Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 243841, 14 pages
doi:10.1155/2012/243841

Research Article

Toward Intelligent Intrusion Prediction for Wireless Sensor Networks Using Three-Layer Brain-Like Learning

Jun Wu,[1] Song Liu,[1] Zhenyu Zhou,[1] and Ming Zhan[2]

[1] *Global Information and Telecommunication Institute, Waseda University, Tokyo 169-0051, Japan*
[2] *National Key Laboratory of Science and Technology on Communications, University of Electronic Science and Technology of China, Chengdu 611731, China*

Correspondence should be addressed to Jun Wu, junwu@akane.waseda.jp

Received 16 June 2012; Accepted 23 August 2012

Academic Editor: Mihui Kim

The intrusion prediction for wireless sensor networks (WSNs) is an unresolved problem. Hence, the current intrusion detection schemes cannot provide enough security for WSNs, which poses a number of security challenges in WSNs. In many mission-critical applications, such as battle field, even though the intrusion detection systems (IDSs) without prediction capability could detect the malicious activities afterwards, the damages to the WSNs have been generated and could hardly be restored. In addition, sensor nodes usually are resource constrained, which limits the direct adoption of expensive intrusion prediction algorithm. To address the above challenges, we propose an intelligent intrusion prediction scheme that is able to enforce accurate intrusion prediction. The proposed scheme exploits a novel three-layer brain-like hierarchical learning framework, tailors, and adapts it for WSNs with both performance and security requirements. The implementation system of the proposed scheme is designed based on agent technology. Moreover, an attack experiment is done for getting training and test data set. Experiment results show that the proposed scheme has several advantages in terms of efficiency of implementation and high prediction rate. To our best knowledge, this paper is the first to realize intrusion prediction for WSNs.

1. Introduction

Wireless sensor networks (WSNs) have become a technology for the new millennium with endless applications ranging from civilian to military [1–3]. A wireless sensor network is consisted of a large number of wireless-capable sensor devices working collaboratively to achieve a common objective. As a matter of fact, WSNs are often deployed in potentially adverse or even hostile environments where adversaries can launch various kinds of attacks [3–5]. The nodes of WSNs are vulnerable to these attackers, because unmanned sensors are often deployed through open medium and dynamic network topology. Intrusion detection is a security technology that attempts to identify those who are trying to break into and misuse a system without authorization and those who have legitimate access to the system but are abusing their privileges. Recently, the problem of intrusion detection in WSNs has received considerable attention [4–15].

In the existing intrusion detection schemes of WSNs [5–15], two approaches have been used: signature-based detection and anomaly detection. Signature-based detection [7–12] lies in the monitoring of system activity and the identification of behaviors which are similar to pattern signatures of known attacks or intrusions stored in a signature database. This category of intrusion detection systems (IDSs) detects accurately known attacks, and the signatures are often generalized in order to detect the many variations of a given known attack. But this generalization leads to the increase of false positives (i.e., false alarms). The main limitation of such IDSs concerns their incapability to detect unknown intrusions that are not already present in the signature database.

On the other hand, anomaly detection systems [6, 13–15] detect attacks by observing deviations from a preestablished normal system or user behavior. This approach makes detecting new or unknown attacks, if these attacks imply an abnormal use of the system. The main difficulty in implementing

reliable anomaly detection systems is the creation of the normal behavior model. Since it is difficult to define correctly these models and only incomplete or incorrect models can be obtained, which leads to false negatives or false positives.

However, there is an important limitation of the existing intrusion detection schemes in WSNs, which is shown as follows. The existing intrusion detection schemes of WSNs have no concept of intrusion prediction. In many mission-critical applications, such as battle field, some attack processes are executed in a very short time [5] when the threat environment for WSNs includes a well-resourced adversary. Even though IDSs can detect these malicious activities afterwards, damages could have been done to the compromised WSNs which could hardly be restored in some cases. Therefore, it is very important to develop algorithms and tools to track and predict attacks in advance to remove potential threats. The intrusion prediction mechanisms for existing applications, such as computer networks, grid computing systems, and automated substation, are developed to predict various attacks, but cannot be applied directly to WSNs [16–21].

Recently, in the intrusion detection community, interest has been growing applying machine learning techniques to get high performances in execution time and overall classification accuracy [22–24]. Machine learning is a technology which is concerned with the design of algorithms that allow systems to evolve behaviors based on empirical data. A learner can take advantage of examples (data) to capture characteristics of interest of their unknown underlying probability distribution. Data can be seen as examples that illustrate relations between observed variables. A major focus of machine learning research is to automatically learn how to recognize complex patterns and to make intelligent decisions based on data. Hence, the learner must generalize from given examples, so as to be able to produce a useful output in new cases. Machine learning based intrusion detection for WSNs [25] has gained limited attention so far, not to mention intrusion prediction or implementation on the current generation of sensor nodes.

From the above discussion, it is clear that achieving prediction with high accuracy using machine learning is still an open challenge in WSNs. Towards addressing this challenge, we proposed in this paper a machine learning based intelligent intrusion prediction scheme. By exploring a three-layer brain-like hierarchical learning model, we proposed a novel intelligent intrusion prediction scheme, namely, BLID, which is specially tailored for WSNs. We based our design on the observation of the inherent nature of WSNs that different nodes own different recourses. Hence, we design this intelligent intrusion prediction as a hierarchical model. In the proposed scheme, supervised learning with relatively low complexity is performed in the resource-restrained sensors. Inversely, unsupervised learning and reinforcement learning are implemented in the sinks and base stations which have powerful resources. The learning modules in different layers can interact with each other. Our solutions have several advantages. First, BLID is efficient in terms of storage, computation, and communication overhead on the sensor side. Most important, BLID can perform intrusion prediction. To the best of our knowledge, the design of intrusion prediction in WSNs has not been addressed in previous work.

In summary, our paper makes the following contributions. (1) It introduces the intrusion prediction problem for the first time in WSNs. (2) The proposed scheme applies and tailors brain-like hierarchical learning to WSNs for achieving intrusion prediction with high accuracy. (3) The implementation of BLID is simulated on the current generation of sensor nodes.

The rest of this paper is organized as follows. Section 2 describes the system model and assumptions as well as some technical preliminaries on which our scheme is based. Section 3 presents the proposed scheme in detail. Section 4 describes the wireless attack experiment through which we get the training and testing data set for evaluating our scheme. In Section 5, we evaluate our scheme in terms of efficiency and accuracy. Finally, we conclude this paper in Section 6.

2. Models and Assumptions

2.1. Network Model. In this work, we consider a WSN with three-layer structure which includes base station layer, sink layer, and sensor layer. This structure is a popular way for deploying WSNs [5, 26–28]. Usually, the sensor nodes are scattered in a sensor field, and each sensor can collect data and deliver the data to the sink or base station (BS). Sensor nodes are static most of the time, whereas mobile nodes can be deployed according to application requirements. One or several base stations (BSs) can be deployed together with the network. BSs and sinks can be either static or mobile. BS acts as an interface between the WSN and the external world.

Data storage and access in WSNs mainly follows two approaches which are centralized and distributed approaches [29]. In centralized scheme, sensed data are collected from individual sensors and transmitted back to the sink, for storage and access. In the distributed case, the sensors store the data locally or at some designated nodes within the network. The stored data can be further accessed in distributed manner by the users of the WSN. BS, sink, and sensor are the access points (AP) when users access the data in the WSN. An access scenario is illustrated in Figure 1. Local users can access WSNs through wireless links directly. However, remote users need to access the WSN through satellite, Internet, or mobile network.

2.2. Intrusion Model. This paper considers that adversaries could be either external intruders or unauthorized network users. Due to lack of physical protection, sensor nodes are usually vulnerable to strong attacks. In particular, we consider the adversary with both passive and active capabilities, which can (1) eavesdrop all the communication traffics in the WSN, and (2) compromise and control a small number of sensor nodes. In addition, (3) unauthorized users may collude to compromise the encrypted data.

2.2.1. Wire Intrusions. The base station can act as an interface between the WSNs and other communication network,

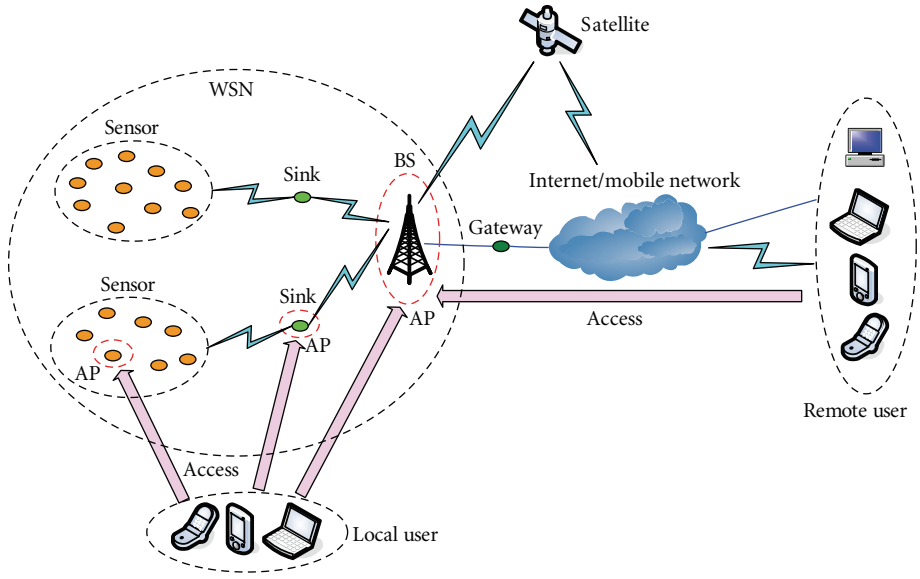

Figure 1: An access scenario of a WSN.

which is most likely to be Internet. In other words, remote users usually access the data in the base station through Internet. Hence, the intrusion attacks of Internet can present to the base station. There have been many studies focused on intrusion detection and prediction for Internet [16, 18, 20–22, 24, 25]. The Internet intrusions present to base station can be resolved on these schemes.

2.2.2. Wireless Intrusions. Comparing with wire networks, wireless networks face more intrusions because the wireless communication medium is open physically to adversaries. Various adversaries can attack wireless networks through wireless links [30]. More seriously, a lot of free tools are available on the Internet that allows novice hackers to exploit wireless protocol weaknesses to deny access to a network. All these facts raise the challenge of intelligent intrusion prediction for WSNs.

The wireless communication infrastructure of WSNs is the choice of application. Because many existing WSNs are deployed by IEEE 802.11 and mote device technologies [31], we consider in this paper IEEE 802.11 as the wireless communication infrastructure of WSNs. In WSNs, all the sensors, the sink, and the base station can act as the wireless access points; hence, all the three kinds of nodes could be intruded by the wireless attacks. There is a free collection of tools to attack 802.11-based networks available for download on Internet [32]. These tools operate on WEP and WPA-protected networks.

In this paper, we take four kinds of attacks, for example, doing experiments, which are ARP replay attack, forgery attack, ongoing dictionary attack, and chopchop attack. These attacks are the common attacks in 802.11 networks [33, 34].

2.3. Preliminaries. This section briefly describes the technique preliminaries on which our scheme is designed.

2.3.1. Brain-Like Hierarchical Learning. Recently, brain-like learning and computation have attracted a lot of attentions in the area of machine learning. Our brain is a highly complicated structure and there have been many studies focused on brain-like learning [35–38]. In this paper, we consider the brain-like learning model in [37], which is developed into a system structure in [38]. This brain-like model is based on the fact that the cerebellum is a specialized organism for supervised learning (SL), the basal ganglia are for reinforcement learning (RL), and the cerebral cortex is for unsupervised learning (UL). In the framework, a particular function, such as the control of arm movement, can be realized by a global network combing different learning modules in the cerebellum, the basal ganglia, and the cerebral cortex. The three learning modules of brain-like learning are described as follows.

Supervised Learning (SL) in the Cerebellum. This learning module is which constructs an input-output mapping. It is characterized by the parameter update based on the correlation between the output error and the presynaptic input.

Unsupervised Learning (UL) in the Cerebral Cortex. This learning module is characterized by the relaxation dynamics for determining the output as well as the Hebbian synaptic rule under a certain regularization.

Reinforcement Learning (RL) in the Basal Ganglia. This learning module is concerned with how an agent ought to take actions in an environment so as to maximize some notion of cumulative reward.

2.3.2. Agent Technology. The agent technology is an important technique in recent researches of the artificial intelligence [39]. In the area of WSNs, a lot of new works

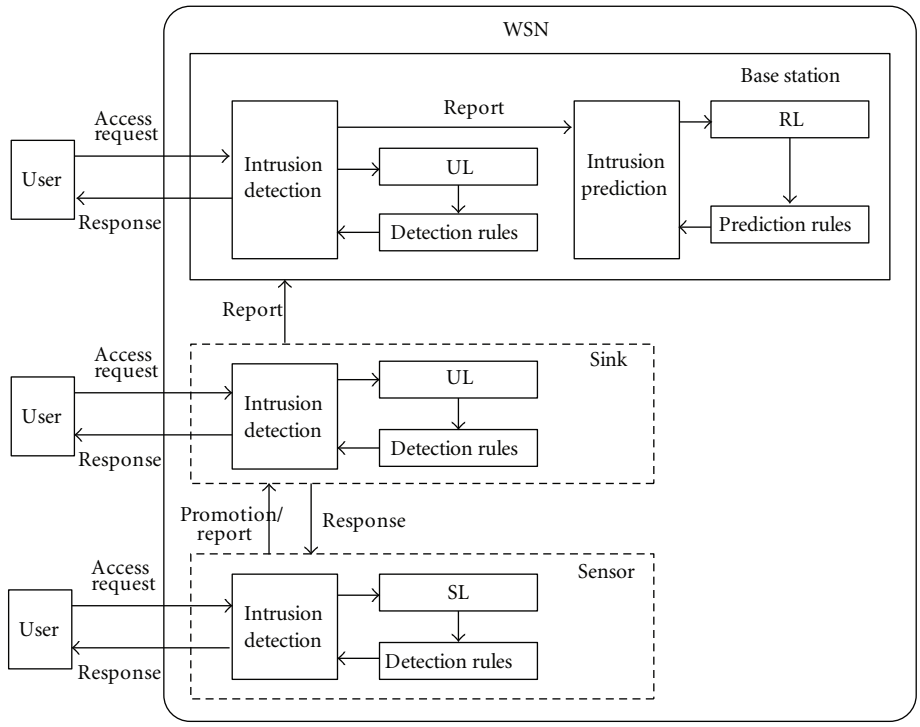

Figure 2: Systemic design of BLID.

introduce agent technology into WSNs [40–42]. Using agent technology in WSNs leads to a number of advantages, such as scalability, event-driven actions, task-orientation, and adaptivity.

3. Brain-Like Hierarchical Learning Intelligent Intrusion Prediction Scheme

This section presents the proposed intelligent intrusion prediction scheme for WSNs. We first introduce the systemic design. Then, we present the detailed description of our basic scheme, which is followed by an advanced design.

3.1. Systemic Design. In this section, we set up the system model of BLID, as shown in Figure 2. The basic idea of the intelligent intrusion prediction scheme is *distributed detection and centralized prediction*. Every node of the WSN can perform intrusion detection. However, only base station can perform intrusion prediction for the whole WSN.

Considering the limited resources of sensors and the powerful resources of sink as well as base station, we define two levels of intrusion detection: (1) supervised learning based detection and (2) unsupervised learning based detection. The supervised learning based detection is a low level detection which is performed in sensors. This part is corresponding with the cerebellum of the brain. On the other hand, the unsupervised learning based detection is a high level detection which is performed both in sinks and base station. This part is corresponding with the cerebral of the brain. If some unknown attacks occur to a sensor, the sensors will send the unknown features to the sink. This operation is marked as "*promotion.*" Then the sink will determine whether the access is an attack or not by its high level rules. If the sink cannot identify based on current rules, it will adaptively update the rules based on unsupervised learning. Then the sink sends the response to the sensor. In short, that sink and the base station perform intrusion detection by theirselves. The sensor performs low level detection by itself, but it needs the help of sink for performing high level detection.

In the WSNs, only the base station can perform intrusion prediction which is based on reinforcement learning. This part is corresponding with the basal ganglia of the brain. In case of an intrusion, the sensors and sinks send the related features of the attack to the base station through a "*report*" operation. Similarly, the detection modular in the base station reports every local intrusion to prediction module. Note that sensor reports every intrusions to base station via sink because sensors cannot communicate directly with the base station.

Through the basic idea above, a particular detection or prediction function can be realized by a global network combing different learning modules in sensor, sink, and base station.

3.2. Supervised Learning Based Intrusion Detection in Sensor. Decision tree is a kind of classifier for supervised learning. In order to perform supervised learning with low complexity, we use decision tree (DT) as a classifier for data analyzing. Usually, there are three criteria for constructing a decision

tree: the information gain, the gain ratio, and the Gini index [43].

There are three steps to design decision tree based intrusion detection. The first step is defining and initializing variables that will be used in the ensuing process. The second step is defining a set of primary detection rules. A detection rule contains a set of keywords that must be checked to trigger an alarm. Finally, the third step is defining a set of primary action rules that describe the behaviour after analyzing the attribute data. The core part is how to construct a decision tree.

The decision tree construction scheme for sensor must have low complexity because the resources of sensors are limited.

The decision tree in our scheme contains three types of nodes: ordinary, leaf, and promotion nodes. Each node is represented by $N\ (A, D, M)$ where A is an attribute set, D is a set of detection rules, and C is a set of countermeasure. The attribute set A denotes the set of attributes already used to decompose the tree and D is the set of detection rules that are matched at that node. The initial root node contains the whole set of detection rules, an empty set of attributes, and an empty set of matched rules. Then, we iteratively decompose each node according to the set of possible attributes using the appropriate inference rules. Leaves are nodes that cannot be transformed anymore. They can be used to report attacks thanks to the detection rules contained in their last field. A promotion node is a node at which can be further processed by the sink as a root node of subtree.

Before we present our construction scheme, we define some notations and auxiliary functions employed in the decision tree construction scheme.

Definition 1. Let $T = \{t_1, t_2, \ldots, t_k\}$ be a set of criterion variable and d be a rule which is $\{(v_1 = t_1) \wedge (v_2 = t_2) \wedge \cdots \wedge (v_k = t_k)\}$. k is the dimension of T. We define the function $\text{Drawn}(d) = \{v_1, v_2, \ldots, v_k\}$. The function can be extended to a set of rules L by

$$\text{Drawn}(D) = \bigcup_{d \in D} \text{Drawn}(d). \tag{1}$$

Definition 2. We define the function $\text{Obtain}(N(F, D, M)) = \{\text{Subtree} \mid N_1(F, D_1, M_1) \cup N_2(F, D_2, M_2) \cdots \cup N_m(F, D_m, M_m)\}$. $N_1, N_2, \ldots, N_m$ are the member nodes of the subtree. This function sends $N(F, D, M)$ to a sink. Then $N(F, D, M)$ can be further processed by the sink and a subtree will be returned to the sensor. The root node of the subtree is $N(F, D, M)$, so that the subtree can be integrated with the current tree.

We use function Drawn to extract the parameters of the local rules, which are low level rules. Also, we use the function Obtain to get a subtree from the sink. In other words, if the sensor cannot deal with some situations, the sink can help to decompose the current node N into a subtree base on high level rules. We assume that the root node of the tree has been selected. For each nonempty branch of the current node, we use the following scheme to construct a decision tree.

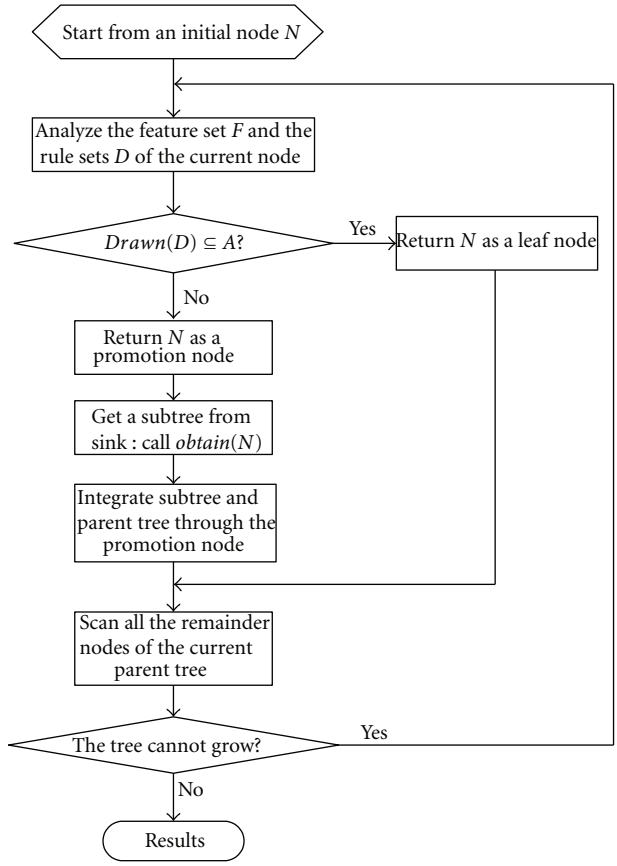

Figure 3: Decision tree learning in sensor.

The scheme of tree construction is shown in Figure 3. The process begins from an initial node N. The current node will become a leaf node if all the attributes have been considered. Otherwise, function Obtain will be used. When $\text{Obtain}(N)$ function is performed, the connection point of the subtree and the parent tree is the current node N. Note that the parent tree is the decision tree in the sensor, and the subtree is generated in the sink. The rule set in the sensor is a subset of the rule set in the sink. All leaf nodes cannot be processed further. The construction process is stopped when all reduced nodes are leaf nodes.

3.3. Unsupervised Learning Based Intrusion Detection in Sink and Base Station. Traditionally, a decision tree is viewed as a supervised learning method, because the splitting is guided by an impurity measure, which depends on the class labels of the data. On the other hand, clustering is an important exploratory data analysis task. It aims to organize objects (data records) into similarity groups or clusters. Clustering is often called unsupervised learning as no classes denoting an a priori partition of the objects are known. This is in contrast with supervised learning (e.g., classification), for which the data records are already labeled with known classes.

As mentioned before, we have designed the supervised learning in sensor based on decision tree. In order to correspond with the learning scheme in sensors, we base our

unsupervised learning on decision tree. As a matter of fact, there have been several studies focused on decision tree based clustering. The scheme in [44] needs to introduce additional data points into the existing points. The operation is complex for WSNs. In [45], an unsupervised decision tree is proposed for information retrieval, which cannot be applied in WSNs directly. In this section, we present the decision tree based clustering for WSNs.

Clustering is required to divide initial sets of objects on many groups (clusters) so that objects inside each group would be the much alike in some sense, while the objects of different groups will be as more as possible "different." It is required to find out such clusters of objects in space of characteristics, which will in the best way satisfy to a criterion of a grouping quality. It is supposed that the characteristics, describing objects, may be both quantitative and qualitative. Various methods of the cluster analysis differ in the ways of understanding of similarity, criterion of quality, and ways of finding groups.

At first, we define a criterion of quality of the grouping. Let characteristic of a request from a user or an attacker be a data sample. All the samples consist of the sample space. The decision tree with L leaves splits space of characteristics into L nonoverlapping subareas $S^1, S^2, \ldots, S^L$. This splitting space corresponds to the splitting of the set of observation Samples into L subsets $\text{Sample}^1, \text{Sample}^2, \ldots, \text{Sample}^L$. Thus, the number of leaves in a tree coincides with the number of groups of objects. We will consider a group of objects Sample^i.

The description of this subset will be the following conjunction of statements:

$$U\left(\text{Sample}^i, V^i\right) = \left(X_1 \in V_1^i\right) \wedge \left(X_2 \in V_2^i\right) \wedge \cdots \wedge \left(X_n \in V_n^i\right), \tag{2}$$

where V_j^i is interval which is calculated as follows:

$$V_j^i = \left[\min_{\text{Sample}^i}\left\{x_j\right\}, \max_{\text{Sample}^i}\left\{x_j\right\}\right], \quad \text{or } V_j^i = \left\{x_j \mid x_j \in \text{Sample}^i\right\}, \tag{3}$$

where the previous equation is for quantitative characteristic, and the second one is for qualitative characteristic.

A characteristic subspace R^i, corresponding to the group's description, we call a taxon (plural taxa). It is important to note, although in a decision tree the part of characteristics can be absent, in the description of each group all available characteristics must participate.

Relative capacity (volume) of taxon can be calculated by

$$\lambda^i = \prod_{j=1}^{n} \frac{\left|V_j^i\right|}{\left|D_j\right|}, \tag{4}$$

where $|V_j^i|$ designates the length of an interval (in case of the quantitative characteristic) or capacity (number of values) of appropriate subset V_j^i (in case of the qualitative characteristic); $|D_j|$ is the length of an interval between the minimal and maximal values of characteristic X_j for all objects from initial sample (for the quantitative characteristic) or the general number of values of this characteristic (for the qualitative characteristic).

When the number of clusters is known, the criterion of quality of a grouping is the amount of the relative volume of taxa:

$$g = \sum_{i=1}^{L} \lambda^i. \tag{5}$$

The grouping with minimal value of the criterion is called optimum grouping.

If the number of clusters is not given beforehand, it is possible to understand the next value as the criterion of quality

$$P = g + aL, \tag{6}$$

where $a > 0$ is a given parameter.

When minimizing this criterion, we receive on the one hand taxa of the minimal size and on the other hand aspire to reduce the number of taxa. Notice that in a case when all characteristics are quantitative, minimization of criterion means minimization of the total volume of multivariate parallelepipeds, which contain the groups.

For the construction of a decision tree, the method of consecutive branching described in Section 3.2 can be used. On each step of this method, a group of the objects corresponding to the leaf of the tree is divided into two new subgroups.

Division occurs with a glance on criterion of quality of a grouping, that is, the total volume of received taxa should be minimal. The node will be divided if the volume of the appropriate taxon is more than a given value. The division proceeds until there is at least one node for splitting or the current number of groups is less than the given number.

Note that learning mechanism in sink not only constructs decision tree for itself but also decomposes the promotion node from sensor to construct a subtree for sensor.

3.4. Reinforcement Learning Based Intrusion Prediction in Base Station. The process of monitoring user behavior and making predictions on network is a nonlinear problem [46]. Especially, comparing with traditional communication networks, WSNs have more dynamic and nonlinear facts. Hence, the linear method cannot work well for intrusion prediction in WSNs. Nonlinear prediction by reinforcement learning [47] and related algorithm can be used to solve intrusion prediction.

Different from supervised learning and unsupervised learning, reinforcement learning is a kind of goal-directed learning which is of great use for a learner (agent) adapting unknown environments [48]. When the environment belongs to Markov decision process (MDP), or partially observable Markov decision process (POMDP), a learner acts some trial-and-error searches according to certain policies and receives reward or punishment. The scheme in [47] uses Stochastic Gradient Ascent (SGA) algorithm [49] as

the reinforcement learning algorithm. However, the main shortcoming of the scheme is that off-policy sampling, as well as nonlinear function approximation, can cause the algorithms to become unstable (i.e., the parameters of the approximator may diverge). Moreover, the unstability will decrease the accuracy of prediction.

By using the convergent temporal-difference learning [50], we develop the nonlinear prediction into a stable scheme. Furthermore, we use the modified scheme as the intrusion prediction scheme base station.

As mentioned before, if an intrusion presents to the sensor, sink, or base station, the attribute parameter of the attack must be reported to the predictor in the base station. Here, the attribute parameter of the attack is denoted as a vector Attack $= [a_1, a_2, \ldots, a_n]$, which includes the concerned node ID, node address, attack type, attack time, and so forth. The selection of the attributes depends on the application scenarios. The architecture of the intrusion prediction system is illustrated in Figure 4.

The neural network in the prediction system is composed by 4 layers: input layer, hidden layer, stochastic layer, and output layer.

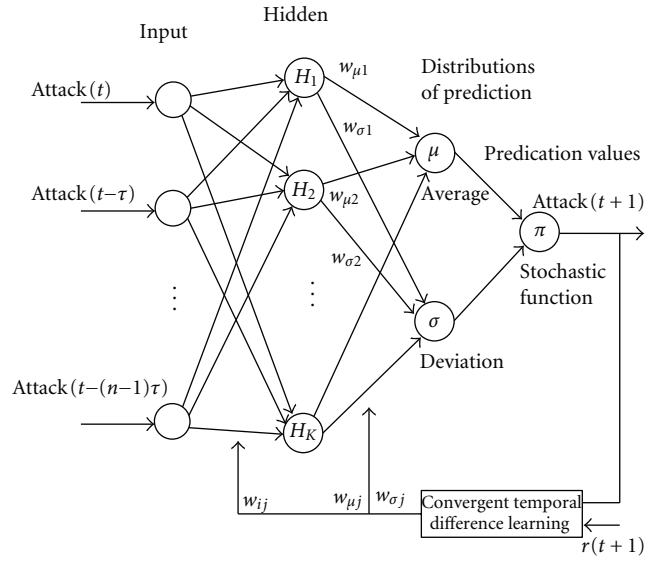

Figure 4: Architecture of intrusion prediction.

Input Layer. The inputs of prediction system on time t can be constructed as an n dimensions vector space $X(t)$, which includes n observed points with same intervals on time series Attack(t).

$$\begin{aligned} X(t) &= (x_1(t), x_2(t), \ldots, x_n(t)) \\ &= (\text{Attack}(t), \text{Attack}(t-\tau), \ldots, \text{Attack}(t-(n-1)\tau)), \end{aligned} \tag{7}$$

where τ is time delay (interval of sampling) and n is the embedding dimension.

Hidden Layer. Multiple nodes accept input with weights w_{ij}, and their output is given by

$$H_j(t) = \frac{1}{1 + e^{-\beta_H \sum a_i(t) w_{ij}}}, \tag{8}$$

where β_H is a constant.

Stochastic Layer. To each hidden node $H_j(t)$ in hidden layer, parameters of distribution function are connected in weight $w_{\mu j}$ and weight $w_{\sigma j}$ when we consider the output is according to Gaussian distribution. Nodes in stochastic layer give their output μ, σ as

$$\begin{aligned} \mu\left(H_j(t), w_{j\mu}\right) &= \frac{1}{1 + e^{-\beta_\mu \sum H_j(t) w_{j\mu}}} \\ \sigma\left(H_j(t), w_{j\sigma}\right) &= \frac{1}{1 + e^{-\beta_\sigma \sum H_j(t) w_{j\sigma}}}, \end{aligned} \tag{9}$$

where β_μ, β_σ is constant, respectively.

Output Layer. The node in output layer means a stochastic policy in reinforcement learning. Here we use a 1-dimension Gaussian function

$$\pi(\text{Attack}(t+1), W, X(t)) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(\text{Attack}(t+1)-\mu)^2/2\sigma^2}, \tag{10}$$

where Attack$(t+1)$ is the value of one-step ahead prediction, produced by regular random numbers. W means weights w_{ij}, $w_{\mu j}$, and $w_{\sigma j}$. This function causes learner's action, so it is called stochastic policy in reinforcement learning.

In order to update weights, we consider a prototypical case of temporal-difference learning, that of learning a linear approximation to the state-value function for a given policy and Markov decision process (MDP) from sample transitions. We take both the MDP and the policy to be stationary, so their combination determines the stochastic dynamics of a Markov chain. The state of the chain at each time t is a random variable, denoted as $s_t = \{1, 2, \ldots, N\}$. On each transition from s_t to s_{t+1}, there is also a reward r_{t+1}, whose distribution depends on both states. We seek to learn the parameter $\theta \in \mathbf{R}^n$ of an approximate value function $V_\theta : S \to \mathbf{R}^n$ such that

$$V_\theta(s) = \theta^T \phi_s \approx V(s) = E\left\{ \sum_{t=1}^{\infty} \gamma^t r_{t+1} \mid s_0 = s \right\}, \tag{11}$$

where $\theta_s \in \mathbf{R}^n$ is feature vector characterizing state s, and $\gamma \in [0, 1)$ is a constant called the discount rate.

Temporal difference error is defined as follows:

$$\delta_k = r_k + \gamma \phi_k^T \phi' - \theta_k^T \phi_k. \tag{12}$$

Following the method in [50], the weight W can calculated by

$$W = E\left[\phi\phi^T\right]^{-1} E[\delta\phi]. \tag{13}$$

Note that that δ depends on θ, hence w depends on θ.

Therefore, w can be updated as follows:

$$w_{k+1} = w_k + \psi_k\left(\delta_k - \phi_k^T w_k\right)\phi_k, \tag{14}$$

where ψ_k factors are step-size parameters, possibly decreasing over time.

3.5. Implementation System Based on Agent Technology

3.5.1. Design of Agent System. We use multiagent to realize the function of intrusion detection and prediction in WSNs. There are four kinds of agents designed in WSNs, which are detection agent (DA), communication agent (CA), database agent (BA), and prediction agent (PA). All the four agents are designed in base station. However, only DA, CA, and BA are designed in sensor and sink. Figure 5 shows the structure of the agent system of a node in wireless sensor network. In Figure 5, prediction agent (PA) is coloured grey, which means that PA does not exist in every node of the WSN but only base station node.

Detection Agent (DA)

(1) The *Detection Learning Module (DLM)* performs the learning algorithm described in Section 3. The module acts as a classifier to perform intrusion detection. It implements the proposed supervised decision tree learning algorithm for sensor. For sink and base station, this module runs the proposed decision tree based cluster algorithm. The rules for making decision are called from detection rule module. The results of learning can be sent to detection rule module for updating rules.

(2) The *Detection Rule Module (DRM)* contains the rule sets for intrusion detection. The rules are the choice of application design. The rules can be updated by the learning algorithm in the DLM.

Communication Agent (CA). This agent provides an interface for the node communicating with other nodes. Also, it preprocesses the raw data into the format required by the data classification techniques. On one hand, this module acts as an interface for the node interoperating with other nodes in WSNs. For sensor, this module sends the packet of *promotion* and reports as well as receives the response from sink. In sink, this module reports every attack, which occurs to the sensor and sink, to the base station. When this module is performed in a base station, it receives the report packets from the sinks. On the other hand, communication agent performs an interface to receive request and send responses for the user who accesses the node. It transfers the parameters of the request to DA and PA for further processing.

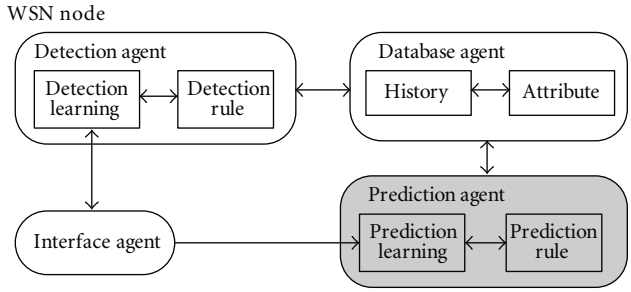

Figure 5: Node model of agent system.

Database Agent (BA)

(1) The *History Module (HM)* provides two distinct functionalities: a convenient mechanism to log events and actions that have occurred and an efficient mechanism to query these logged events. This module provides history data for detection learning and prediction learning.

(2) The *Attribute Module (AM)* provides an interface for the detection agent (DA) and prediction agent (PA) to query and update attributes of the data and users.

Prediction Agent (DA)

(1) *The Prediction Learning Module (PLM)* is designed only in base station. It performs the reinforcement learning algorithm described in Section 3. The module acts as a classifier to perform intrusion prediction. The rules for making decision are called from prediction rule module. The results of learning can be sent to prediction rule module for updating rules.

(2) *The Prediction Rule Module (PRM)* contains the rule sets for intrusion prediction. The rules can be updated adaptively by the learning algorithm in the PLM.

3.5.2. Sequence Model of Interaction. In BLID, there are several cases of interoperation among the learning modules in different nodes in WSNs, which is similar with the cerebellum, the basal ganglia, and the cerebral cortex. On one hand, when unknown request presents to the sensor, the sensor performs a *promotion* action for further detection by helping of sink. After the sink finished the further detection by using the clustering algorithm, it sends the responses to the sensor. The response includes the subtree (see Section 3.2) for updating the detection rules in the sensor. On the other hand, for both sink and sensor, they must report the features of the request to the base station if they find the request comes from an attacker. Note that sensors report the attack to base station via the sink, because sensor cannot communicate with the base station directly. Because report is an operation which is easy to understand, we just illustrate the sequence of *promotion* operation in Figure 6.

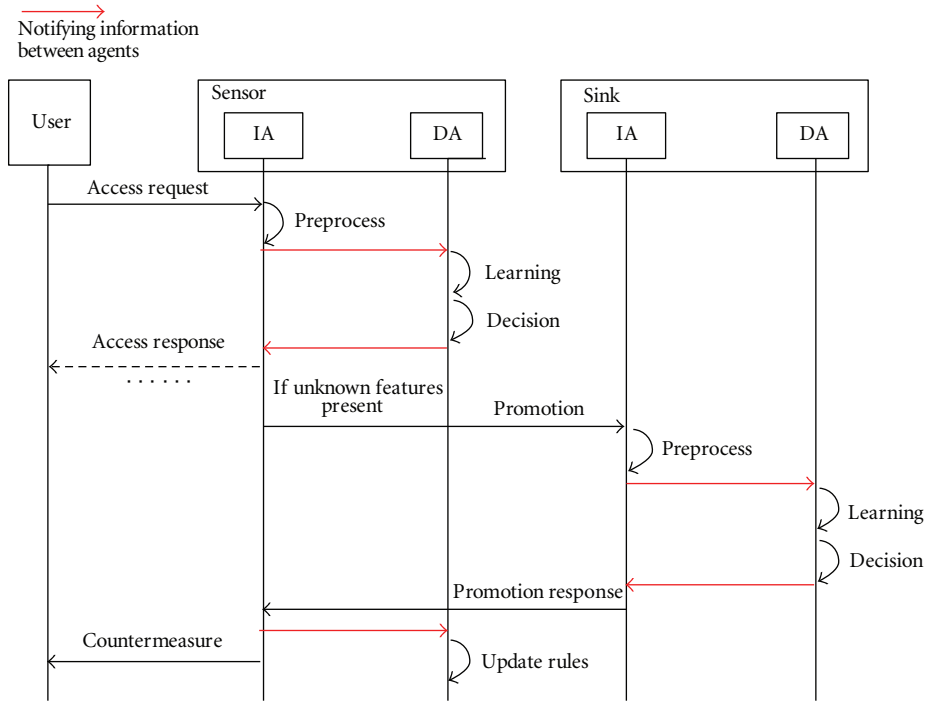

Figure 6: Sequence model of *promotion* operation.

4. Getting Data Set via Wireless Attack Experiment

In this section, we report the attack experimentation, through which we can get the data set for training and test. Because many existing WSNs are deployed by IEEE 802.11 and mote device technologies [31], we use IEEE 802.11 based wireless link for our experiment. Moreover, for access control, a role-based access control (RBAC) policy is used.

Feature selection is an important issue for intrusion prediction. In order to enhance the prediction accuracy for the attack from different layers, we consider both the application layer features and MAC layer features to construct the data set. We combine the features of access control and 802.11 wireless traffics [51] to construct the feature data set. On one hand, we select the important features of access control based on the feature selection method for access control in [52]. On the other hand, according to IEEE 802.11 standard [51], the fields of the MAC header can be extracted. We used the information gain ratio (IGR) as a measure to determine the relevance of each feature [53]. We can order the features according to the score assigned by the IGR measure. The IGR measure is based on the data set of frames collected from our testing network. The features of access control and 802.11 traffics which we used for experiment are shown in Tables 1 and 2, respectively. The number of the selection features depends on the requirements of security and the recourses of the system. As a case study for resource-constrained WSNs, we select 5 access control features and 7 traffic features of 802.11 for test.

We did the attack experiment in an 802.11 network. We take ARP replay attack, forgery attack, ongoing dictionary attack [32], and chopchop attack [33], which are the common attacks in 802.11 networks, as the examples for evaluation. The tool we use to generate attacks is Backtrack, which is available from the website [34].

In our experiment, the network was composed of three wireless stations. We use one machine as a server node (access point). Then, we use another machine to generate normal traffic firstly and later attacks. The last machine was used

Table 1: Features of access control.

Order	Features	Description
1	LoginResult	Access decision results before access
2	NumbWr	Number of write operation on access control files
3	NumbCrea	Number of create operation on rule file
4	NumbAccess	Number of access
5	NumbDe	Number of delete operation on access control files

Table 2: Features of traffic.

Order	Features	Description
1	WepResult	The result of WEP ICV check
2	Duration	The time the medium is expected to be busy
3	More_Frag	Whether a frame is nonfinal fragment or not
4	Desti_Addr	The MAC address of the receiving node
5	Fram_Type	The type of the frame
6	IfRetransmit	If the frame is a retransmitted frame
7	Sour_Addr	The MAC address of sending node

Table 3: Data set.

Traffic type	Training set	Test set
ARP replay attack	200	200
Forgery attack	200	200
Ongoing dictionary attack	200	200
Chopchop attack	0	200
Normal	1200	1200

to collect and record both normal and intrusion traffic. The number of related records in the data set is shown in Table 3. There is no training set for chopchop attack, because we use this attack as unknown attack for test. The other three kinds of attacks can be regarded as usual attacks.

5. Evaluation and Comparisons

This section evaluates BLID in terms of overhead and accuracy.

5.1. Overhead and Complexity Evaluation

5.1.1. Time Overhead and Memory Consumption in Sensor. Usually the resources of sensor are limited, but the resources of sink and base station are powerful. Hence, the evaluation of sensor is crucial and typical. We focus on the time overhead and memory consumption caused by our scheme on sensor. We have implemented BLID for TinyOS and tested it using TOSSIM [54]. The mote that TOSSIM simulates is MicaZ.

There are two phases of the learning, training phase and test set. Before the sensors being deployed, the training process can be performed on some other well-resourced devices, such as laptop, because the recourses of sensors are limited. Hence, the initial detection rules can be constructed on well-resourced devices and then loaded into sensors. In this paper, the initial detection rules training is based on the training data set in Section 4. Based on the above reasons, we just focus on the test phase. The overhead caused by BLID and related schemes during detection is reported in Figure 7, which is the time needed by a sensor from receiving a request to making a local detection decision.

In Figure 7, the vertical coordinates denote the overhead caused by intrusion detection system. Four groups of columns denote four cases which are corresponding with four kinds of attacks described in Section 4. As shown in Figure 7, the time overhead caused by BLID is lower than that of the schemes in [12, 25]. The results show that detecting unknown attacks usually needs more time than detecting known attacks.

Loading the rules intrusion detection requires memory. The memory consumption of our scheme is an important measure of its feasibility and usefulness on memory-constrained sensor nodes. The memory consumption is shown in Table 4. Because MicaZ has 128 KB of instruction memory and 512 KB of flash memory, the experiment results mean that BLID leaves enough space in the mote's memory for user applications. In addition, we use PC act as the sink and base station nodes. The memory consumption of high level detection and prediction is shown in Table 5.

Table 4: Memory consumption in sensor.

Agent	Size (bytes)
Detection agent	10274
Database agent	21857
Communication agent	3216
Total	35347

Table 5: Memory consumption in sink and base station.

Agent	Size (bytes)
Prediction agent in BS	535341
Database agent in BS	732219
Detection agent in sink	152796
Database agent in sink	225678
Communication agent	3216

5.1.2. Energy Consumption of Sensor. Energy cost is one of the most critical problems in resource-constrained sensors. In this subsection, we estimate the energy consumption of sensor using PowerTOSSIM [55], which is an energy modeling extension of TOSSIM. PowerTOSSIM is often used to evaluate the energy consumption of WSNs [56–58]. The energy consumption is measured for five components: CPU, RADIO, LED, SENSOR, and EEPROM. According to the attack experiment in Section 4, ARP reply attack, forgery attack, and ongoing dictionary attack act as the known attacks, while chopchop attack acts as unknown attack. In addition, based on the time overhead in Figure 7, the time overhead of ongoing attack has higher time complexity than ARP replay attack and forgery attack. Therefore, here we take ongoing attack as the example of known attack for energy consumption evaluation. Chopchop attack is still used as unknown attack for the energy consumption evaluation. Then, we fix the time of execution equal to 1200 simulated seconds, which is because the motes in PowerTOSSIM take boot time of 10 seconds. Here we consider three cases, which are continuous known attack, continuous unknown attack, and continuous normal access. In the simulation, the radio is set as sleep mode if there is no *promotion* or *report* message being transferred. The energy costs of different cases are shown in Figure 8.

In our proposed system, storing feature and rule data performed by EEPROM component and classification analysis performed by CPU component course the corresponding additive energy consumption in EEPROM and CPU, while radio transmission is not always necessary which depends on the *promotion* and *report* operations. As shown in Figure 8, the CPU energy costs for unknown and known attacks are higher than that of normal access and known attack. This is because that the attacks course more computations of the features analysis and must perform report operation to the base station. RADIO energy consumption of unknown

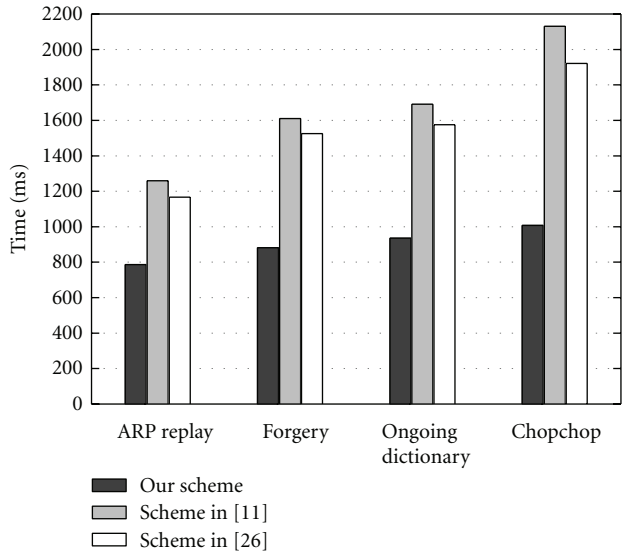

Figure 7: Time overhead in sensor.

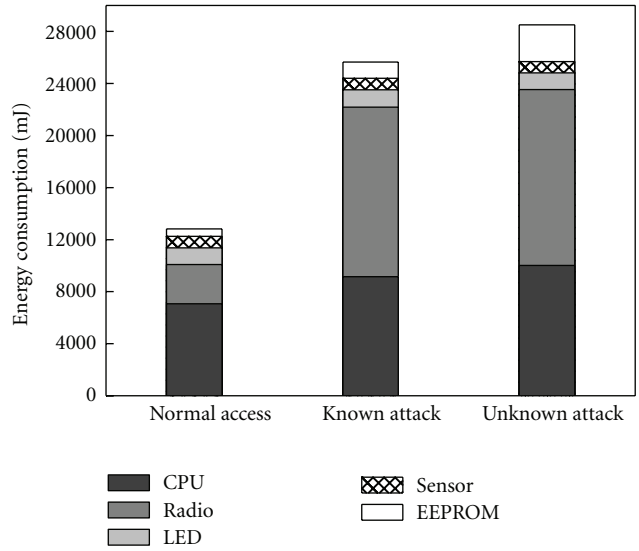

Figure 8: Energy consumption of sensor.

attack detection is higher than that of known attack, because of the additive *promotion* messages. Energy costs of the all the three cases are lower than that of related application schemes of WSNs, such as some ECG monitoring schemes in [57] and some data-stream protocol in [58]. Therefore, the energy consumption of our scheme is acceptable for resource-constrained WSNs.

5.1.3. Communication Overhead. BLID can cause communication overhead into WSNs. In a WSN, the number of sensor is usually much more than that of sink and base station, and some sensors usually are deployed far from base station and sink. In other words, the communication overhead is mainly caused by sensors. Hence, we evaluate the case that the attacks occur to sensors. Figure 9 depicts the communication cost of BLID measured in overhead packets in WSNs.

As shown in Figure 9, the communication overhead in case of unknown attack is higher than that in case of known attack, because the sink needs to return a subtree to the sensor in case of unknown attacks. The communication overhead also depends on the number of hop from the intruded sensor to the sink. For small scale WSNs, such as the number of hop is 3, the communication cost is only 4 for known attacks and 7 for known attacks, respectively. Moreover, for larger scale WSNs, such as the case of 7 hop, the overhead still remains low (8 packets for known attacks and 15 packets for unknown attacks). The communication cost of BLID is lower than that of cooperative intrusion detection scheme in [59]. For the scheme in [59], the communication overhead is 12 for 4 sensors cooperate for detection, and the overhead increase to 19 when 8 sensors cooperate.

5.2. Prediction Rate. The evaluation of the accuracy of prediction was obtained using Matlab and NeuroSolutions [60]. The detection accuracy of BLID depends on the learning algorithm in sink and base station, because "*promotion*"

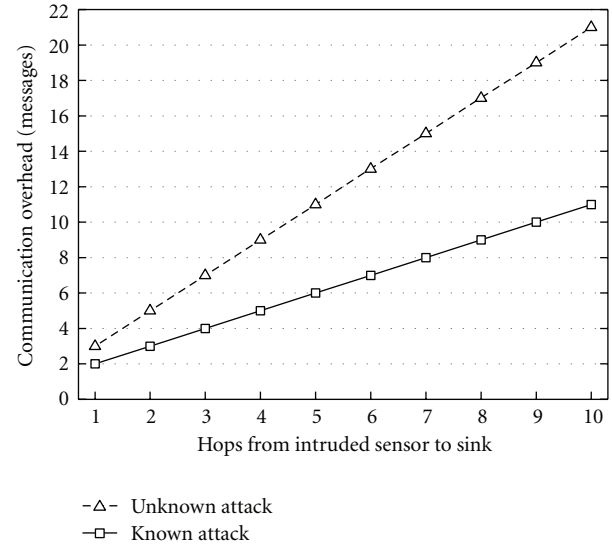

Figure 9: Communication overhead.

operation exists in the low level detection in sensor. The prediction accuracy depends on the reinforcement learning scheme performed in base station.

We use two metrics to evaluate the intrusion prediction performance, namely, prediction rate q and false alarm rate η. The prediction rate is formally defined by

$$q = \frac{d}{n}, \tag{15}$$

where d is the number of prediction attacks, and n is the total number of actual attacks.

We evaluate the prediction rate in this section. Because the real sample cannot be gotten in WSNs for intrusion prediction, DARPA Intrusion Detection Evaluation Data [61] is used as the training and test data set to verify the prediction rate of related schemes. The training data consist of five

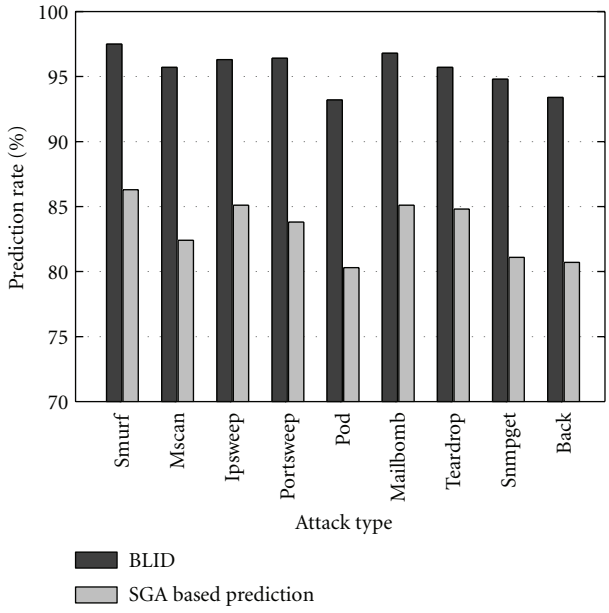

Figure 10: Prediction rate.

weeks of network-based attacks in the midst of normal background data. Attacks are labeled in training data. The test data consist of two weeks network-based attacks and normal background data. We also use the features defined in DARPA 1998 as the feature parameters of our prediction scheme. The prediction rate is shown in Figure 10. In Figure 10, the vertical coordinates denote the prediction rate. Nine groups of columns denote nine cases which are corresponding with nine kinds of attacks in DARPA Intrusion Detection Evaluation Data. As shown in Figure 10, the prediction rate of BLID is on average 12 percentages higher than that of SGA based scheme in [48].

6. Conclusion

In this paper, we analyzed the important issues of accurate intrusion detection and prediction in WSNs. To address the problems, we proposed a brain-like hierarchical learning based intelligent intrusion prediction scheme, called BLID, in which the sensor, sink, and base station perform different kinds of learning algorithms and interoperate optimally with each other. Referring to brain-like hierarchical learning model, we designed a relatively simple decision tree learning algorithm in the sensor for low level intrusion detection, which is corresponding with the supervised learning of cerebellum. Then, we proposed a decision tree based clustering mechanism in sink and base station for intrusion detection, which has a correspondence with unsupervised learning of cerebral cortex. Furthermore, we developed a stable reinforcement learning model in base station for high level intrusion prediction, which is referenced to reinforcement learning of the basal ganglia. Through combing and connecting different learning modules in the sensor, the sink, and the base station as a global network, the function of distributed detection and centralized prediction can be realized. The implementation system of BLID is designed based on agent technology. Our experiment shows that the proposed scheme has several advantages in terms of efficiency of implementation and high prediction rate. Although we assume in this paper that WSNs is deployed through the three-layer architecture, BLID can also be applicable for the WSNs deployed in two-layer architecture, which only includes base station and sensor. This is because both unsupervised learning and reinforcement learning modules are designed in base station, then sensor can interoperate directly with base station for *promotion* operation. An interesting future work of BLID may be on the efficiently distributed intrusion prediction of WSNs.

Acknowledgments

The authors thank Dr. Rana Ashour for the invitation. This work was supported by Japan Society for the Promotion of Science (JSPS) under Grant-in-Aid for Scientific Research (C) (no. 20560373) and China Scholarship Council (CSC) under Grant no. 2008638003.

References

[1] E. Fontana, J. F. Martins-Filho, S. C. Oliveira et al., "Sensor network for monitoring the state of pollution of high-voltage insulators via satellite," *IEEE Transactions on Power Delivery*, vol. 27, no. 2, pp. 953–962, 2012.

[2] J. Valverde, V. Rosello, G. Mujica, J. Portilla, A. Uriarte, and T. Riesgo, "Wireless sensor network for environmental monitoring: application in a coffee factory," *International Journal of Distributed Sensor Networks*, vol. 2012, pp. 1–18, 2012.

[3] J. Wu and S. Shimamoto, "Usage control based security access scheme for wireless sensor networks," in *Proceedings of the IEEE International Conference on Communications (ICC '10)*, Cape Town, South Africa, May 2010.

[4] B. Sun, L. Osborne, Y. Xiao, and S. Guizani, "Intrusion detection techniques in mobile ad hoc and wireless sensor networks," *IEEE Wireless Communications*, vol. 14, no. 5, pp. 56–63, 2007.

[5] J. Wu and S. Shimamoto, "Integrated UCON-based access control and adaptive intrusion detection for wireless sensor networks," in *Proceedings of the 53rd IEEE Global Communications Conference (GLOBECOM '10)*, Miami, Fla, USA, December 2010.

[6] S. Budhaditya, D. S. Pham, M. Lazarescu, and S. Venkatesh, "Effective anomaly detection in sensor networks data streams," in *Proceedings of the 9th IEEE International Conference on Data Mining (ICDM '09)*, Miami, Fla, USA, December 2009.

[7] F. Bao, I. R. Chen, M. J. Chang, and J. H. Cho, "Hierarchical trust man-agement for wireless sensor networks and its applications to trust-based routing and intrusion detection," *IEEE Transactions on Network and Service Management*, vol. 9, no. 2, pp. 169–182, 2012.

[8] L. M. Wang, T. Jiang, and X. Y. Zhu, "Updatable key management scheme with intrusion tolerance for unattended wireless sensor network," in *Proceedings of the IEEE Global Communications Conference (GLOBECOM '11)*, December 2011.

[9] R. Roman, J. Zhou, and J. Lopez, "Applying intrusion detection systems to wireless sensor networks," in *Proceedings of*

the 3rd IEEE Consumer Communications and Networking Conference (CCNC '06), Las Vegas, Nev, USA, January 2006.

[10] E. C. H. Ngai, J. Liu, and M. R. Lyu, "On the intruder detection for sinkhole attack in wireless sensor networks," in *Proceedings of the IEEE International Conference on Communications (ICC '06)*, Istanbul, Turkey, July 2006.

[11] C. E. Loo, M. Y. Ng, C. Leckie, and M. Palaniswami, "Intrusion detection for routing attacks in sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2, no. 4, pp. 313–332, 2006.

[12] K. Q. . Yan, S. C. Wang, and C. W. Liu, "A hybrid intrusion detection system of cluster-based wireless sensor networks," in *Proceedings of the International MultiConference of Engineers and Computer Scientists (IMECS '09)*, Hong Kong, China, March 2009.

[13] I. C. Paschalidis and Y. Chen, "Anomaly detection in sensor networks based on large deviations of markov chain models," in *Proceedings of the 47th IEEE Conference on Decision and Control (CDC '08)*, Cancun, Mexico, December 2008.

[14] S. Rajasegarar, C. Leckie, M. Palaniswami, and J. C. Bezdek, "Quarter sphere based distributed anomaly detection in wireless sensor networks," in *Proceedings of the IEEE International Conference on Communications (ICC '07)*, Scotland, UK, June 2007.

[15] H. Alipour, M. Gholami, and A. Vahdani, "A base-station oriented anomaly detection for wireless sensor networks," in *Proceedings of the 4th IEEE/IFIP International Conference in Central Asia on Internet (ICI '08)*, pp. 1–5, Tashkent, Uzbekistan, September 2008.

[16] F. Jemili, M. Zaghdoud, and M. B. Ahmed, "Hybrid intrusion detection and prediction multiAgent system, HIDPAS," *International Journal of Computer Science and Information Security*, vol. 5, no. 1, pp. 62–71, 2009.

[17] Z. Zhang, Z. Peng, and Z. Zhou, "The study of intrusion prediction based on HsMM," in *Proceedings of the IEEE Asia-Pacific Services Computing Conference (IEEE APSCC '08)*, Yilan, Taiwan, December 2008.

[18] F. Ozgul, Z. Erdem, and C. Bowerman, "Prediction of past unsolved terrorist attacks," in *Proceedings of the IEEE International Conference on Intelligence and Security Informatics (ISI '09)*, Dallas, Tex, USA, June 2009.

[19] M. R. Ahmadi, "An intrusion prediction technique based on co-evolutionary immune system for network security (CoCo-IDP)," *International Journal of Network Security*, vol. 9, no. 3, pp. 290–300, 2009.

[20] N. Ye, Q. Chen, and C. M. Borror, "EWMA forecast of normal system activity for computer intrusion detection," *IEEE Transactions on Reliability*, vol. 53, no. 4, pp. 557–566, 2004.

[21] K. Haslum, A. Abraham, and S. Knapskog, "DIPS: a framework for distributed intrusion prediction and prevention using hidden Markov models and online fuzzy risk assessment," in *Proceedings of the 3rd Internationl Symposium on Information Assurance and Security (IAS '07)*, Manchester, UK, August 2007.

[22] S. Li and Y. Luo, "Discernibility analysis and accuracy improvement of machine learning algorithms for network intrusion detection," in *Proceedings of the IEEE International Conference on Communications (ICC '09)*, Dresden, Germany, May 2009.

[23] C. F. Tsai, Y. F. Hsu, C. Y. Lin, and W. Y. Lin, "Intrusion detection by machine learning: a review," *Expert Systems with Applications*, vol. 36, no. 10, pp. 11994–12000, 2009.

[24] T. Pietraszek and A. Tanner, "Data mining and machine learning—towards reducing false positives in intrusion detection," *Information Security Technical Report*, vol. 10, no. 3, pp. 169–183, 2005.

[25] Z. Yu and J. J. P. Tsai, "A framework of machine learning based intrusion detection for wireless sensor networks," in *Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC '08)*, Taichung, Taiwan, June 2008.

[26] C. Chen, J. Ma, and K. Yu, "Designing energy-efficient wireless sensor networks with mobile sinks," in *Proceedings of the ACM Sensys'06 Workshop WSW*, 2006.

[27] J. Zhang and V. Varadharajan, "A new security scheme for wireless sensor networks," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '08)*, New Orleans, La, USA, December 2008.

[28] I. Bisio and M. Marchese, "Efficient satellite-based sensor networks for information retrieval," *IEEE Systems Journal*, vol. 2, no. 4, pp. 464–475, 2008.

[29] B. Thuraisingham, "Secure sensor information management and mining," *IEEE Signal Processing Magazine*, vol. 21, no. 3, pp. 14–19, 2004.

[30] P. Goyal, S. Batra, and A. Singh, "A literature review of security attack in mobile Ad-hoc networks," *International Journal of Computer Applications*, vol. 9, no. 12, pp. 11–15, 2010.

[31] G. Anastasi, E. Borgia, M. Conti, E. Gregori, and A. Passarella, "Understanding the real behavior of Mote and 802.11 ad hoc networks: an experimental approach," *Pervasive and Mobile Computing*, vol. 1, no. 2, pp. 237–256, 2005.

[32] http://www.backtrack-linux.org/.

[33] J. Lei, X. Fu, D. Hogrefe, and J. Tan, "Comparative studies on authentication and key exchange methods for 802.11 wireless LAN," *Computers and Security*, vol. 26, no. 5, pp. 401–409, 2007.

[34] A. Bittau, M. Handley, and J. Lackey, "The final nail in WEP's coffin," in *Proceedings of the IEEE Symposium on Security and Privacy*, May 2006.

[35] A. Roy, "On connectionism, rule extraction, and brain-like learning," *IEEE Transactions on Fuzzy Systems*, vol. 8, no. 2, pp. 222–227, 2000.

[36] M. A. Sharbafi, C. Lucas, and R. Daneshvar, "Motion control of omni-directional three-wheel robots by brain-emotional-learning-based intelligent controller," *IEEE Transactions on Systems, Man and Cybernetics Part C*, vol. 40, no. 6, pp. 630–638, 2010.

[37] K. Doya, "What are the computations of the cerebellum, the basal ganglia and the cerebral cortex?" *Neural Networks*, vol. 12, no. 7-8, pp. 961–974, 1999.

[38] J. Hu, T. Sasakawa, K. Hirasawa, and H. Zheng, "A hierarchical learning system incorporating with supervised, unsupervised and reinforcement learning," in *Proceedings of the 4th International Symposium on Neural Networks (ISNN '07)*, Nanjing, China, June 2007.

[39] J. M. Bradshaw, "Introduction to software agents," in *Soft Agents*, J. M. Bradshaw, Ed., AAAI Press/MIT Press, Cambridge, Mass, USA, 1997.

[40] F. Bai, K. S. Munasinghe, and A. Jamalipour, "An ecologically inspired intelligent agent assisted wireless sensor network for data reconstruction," in *Proceedings of the IEEE International Conference on Communications (ICC '10)*, May 2010.

[41] A. Rogers, N. R. Jennings, and D. D. Corkill, "Agent technologies for sensor networks," *IEEE Intelligent Systems*, vol. 24, no. 2, pp. 13–17, 2009.

[42] Z. Deng and W. Zhang, "Localization and dynamic tracking using wireless-networked sensors and multi-agent technology: first steps," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 85, no. 11, pp. 2386–2395, 2002.

[43] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986.

[44] B. Liu, Y. Xia, and P. S. Yu, "Clustering through decision tree construction," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, Dallas, Texas, USA, May 2000.

[45] P. Bellot and M. El-Bèze, "Clustering by means of unsupervised decision trees or hierarchical and K-means-like algorithm," in *Proceedings of the RIAO Conference, Collège de France (RIAO '00)*, Paris, France, April 2000.

[46] V. Kodogiannis and A. Lolis, "Forecasting financial time series using neural network and fuzzy system-based techniques," *Neural Computing and Applications*, vol. 11, no. 2, pp. 90–102, 2002.

[47] T. Kuremoto, M. Obayashi, and K. Kobayashi, "Nonlinear prediction by reinforcement learning," in *Proceedings of the International Conference on Intelligent Computing (ICIC '05)*, Hefei, China, August 2005.

[48] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, The MIT Press, 1998.

[49] H. Kimura and S. Kobayashi, "Reinforcement learning for continuous action using stochastic gradient ascent," in *Proceedings of the 5th International Conference on Intelligent Autonomous Systems (IAS '98)*, 1998.

[50] H. R. Maei, C. Szepesvari, S. Bhatnagar, D. Precup, D. Silver, and R. S. Sutton, "Convergent temporal-difference learning with arbitrary smooth function approximation," in *Proceedings of the 23rd Annual Conference on Neural Information Processing Systems (NIPS '09)*, Vancouver, Canada, December 2009.

[51] IEEE 802.11-1999, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Higher Speed Physical Layer Extension in the 2.4 GHz Band.

[52] W. Lee and S. J. Stolfo, "A framework for constructing features and models for intrusion detection systems," *ACM Transactions on Information and System Security*, vol. 3, no. 4, pp. 227–261, 2000.

[53] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986.

[54] P. Levis, N. Lee, M. Welsh, and D. Culler, "TOSSIM: accurate and scalable simulation of entire TinyOS applications," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (ACM SenSys '03)*, pp. 126–137, November 2003.

[55] V. Shnayder, M. Hempstead, B. R. Chen, G. W. Allen, and M. Welsh, "Simulating the power consumption of large-scale sensor network applications," in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, November 2004.

[56] E. T. H. Chu, H. J. Lee, T. Y. Huang, and C. T. King, "Sample assignment for ensuring sensing quality and balancing energy in wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 9, pp. 1578–1584, 2011.

[57] M. Zeng, I. Y. Chung, J. A. Lee, and J. G. Lee, "An on-node intelligent based energy efficient ECG monitoring system," in *Proceedings of the International Conference on ICT Convergence (ICTC '11)*, September 2011.

[58] N. Erratt and Y. Liang, "Compressed data-stream protocol: an energy-efficient compressed data-stream protocol for wireless sensor networks," *IET Communications*, vol. 5, no. 18, pp. 2673–2683, 2011.

[59] I. Krontiris, Z. Benenson, T. Giannetsos, F. C. Freiling, and T. Dimitriou, "Cooperative intrusion detection in wireless sensor networks," in *Proceedings of the 6th European Conference on Wireless Sensor Networks (EWSN '09)*, Cork, Ireland, February 2009.

[60] NeuroSolutions, Inc., 2010, http://www.neurosolutions.com/.

[61] DARPA, DARPA, Intrusion Detection Evaluation, 1998, http://www.ll.mit.edu/mission/communications/ist/corpora/ideval/data/1998data.html.