Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 343050, 10 pages
http://dx.doi.org/10.1155/2015/343050

Research Article

Cellular Neural Network-Based Methods for Distributed Network Intrusion Detection

Kang Xie,[1] Yixian Yang,[1,2] Yang Xin,[2] and Guangsheng Xia[3]

[1]*College of Information Science and Engineering, Shandong University, Jinan 250100, China*
[2]*Information Security Center, Beijing University of Posts and Telecommunications, Beijing 100876, China*
[3]*National Cybernet Security Ltd., Beijing 100088, China*

Correspondence should be addressed to Kang Xie; xiekang1987@sina.com

Received 8 July 2014; Accepted 13 January 2015

Academic Editor: Alessandro Gasparetto

According to the problems of current distributed architecture intrusion detection systems (DIDS), a new online distributed intrusion detection model based on cellular neural network (CNN) was proposed, in which discrete-time CNN (DTCNN) was used as weak classifier in each local node and state-controlled CNN (SCCNN) was used as global detection method, respectively. We further proposed a new method for design template parameters of SCCNN via solving Linear Matrix Inequality. Experimental results based on KDD CUP 99 dataset show its feasibility and effectiveness. Emerging evidence has indicated that this new approach is affordable to parallelism and analog very large scale integration (VLSI) implementation which allows the distributed intrusion detection to be performed better.

1. Introduction

Intrusion detection systems (IDS), which use classification algorithms, effectively discriminate normal behavior from abnormal behavior and play a major role in providing security to networks [1]. Traditional IDS can be divided into three categories [2]: integral structure, which integrates all functions but is weak on system prevention [3], and hierarchal structure, which is composed of detector and controller like a tree and shares information of all subsystems for intrusion detection, but the disadvantage of this structure is single-point failure [4]. Distributed structure IDS can be divided into several modules which distribute in the heterogeneous network environment, receive different information, and report the results to the higher function units. According to the increase of network size and complexity, the distributed structure IDS is matched with the characters of modern network environment and become one of the most important problems in network information security [5, 6]. The comparison of current IDS policy is shown as Table 1.

The model of distributed IDS (DIDS) can be defined either centralized or distributed. The centralized DIDS is a combination of individual sensors which collect the network data to the central analyzer component where the collected data is stored and processed. In [7], many artificial intelligence techniques are being used for threats detection, such as genetic algorithm, artificial immune and artificial neural network (ANN). Hosseinpour et al. [8] propose a multilayered framework based on Artificial Immune System (AIS) to enhance the detection performance; in their design, the genetic algorithm is used for enhancing the secondary immune response. The AIS-based IDS consists of two main components: IDS central engine and detection sensors. Each of these components is composed of some agents which correlate with each other in order to detect the anomalies and intrusions. Bartos et al. presents a distributed self-organized model for collaboration of multiple heterogeneous IDS sensors in [9]. In order to optimize behavior of each IDS sensor with respect to other sensors in highly dynamic environments, the distributed model is based on a Game-theoretical approach and introduces E-FIRE which is a solution concept suitable for solving the game. In the above centralized intrusion detection model, all the network data are sent to a central unit for processing; the raw data

Table 1: The comparison of current IDS policy.

Title	Type	Characteristics	Limitations
IDS	Integral structure	Integrates all functions	Weak on system prevention
	Hierarchal structure	Shares information of all subsystems	Single-point failure
	Distributed structure	Combination of individual sensors	Data communications are too large and complex

Table 2: The comparison of current DIDS policy.

Title	Type	Characteristics	Limitations
DIDS	Centralized architecture DIDS	Artificial intelligence techniques [7]	(1) Data communications may occupy considerable network bandwidth. (2) The privacy of the data cannot be protected and is very sensitive to Denial of service attack.
		Multilayered framework [8]	
		Self-organized model [9]	
	Distributed architecture DIDS	ANN [12]	There is a large number of communications between local nodes.
		Signature-based multilayer IDS [13]	
		Dynamic Election Algorithm [14]	

communications may occupy considerable network bandwidth and cause a computational burden in the central site. The privacy of the data obtained from the local nodes cannot be protected and is very sensitive to denial of service attack.

The distributed architecture DIDS includes one or more devices that cooperate in order to perform data gathering and processing reporting functions. Agent-based systems are widely used in distributed model, which is defined as a distinct software process that can be able to carry out activities in an intelligent manner to responsive changes in the environment and cooperate with other agents [11]. El Kadhi et al. [12] propose a global agent architecture using ANN as a major decision algorithm. Uddin et al. [13] propose a signature-based multilayer IDS model, which can detect imminent threats by automatically creating and using small multiple databases and provide mechanism to update these small signature databases at regular intervals using mobile agents. In [14], an agent based distributed adaptive IDS is put forward which employs joint detection mechanism for data mining algorithm and dynamic election algorithm for the recovery mechanism. In an agent-based DIDS, there is no central station and central point of failure, therefore overcoming the deficiency of centralized structure. But the limitation is that many raw network data still need to be shared among distributed nodes which make a large number of communications between local nodes. The comparison of current DIDS policy is shown as Table 2.

In each detection node of DIDS, there are various machine learning based anomaly detection algorithms proposed in the literature. These algorithms can be classified as statistics based, data mining based, and classification based [15]. Statistics based algorithms construct statistical models of network connections to determine whether a new connection is an attack. In [16], an adaptive blacklist-based packet filter using a statistic-based approach is proposed, the filter employs a blacklist technique to help filter out network packets based on IP confidence and the statistic-based approach allows the blacklist generation and update periodically in an adaptive way. Data mining based algorithm determines whether a new connection is an attack by using mine rules. For example, Mao et al. [17] propose two models based on data mining technology, respectively called frequency patterns (FP) and tree patterns (TP). FP based on frequency analysis and uses a short sequence model to find out frequent sequential patterns in the training system-call. TP make use of tree pattern mining and can get a quality profile from the training system-call sequences. Classification based IDS algorithm construct a classifier that is used to classify network connection as either attacks or normal connections. Pani and de Toro [18] propose an additive decision rules binary classifier which is optimized by a multiobjective evolutionary algorithm in order to maximize the classification accuracy and the coverage level. In [19], a solution for IDS based on the Coarse-to-refined Grid Search Support Vector Machine (GS-SVM) is presented to identify massive intrusive behaviors. In order to classify online traffic data efficiently, Mitra et al. [20] propose a rule based lazy classifier by using genetic algorithm. Sravani and Srinivasu classify the network traffic data by using Naïve Bayes and Neural network in [21]. Although there is much more work on anomaly detection algorithms, several issues still require further research, especially in the following areas: as new type of attack emerge, the size of intrusion training data increases rapidly over time. The intrusion detector must be retrained periodically in order to keep up with the changes in the network; in this way, it is time consuming. In addition, new attack data are used to update the detector and are then discarded. The key issue of detection algorithm in each node is to improve the processing time and maintain the accuracy of intrusion detector. The comparison of current machine learning based IDS policy in each detection node is shown as Table 3.

In this paper, we address the above challenges and propose a cellular neural network (CNN) based classification framework for the dynamic distributed network intrusion detection by using the discrete-time (DT) CNN [22] and state-controlled (SC) CNN algorithm [23]. These networks take most of their inspiration from the CNN paradigm but have some different peculiarities that make them preferable

Table 3: The comparison of current machine learning based IDS policy in each detection node.

Title	Type	Characteristics	Limitations
Machine learning based IDS in each detection node	Statistics based	Adaptive blacklist-based packet filter [16]	Processing time should be improved and the accuracy of intrusion detector should be maintained
	Data mining based	Frequency patterns (FP) and tree patterns (TP) [17]	
	Classification based	An additive decision rules binary classifier [18]	
		Coarse-to-refined Grid Search Support Vector Machine (GS-SVM) [19]	
		Lazy classifier by using genetic algorithm [20]	
		Naïve Bayes and Neural network [21]	

for some applications [10, 24, 25]. CNN is one of the most popular machine learning algorithms [26]. The key features of CNN are asynchronous parallel processing, continuous time dynamics, and local interactions among network elements which are called cells. The CNN system can be implemented as a mixed signal very large scale integration (VLSI) chip, the CNN universal machine (CNNUM). Taking advantage of these characteristics, CNN are widely used in research applications that are particularly demanding in terms of computational and time requirements, parallelization of sensing and processing, such as real-time image processing [27], pattern recognition [28], multisensory fusion, and control of complex systems [25, 29, 30].

In the proposed architecture, the system includes local nodes based on DTCNN and global model based on SCCNN. Each local node is responsible for detecting signs of intrusion independently, then exchanging their local information by sending an update message periodically to their neighboring nodes. On the other hand, the global model could be seen as a complete route picture of the whole network.

The rest of this paper is organized as follows. Section 2 introduces the theoretical foundation. In Section 3, the proposed algorithm of DTCNN-based local intrusion detection model is discussed. Section 4 presents the method for constructing the SCCNN-based global detection model and explains the details for designing SCCNN parameters via solving the Linear Matrix Inequality (LMI). After that, Section 5 presents the experimental results, followed by the conclusions in Section 6.

2. Theoretical Foundation

Consider a linear, time invariant system G described by state-space equations:

$$G: \begin{cases} \dot{x} = Ax + Bu, \\ y = Cx, \end{cases} \tag{1}$$

where $x \in R^n$ is the state vector, $u \in R^m$ is the control input, and $y \in R^r$ is the measured output. The pairs (A, B) and (A, C) are assumed to be, respectively, stabilization and measurable, and we assume that

(A1): $m \le r$ and $\operatorname{rank}(CB) = m$.

Lemma 1 (see [31]). *System G is asymptotically stable by a static output feedback if and only if there exist matrix $P > 0$, $\sigma_1 > 0$, and $\sigma_2 > 0$ such that*

$$\begin{aligned} A^T P + PA - \sigma_1 PBB^T P < 0, \\ A^T P + PA - \sigma_2 C^T C < 0. \end{aligned} \tag{2}$$

Lemma 2 (see [32]). *For the system G satisfies (A1), there exist a liner transformation, which makes input matrix and output matrix have new structures as follows:*

$$\begin{aligned} B &= \begin{bmatrix} 0 \\ I_m \end{bmatrix}, \\ C &= \begin{bmatrix} C_1 & C_2 \end{bmatrix}, \end{aligned} \tag{3}$$

where $C_2 \in R^{r \times m}$ is full rank. R^{m*n} represents real $m \times n$ matrices and I_m is $m \times m$ identity matrix. $M > 0$, where M is symmetric and positive definite, and $A + (*)^T = A + (A)^T$.

Formula (3) are definite as the specification style. Without any lose of generality, we regard the system have such specification style.

Lemma 3 (see [32]). *For the system G satisfies (A1), when μ tends to zero if there exist matrix $Q_1 \in R^{(n-m)\times(n-m)}$, $W \in R^{r\times r}$ and $Z \in R^{m\times(n-m)}$ satisfy follow inequalities:*

$$\begin{aligned} Q_1 &> 0, \\ \begin{bmatrix} (A_{11}Q_1 - A_{12}Z) + (*)^T & (C_1Q_1 - C_2Z)^T \\ C_1Q_1 - C_2Z & -W \end{bmatrix} &< 0, \\ 0 < W &< \mu I. \end{aligned} \tag{4}$$

There must exist T and $\widehat{P}$ satisfying formula (2), where $N = ZQ_1^{-1}$.

3. Local Detection Models Based on DTCNN

Intrusion detection problem can be viewed as a pattern classification problem. By building profiles of authorized behaviors, the computer behavior will be classified into authorized or intrusive behavior. DTCNN, as introduced in [22], is a combination from general linear threshold networks, where

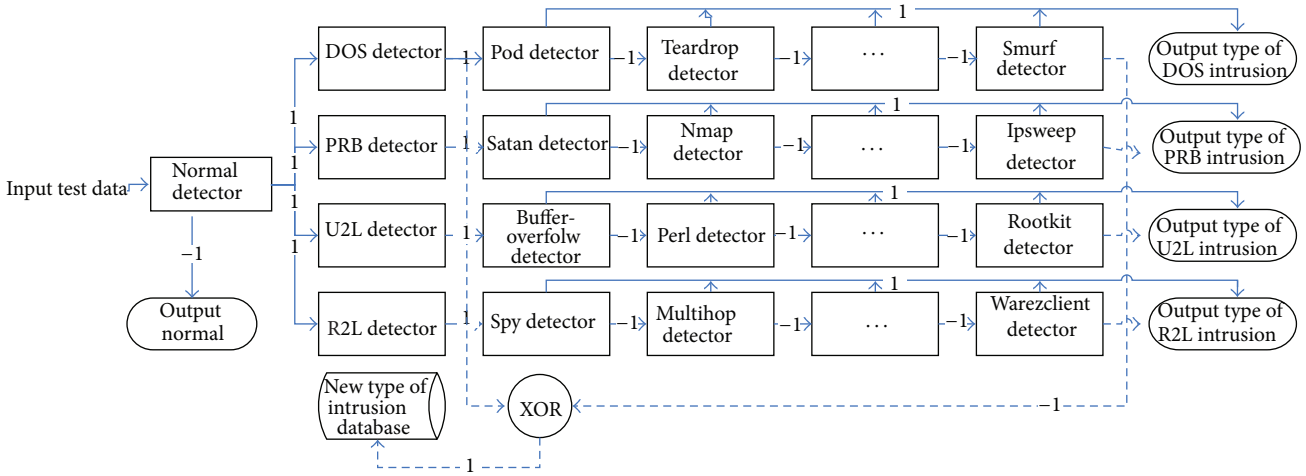

Figure 1: The Hierarchical Intrusion Detection model based on DTCNN for local node [10].

the local cell connectivity and translational invariance of the weights were transferred from CNN, represents an efficient architecture for pattern recognition. Because a cell in the DTCNN has a stable equilibrium point at the two saturation regions of the piece wise linear output function after the transient has decayed to zero. The output at the two saturation regions corresponds to 1 and −1. This significant characteristic of DTCNN shows a possibility as a classifier. In the following, we introduce the classifiers based on DTCNN for intrusion detection.

The circuit equations of a DTCNN cell which satisfies *KCL* and *KVL* are easily derived as follows.

State equation:

$$X(n) = \sum_{d \in N_r(c)} a_d y^d(n) + \sum_{d \in N_r(c)} b_d u^d + i = AY(n) + BU + I. \quad (5)$$

Output equation:

$$Y(n) = F(X(n-1)) = \begin{cases} 1, & \text{for } X(n-1) > 0, \\ -1, & \text{for } X(n-1) < 0, \end{cases} \quad (6)$$

where $U = [u_0 \quad u_1 \quad \cdots \quad u_n]^T$ is the input numeric value and $X = [x_0 \quad x_1 \quad \cdots \quad x_n]^T$ is the state. $Y = [y_0 \quad y_1 \quad \cdots \quad y_n]^T$ is the output which is binary and is determined by the sign of $X(n-1)$, $Y = 1$ denotes an intrusive behavior, $Y = -1$ denotes an authorized behavior, and $F = [f(x_0) \quad f(x_1) \quad \cdots \quad f(x_n)]^T$ contains the output functions. The value of I is constant and used for adjusting a bias, the matrices **A** and **B** contain the feedback and control parameters, respectively. A set of feedback, control coefficients, and the bias is called a template which is translational invariant. It means that each cell is influenced identically by its neighbors. This reduces the different weights to a small number that can be implemented by global bus lines in VLSI realizations.

The DARPA 1998 and 1999 ID evaluation data sets consist of comprehensive technical evaluation of research IDS. The network connection records fall into four main categories: the denial of service (DOS) includes pod, teardrop, neptune, and smurf. The surveillance and other probing (PRB) consist of satan, portsweep, nmap, and ipsweep. The unauthorized access to local super root privileges (U2R) includes loadmodule, perl, buffer-overflow, and rootkit. The unauthorized access from a remote machine (R2L) consists of spy, guess-passwd, multihop, and warezclient. Each of these data record has 41 unique features, including 34 numeric and 7 discrete features that its value can be converted to numeric value, and the last is the flag of attribute.

The structure of the proposed DTCNN model is shown in Figure 1. The first level is a DTCNN detector which is trained by using normal data, distinguishing abnormal attacks from normal ones. If the first level iteration result $y_{\text{fl}}(n)$ is equal to −1, the test data is reported as normal. When output value $y_{\text{fl}}(n)$ is reported as 1, the lower levels of the HDCNN model are joined in parallel ways which will give the particular type of attacks according to the lower level result $y^i_{\text{ll}m}(n)$, where m is the number of layers, i is the kind of attacks, such as DOS, R2L, and so forth.

If $y_{\text{fl}}(n) = 1$, but $z_{\text{ll2}}(n) = y^{\text{DOS}}_{\text{ll2}}(n) \oplus y^{\text{PRB}}_{\text{ll2}}(n) \oplus y^{\text{U2L}}_{\text{ll2}}(n) \oplus y^{\text{R2L}}_{\text{ll2}}(n) = -1$, the test data will be determined as new category of attacks. Otherwise, $y^{\text{DOS}}_{\text{ll2}}(n) = 1$, but $z_{\text{DOS}}(n) = \sum_{m=3}^{i} y^{\text{DOS}}_{\text{ll}m} = -1$, the data can be identified as the new type intrusion of DOS attacks. Finally, the new category of intrusions will be added to the database to train new detectors. The output type of intrusion is labeled as $1, 2, \ldots$ depending on attack samples, and the normal samples are labeled as −1.

The above design of classifiers for intrusion detection has the following advantages. There is only iteration operation in a detection processing, the computation complexity for constructing the decision stumps is very low and online updating of new type of intrusion database can be easily implemented when new intrusion samples are obtained. Furthermore, the output decision stumps fully take into account the type of each attack or the normal samples. Finally, a suitable choice of the template coefficients allows for

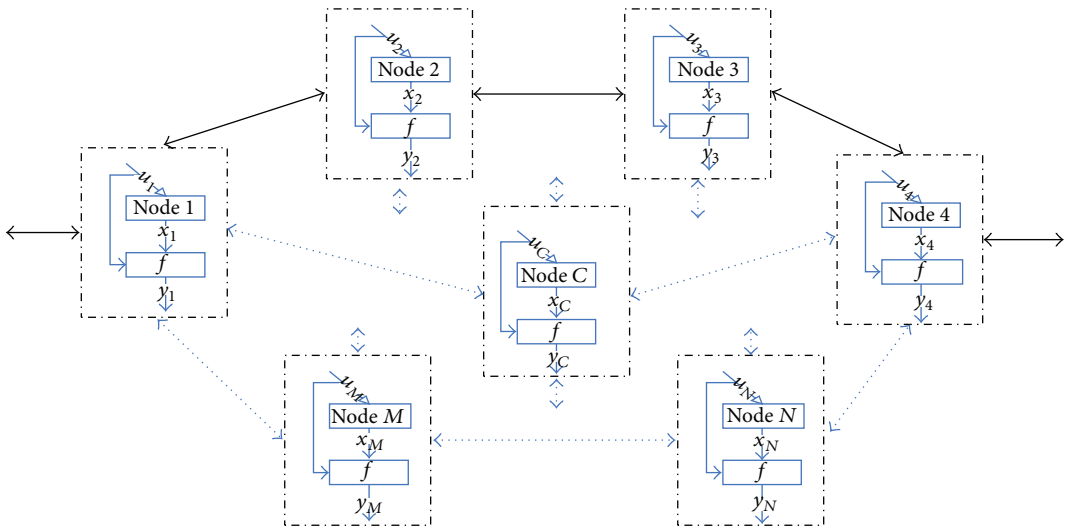

Figure 2: The DIDS architecture based on the SCCNN paradigm.

determining the behavior of the whole system. The improved PSO algorithm carries out the training task and proved in our other paper [10].

4. Global Detection Models Based on SCCNN

In the distributed intrusion detection framework, each node constructs its own local IDS model independently according to its own data. By combining all the local models, a global model trained using a small number of detection results from each node, without sharing any of the original training data between nodes. The global intrusion detector is more accurate than the local detectors that may be only adequate for each node output, specific attack types, or normal samples, due to the limited training data available at each node. Once local nodes gain their own models, the global models are used to detect intrusions, for the SCCNN connection, the vector of the results from the local models is used as the input to the global whose result determines whether the current network connection is an attack.

Furthermore, DIDS refers to the processing of a large set of data in order to derive hacking information that cannot be obtained by using each node alone. Therefore, real-time processing, high computational capability, and a space distributed parallel structure are required to gather and manage the huge amount of information carried out from the local node IDS system efficiently. This is satisfied for SCCNN as introduced in [23], whose main advantage lies in the possibility to control each node on the basis of global information regardless of having only local connections among neighboring nodes. So, by combining the ability of the SCCNN for sample sets, a global detection model can be constructed effectively in each node.

SCCNN are arrays of locally interconnected analog cells arranged in a regular grid, and the processing is controlled by the values of the templates. The main difference between CNN and SCCNN is that each cell of SCCNN contains the sensing and the actuation part inside its structure. In the following, linear characteristics for DIDS will be considered and some sufficient conditions for the stability of the whole structure will be derived, together with a design methodology for the template coefficient selection. Figure 2 gives the framework that consists of the modules: local models and global model.

As Figure 2 shown, SCCNN-DIDS architectures allow implementing global functions on data obtained from distributed IDS node systems having only local connectivity. In particular, due to the intrinsic characteristics of SCCNN, each node as a cell which output is a function of all the SCCNN cells inputs without a direct global cell interconnection. Following Arena et al. [25, 30], the multilayer version of the SCCNN come in a straightforward way, the architecture proposed for one-dimensional DIDS is a circular, and each cell input u_i represents the output for the node N_i, that is, a vector $(y_1, y_2, y_3, \ldots, y_j)$ is constructed, where y_j is the result of the jth local detection model for the sample; each node is connected via the state template C and the control template B to the neighboring nodes. Symmetric templates are considered due to the circular topology adopted: $A = [a_1, a_0, a_1]$ and $B = [b_1, b_0, b_1]$. The generalized linear circular SCCNN equations describing system dynamics can be written for N cells as follows:

$$\begin{aligned}
\dot{x_1} &= -x_1 + a_1 x_N + a_0 x_1 + a_1 x_2 + b_1 u_N \\
&\quad + b_0 u_1 + b_1 u_2, \\
\dot{x_2} &= -x_2 + a_1 x_1 + a_0 x_2 + a_1 x_3 + b_1 u_N \\
&\quad + b_0 u_1 + b_1 u_2, \\
&\vdots \\
\dot{x_N} &= -x_N + a_1 x_{N-1} + a_0 x_N + a_1 x_1 \\
&\quad + b_1 u_{N-1} + b_0 u_N + b_1 u_1,
\end{aligned}$$

$$y_i = F\left(N, a_0, a_1, b_0, b_1\right) = \sum_{j=-n}^{n} C\left(j,n\right) u_j, \quad (i = 1, \ldots, N), \tag{7}$$

where $N = 2n + 1$, with the following periodic boundary conditions: $u_0 = u_N$ and $u_{N+1} = u_1$. It should be remarked that the considered SCCNN is linear, and $F(*)$ represents the state value of the generic cell. The weighting function $C(j,n)$ has the following form:

$$C\left(j,n\right) = \frac{X\left(j,n\right)}{\left(a_0 + 2a_1 - 1\right) Y\left(n\right)}, \tag{8}$$

where j ($j = 0, \ldots, n$) represents the distance between the cell considered and the cell whose considered input is connected. Moreover, the function $C(j,n)$ is symmetric with respect to j. In order to ensure a control law for the structure, the desired coefficient parameter vector $C^* = [C^*(0,n)]$ is imposed to be the same for all the nodes, which is the same quantity, is available at each node regardless of the network dimension. Furthermore, in order to ensure the asymptotic stability of the system, the formula (9) must be satisfied [25], this result ensures that the multinode system will converge to an asymptotically stable state:

$$a_0 + 2a_1 - 1 < 0. \tag{9}$$

It is clear that a suitable choice of the template coefficients which determine the behavior of the system must be satisfied, in order to obtain the asymptotic stability of the system. The design problem consists in determining the template coefficients which is still an open issue in current literature [10, 33]. In the following, we introduce a new methodology for the design of the template coefficients based on LMI.

For the DIDS architecture shown as Figure 2 with N cells, the asymptotic behavior can be written in matrix form as follows:

$$\dot{X} = AX + BU, \quad Y = CU, \tag{10}$$

where **X**, **U** are vectors of dimension equal to the number N of nodes, while **A**, **B** are square real matrices $N \times N$.

Without any loss of generality, let

$$P = P^T \in R^{n\times n}, \qquad P = \begin{bmatrix} P_1 + N^T P_2 N & N^T P_2 \\ P_2 N & P_2 \end{bmatrix} \tag{11}$$

with $P_1 \in R^{(n-m)\times(n-m)}$, $P_2 \in R^{m\times m}$, and $N \in R^{m\times(n-m)}$. Define

$$T = \begin{bmatrix} I_{n-m} & 0 \\ N & I_m \end{bmatrix}, \qquad \widehat{P} = \begin{bmatrix} P_1 & 0 \\ 0 & P_2 \end{bmatrix}; \tag{12}$$

then, we get

$$P = T^T \widehat{P} T. \tag{13}$$

Through (11) and (12), inequality group (2) can be written as

$$\widehat{P}\widehat{A} + \widehat{A}^T\widehat{P} - \sigma_1 \widehat{P}\widehat{B}\widehat{B}^T\widehat{P} < 0, \quad \widehat{P}\widehat{A} + \widehat{A}^T\widehat{P} - \sigma_2 \widehat{C}^T\widehat{C} < 0, \tag{14}$$

where

$$\widehat{A} = TAT^{-1} = T\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} T^{-1}, \tag{15}$$

$$\widehat{B} = TB = \begin{bmatrix} 0 \\ I_m \end{bmatrix}, \tag{16}$$

$$\widehat{C} = CT^{-1} = \begin{bmatrix} C_1 - C_2 N & C_2 \end{bmatrix}. \tag{17}$$

Theorem 4. *System G is asymptotically stable by a static output feedback if and only if there exist matrix $\widehat{P} > 0$, $Y \in R^{m\times m}$, and $N \in R^{m\times(n-m)}$ such that*

$$\widehat{P}\widehat{A} + \widehat{B}Y\widehat{C} + (*)^T < 0, \tag{18}$$

with $K = P_2^{-1}Y$.

Proof. In order to compute a static output feedback law $u = Ky$ that ensures the stability of the system, there exists a matrix $P > 0$, which makes

$$P\left(A + BKC\right) + \left(A + BKC\right)^T P < 0. \tag{19}$$

Corresponding to (14), (19) can be written as

$$T^T\widehat{P}TA + T^T\widehat{P}TBKC + (*)^T < 0. \tag{20}$$

Making congruent transformation for (20), we get

$$\widehat{P}TAT^{-1} + BP_2KT^{-1} + (*)^T < 0. \tag{21}$$

Letting $Y = P_2K$, we get

$$\widehat{P}\widehat{A} + \widehat{B}Y\widehat{C} + (*)^T < 0. \tag{22}$$

This completes the proof. □

Corresponding to Theorem 4, it is clear that Lyapunov inequality (18) changes into two inequalities containing $\widehat{P} > 0$ and Y, when N is fixed. And our aim is to compute a static output feedback law $u = Ky$ that ensures the stability of the system. There exists a matrix K makes (19) holds. Lemma 3 gives us a solution to determine matrix A, B, C, and K. According to this, in the following, we propose a method based on LMI to solve the template parameters.

Step 1. Corresponding to Lemma 2, change the system to specification style structured like (3).

Step 2. Solve inequalities (4) and get the solution matrix N when μ reach the minimum value.

Step 3. Corresponding to (12), we get the matrix T and the system matrix $\{\widehat{A}, \widehat{B}, \widehat{C}\}$.

Step 4. Solve the inequalities $\widehat{P}\widehat{A} + \widehat{B}Y\widehat{C} + (*)^T < 0$, where $\widehat{P} > 0$ and structured like (12).

Finally, we can get $K = P_2^{-1}Y$.

Table 4: Experimental data.

Categories	DOS	PRB	U2R	R2L	Normal	Others	Total
Training data	1596	980	52	470	1065	0	4163
Test data	1398	796	228	689	850	729	4690

Table 5: The results of handle mixed features.

Features	Training data		Test data	
	DR (%)	FAR (%)	DR (%)	FAR (%)
Continuous	98.76	6.50	91.35	10.38
Continuous and categorical	99.31	3.78	94.7	7.67

5. Experiments

Our algorithms are implemented on a Pentium IV computer with 2.6 GHZ CPU and 256 M RAM, using MATLAB7.0. The knowledge discovery KDD CUP 1999 dataset, which is still the most trustful and credible public benchmark dataset for evaluating network intrusion detection algorithms, are being used to test our algorithms. In the dataset, 41 features describe the basic information about the network packet, network traffic, host traffic, and content information, including nine categorical features, which is discrete type variables, and 32 continuous features are extracted for each network connection, and attacks in the dataset fall into four main categories: DOS, Probe, R2L and U2R. We take part of the kddcup.data_10_percent as the data resource, in which the test dataset includes some attack types that do not exist in the training dataset. The numbers of normal connections and each category of attacks in the training and test datasets are listed in Table 4.

In the following, we first introduce the performances of our local intrusion detection models based on DTCNN, including the ability to handle mixed features and the ability to learn new types of attacks. Then, the performance of our SCCNN-based distributed intrusion detection algorithm is evaluated, such as the effects of parameters on detection performance, the comparison results with other published existing algorithms.

5.1. Local Models. For the local models, the results of handle mixed features are shown as Table 4, when only continuous features or both continuous features and categorical features are used, test on the training and test datasets, respectively. And the percentage of detection rate (%DR) and false alarm rate (%FAR) are chosen to evaluate the performance for anomaly detection. It can be seen that the results obtain by using both continuous and categorical features are more accurate than the results obtained by only using continuous features, which shows the ability of our algorithms to handle mixed features in network connection data. Furthermore, the data set has high dimensionality and contains a lot of irrelevant attributes. After the analysis, we only need to extract the first 10 eigenvectors which can be converted to numeric value as the input principal component (Table 5).

The ability to detect the new types of attacks correctly is an attractive and important characteristic of our DTCNN-based intrusion detection algorithms. Figure 3 show the online processing of the classifier with respect to four types of attack, that is, "neptune," "satan," "buffer overflow," and "multihop," which belong to the four main categories, respectively. They all appeared before in the training data.

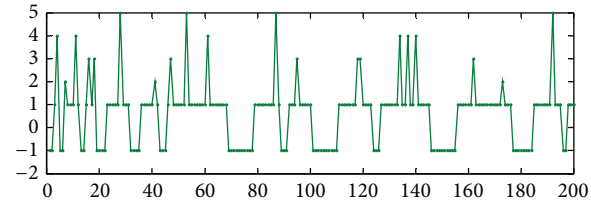

Figure 3: Online processing of the classifier for "neptune," "satan," "buffer overflow," and "multihop" attack.

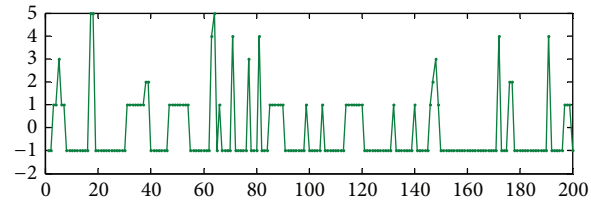

Figure 4: Online processing of the classifier for "mailbomb," "mscan," "ps," and "snmpguess" attacks.

In Figures 3 and 4, the horizontal coordinate indicates the numbers of samples, and the vertical coordinate indicates the class label predicted by the detector for a sample, where $y = 1$ means that it is correctly classified as a network attack. Corresponding to the four kind of attack, $y = 2, 3, 4, 5$ means that the sample is mistakenly classified as a normal connection or other type of attack, and $y = -1$ means that it is correctly classified as a normal connection. It is seen that at the part of processing for four types of attacks, the classifier classifies samples as attacks correctly, and the false alarm rate is low. But the performance of our DTCNN classifier for "buffer overflow" and "multihop" attack which belong to U2R and R2L achieves lower detection rate than other two types of intrusion. Corresponding to the four kind of sample attack, the detection rate are 99.63%, 98.89%, 72.73%, and 77.78%. This is mainly due to the few training data sets for "buffer overflow" and "multihop" attack. But this experiment

Table 6: The results of various template matrices coefficient values.

Model	a_0	a_1	b_0	b_1	K
1	−10	7	10	10	1.0618
2	−10	7	1	0.5	0.7825
3	−2	1	1	0.5	0.4872
4	−0.3	0.7	0.2	0.1	No exit

Table 7: Test datasets in the seven nodes.

Attack	Node 1	Node 2	Node 3	Node 4	Node 5	Node 6	Node 7
Neptune	500	500	500	0	500	300	250
Satan	500	500	0	500	500	250	150
Httptunnel	150	0	150	150	150	100	50
Warezmaster	0	250	250	250	250	150	100
Normal	750	750	750	750	0	500	300

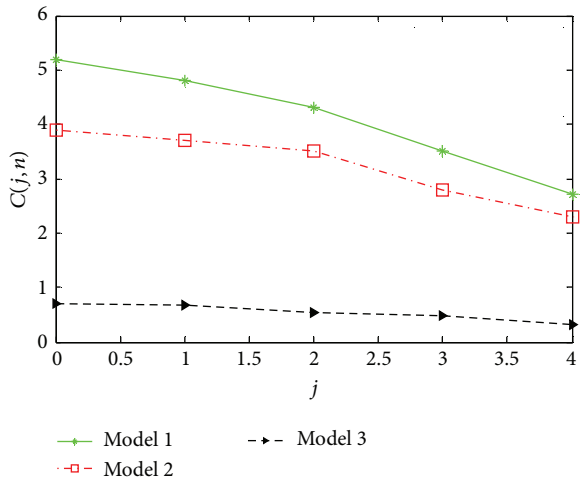

Figure 5: $C(j,n)$ for various template matrices coefficient values.

still shows the effectiveness of the online processing of the algorithm for four type of attacks.

Figure 4 shows the online process of the classifier with respect to four new types of attack, "mailbomb," "mscan," "ps," and "snmpguess" which have not appeared before in the training data. It is seen that for new types of attacks, the proposed DTCNN-based intrusion detection model also can detected the attack accurately. The false alarm rate is low and emerging experiment results have indicated that our DTCNN-based algorithms are also effective for new kinds of attacks.

5.2. Global Models. (a) According to LMI, the coefficients must be chosen as to minimize the quantity $|C^* - C|$ and imposing the stability constraint (17). Table 6 and Figure 5 give us the computation results for various template matrices coefficient values ($n = 4$).

For different model, it can be derived that the closer the A template coefficients are to the stability condition (8), the smaller is $c(j,n)$. And in some case, like model 4, K is not exit. Under that condition, it means the system is not stable. Finally, the cell parameters A and B are chosen based on both dynamic and stability considerations, while the C allows customizing the cell output. The desired weighting function was fixed to $C = [1.1034, 0.7945, 1.1034, 0.7945]$, and the template coefficients were determined as $a_0 = -0.3187$, $a_1 = 0.1496$, $b_0 = 0.007$, and $b_1 = 0.256$.

(b) The proposed SCCNN-based distributed intrusion detection algorithm is tested with seven nodes. To simulate a distributed intrusion detection environment, neptune, satan, httptunnel, and warezmaster which belong to the four main kinds of attacks in the KDD CUP 99 dataset are used for constructing local detection nodes, and a global model is constructed using the seven local models. Table 7 shows the test datasets used for constructing the global models in the seven nodes. It is seen that the size of the test datasets are comparatively small. Node 1 has not the warezmaster attack samples; Node 2 has not the httptunnel samples; Node 3 has not the satan attack samples; and Node 4 has not the neptune samples. Table 8 show the comparison results by using our algorithm and other traditional methods though DR and FAR.

Overall, it is seen that our DTCNN-based and SCCNN-based algorithm greatly increases the detection rate and decreases the false alarm rate for each model after the local node intrusion detection. Compared with centralized architecture DIDS algorithms, the distributed architecture DIDS algorithms not only gain high DR while keeping low FAR, but also adapt to the local models in online mode. This adaptability is very important for heterogeneous network environment. The detection accuracies of the SCCNN-based algorithms are comparable with the ANN-based algorithm in [12], Signature-based multilayer IDS in [13], and the Dynamic Election algorithm-based in [14]. In particular, lower FAR is obtained. The reason for this is that the design processing of SCCNN parameters is based on solving the LMI which come from the equations describing system dynamics that obtains more accurate weights for the global detection model. On the other hand, the computational complexity of our algorithm is relatively low, because the proposed detection approach only

Table 8: The comparison results about DIDS with different methods.

	Algorithms	DR (%)	FAR (%)
Centralized architecture DIDS	Genetic algorithm, artificial immune, and ANN [7]	84.52–97.5	2.16–3.64
	Artificial Immune System [8]	89.5–96.9	1.2–2.7
	Game-theoretical approach [9]	92.3–97.67	0.77–2.9
	Our DTCNN-based local detection method	87.83–98.16	0.71–2.94
Distributed architecture DIDS	ANN [12]	Average 96.24	None
	Signature-based multilayer IDS [13]	Average 96.7	Average 1.83
	Dynamic Election Algorithm [14]	96–97.5	0.93–1.97
	Our SCCNN-based global detection method	96.19–97.76	0.795–1.845

executes iterations to perform the local and global detection phase which reduces the time consumption and then achieves a better performance.

6. Conclusion

In this paper, we proposed a new DIDS algorithm, in which DTCNNs were used as weak classifiers in local nodes and SCCNN was used as global detection method, respectively. We further proposed a method for design SCCNN parameters based on LMI. And the global detection procedure can be emerged as locally connected, nonlinear processor arrays, while the circular equations describing system dynamics reach their steady state at an equilibrium point which represents a desirable feature in view of VLSI hardware implementations of real time networks. Experiments are carried out to demonstrate the performance of the proposed model, and the comparative results with other traditional centralized or distributed architecture DIDS methods show that the proposed model exhibit superior performance.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (nos. 61121061 and 61161140320) and the National Key Technology R&D Program (2012BAH37B05).

References

[1] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Network anomaly detection: methods, systems and tools," *IEEE Communications Surveys and Tutorials*, vol. 16, no. 1, pp. 303–336, 2014.

[2] I. Corona, G. Giacinto, and F. Roli, "Adversarial attacks against intrusion detection systems: taxonomy, solutions and open issues," *Information Sciences*, vol. 239, pp. 201–225, 2013.

[3] Y.-B. Chen, C. Feng, Q. Zhang, and C.-J. Tang, "Integrated artificial immune system for intrusion detection," *Journal on Communications*, vol. 33, no. 2, pp. 125–131, 2012.

[4] L. Li, Y. H. Li, D. Y. Fu, and M. Wan, "Intrusion detection model based on hierarchical structure in wireless sensor networks," in *Proceedings of the International Conference on Electrical and Control Engineering (ICECE '10)*, pp. 2816–2819, June 2010.

[5] E. J. Cho, C. S. Hong, S. Lee, and S. Jeon, "A partially distributed intrusion detection system for wireless sensor networks," *Sensors*, vol. 13, no. 12, pp. 15863–15879, 2013.

[6] D. Zhang and C. K. Yeo, "Distributed Court System for intrusion detection in mobile ad hoc networks," *Computers and Security*, vol. 30, no. 8, pp. 555–570, 2011.

[7] N. A. Alrajeh and J. Lloret, "Intrusion detection systems based on artificial intelligence techniques in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2013, Article ID 351047, 6 pages, 2013.

[8] F. Hosseinpour, K. A. Bakar, A. H. Hardoroudi, and A. F. Dareshur, "Design of a new distributed model for intrusion detection system based on artificial immune system," in *Proceedings of the 6th International Conference on Advanced Information Management and Service (IMS '10)*, pp. 378–383, Seoul, Republic of Korea, December 2010.

[9] K. Bartos, M. Rehak, and M. Svoboda, "Self-organized collaboration of distributed IDS sensors," in *Detection of Intrusions and Malware, and Vulnerability Assess*, vol. 7591 of *Lecture Notes in Computer Science*, pp. 214–231, Springer, Berlin, Germany, 2013.

[10] K. Xie, Y. Yang, L. Zhang, W. Li, and Y. Xin, "Research of hierarchical intrusion detection model based on discrete cellular neural networks," *Journal of Information and Computational Science*, vol. 10, no. 17, pp. 5569–5578, 2013.

[11] Z. Hou, Z. Yu, W. Zheng, and X. Zuo, "Research on distributed intrusion detection system based on mobile agent," *Journal of Computers*, vol. 7, no. 8, pp. 1919–1926, 2012.

[12] N. El Kadhi, K. Hadjar, and N. El Zant, "A mobile agents and artificial neural networks for intrusion detection," *Journal of Software*, vol. 7, no. 1, pp. 156–160, 2012.

[13] M. Uddin, A. A. Rehman, N. Uddin, J. Memon, R. Alsaqour, and S. Kazi, "Signature-based multi-layer distributed intrusion detection system using mobile agents," *International Journal of Network Security*, vol. 15, no. 2, pp. 97–105, 2013.

[14] Y. B. Liu, L. Shi, B. Z. Wang, Y. Q. Wu, and P. H. Wang, "An new agent based distributed adaptive intrusion detection system," *Advanced Materials Research*, vol. 532-533, pp. 624–629, 2012.

[15] W. Hu, J. Gao, Y. Wang, O. Wu, and S. Maybank, "Online adaboost-based parameterized methods for dynamic distributed network intrusion detection," *IEEE Transactions on Cybernetics*, vol. 44, no. 1, pp. 66–82, 2014.

[16] Y. Meng and L.-F. Kwok, "Adaptive blacklist-based packet filter with a statistic-based approach in network intrusion detection," *Journal of Network and Computer Applications*, vol. 39, no. 1, pp. 83–92, 2014.

[17] G. Mao, X. Wu, and X. Jiang, "Intrusion detection models based on data mining," *International Journal of Computational Intelligence Systems*, vol. 5, no. 1, pp. 30–38, 2012.

[18] T. Pani and F. de Toro, "An additive decision rules classifier for network intrusion detection," in *Advances in Computational Intelligence*, vol. 6691 of *Lecture Notes in Computer Science*, pp. 105–112, Springer, Berlin, Germany, 2011.

[19] X. Lei and P. Zhou, "An intrusion detection model based on GS-SVM classifier," *Information Technology Journal*, vol. 11, no. 7, pp. 794–798, 2012.

[20] R. Mitra, S. Mazumder, T. Sharma, N. Sengupta, and J. Sil, "Dynamic network traffic data classification for intrusion detection using genetic algorithm," in *Swarm, Evolutionary, and Memetic Computing*, vol. 7677 of *Lecture Notes in Computer Science*, pp. 509–518, Springer, Berlin, Germany, 2012.

[21] K. Sravani and P. Srinivasu, "Comparative study of machine learning algorithm for intrusion detection system," in *Proceedings of the International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA) 2013*, vol. 247 of *Advances in Intelligent Systems and Computing*, pp. 189–196, Springer, Cham, Switzerland, 2014.

[22] A. Yavuz Oruc, "Designing cellular permutation networks through coset decompositions of symmetric groups," *Journal of Parallel and Distributed Computing*, vol. 4, no. 4, pp. 404–422, 1987.

[23] K. Halonen, V. Porra, T. Roska, and L. O. Chua, "Programmable analogue VLSI CNN chip with local digital logic," *International Journal of Circuit Theory and Applications*, vol. 20, no. 5, pp. 573–582, 1992.

[24] R. Caponetto, L. Fortuna, L. Occhipinti, and M. G. Xibilia, "Hyperchaotic dynamic generation via SC-CNNs for secure transmission applications," in *Proceedings of the IEEE International Joint Conference on Neural Networks*, vol. 1, pp. 492–496, May 1998.

[25] P. Arena, S. Baglio, L. Fortuna, and S. Graziani, "Analog cellular networks for multisensor fusion and control," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 47, no. 9, pp. 1378–1382, 2000.

[26] L. O. Chua and L. Yang, "Cellular neural networks: applications," *IEEE Transactions on Circuits and Systems*, vol. 35, no. 10, pp. 1273–1290, 1988.

[27] S. Parmaksizoglu, E. Gunay, and M. Alci, "Determining clonning templates of CNN via moga: edge detection," in *Proceedings of the IEEE 19th Conference on Signal Processing and Communications Applications (SIU '11)*, pp. 758–761, 2011.

[28] L. Li, G. Ma, and X. Du, "New method of horizon recognition in seismic data," *IEEE Geoscience and Remote Sensing Letters*, vol. 9, no. 6, pp. 1066–1068, 2012.

[29] S. Baglio, S. Graziani, G. Manganaro, and N. Pitrone, "Cellular neural networks: a new paradigm for multisensor data fusion," in *Proceedings of the 8th Mediterranean Electrotechnical Conference (MELECON '06)*, pp. 509–512, May 1996.

[30] B. Ando, S. Baglio, S. Graziani, and N. Pitrone, "A novel analog modular sensor fusion architecture for developing 'smart' structures," in *Proceedings of the IEEE Instrumentation and Measurement Technology Conference (IMTC '97)*, vol. 2, pp. 1221–1224, Ottawa, Canada, May 1997.

[31] J. Benton Jr. and D. Smith, "Static output feedback stabilization with prescribed degree of stability," *IEEE Transactions on Automatic Control*, vol. 43, no. 10, pp. 1493–1496, 1998.

[32] J. Xiang, X. Y. Zhang, H. Y. Su, and J. Chu, "Static output feedback stabilization based on structured Lyapunov Matrix," *Control and Decision*, vol. 19, no. 9, pp. 978–982, 2004.

[33] J. J. Martínez, J. Garrigós, J. Toledo, and J. Manuel Ferrández, "An efficient and expandable hardware implementation of multilayer cellular neural networks," *Neurocomputing*, vol. 114, pp. 54–62, 2013.