Brain-Like Two-Layer Learning based Efficient Attack Detection for Wireless Sensor Networks

Jun WU and Shigeru SHIMAMOTO

1 Introduction

A wireless sensor network (WSN) is consisted of a large number of wireless-capable sensor devices working collaboratively to achieve a common objective. Nowadays, wireless sensor networks (WSNs) have become a technology for the new millennium with endless applications ranging from civilian to military. As a matter of fact, WSNs are often deployed in potentially adverse or even hostile environments where adversaries can launch various kinds of attacks (Wu 2010). These attacks can disturb the deployment purpose of the WSN. How to defense against various kinds of the attacks is a very important issue for WSNs.

In the area of computer science, 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.

In current intrusion detection schemes of WSNs (Wu 2010a, Khanna 2009, Wu 2010b, Yan 2009, Yu 2008, Sun 2007, Su 2005, Loo 2006, Ngai 2006, Paschalidis 2008, Rajasegarar 2007), two approaches have been used: signature-based detection and anomaly detection. Signature-based detection 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 detect attacks by observing deviations from a pre-established 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 the 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.

The existing attack detection schemes can provide security for WSNs to some extent. However, the detection accuracy of most existing schemes is relative low, especially for defensing against unknown attacks. Sensor nodes usually have severe constraints in computational power, memory size, and energy. Because of those limited resources of WSNs, the existing intrusion detection schemes did not consider many effective security defense techniques, such as public key cryptography. Moreover, due to the limited memory in sensors, there are only few features that could be selected to detect intrusion. All the above facts can reduce the detection rate and enhance false alarm rate.

Recently, in the intrusion detection community, interest has been growing applying machine learning techniques to get high performances in classification accuracy. Machine learning based intrusion detection for WSNs (Yu 2008) has gained limited attention so far. WSNs are usually deployed as a hierarchical structure. The nodes in different layer own different resources. We further find hierarchical learning is an existing concept in the area of machine learning. The complexity of different kind learning is different. Based on the interoperation among the learning of different layer, the system can get a good tradeoff between efficient and accuracy. Hence, we consider design the attack detection for WSNs based on hierarchical learning.

Based on above discussion, it is clear that achieving intrusion detection with high accuracy using machine learning is still an open challenge in WSNs. In order to addressing this challenge, we proposed in this paper machine learning based efficient attack detection scheme. By exploring a brain-like two-layer learning model, we

Manuscript received 15th March 2011; revised 13th Sept. 2011; accepted 6th Oct. 2011

Graduate School of Global Information and Telecommunication Studies, Waseda University

propose a novel accurate attack detection scheme, which is specially tailored for WSNs. Our solutions have several advantages. First, our scheme is efficient in terms of storage, computation and communication overhead on the sensor side. Second, it has a significant impact on the accuracy of the intrusion detection due to the brain-like hierarchical learning architecture.

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. Section 6 analyzes related important features. Finally, we conclude this paper in Section 7.

2 Models and Assumptions

2.1 Network Model

In this work, we consider a WSN with two-layer structure which includes base station layer, sink layer and sensor layer. This structure is a popular way for deploying WSNs. Usually each sensor can collect data and delivery the data to the sink or base station （BS）. The resources of sensors are limited. However, the resources of sink and bases station are powerful.

2.2 Brain-like hierarchical learning

Recently, brain-like learning and computation has attracted a lot of attentions in the area of machine learning. In this paper, we consider the brain-like learning model in （Doya 1999）, which is developed into a system structure in （Hu 2005）. 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. We design the related learning scheme for the sensor, sink and base station layers respectively, and base our design on the concept of brain-like hierarchical learning.

Note that in this paper we use the supervised learning and unsupervised learning as a two-layer learning model in the brain-like learning framework.

3 Brain-like Two-layer Learning Attack Detection Scheme

3.1 Systemic Design

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. Then the sink sends the response to the sensor. In short, that sink and the base station perform intrusion detection by themselves. The sensor performs low level detection by itself, but it needs the help of sink for performing high level detection.

In this paper, for simplification of presentation, we assume that only one base station and several sink is in the WSN. In our scheme, we assume that sensors cannot communicate with base station directly. They can only communicate with base station via sink. For example, sensors can send unknown features to the sink, and then the sink sends back the detection results. However, the sensor can communicate with the sink which is in charge of the corresponding area.

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

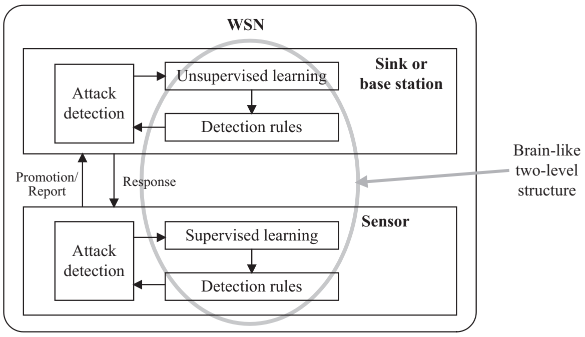

Fig. 1 Systemic design

3.2 Supervised Learning based Intrusion Detection in Sensor

Decision tree is a kind of classifier for supervised learning. Based on this kind of learning, we design the intrusion detection in sensors. 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 M 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 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 $\mathrm{T}=\{t_1,t_2,\cdots,t_k\}$ 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 $Drawn(d)=\{v_1, v_2,\cdots,v_k\}$. The function can be extended to a set of rules D by

$$Drawn(D)=\bigcup_{d\in D} Drawn(d)$$

Definition 2: We define the function $Obtain(N(A, D, M))=\{Subtree|\ N_1(A, D_1, M_1)\cup N_2(A, D_2, M_2)\ldots\cup N_m(A, D_m, M_m)\}\}$. N_1, N_2,…,N_m are the member nodes of the subtree. This function send *N(A, D, M)* to a sink. Then *N(A, 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(A, 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 based 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 follow scheme to construct a decision tree.

The scheme of tree construction is shown in Fig.2. 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 *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.

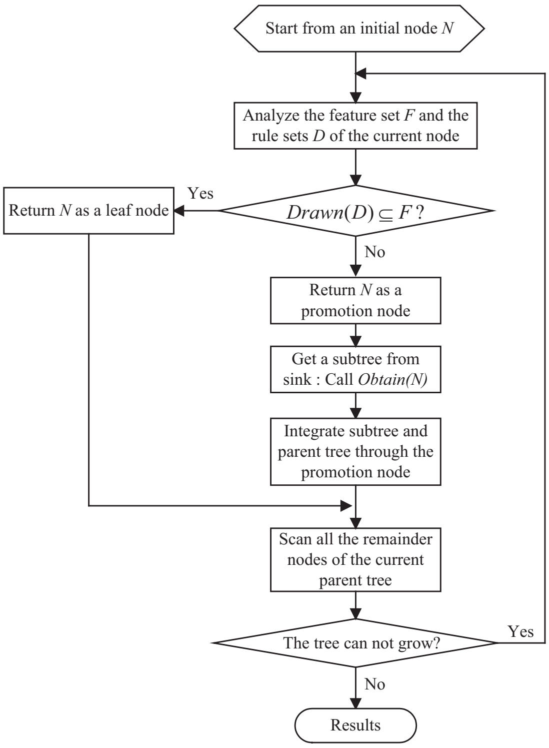
Start from an initial node N
Analyze the feature set F and the rule sets D of the current node
Return N as a leaf node
Yes
Drawn(D) ⊆ F ?
No
Return N as a promotion node
Get a subtree from sink : Call Obtain(N)
Integrate subtree and parent tree through the promotion node
Scan all the remainder nodes of the current parent tree
The tree can not grow?
Yes
No
Results

Fig. 2 Decision Tree learning in sensor

3.3 Unsupervised Learning based Intrusion Detection in Sink and Base Station

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.

At first we define a criterion of quality of the cluster. Assume the characteristic of a request from a user or an attacker is a data sample. All the samples consist of the data space. The decision tree with L leaves splits space of characteristics into L non overlapping subareas S^1, S^2,..., S^L. This splitting space corresponds to the splitting of the set of observations samples into L subsets $Sample^1$, $Sample^2$, ... , $Sample^L$. Thus, the number of leaves in a decision tree coincides with the number of clusters. A cluster of samples is denoted as $Sample^i$.

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

$$U(Sample^i,V^i)=(X_1\in V_1^i)\wedge(X_2\in V_2^i)\wedge\cdots\wedge(X_n\in V_n^i)$$

where V_j^i is interval of the sample space of the attacks, which is calculated by follows.

$$V_j^i=[\min_{Sample^i}\{x_j\},\max_{Sample^i}\{x_j\}]\ \text{, or}$$

$$V_j^i=\{x_j\mid x_j\in Sample^i\}$$

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

Relative capacity（volume）of characteristic subspace

R^i can be calculated by

$$\lambda^i = \prod_{j=1}^{n} \frac{|V_j^i|}{|D_j|}$$

where $|V_j^i|$ denotes 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 samples from initial sample (for the quantitative characteristic) or the general number of values of this characteristic (for the qualitative characteristic). Here initial sample means the samples space which has not been divided into the subspace by the learning algorithm. General number means the number which denotes the quantitative characteristics, because we use number to denote the quantitative characteristic for processing.

When the number of clusters is known, the criterion of quality of a cluster is the amount of the relative volume of characteristic subspace R^i:

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

If the number of clusters is not given beforehand, the next value as the criterion of quality is as follow,

$$P=g+aL$$

where $a>0$ is a given parameter.

When minimizing this criterion, we receive the characteristic subspace of the minimal size. Meanwhile, we can aspire to reduce the number of characteristic subspaces.

For the construction of a decision tree, the method of consecutive branching described in paragraph 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. The total volume of received characteristic subspace should be minimal. The node will be divided if the volume of the appropriate characteristic subspace 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 Implementation System based on Agent Technology

We use multi-agent to realize the function of intrusion detection in WSNs. There are four kinds of agents designed in WSNs, which are detection agent *(DA)*, communication agent *(CA)*, and database agent *(BA)*. Figure 3 shows the structure of the agent system of a node in wireless sensor network. DA is distributed in each node of the WSN.

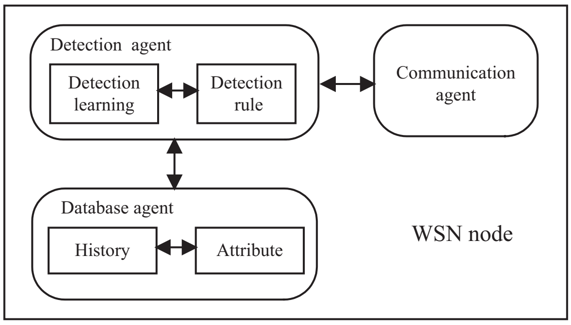

Fig. 3 Node model of agent system

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.

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 pre-process the raw data into the format required by the data classification techniques. On one hand, this module acts an interface for the node interoperating with other nodes in WSNs. On the other hand, communication agent performs an interface to receive request and send responses for the user who accesses the node.

Database agent *(BA)*:

1) The *History Module (HM)* 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.

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

Note that there are two cases of interoperation among the learning modules in different kind nodes in WSNs. These interoperations include *promotion* operation and report operation. The sequence of *promotion* operation of an access to a sensor is illustrated in Fig. 5. The sequence of report operation is similar with that of the *promotion*.

In Fig. 4, the sequence model of *promotion* operation can be described in 9 steps:

Step1: The legal or illegal user sends an access request the sensor.

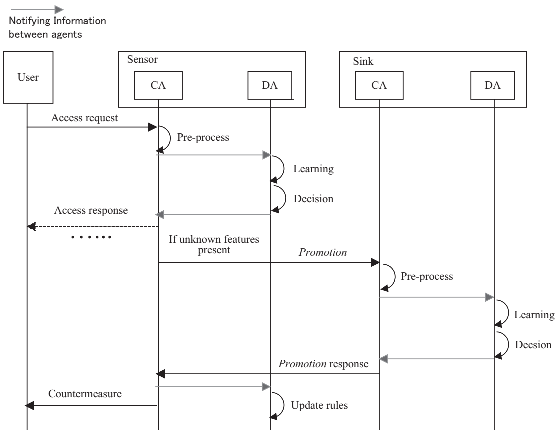

Fig. 4 Sequence model of promotion operation

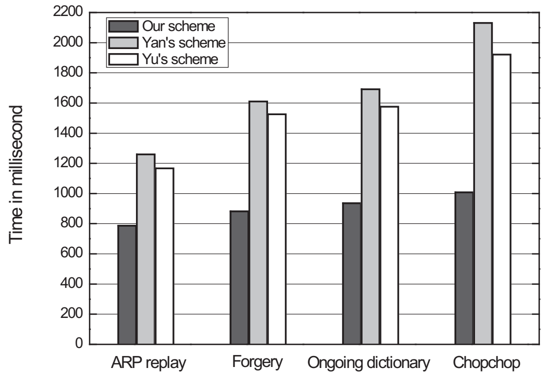

Fig. 5. Time overhead in sensor

Step2: The communication agent in the sensor receives the access request and pre-processes the request data and transfer the notification to the corresponding detection agent in the sensor.

Step3: The detection agent looks up the features of the request based on the rules in the existing decision tree using supervised learning.

Step4: If the features match the rule of the existing decision tree, the communication agent sends an access response to the user.

Step4: If there are unknown features which do not match the rule of the decision tree, the sensor will start the *promotion* process and send the features of the unknown access to the sink.

Step5: The communication agent in the sink receives the parameters of the *promotion* operation and then pre-processes them.

Step6: The communication agent in the sink transfers the notification of the *promotion* operation to corresponding detection agent in the sink.

Step7: The detection agent in the sink performs the unsupervised learning based on the new features of the access and the communication agent send back the results of the unsupervised learning to the sensor.

Step8: The detection agent in the sensor update the decision tree structure and make decision based on the new decision tree.

Step9: The communication agent in the sensor sends the response of the unknown access to the user.

4 Wireless Attack Experiment and Data Set

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 devices technologies, 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 detection. In order to enhance the detection 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 to construct the feature data set. On one hand, we select the important features of access control. On the other hand, according to IEEE 802.11 standard, the fields of the MAC header can be extracted. We used the Information Gain Ratio (IGR) (Quinlan 1986) as a measure to determine the relevance of each feature. 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 Table 1 and Table 2 respectively. The number of the selection features

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 non final 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.

depends on the requirements of security and the resources 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, chopchop attack, 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 (Backtrack 2010).

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 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.

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

5 Evaluation and Comparisons

5.1 Time and Memory Overhead 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 our scheme for TinyOS and tested it using TOSSIM. The mote that TOSSIM simulates is MicaZ.

Moreover, the number of cluster is unknown in the experiment.

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 resources 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

Based on the above reasons, we just focus on the test phase. We evaluate the average overhead of every test data sample. The average overhead caused by the proposed scheme and related schemes during detection is reported in Fig. 6, which is the time needed by a sensor from receiving a request to making a local detection decision.

Table 4 Memory Consumption in Sensor

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

As shown in Fig. 5, the time overhead caused by the proposed scheme is lower than that of the Yan's and Yu's schemes. The results show that detecting unknown attacks usually need more time than detecting known attack.

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 means that the proposed scheme leaves enough space in the mote's memory for user applications. On the other hand, the memory consumption of Yan's and Yu's are 74673 Bytes and 53782 Bytes. Our scheme shows the advantage on memory consumptions on sensor.

Note that for the sensor, the subtree for chopchop attack get from the sink is 3729 Bytes.

5.2 Communication Overhead

The proposed scheme 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 focus on the case that the attacks occur to sensors. Figure 6 depicts the communication cost of the proposed scheme measured in overhead packets in WSNs.

As shown in Fig. 6, the communication overhead in

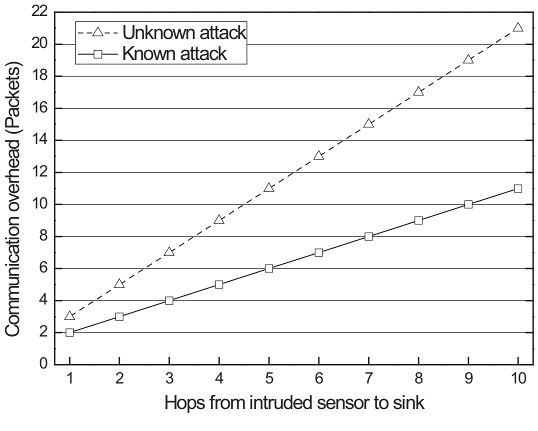

Fig. 6 Communication overhead

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.

5.3 Detection Rate

The evaluation of the accuracy of detection was obtained using Matlab and NeuroSolutions (NeuroSolutions, 2010). The detection accuracy of the proposed scheme depends on the learning algorithm in sink and base station, because "*promotion*" operation exists in the low level detection in sensor.

We use a metrics to evaluate the attack detection performance, namely, detection rate q. The detection rate is formally defined by

$$q = d/n$$

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

The false alarm rate measures the percentage of false positives among all normal traffic events. A formal definition is given by

$$\eta = m / k$$

where m is the total number of false positive alarms and k is the total number of connection event.

For evaluating the accuracy of detection, the training and test data are the data set in section 4. The experiment results of detection rate and the comparison with Yan's as well as Yu's schemes are shown in Fig. 7.

Fig. 8 shows the experimental results of false alarm rate of detection. Comparing with Yan's and Yu's schemes, the detection false alarm rate of our scheme is 5.7 and 8.5 percentages lower respectively.

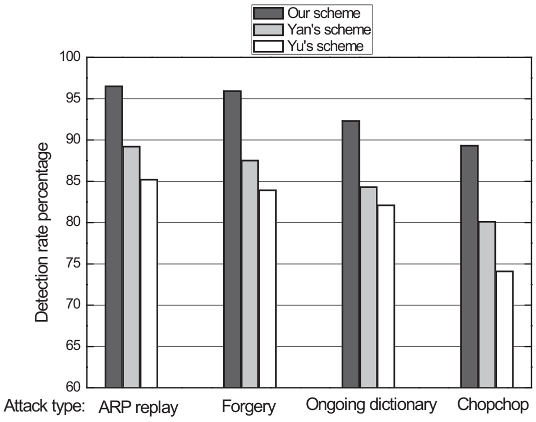

Fig. 7. Detection rate

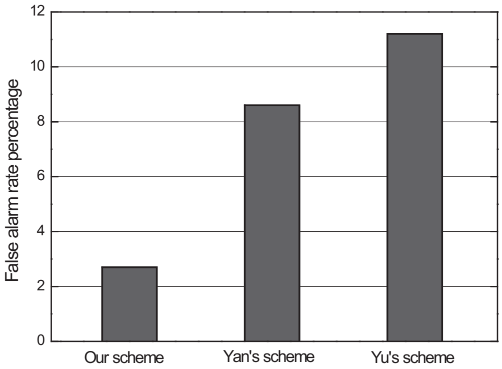

Fig. 8. False alarm rate

6 Analysis

In our scheme, the same point between sink and base station is that both the sink and base station can perform the high level intrusion detection using unsupervised learning algorithm. However, there is different point between sink and base station. Sink node not only perform the high level detection itself, but also perform the high level detection for the sensors which is covered by the sink.

Feature selection is a very important for reducing the number of characteristic subspaces. In this paper, we just select some important features of the traffic and access control, which is refer to the related papers. We think how to further reducing the number of characteristic subspaces is an important future work. The given n value for the volume of the appropriate subspace depends on the features selection and the classification accuracy of the learning algorithm.

In WSNs, the resources of sensors are limited, and the resources of sink and base station are powerful. Therefore, we focus the time overhead of sensors in our evaluation. Because only related simple supervised learning algorithm is performed as low level detection in sensors, which has lower complexity than Yan's and Yu's schemes. Therefore, in Fig.5, the time overhead caused by the proposed scheme is lower than that of Yan's and Yu's schemes.

On the other hand, in our scheme, the detection rate depends on the high level detection. We use unsupervised learning to perform the high level detection, which can deal with complex samples. Therefore, our scheme has higher detection rate than Yan's and Yu's schemes.

In the evaluation, because we just use chopchop attack as the unknown attack, only one promotion operation is performed.

7 Conclusion

In this paper, we analyzed the important security issue of accurate intrusion detection in WSNs. In order to resolve this problem, we proposed the brain-like tow-layer learning based attack detection scheme, in which the sensor, sink/

base station perform different kinds of learning algorithms and interoperate optimally with each other. Referencing to the 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/base station for intrusion detection, which has a correspondence with unsupervised learning of cerebral cortex. Through combing and connecting different learning modules in the sensor, the sink and the base station as a global network, the function of distributed attack detection can be realized. The implementation system of the proposed scheme is designed based on the agent technology. Our evluation based on the attack experiment shows that the proposed scheme has several advantages in terms of efficiency of implementation, high detection rate. Although we assume in this paper that WSNs is deployed through the three-layer architecture, the proposed scheme can also be applicable for the WSNs deployed in two-layer architecture, which only includes base station and sensor. This is because based on our model the sensor can interoperate directly with base station for *promotion* operation.

REFERENCES

J. Wu and S. Shimamoto, "Usage control based security access scheme for wireless sensor networks" Proc. IEEE International Conference on Communications (ICC 2010), Cape Town, South Africa, May 2010.

R. Khanna, H. Liu and H. Chen, "Reduced complexity intrusion detection in sensor networks using genetic algorithm," Proc. IEEE International Conference on Communications (ICC 2009), Dresden, Germany, May 2009.

J. Wu and S. Shimamoto, "Integrated UCON-based access control and adaptive intrusion detection for wireless sensor networks," Proc. IEEE Global Communication Conference (GLOBECOM 2010), Miami, USA, Dec. 2010.

K. Q. Yan, S. C. Wang, and C.W. Liu, "A hybrid intrusion detection system of cluster-based wireless sensor networks," Proc. International MultiConference of Engineers and Computer Scientists (IMECS 2009), Hong Kong, China, Mar. 2009.

Z. Yu and J. J. P. Tsai, "A framework of machine learning based intrusion detection for wireless sensor networks," Proc. IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC 2008), Taichung, Taiwan, Jun. 2008.

B. Sun, Y. Xiao, and S. Guizani, "Intrusion detection techniques in mobile ad doc and wireless sensor networks," IEEE Wireless Communication, pp. 56-63, Oct. 2007.

C. C. Su, K. M. Chang, Y. H. Kuo, "The new intrusion prevention and detection approaches for clustering-based sensor networks," Proc. IEEE Wireless Communications and Networking Conference (WCNC 2005), New Orleans, USA, Mar. 2005 .

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, Oct. 2006.

E. Ngai, J. Liu, and M. Lyu. "On the intruder detection for sinkhole attack in wireless sensor networks", Proc. IEEE International Conference on Communications (ICC 2006), Istanbul, Turkey, Jun. 2006.

I. Ch. Paschalidis and Y. Chen, "Anomaly detection in sensor networks based on large deviations of Markov chain models," Proc. 47th IEEE Conference on Decision and Control Cancun, Mexico, Dec. 9-11, 2008.

S. Rajasegarar, C. Leckie, M. Palaniswami, and J. C. Bezdek, "Quarter sphere based distributed anomaly detection in wireless sensor networks," Proc. IEEE International Conference on Communications (ICC 2007), Glasgow, Scotland, UK, May 2007.

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, Oct. 1999.

J. Hu, T. Sasakawa, K. Hirasawa, and H. Zheng, "A hierarchical learning system incorporating with supervised, unsupervised and reinforcement learning," Proc. International Symposium on Neural Networks (ISNN 2007), Nanjing, China, Jun. 2007.

J. R. Quinlan, "Induction of decision trees," Machine Learning, vol. 1, pp. 81-106, 1986.

Backtrack 4, [Online]: http://www.backtrack-linux.org/, accessed on Jul. 2010.

NeuroSolutions, Inc., [Online] http://www.neurosolutions. com/, accessed in Aug. 2010.