applied to cyber intrusion prevention technology, which effectively increases the detection rate, reduces the false alarm rate, and improves the performance of cyber intrusion prevention systems.

In summary, this paper proposes a semi-supervised deep learning solution based on local and non-local regularization and a new regularized DDBN deep learning method. The proposed method is validated on two standard datasets (KDD Cup99 and NSL-KDD). Through performance evaluation and comparative experiments, we demonstrated that the features extracted by our method are more discriminative and have significant advantages for cyber intrusion protection.

II. DDBN BASED ON SEMI-SUPERVISED LEARNING

Considering the classical learning method, a new semi-supervised learning method based on DDBN is proposed. First, the semi-supervised learning problem that needs to be solved using DDBN is introduced. Next, the structure of DDBN is described. Then, the supervised and unsupervised learning methods of DDBN are discussed. Finally, the algorithm flow of DDBN is presented.

A. SEMI-SUPERVISED LEARNING PROBLEM REPRESENTATION

Let X denote a sample dataset, which can be expressed as

$$X = [x^1, x^2, \cdots, x^{L+U}] = \begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_1^{L+U} \\ x_1^2 & x_2^2 & \cdots & x_2^{L+U} \\ \vdots & \vdots & \ddots & \vdots \\ x_D^1 & x_D^2 & \cdots & x_D^{L+U} \end{bmatrix} \tag{1}$$

where L denotes the number of labeled data, U denotes the number of unlabeled data, and D denotes the number of features of each data X. Each column of X represents a data object. A data object that has all the features can be considered as a vector in space R^D, where the first j coordinates correspond to the *jth* feature.

Further, let Y denote the tag dataset corresponding to L, which can be expressed as

$$Y = [y^1, y^2, \cdots, y^L] = \begin{bmatrix} y_1^1 & y_1^2 & \cdots & y_1^L \\ y_2^1 & y_2^2 & \cdots & y_2^L \\ \vdots & \vdots & \ddots & \vdots \\ y_C^1 & y_C^2 & \cdots & y_C^L \end{bmatrix} \tag{2}$$

where C denotes the number of categories in the dataset. Each column of Y is a vector in space R^C, where the first j coordinates correspond to the *jth* category.

$$y_j^i = \begin{cases} 1 & if\ x^i \in jth\ \text{category} \\ -1 & if\ x^i \notin jth\ \text{category} \end{cases} \tag{3}$$

In this study, the deep framework uses L labeled data and U unlabeled data to train the mapping function $X \rightarrow Y$. After training, when the new data X is input, the deep schema can use the mapping function to determine the corresponding labeled X of Y.

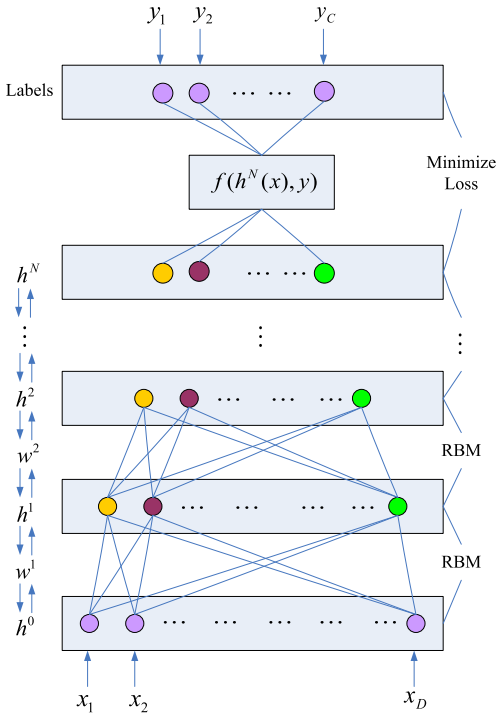

FIGURE 2. Structure of DDBN.

B. FRAMEWORK OF DDBN

The structure of DDBN is shown in Fig. 2. DDBN consists of hidden layers from h^1 to h^N, one input layer x, and one labeled layer y. The hidden layers h^i are constructed using both labeled and unlabeled data. The function $f(h^N(x), y)$ is learned using the labeled data.

DDBN is a fully connected and directional multilayer neural network that consists of an input layer h^0, N hidden layers $h^1, h^2, \cdots h^N$, and a label layer at the top. The input layer h^0 has D units, equal to the number of characteristics of the data x. The label layer has C units, equal to the number of categories of the labeled data y. Further, $W = \{w^1, w^2, \cdots w^{N+1}\}$ is the parameter needs to learned in the deep framework. The number of hidden layers and number of nodes in each hidden layer need to be set empirically. The problem of finding the mapping function $X \rightarrow Y$ can be converted into the problem of finding the deep framework space W.

The DDBN training process is divided into two steps:

(1) DDBN uses (restricted boltzmann machine) RBM as its basic module, and it uses greedy unsupervised methods to build the deep architecture layer by layer.

(2) DDBN uses the gradient descent method to train the deep structure on the basis of the exponent loss function. The space W is further optimized by L labeled data.

C. UNSUPERVISED LEARNING METHOD OF DDBN

The deep architecture of DDBN is built as layers of RBM. RBM is a two-layer recurrent neural network [50]. Random binary inputs are connected to the random binary outputs via symmetric weights.

In the deep architecture of DDBN, the energy state is defined as

$$E(h^{k-1}, h^k; \theta) = -\sum_{s=1}^{D_{k-1}}\sum_{t=1}^{D_k} w_{st}^k h_s^{k-1} h_t^k - \sum_{s=1}^{D_{k-1}} b_s h_s^{k-1} - \sum_{t=1}^{D_k} c_t h_t \quad (4)$$

where $\theta(w, b, c)$ denotes the model parameter; w_{st}^k is the symmetric connection parameter between unit s in hidden layer h^{k-1} and unit t in hidden layer h^k, $k = 1, 2, \cdots, N-1$; b_s is the sth bias in the hidden layer h^{k-1}; c_t is the tth bias in the hidden layer h^k; and D_k is the number of nodes in layer k.

The probability of h^{k-1} can be expressed as

$$P(h^{k-1}; \theta) = \frac{1}{Z(\theta)} \sum_{h^k} \exp(-E(h^{k-1}, h^k; \theta)) \quad (5)$$

$$Z(\theta) = \sum_{h^{k-1}} \sum_{h^k} \exp(-E(h^{k-1}, h^k; \theta)) \quad (6)$$

where $Z(\theta)$ denotes the normalization constant.

The conditional probabilities for h^k and h^{k-1} are expressed as

$$p(h^k \Big| h^{k-1}) = \prod_t p(h^k \Big| h^{k-1}) \quad (7)$$

$$p(h^{k-1} \Big| h^k) = \prod_s p(h_s^{k-1} \Big| h^k) \quad (8)$$

The probability that the tth unit equals 1 is the logical function containing h^{k-1} and w_{st}^k:

$$p(h_t^k = 1 \Big| h^{k-1}) = sigm(c_t + \sum_s w_{st}^k h_s^{k-1}) \quad (9)$$

The probability that the sth unit equals 1 is the logical function containing h^k and w_{st}^k:

$$p(h_s^k = 1 \Big| h^k) = sigm(c_s + \sum_t w_{st}^k h_s^k) \quad (10)$$

where the logical function can be expressed as

$$sigm(\eta) = 1/(1 + \exp(-\eta)) \quad (11)$$

The logarithm of the probability of the hidden layer is derived from the model parameter w^k using the contrastive divergence (CD) method:

$$\frac{\partial \log p(h_s^{k-1})}{\partial w_{st}^k} = \left\langle h_s^{k-1} h_t^k \right\rangle_{P_0} - \left\langle h_s^{k-1} h_t^k \right\rangle_{P_M} \quad (12)$$

where $\langle \cdot \rangle_{P_0}$ represents the expectation of data distribution and $\langle \cdot \rangle_{P_M}$ represents the data distribution obtained by performing Gibbs sampling M times after the data input.

The parameter w^k can be adjusted by the following equation:

$$w_{st}^k = \vartheta w_{st}^k + \eta \frac{\partial \log p(h^{k-1})}{\partial w_{st}^k} \quad (13)$$

where ϑ is the momentum and η is the learning rate.

The above-mentioned procedure is discussed with regard to sample data x. In the DDBN system, the deep structure is constructed by training all the labeled and unlabeled data input into h^0 one by one.

The input data build the network from one layer to the next. At each level, the parameter space w^k is constructed using data calculated by layer $k-1$.

After obtaining the parameter w^k using the above-mentioned calculation method, the hidden layer can be calculated using the following formula after data x is input into h^0:

$$h_t^k(x) = sigm\left(c_t^k + \sum_{i=1}^{D_{k-1}} w_{st}^k h_x^{k-1}(x)\right) \quad t = 1, 2, \cdots, D_k; \quad k = 1, 2, \cdots, N-1 \quad (14)$$

As with the classical backpropagation method, the parameter space w^k is initialized by a random number ccording to the standard normal distribution:

$$h_t^N(x) = c_t^N + \sum_{i=1}^{D_{N-1}} w_{st}^N h_s^{N-1}(x) \quad t = 1, 2, \cdots, D_N \quad (15)$$

D. SUPERVISED LEARNING METHOD OF DDBN

After greedy unsupervised training, $h^N(x)$ is the abstract representation of x. Here, L labeled data are used to optimize the parameter space W so that the deep architecture has better discriminative capability. This task can be transformed into an optimization problem:

$$\arg\min_w f(h^N(X), Y) \quad (16)$$

where

$$f(h^N(X), Y) = \sum_{i=1}^{L}\sum_{j=1}^{C} T(h^N(x_j^i) y_j^i) \quad (17)$$

In the above-mentioned equation, T indicates the loss function. The key problem is how to define a proper loss function to improve the discrimination ability of the classifier.

The following mean square loss function [51] is a natural choice because it has been widely used in backpropagation methods:

$$T_{square}(r) = (r-1)^2 \quad (18)$$

where $r = h^N(x_j^i) y_j^i$.

The hinge loss function [52] applied to SVM is another possible choice:

$$T_{hinge}(r) = \max(1 - r, 0) \quad (19)$$

DDBN uses the exponent loss function [51], which has been applied to the boosting algorithm. It performs well in an actual application dataset.

$$T_{\exp onent}(r) = \exp(-r) \quad (20)$$

After the loss function is determined, the gradient descent method is used to optimize the parameter space of the entire

deep layer architecture. The random activation mechanism in the unsupervised learning phase is replaced by the determined real probability in the supervised learning phase.

E. ALGORITHM FLOW OF DDBN

The algorithm flow of DDBN is as follows.

Input: Dataset X, labeled dataset Y

Hidden layer h, Layer number N, number of units per layer $D_1, D_1, \ldots, D_N$, iteration number Q

Parameter space $W = \{w^1, w^2, \cdots, w^N\}$, bias b, c, momentum ϑ, learning rate η

Number of labeled data L, number of unlabeled data U

Output: a deep framework containing the trained parameter space W

(1) Greedy unsupervised method constructs the network layer by layer

For $k = 1; k \leq N - 1$ do

For $q = 1; q \leq Q$ do

For $u = 1; u \leq L + U$ do

Calculate the nonlinear forward and reverse states:

$$p(h_{t,u}^k = 1 \Big| h^{k-1}) = sigm\left(c_t + \sum_s w_{st}^k h_{s,u}^{k-1}\right) \tag{21}$$

$$p(h_{t,u}^{k-1} = 1 \Big| h^k) = sigm\left(b_s + \sum_t w_{st}^k h_{s,u}^k\right) \tag{22}$$

Update the parameters and offsets:

$$w_{st}^k = \vartheta w_{st}^k + \eta\left(\left\langle h_{s,u}^{k-1} h_{tu}^k \right\rangle_{P_0} - \left\langle h_{s,u}^{k-1} h_{s,u}^k \right\rangle_{P_1}\right) \tag{23}$$

End

End

End

(2) Supervised learning based on gradient descent method

$$\arg_w \min \sum_{i=1}^{L} \sum_{j=1}^{C} \exp(-h^N(x_j^i) y_j^i) \tag{24}$$

First, the greedy unsupervised method builds the network layer by layer. RBM is used to build the deep architecture layer by layer. $L + U$ training data are iterated Q times to initialize the parameter space $w^1, w^2, \cdots, w^{N-1}$. The output layer W^N is initialized using random numbers subject to a normal distribution. Second, supervised learning based on the gradient descent method is adopted. In this global optimization stage, the stochastic activation mechanism is replaced by the real determined probability value. The conjugate gradient algorithm is used to globally optimize the entire network. Once the training is complete, when new data x in input, the category of X can be determined according to the value of the output $h^N(x)$ in the deep layer schema [53]–[55].

III. SEMI-SUPERVISED DDBN BASED ON LOCAL AND NON-LOCAL REGULARIZATION

A suitable feature representation can reveal the implicit structures in the data. The intrinsic geometric structure of the data is an ideal implicit structure in feature learning. In other words, samples with the local neighborhood of the original data can still be close to each other in the feature space. Samples in a non-local relationship in the original data should be kept as far as possible in the feature space. The features thus obtained can play a positive role in further application processing. DBN as a powerful learning tool can extract the characteristics of the original data. However, neither standard DBN nor regular DBN focuses on the intrinsic geometric structure of the data or the discriminant structure in the data space, which are useful for semi-supervised learning tasks. This paper presents a semi-supervised deep learning solution based on local and non-local regularization.

The discriminant power of the features obtained by the model mainly depends on the design of the local and non-local weight matrices between the data samples. As the design of the weight matrix is directly related to the local and non-local topological structure information of the encoding data space, it is also related to the validity and discrimination of the final feature.

For the ultimate classification purpose in cyber intrusion prevention, the objective is to find a series of discriminant feature representations that preserve the local and non-local topological structure of the original data.

The objective function of the semi-supervised regularization term proposed in this paper can be integrated into the equation obtained by minimizing the following equation:

$$\begin{aligned} J(\theta) &= \alpha S(\theta) + \beta U(\theta) \\ &= \alpha(S^w - S^b) + \beta(S^L - S^N) \\ &= \alpha F(FL^w F^T - FL^b F^T) + \beta(FL^L F^T - FL^N F^T) \\ &= \alpha F(L^w - L^b)F^T + \beta F(L^L - L^N)F^T \end{aligned} \tag{25}$$

where α and β are used to balance the scaling parameters of the corresponding regularization contributions. The first and second terms in the above-mentioned formula represent the constraints of labeled and unlabeled samples, respectively, in the feature space.

Combined with the original objective function of the deep model, the objective function of the semi-supervised DDBN based on local and non-local regularization is defined as

$$\theta^* = \arg\min_{\theta} \sum_{i=1}^{l} l(f(x_i), y_i) + J(\theta) \tag{26}$$

where $f((x_i), y_i)$ is the loss function of DDBN, x_i is the input vector of the deep model, y_i is the label corresponding to sample x_i, $f(x_i)$ is the output of sample x_i in the model (the softmax method is used here), and θ is a parameter of the deep learning model. Note that the semi-supervised regularization item is embedded in the supervisory loss function of the last hidden layer of the DDBN model. Then, the follow-up backward transfer supervisory training is performed.

The detailed learning process of the proposed method, as shown in the following algorithm, is roughly the same as the training process of the traditional DBN, which is divided into two steps.

(1) Unsupervised pre-training: The CD method is used to greedily pre-train the DDBN layer by layer.
(2) Algorithm: Semi-supervised DDBN based on local and non-local regularization

Input: Training dataset X, including labeled data X^l and unlabeled data X^u

Labeled set Y corresponding to labeled data

Layer number N and epoch number E of DDBN

Labeled data number l

Interclass divergence matrix S^b with labeled data and intraclass divergence matrix S^w

Local divergence matrix S^L and non-local divergence matrix S^N for unlabeled data

Initialized weighted connection parameters w, deviation parameters a and b

Momentum ϑ and learning rates ζ_w, ζ_a, and ζ_b

Scaling parameters α and β corresponding to the regularization items

Output: Optimal parameters $\theta^* = [w^*, a^*, b^*]$

Training process:

Greedy unsupervised layer-by-layer pre-training:

For $n = 1, \ldots, N$, do

For $e = 1, \ldots, E$, do

Complete an alternate Gibbs sampling to calculate the state of the next layer:

$$p(h_j^{n+1} = 1 \mid h^n) = \sigma\left(b_j^n + \sum_i h_i^n w_{ij}^n\right) \tag{27}$$

$$p(h_i^n = 1 \mid h^{n+1}) = \sigma\left(a_i^n + \sum_j h_j^n w_{ij}^n\right) \tag{28}$$

Update the connection weights and unit deviations

$$w_{ij}^n = \vartheta w_{ij}^n + \zeta_w\left(\left\langle h_i^n h_j^{n+1}\right\rangle_{data} - \left\langle h_i^n h_j^{n+1}\right\rangle_{recon}\right) \tag{29}$$

$$a_i^n = \vartheta a_i^n + \zeta_a\left(\left\langle h_i^n\right\rangle_{data} - \left\langle h_i^n\right\rangle_{recon}\right) \tag{30}$$

$$a_j^n = \vartheta b_j^n + \zeta_b\left(\left\langle h_j^{n+1}\right\rangle_{data} - \left\langle h_j^{n+1}\right\rangle_{recon}\right) \tag{31}$$

End for

End for

Supervisory fine-tuning of a neural networkčž

The final objective function is optimized by gradient descent through backward transfer algorithm, and the optimal parameters are calculated.

$$\theta^* = \arg\min_{\theta} \sum_{i=1}^{l} l(f(x_i), y_i) + J(\theta) \tag{32}$$

(3) Supervised fine-tuning of DDBN

The final objective function is optimized by gradient descent using the backward transfer algorithm, and the optimal parameters are calculated. In this step, the DDBN is fine-tuned by optimizing the objective function (21) to extract the effective features that are more beneficial for classification. The conjugate gradient method is used to optimize the objective, where the gradient solution is computed by the backward transfer method [32].

TABLE 1. Experimental KDD Cup99 dataset for known attacks.

Type	Class	10percent dataset	corrected dataset
DOS	1	223749	123628
R2L	3	884	13957
U2R	5	50	192
PROBE	4	1862	1938
NORMAL	2	64372	46836
Total	15	290917	186551

IV. EXPERIMENTAL RESULTS AND SIMULATION ANALYSIS

A. SOFTWARE AND HARDWARE ENVIRONMENT OF THE EXPERIMENT

The experiments were performed in the following hardware and software environments. The processor used was the Intel i5 8500 with 8.00 GB memory and a 1000 GB hard disk drive. Microsoft Windows 10 was used as the operating system. The effectiveness of the DDBN algorithm was verified via simulation in MATLAB.

B. EXPERIMENTS ON LARGE-SCALE NETWORK-CONNECTED INFORMATION SETS

The experimental data (see Table 1) used in this study were the KDD Cup99 10% dataset and the KDD Cup99 corrected dataset for known attacks, which include large amounts of normal cyber traffic and various attacks. The 10% dataset was used as the training set and the corrected dataset was used as the testing set. Abnormal data can be categorized into four types: denial of service (DOS) attacks, unauthorized access from a remote machine (R2L), unauthorized access to local superuser privileges (U2R), and surveillance and probing (PROBE) attacks. In the experiment, the DDBN architecture was 41-300-300-300-1800-5, i.e., the number of nodes in the input layer was 41, the number of nodes in the output layer was 5, and the numbers of nodes in the four hidden layers were 300, 300, 300, 1800.

As shown in Fig. 3, on the training and test datasets, DDBN achieved good results with 50 iterations. When the number of DDBN iterations exceeded 100, as the number of iterations increased, the cyber intrusion prevention classification error rate tended to be stable. Thus, the weight matrix of DDBN was optimized. At this time, if the number of iterations of DDBN were increased, the classification error rate would not be improved significantly, whereas the running time of the program would increase.

The number of training labeled data of cyber traffic was set as 50, 100, 200, 400, 800, and 1600. Each class had at least one labeled data object while other large-scale samples were used as unlabeled data. The Hopfield, SVM, GAN, and DBN-RFS classifiers were used in the experiment for comparison with DDBN. The experimental results of each

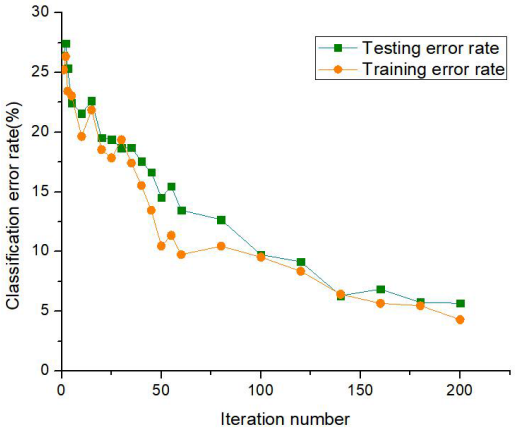

FIGURE 3. DDBN iteration number and classification error rate (%).

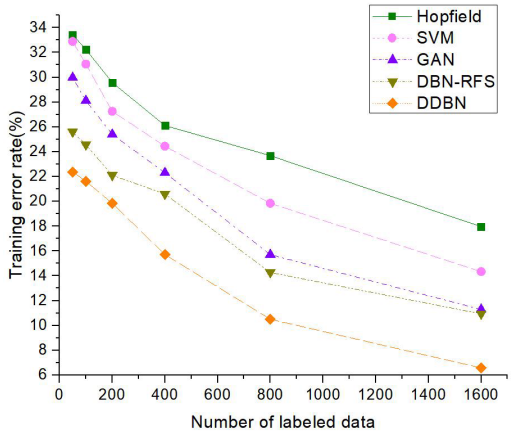

FIGURE 4. Training error rates on large-scale cyber connection information sets using different amounts of labeled data.

classifier with different amounts of training labeled data are shown in Fig. 4. The results indicate that the training performance of DDBN is better than that of the Hopfield, SVM, GAN, and DBN-RFS classifiers in terms of network intrusion prevention. When the number of labeled data was 1600, the training error rate of DDBN was 6.58%.

The number of testing labeled data of cyber traffic was set as 40, 80, 150, 300, 600, and 1200. The experimental results of each classifier with different numbers of testing labeled data are shown in Fig. 5. The results indicate that the testing performance of DDBN is better than that of the other classifiers in terms of cyber intrusion prevention. When the number of labeled data was 1200, the testing error rate of DDBN was 7.82% and that of DBN-RFS was 11.68%. Thus, the testing error rate of DDBN was 3.86% lower than that of DBN-RFS.

Table 2 shows the time performance comparison of different classifiers on the KDD cup99 dataset in terms of both the training error rate and the testing error rate. It can been seen that the training error rate and testing error rate decrease in the order of Hopfield, SVM, GAN, DBN-RFS, and DDBN.

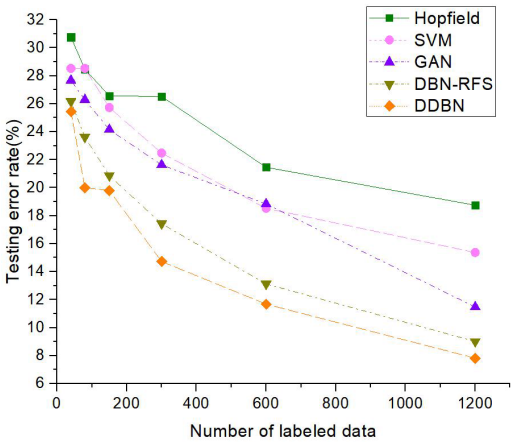

FIGURE 5. Testing error rates on large-scale cyber connection information sets using different amounts of labeled data (KDD Cup99 dataset).

TABLE 2. Time performance comparison of different classifiers on KDD Cup99 dataset in terms of both training error rate and testing error rate.

Testing data	Detection rate(%)	False alarm rate (%)
Dataset 1	86.43	1.28
Dataset 2	87.92	1.24
Dataset 3	93.47	0.72
Dataset 4	95.68	0.63
Dataset 5	97.53	0.57
Average result	92.21	0.89

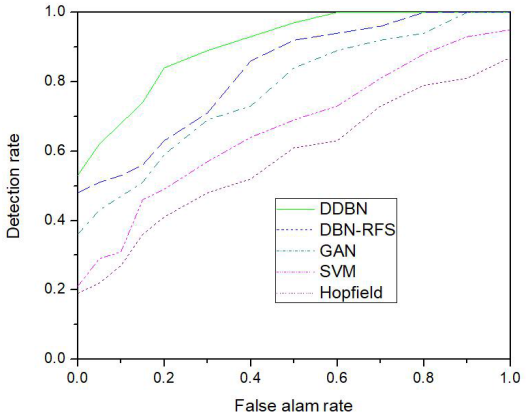

FIGURE 6. ROC curves of different methods.

The training time and testing time decrease in the order of GAN, DBN-RFS, DDBN, Hopfield, and SVM. This method is shown to achieve high recognition accuracy with relatively low time consumption.

After many experiments, the ROC curves (Fig. 6) based on the detection rate and false alarm rate were plotted to reflect the dynamic effects. The effect of DDBN was found to be obviously stronger than that of the DBN-RFS, GAN, SVM, and Hopfield classifiers in terms of cyber intrusion prevention.

As shown in Fig. 7, the areas under the curves (AUCs) of the Hopfield, SVM, GAN, DBN-RFS, and DDBN classifiers

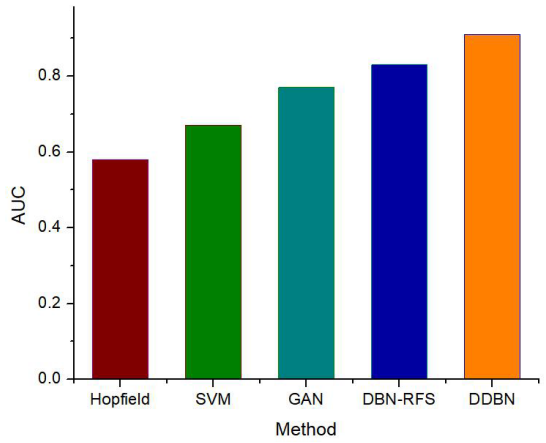

FIGURE 7. AUC of different methods.

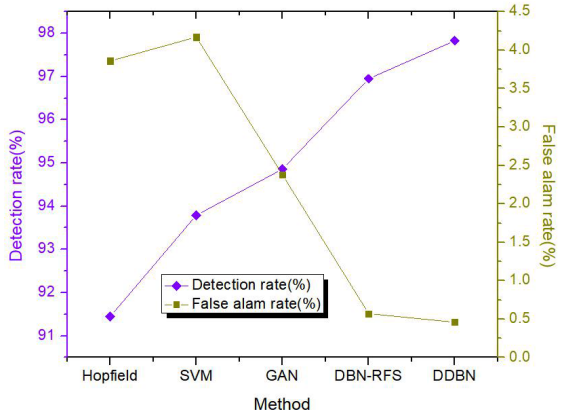

FIGURE 8. Comparison of detection results of different methods.

were 0.58, 0.67, 0.77, 0.83, and 0.91, respectively. It can be seen that the performance of DDBN in terms of cyber intrusion prevention was better than that of the Hopfield, SVM, GAN, and DBN-RFS classifiers.

As the labeled data increased, the detection rate and false alarm rate tended to be stable, which indicates that the system performance was reliable. In the case of complete marking, the detection rate and false alarm rate are shown in Fig. 8. By comparison, the application of the proposed DDBN algorithm to the intrusion prevention system improved the performance of the system to a certain extent, effectively improved the detection rate, and reduced the false alarm rate.

Semi-supervised learning is a method of learning that relies on a small amount of labeled data and a large number of unlabeled data. The labeled datasets used in this paper account for 1/5, 1/4, 1/3, 1/2, and 3/4 of the total datasets. In the case of dataset 1(1/5 of the total dataset), dataset 2 (1/4 of the total dataset), dataset 3 (1/3 of the total dataset), dataset 4 (1/2 of the total dataset), and dataset 5 (3/4 of the total dataset), the classification error rates in terms of cyber intrusion prevention are summarized in Table 3.

TABLE 3. Detection rate and false alarm rate of different dataset(%).

Classifier	Training time (s)	Training error rate(%)	Testing time (s)	Testing error rate(%)
Hopfield	932	23.47	577	24.32
SVM	894	20.63	607	22.58
GAN	1848	18.78	1219	19.83
DBN-RFS	1638	14.74	1064	16.52
DDBN	1613	11.92	1016	12.23

TABLE 4. Detection rate and false alarm rate of different testing datasets (%).

Testing data	Detection rate (%)	False alarm rate (%)
Dataset 1	86.43	1.28
Dataset 2	87.92	1.24
Dataset 3	93.47	0.72
Dataset 4	95.68	0.63
Dataset 5	97.53	0.57
Average result	92.21	0.89

The proposed cyber intrusion prevention algorithm is a semi-supervised clustering algorithm, and the detection rate and false alarm rate were used to verify the classification effect.

From the experimental results listed in Table 4, it can be seen that the detection rate and false alarm rate of the cyber intrusion prevention algorithm based on deep learning and semi-supervised clustering are not ideal with less training data. The reason for this shortcoming is that DDBN cannot implement the corresponding comprehensive training under the condition of insufficient data or more disturbed data, and the weight matrix and bias value obtained are not perfect. From the training results of datasets 3, 4, and 5, it can be concluded that the classification effects of the algorithm in terms of in cyber intrusion prevention are good after more comprehensive training.

C. DDBN CYBER PROTECTION METHOD FOR SEMI-SUPERVISED LEARNING BASED ON LOCAL AND NON-LOCAL REGULARIZATION

In the final optimization of the deep learning model, a regularization term is added to the objective function (loss function) to realize a regularization method for deep learning, but the intrusion prevention performance of this method is not ideal.

This study designed a novel regularization method that integrates local and non-local constraint information of labeled and unlabeled cyber traffic data. Using this information, we can extract abstract features for class separability that are effectively preserved in the original samples. For labeled data, the class labels can be used to define local and non-local information. Then, the topology regularization term can be obtained by minimizing the compactness within the class (locality) and maximizing the separability between classes (non-locality). For unlabeled data, the sample's neighbor and non-neighbor samples can be determined

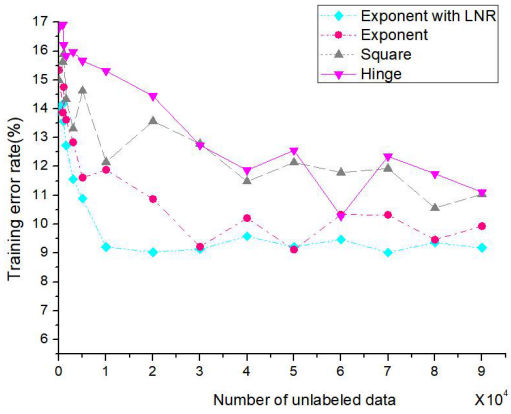

FIGURE 9. Training error rates using different amounts of unlabeled data and different loss functions.

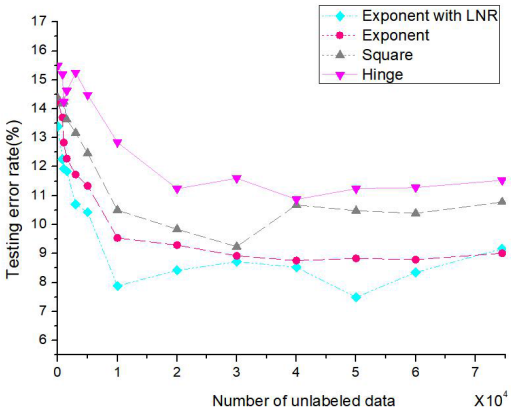

FIGURE 10. Testing error rates using different amounts of unlabeled data and different loss functions.

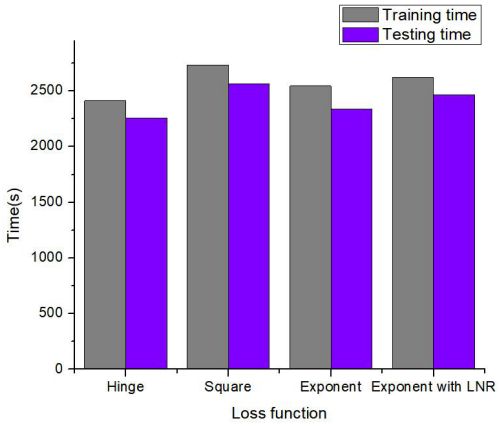

FIGURE 11. Training and testing times of different loss functions of DDBN algorithm.

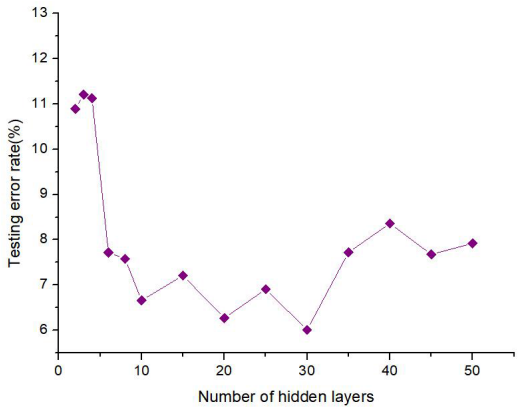

FIGURE 12. Testing error rates using different numbers of hidden layers.

using the average distance from one sample to another as a threshold. Then, the topological regularization term maximizes the non-local divergence and minimizes the local divergence [32].

The experiment investigated the impact of unlabeled data and different loss functions on semi-supervised deep learning. Four types of loss functions were used in the experiment: exponent loss function with local and non-local regularization (exponent with LNR), exponent loss function, square loss function, and hinge loss function. The number of labeled data was unchanged (3000) while the number of unlabeled data took different values, and their error rates were measured.

As shown in Fig. 9 and 10, the training and testing error rates fluctuated with the increase in the unlabeled data. The number of iterations in the experiment is 200. The training and testing error rates obtained using the exponent loss function with LNR were lower than those obtained using the exponent, square, and hinge loss functions.

Fig. 11 shows the training and testing times of different loss functions of the DDBN algorithm. The number of unlabeled data in the training dataset was 9×10^4 and the number of unlabeled data in the testing dataset was 7.45×10^4. The training times of hinge, square, exponent, and exponent with LNR were 2411, 2733, 2545, and 2621 s, respectively, while their testing times were 2258, 2562, 2336, and 2467 s, respectively. The training time and testing time of the exponent loss function are 2545 s and 2336 s, respectively, while those of the exponent loss function with LNR are 2621 s and 2467 s, respectively. The square loss function took the most time, while the exponent loss function with LNR took only a little more time than the exponent loss function.

The next subsection describes experiments conducted using the exponent loss function with local and non-local regularization (exponent with LNR).

D. INTRUSION PREVENTION FOR DEEP ARCHITECTURES OF DIFFERENT DEPTHS

This experiment was performed with a similar 41-300-1800-5 architecture with two hidden layers, and a hidden layer with 300 nodes was continuously inserted into the deep architecture until the number of hidden layers reached 50 and the number of hidden layer nodes increased from 2100 to 16500.

The number of fixed labeled data was 3000, and the number of unlabeled data was 74500. The testing error rates with different numbers of hidden layers are shown in Fig. 12. When the number of hidden layers was greater than or equal

TABLE 5. A deep framework using different numbers of hidden layers and the same number of nodes.

Input layer	Hidden layer	Output layer	Input layer	Hidden layer	Output layer
41	4096	5	41	2048 1024 1024	5
41	2048 2048	5	41	2048 1024 512⋯	5

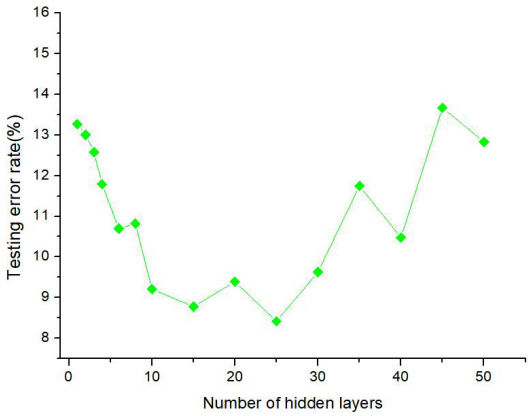

FIGURE 13. Testing error rate using different numbers of hidden layers and 4096 nodes.

to 6, the testing error rate obviously decreased. When the number of hidden layers was 30, the testing error rate of DDBN was the lowest and the effect was the best. Specifically, 10500 hidden nodes were used in this case.

E. INTRUSION PREVENTION FOR DEEP ARCHITECTURES WITH FIXED NUMBER OF NODES AND VARYING DEPTH OF THE DEEP LAYER

In another experiment, the total number of nodes in the fixed hidden layer was 4096 and the depth of the deep structure changed as shown in Table 5.

Fig. 13 shows the testing error rates for different deep architectures using 3000 labeled data and 74500 unlabeled data. When the number of hidden layers was 25, the performance of DDBN was the best.

Fig. 14 shows the relationship between the number of hidden layers and the running time when the total number of hidden layer nodes was 8192, 6144, and 4096. Thus, DDBN is effective and efficient in network intrusion prevention. When the total number of hidden layer nodes was 8192, 6144, and 4096, the running time of DDBN with 10 hidden layers was 2568 s, 2529 s, and 1943 s, respectively. When the total number of hidden layer nodes was 8192, 6144, and 4096, the running time of DDBN with 25 hidden layers was 1894 s, 1856 s, and 1658 s, respectively. When the total number of hidden layer nodes was 8192, 6144, and 4096, the running time of DDBN with 25 hidden layers was 22.74%, 26.61%, and 14.67 % shorter than those of DDBN with 10 hidden layers. The reduced time consumption was not obvious. With the same scale of the deep architecture and the same amount of labeled and unlabeled data, the DDBN with 25 hidden

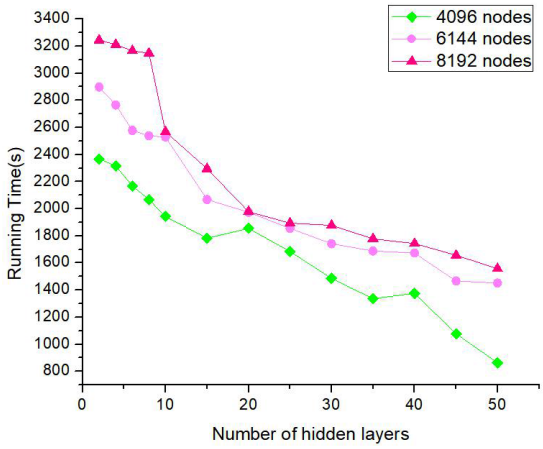

FIGURE 14. Running time using different numbers of hidden layers and different numbers of nodes.

TABLE 6. Experimental NSL-KDD dataset.

Type	Training dataset	Testing dataset
DOS	43826	7259
R2L	873	2368
U2R	49	184
PROBE	10936	2163
TOTAL	55684	11974

layers consumed less time than the DDBN with 10 hidden layers, and it achieved better results in terms of cyber intrusion prevention. The experimental results also showed that the running time slightly decreases as the number of hidden layers increases, especially with 6144 and 4096 hidden layer nodes.

F. ALGORITHM PERFORMANCE COMPARISONS USING DIFFERENT DATASETS

The above-mentioned experiments were based on the KDD Cup99 dataset. Next, we used the NSL-KDD dataset to perform some experiments with different methods and compared the results with those of the KDD Cup99 dataset.

As shown in Table 6, there were 55684 training data and 11974 test data in the NSL-KDD cyber intrusion protection dataset.

We used the NSL-KDD dataset to test different algorithms. The number of testing labeled data of cyber traffic was set as 10, 20, 40, 80, 160, and 320. The experimental results of each classifier with different numbers of testing labeled data are shown in Fig. 15. The results indicate that the testing performance of DDBN is better than that of the other classifiers in terms of cyber intrusion prevention. When the number of labeled data was 320, the testing error rate of DDBN was 11.53% and that of GAN was 13.84%. Thus, the testing error rate of DDBN was 2.31% lower than that of GAN.

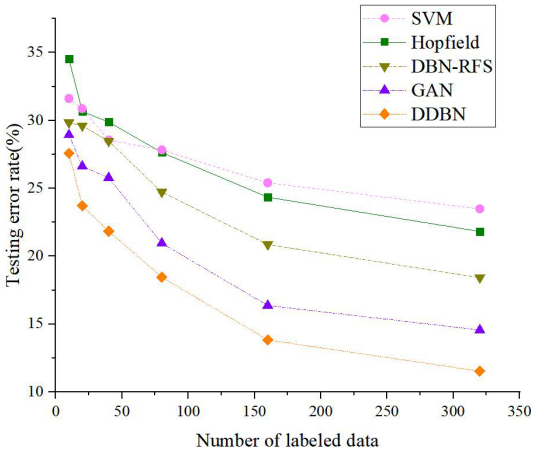

FIGURE 15. Testing error rates on large-scale cyber connection information sets with different numbers of labeled data (NSL-KDD dataset).

TABLE 7. Detection rate and false alarm rate of different datasets for different algorithms(%).

Dataset	Algorithm	Detection rate(%)	False alarm rate(%)
KDD Cup99	Hopfield	88.78	3.12
	SVM	88.93	2.78
	GAN	91.67	2.14
	DBN-RFS	92.73	1.58
	DDBN	96.56	0.64
NSL-KDD	Hopfield	87.31	2.96
	SVM	86.36	3.32
	GAN	91.73	1.34
	DBN-RFS	90.39	2.36
	DDBN	94.58	0.83

Table 7 lists the detection rate and false alarm rate of different datasets for different algorithms. Using the detection rate and false alarm rate as indicators for evaluation on the KDD Cup99 dataset, we found that DDBN has the best effect, followed by DBN-RFS, GAN, SVM, and Hopfield. On the NSL-KDD dataset, we found that DDBN has the best effect, followed by GAN, DBN-RFS, Hopfield, and SVM. Thus, DDBN achieved the best recognition results in terms of cyber intrusion protection on both the KDD Cup99 and the NSL-KDD datasets.

G. RECOGNITION EFFECTS IN UNKNOWN ATTACKS

The KDDCUP99 dataset comprising the test set and training set (as shown in Table 8) was used to test the detection rate of known and unknown attacks [56].

The number of instances including DOS attacks in the testing set was 108487, of which 103280 were known attacks, accounting for 95.2% of the total instances of DOS, while 5207 were unknown attacks, accounting for 4.8% of the total instances of DOS. The numbers of instances of known and unknown attacks for the other three types of attacks are listed in the Table 8.

TABLE 8. Experimental KDD Cup99 dataset for known and unknown attacks.

Type	Training dataset	Testing dataset
DOS	169351 0	103280(95.2%) 5207(4.8%)
R2L	662 0	3500(41%) 5032(59%)
U2R	46 0	15(19.4%) 63(80.6%)
PROBE	1542 0	920(53.2%) 809(46.8%)
NORMAL	46816	37815
Total	218417	156645

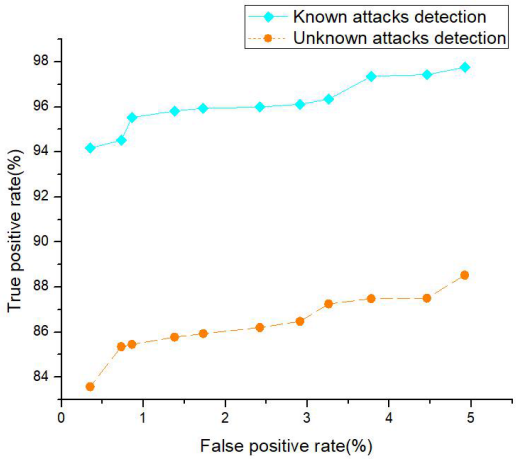

FIGURE 16. ROC curve of DDBN cyber intrusion prevention model for detecting known and unknown attacks.

As can be seen from Fig.16, the DDBN cyber intrusion prevention model achieved high ROC scores in detecting known and unknown attacks, and achieved good results in terms of cyber intrusion prevention detection. When the false positive rate was 4.92%, the true positive rates of known attacks and unknown attacks were 97.76% and 88.53%, respectively.

V. CONCLUSION

This paper proposed a novel and effective cyber intrusion prevention method using DDBN for large-scale semi-supervised deep learning based on local and non-local regularization. Semi-supervised feature selection uses both labeled and unlabeled cyber traffic data to evaluate feature relevance. Specifically, deep learning was applied to cyber intrusion prevention to detect implicit attack behavior based on intrusion data and to reduce the error rate in cyber intrusion prevention. DDBN can considerably enhance its learning ability for cyber intrusion prevention using a large amount of unlabeled data when labeled data are sparse. Semi-supervised deep learning with DDBN using the exponent loss function with local and non-local regularization was found to be more discriminative with performance advantages in cyber intrusion prevention tasks. In addition, an ROC comparison showed that the cyber

intrusion prevention effect of DDBN is better than those of the DBN-RFS, GAN, SVM, and Hopfield classifiers. The comparative experiments demonstrated that the proposed approach can effectively reduce the classification error rate in cyber intrusion prevention.

REFERENCES

[1] D. Berman, A. Buczak, J. Chavis, and C. Corbett, "A survey of deep learning methods for cyber security," *Information*, vol. 10, no. 4, p. 122, Apr. 2019.
[2] H. Wang, J. Ruan, Z. Ma, B. Zhou, X. Fu, and G. Cao, "Deep learning aided interval state prediction for improving cyber security in energy Internet," *Energy*, vol. 174, pp. 1292–1304, May 2019.
[3] D. Santoro, G. Escudero-Andreu, K. G. Kyriakopoulos, F. J. Aparicio-Navarro, D. J. Parish, and M. Vadursi, "A hybrid intrusion detection system for virtual jamming attacks on wireless networks," *Measurement*, vol. 109, pp. 79–87, Oct. 2017.
[4] G.-Y. Hu, Z.-J. Zhou, B.-C. Zhang, X.-J. Yin, Z. Gao, and Z.-G. Zhou, "A method for predicting the network security situation based on hidden BRB model and revised CMA-ES algorithm," *Appl. Soft Comput.*, vol. 48, pp. 404–418, Nov. 2016.
[5] A. AlEroud and I. Alsmadi, "Identifying cyber-attacks on software defined networks: An inference-based intrusion detection approach," *J. Netw. Comput. Appl.*, vol. 80, pp. 152–164, Feb. 2017.
[6] H. Wang, J. Gu, and S. Wang, "An effective intrusion detection framework based on SVM with feature augmentation," *Knowl.-Based Syst.*, vol. 136, pp. 130–139, Nov. 2017.
[7] A. S. A. Aziz, S. E.-O. Hanafi, and A. E. Hassanien, "Comparison of classification techniques applied for network intrusion detection and classification," *J. Appl. Log.*, vol. 24, pp. 109–118, Nov. 2017.
[8] A. Patel, H. Alhussian, J. M. Pedersen, B. Bounabat, J. C. Júnior, and S. Katsikas, "A nifty collaborative intrusion detection and prevention architecture for smart grid ecosystems," *Comput. Secur.*, vol. 64, pp. 92–109, Jan. 2017.
[9] S. Iqbal, M. L. Mat Kiah, B. Dhaghighi, M. Hussain, S. Khan, M. K. Khan, and K.-K. Raymond Choo, "On cloud security attacks: A taxonomy and intrusion detection and prevention as a service," *J. Netw. Comput. Appl.*, vol. 74, pp. 98–120, Oct. 2016.
[10] P. Mishra, E. S. Pilli, V. Varadharajan, and U. Tupakula, "Intrusion detection techniques in cloud environment: A survey," *J. Netw. Comput. Appl.*, vol. 77, pp. 18–47, Jan. 2017.
[11] B. B. Zarpelão, R. S Miani, C. T. Kawakani, and S. C. de Alvarenga, "A survey of intrusion detection in Internet of Things," *J. Netw. Comput. Appl.*, vol. 84, pp. 25–37, Apr. 2017.
[12] S. Niksefat, P. Kaghazgaran, and B. Sadeghiyan, "Privacy issues in intrusion detection systems: A taxonomy, survey and future directions," *Comput. Sci. Rev.*, vol. 25, pp. 69–78, Aug. 2017.
[13] S. Z. Wang, "Research on intrusion prevention technology based on deep learning and semi-supervised clustering," M.S. thesis, School Comput. Sci. Technol., Jiangsu Univ. Sci. Technol., Jiangsu, China, 2016.
[14] N. Milosevic, A. Dehghantanha, and K.-K.-R. Choo, "Machine learning aided Android malware classification," *Comput. Electr. Eng.*, vol. 61, pp. 266–274, Jul. 2017.
[15] Y. Zhang, J. Wu, C. Zhou, and Z. Cai, "Instance cloned extreme learning machine," *Pattern Recognit.*, vol. 68, pp. 52–65, Aug. 2017.
[16] F. Arcelli Fontana and M. Zanoni, "Code smell severity classification using machine learning techniques," *Knowl.-Based Syst.*, vol. 128, pp. 43–58, Jul. 2017.
[17] G.-M. Xian, "An identification method of malignant and benign liver tumors from ultrasonography based on GLCM texture features and fuzzy SVM," *Expert Syst. Appl.*, vol. 37, no. 10, pp. 6737–6741, Oct. 2010.
[18] F. Barboza, H. Kimura, and E. Altman, "Machine learning models and bankruptcy prediction," *Expert Syst. Appl.*, vol. 83, pp. 405–417, Oct. 2017.
[19] S. Cramer, M. Kampouridis, A. A. Freitas, and A. K. Alexandridis, "An extensive evaluation of seven machine learning methods for rainfall prediction in weather derivatives," *Expert Syst. Appl.*, vol. 85, pp. 169–181, Nov. 2017.
[20] C. Amrit, T. Paauw, R. Aly, and M. Lavric, "Identifying child abuse through text mining and machine learning," *Expert Syst. Appl.*, vol. 88, pp. 402–418, Dec. 2017.
[21] S. H. Lee, C. S. Chan, S. J. Mayo, and P. Remagnino, "How deep learning extracts and learns leaf features for plant classification," *Pattern Recognit.*, vol. 71, pp. 1–13, Nov. 2017.
[22] Q. Zhang, X. Chen, Q. Zhan, T. Yang, and S. Xia, "Respiration-based emotion recognition with deep learning," *Comput. Ind.*, vols. 92–93, pp. 84–90, Nov. 2017.
[23] J. Evermann, J.-R. Rehse, and P. Fettke, "Predicting process behaviour using deep learning," *Decis. Support Syst.*, vol. 100, pp. 129–140, Aug. 2017.
[24] C. Luo, D. Wu, and D. Wu, "A deep learning approach for credit scoring using credit default swaps," *Eng. Appl. Artif. Intell.*, vol. 65, pp. 465–470, Oct. 2017.
[25] R. Sheikhpour, M. A. Sarram, S. Gharaghani, and M. A. Z. Chahooki, "A survey on semi-supervised feature selection methods," *Pattern Recognit.*, vol. 64, pp. 141–158, Apr. 2017.
[26] J. Wang, G. Yao, and G. Yu, "Semi-supervised classification by discriminative regularization," *Appl. Soft Comput.*, vol. 58, pp. 245–255, Sep. 2017.
[27] G. Lin, K. Liao, B. Sun, Y. Chen, and F. Zhao, "Dynamic graph fusion label propagation for semi-supervised multi-modality classification," *Pattern Recognit.*, vol. 68, pp. 14–23, Aug. 2017.
[28] W. Hang, K.-S. Choi, S. Wang, and P. Qian, "Semi-supervised learning using hidden feature augmentation," *Appl. Soft Comput.*, vol. 59, pp. 448–461, Oct. 2017.
[29] Y. Wang, Y. Meng, Y. Li, S. Chen, Z. Fu, and H. Xue, "Semi-supervised manifold regularization with adaptive graph construction," *Pattern Recognit. Lett.*, vol. 98, pp. 90–95, Oct. 2017.
[30] M. Tang, F. Nie, S. Pongpaichet, and R. Jain, "Semi-supervised learning on large-scale geotagged photos for situation recognition," *J. Vis. Commun. Image Represent.*, vol. 48, pp. 310–316, Oct. 2017.
[31] Y. Han, Y. Yang, Y. Yan, Z. Ma, N. Sebe, and X. Zhou, "Semisupervised feature selection via spline regression for video semantic recognition," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 2, pp. 252–264, Feb. 2015.
[32] G. C. Cheng, "Local and non-local regulation for semi-supervised deep learning," M.S. thesis, School Comput. Sci. Technol., Tianjin Univ., Tianjin, China, 2014.
[33] M. Qin, Z. Li, and Z. Du, "Red tide time series forecasting by combining ARIMA and deep belief network," *Knowl.-Based Syst.*, vol. 125, pp. 39–52, Jun. 2017.
[34] F. Wu, Z. Wang, W. Lu, X. Li, Y. Yang, J. Luo, and Y. Zhuang, "Regularized deep belief network for image attribute detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 7, pp. 1464–1477, Jul. 2017.
[35] K. Peng, R. Jiao, J. Dong, and Y. Pi, "A deep belief network based health indicator construction and remaining useful life prediction using improved particle filter," *Neurocomputing*, vol. 361, pp. 19–28, Oct. 2019.
[36] L. Zhao, Y. Zhou, H. Lu, and H. Fujita, "Parallel computing method of deep belief networks and its application to traffic flow prediction," *Knowl.-Based Syst.*, vol. 163, pp. 972–987, Jan. 2019.
[37] M. Pan, J. Jiang, Q. Kong, J. Shi, Q. Sheng, and T. Zhou, "Radar HRRP target recognition based on t-SNE segmentation and discriminant deep belief network," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 9, pp. 1609–1613, Sep. 2017.
[38] Y. Liu, S. Zhou, and Q. Chen, "Discriminative deep belief networks for visual data classification," *Pattern Recognit.*, vol. 44, nos. 10–11, pp. 2287–2296, Oct. 2011.
[39] S. Huang, H. Xu, and X. Xia, "Active deep belief networks for ship recognition based on BvSB," *Optik*, vol. 127, no. 24, pp. 11688–11697, Dec. 2016.
[40] Z. Luo, L. Liu, J. Yin, Y. Li, and Z. Wu, "Deep learning of graphs with ngram convolutional neural networks," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 10, pp. 2125–2139, Oct. 2017.
[41] M. Kim, D.-G. Lee, and H. Shin, "Semi-supervised learning for hierarchically structured networks," *Pattern Recognit.*, vol. 95, pp. 191–200, Nov. 2019.
[42] S. H. Park and S. B. Kim, "Active semi-supervised learning with multiple complementary information," *Expert Syst. Appl.*, vol. 126, pp. 30–40, Jul. 2019.
[43] Y. Li, Q. Pan, S. Wang, H. Peng, T. Yang, and E. Cambria, "Disentangled variational auto-encoder for semi-supervised learning," *Inf. Sci.*, vol. 482, pp. 73–85, May 2019.

[44] N. Zrira, H. A. Khan, and E. H. Bouyakhf, "Discriminative deep belief network for indoor environment classification using global visual features," *Cognit. Comput.*, vol. 10, no. 3, pp. 437–453, Jun. 2018.
[45] M. Ma, C. Sun, and X. Chen, "Discriminative deep belief networks with ant colony optimization for health status assessment of machine," *IEEE Trans. Instrum. Meas.*, vol. 66, no. 12, pp. 3115–3125, Dec. 2017.
[46] M. Karabulut and T. Ibrikci, "Discriminative deep belief networks for microarray based cancer classification," *Biomed. Res.*, vol. 28, no. 3, pp. 1016–1024, 2017.
[47] R. Ito, K. Nakae, J. Hata, H. Okano, and S. Ishii, "Semi-supervised deep learning of brain tissue segmentation," *Neural Netw.*, vol. 116, pp. 25–34, Aug. 2019.
[48] X. Tang, F. Guo, J. Shen, and T. Du, "Facial landmark detection by semi-supervised deep learning," *Neurocomputing*, vol. 297, pp. 22–32, Jul. 2018.
[49] C. Chen, Y. Liu, M. Kumar, J. Qin, and Y. Ren, "Energy consumption modelling using deep learning embedded semi-supervised learning," *Comput. Ind. Eng.*, vol. 135, pp. 757–765, Sep. 2019.
[50] C. Chen, R. Zhuo, and J. Ren, "Gated recurrent neural network with sentimental relations for sentiment classification," *Inf. Sci.*, vol. 502, pp. 268–278, Oct. 2019.
[51] R. Collobert, F. Sinz, and J. Weston, "Large scale transductive SVMs," *J. Mach. Learn. Res.*, vol. 7, pp. 1687–1712, Aug. 2006.
[52] K. Takahashi, K. Kim, T. Ogata, and S. Sugano, "Tool-body assimilation model considering grasping motion through deep learning," *Robot. Auto. Syst.*, vol. 91, pp. 115–127, May 2017.
[53] N. Zhang, S. Ding, J. Zhang, and Y. Xue, "Research on point-wise gated deep networks," *Appl. Soft Comput.*, vol. 52, pp. 1210–1221, Mar. 2017.
[54] Z. Zhao, L. Jiao, J. Zhao, J. Gu, and J. Zhao, "Discriminant deep belief network for high-resolution SAR image classification," *Pattern Recognit.*, vol. 61, pp. 686–701, Jan. 2017.
[55] D. Borbor, L. Wang, S. Jajodia, and A. Singhal, "Optimizing the network diversity to improve the resilience of networks against unknown attacks," *Comput. Commun.*, vol. 145, pp. 96–112, Sep. 2019.

GUANGMING XIAN (Member, IEEE) received the B.Eng., M.Eng., and Ph.D. degrees from the South China University of Technology, Guangzhou, China, in 1998, 2003, and 2007, respectively. From 2009 to 2011, his postdoctoral scientific project research was jointly carried out in the computer science and technology postdoctoral scientific research flow station of South China University of Technology and the postdoctoral scientific research workstation of Guangzhou Tianhe Software Park Management Committee. The content of his postdoctoral research report was the application of machine learning in financial time series prediction. He is currently an Associate Professor with the School of Software, South China Normal University, Foshan, China. He has published a series of articles in academic journals and conferences. His current research interests include artificial intelligence, pattern recognition, machine learning, deep learning, and cyberspace security.