A Hybrid Intrusion Detection System of Cluster-based Wireless Sensor Networks

K.Q. Yan, S.C. Wang, C.W. Liu

Abstract—**Recent advances in Wireless Sensor Networks (WSNs) have made them extremely useful in various applications. WSNs are susceptible to attack, because they are cheap, small devices and are deployed in open and unprotected environments. In this paper, we propose an Intrusion Detection System (IDS) created in Cluster-based Wireless Sensor Networks (CWSNs). According to the capability of Cluster Head (CH) is better than other Sensor Nodes (SNs) in CWSN. Therefore, a Hybrid Intrusion Detection System (HIDS) is designed in this research. The CH is used to detect intruders that not only decreases the consumption of energy, but also efficiently reduces the amount of information in the entire network. However, the lifetime of network can be prolonged by the proposed HIDS.**

Index Terms—**Wireless sensor network, Intrusion detection, Hybrid IDS, Anomaly detection, Misuse detection.**

I. Introduction

Advances in wireless communication and miniature electronics have enabled the development of small, low-cost, low-power sensor nodes (SNs) with sensing, computation and communication capabilities. Therefore, the issues of Wireless Sensor Networks (WSNs) have become popular research subjects. WSN is a non-infrastructure network, and through the mass deployment of SNs, a WSN is formed. The major function of WSN is to collect and monitor the related information which about the specific environment. The SNs detect the surrounding environment or the given target and deliver the data to the sink using wireless communication. The data is then analyzed to understand the state of the target. However, due to the design of their hardware, WSNs suffer from many resource constraints, such as low computation capability, small memory and limited energy.

Two network topologies occur in WSNs. One is flat-based WSN, and the other is Cluster-based WSN (CWSN). However, a large amount of the information is generated by multi-hop communication and the energy consumption is raised in flat-based WSN, such as SPIN [3]. In CWSN, all SNs are clustered, and a Cluster Head (CH) is elected to manage the operation of its own cluster [4,8-10]. CH should aggregate data from all SNs sensed from a specific target. Therefore, CWSN efficiently reduces the amount of information in the entire network. The advantages of CWSN are a decrease in the consumption of energy, an increase in the network scale, and a prolonged network lifetime. Many protocols of CWSN have been proposed, such as LEACH [4], TEEN [9], APTEEN [10], and PEGASIS [8].

Because WSNs consist of many low-cost, small devices, and usually deploy to an open and unprotected region, they are vulnerable to various types of attacks. For example, when WSN is applied to the battlefield, SNs are invaded by the enemy and destroyed. Thus, the security of the WSN needs to be considered. A prevention mechanism is used to counteract well-known attacks. It establishes a corresponding prevention method, according to the characteristics of an attack. However, prevention mechanisms cannot resist overall attacks. Therefore, it is necessary to detect the attacks, using an Intrusion Detection System (IDS). IDS is used to detect the packets in a network, and determine whether they are attackers. Additionally, IDS can help to develop the prevention system through acquired natures of attack.

The IDS acts as a network monitor or an alarm. It prevents destruction of the system by raising an alarm before the intruder begins to attack. The two major models of intrusion detection include anomaly detection and misuse detection [7]. Anomaly detection builds a model of normal behavior, and compares the model with detected behavior. Anomaly detection has a high detection rate, but the false positive rate is also high. The misuse detection model is built, so that the attack type is determined by comparison with the attack behavior. The misuse detection has high accuracy, but the detection rate is lower. The misuse detection cannot detect unknown attacks, which are not in the model base. Many researchers discuss the model of hybrid detection to gain both the advantages of anomaly detection and of misuse detection [1,12]. This combination detects unknown attacks, using the detection rate of anomaly detection, and the accuracy of misuse detection. The Hybrid Intrusion Detection System (HIDS) achieves the goals of high detection rate and low false positive rate.

In this study, a HIDS is discussed in a heterogeneous CWSN. CH is a one of SNs in the CWSN but the capability of CH is better than other SNs [2]. Additionally, the CH aggregates the sensed data from other SNs in its own cluster. This makes it a target for attackers. However, the CH is used to detect the intruders in our proposed HIDS. This not only decreases the consumption of energy, but also efficiently reduces the amount of information. Therefore, the lifetime of WSN can be prolonged.

The remainder of this paper is organized as follows: In Section 2, works relevant to the common attacks in WSN and the analytic tools of intrusion detection, used in our research, are introduced. In Section 3, the proposed methods and architecture of our research are introduced. The simulation results used to evaluate the performance of the proposed

Manuscript received December 26, 2008.

K.Q. Yan is with the Chaoyang University of Technology, Taiwan, ROC (e-mail: kqyan@cyut.edu.tw).

S.C. Wang is with the Chaoyang University of Technology, Taiwan, ROC (corresponding author to provide phone: 886-4-2332-3000; fax: 886-2374-2337; e-mail: scwang@cyut.edu.tw).

C.W. Liu is with the Chaoyang University of Technology, Taiwan, ROC (e-mail: s9614640@cyut.edu.tw).

system are presented in Section 4. Finally, the conclusion and future work is discussed in Section 5.

II. RELATED WORK

A. Attacks in WSN

Attacks can be classified into two main categories, based on the objectives of intrusion [14]. The comparison of attacks in WSN is shown in Table 1 [6,15,16]. However, the majority of attack behavior consists of the route updating misbehavior, which influences data transmission. In the application of CWSN, the data is sensed and collected by SNs, and is delivered to CH to aggregate. The aggregated data is then sent to sink from CH. Therefore, CH is a main target for attack.

Table 1. The comparison of different attacks in WSN

Attack name	Behavior
Spoofed, Altered, or Replayed routing information	Route updating misbehavior
Select forward	Data forwarding misbehavior
Sinkhole	Route updating misbehavior
Sybil	Route updating misbehavior
Wormholes	Route updating misbehavior
Denial of Service	Data forwarding misbehavior
Hello floods	Route updating misbehavior
Acknowledgment spoofing	Route updating misbehavior

B. Analytic Tool of Intrusion Detection

The proposed HIDS in our research not only efficiently detects attack, but also avoids the waste of resources. First, it filters a large number of packet records, using the anomaly detection model, and then completes a second detection, using the misuse detection model. By training the mode of normal behavior, the anomaly detection model detects the normalcy of current behavior, as determined by the rules. The misuse detection model determines if the current behavior is an attack, and the BPN is used to classify the attacks.

1) Rule-based

Rule-based presents the thoughts of expert [11]. Because human thought is very complicated, the knowledge could hardly be presented by algorithms. Therefore, a rule-based method is used to analyze results. The rules are defined by an expert, through his experience and observation. Additionally, the rules are logged in a rule base after they have been defined. The basic method of expression of rule is if…then, that means if "condition" is established and then the "conclusion" will occur.

2) Back Propagation Network

Back Propagation Network (BPN) is the most typical and the most general model to use in a neural network [13]. BPN is a model of supervised learning, through the specific environment to get the training data, which includes input and output variables. However, BPN learns the corresponding relations between input and output variables to infer the kind of output variables that a new input variable belongs to. Therefore, it is more adaptable works such as diagnosis, prediction, etc.

A network structure of BPN includes many layers, and each layer has several processing units. The network structure of BPN is shown in Figure 1. It consists of three layers, including an input layer, a hidden layer, an output layer and many links between each layer. The input layer is used to input the outer environmental messages, and by the intersect computing in the hidden layer, a corresponding output is gotten from output layer.

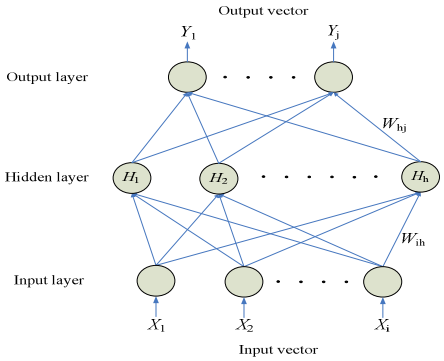

Figure 1. Network structure of BPN

In the progression of BPN training, when all training data have been trained, then the procedure is completed and it is called one epoch. The BPN learns training data repeatedly, and tunes the weights between layers continuously, through many epochs, until the output of the network is similar to the target value and a convergence is achieved.

III. RESEARCH METHOD AND ARCHITECTURE

In CWSN, due to the heterogeneous nature of SNs, the capability of CH is greater than general SN. Additionally, because CH aggregates sensed data from SNs, it therefore often suffers attack. The CH used to detect intruders is not only decreases the consumption of energy, but also efficiently reduces the amount of information in the entire network.

The proposed HIDS in this research consists of three models is shown in Figure 2. The anomaly detection and misuse detection model is used to detect intrusion that to filter a large number of packet records using the anomaly detection model and to make a further detection with the misuse detection model. Finally, the decision making model integrates the outputs of anomaly detection and misuse detection models. It determines if an intrusion occurred, and classifies the type of attack. The output of the decision making model is then reported to the administrator for follow-up work.

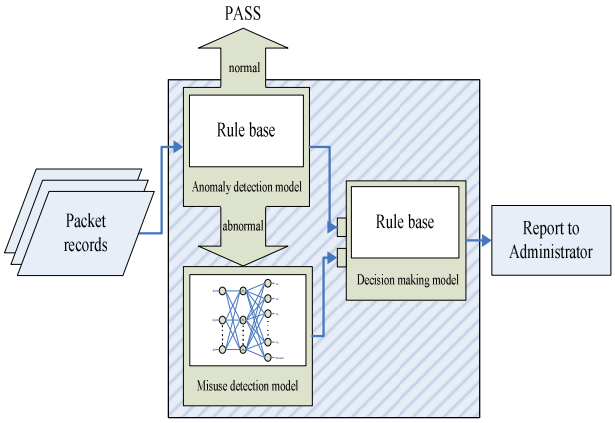

Figure 2. System architecture

A. Anomaly Detection Model

The anomaly detection model plays a role like a filter in this research. Abnormal packets are delivered to the misuse detection model for further detection. Because the anomaly detection uses a defined model of normal behavior, a packet is determined to be abnormal by the system when the current behavior varies from the model of normal behavior. As a result, the anomaly detection usually determines the normal communication as abnormal communication, and creates the problem of erroneous classification. However, it seldom marks an abnormal communication as a normal communication. Therefore, the anomaly detection model is used to filter a large number of packet records first, and make further detection with the misuse detection model, when the amount of information decreases.

Our anomaly detection model adopts a rule-based method, using the rule base to analyze the packets, and distinguish which packets are abnormal. Therefore, a model of normal behavior is established. In our research, we use the rule-based method to construct the anomaly detection model. The flow of construction can be divided into three steps, as follows:

Step 1: **Analyzing the packet's historical records of CWSN.** In CWSN, the packets, which pass through CH, are sent from: (1) the member node of its own cluster; (2) the neighbor CH, which chooses this CH as the transmission path. Therefore, we collect the past packets which communicate on CH to analyze, dividing the packets into normal and abnormal.

Step 2: **Feature selection.** To find the features, which have identifiable properties, we compare the normal and abnormal packets to find the features, which determine normalcy, and develop the rules in our anomaly detection model.

Step 3: **Establishing the rules in anomaly detection model.** Because the anomaly detection determines attack occurrences, according to a defined model of normal behavior, we use the rule-based method to define the state of normal packets. The rules are defined, according to normal packets and the selected features. In addition, the defined rules are saved in a rule base. The established rule base is our anomaly detection model. In CWSN, all packets, which pass through the CH, have to be detected by anomaly detection model. The misuse detection model makes further detection when it is abnormal. The detection flow chart of anomaly detection model is shown in Figure 3.

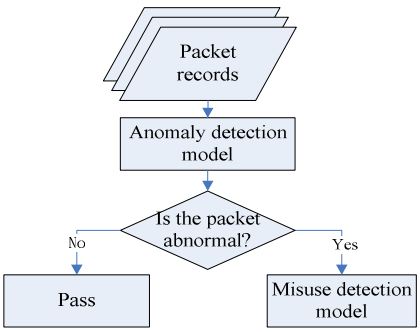

Figure 3. The detection flow chart of anomaly detection model

B. Misuse Detection Model

The misuse detection model utilizes various models of well-known attack behaviors, so we should build a model base according to these behaviors. In this research, we adopt BPN to construct our misuse detection model, because the performance in most techniques of intrusion detection is promised through training data. Through the supervised learning of BPN, learns the corresponding relations between input and output variables, and tunes the corresponding weight. It can result in the error for inference is minimal, so as to high accuracy. Therefore, BPN achieves high accuracy for our HIDS through mass trainings. We embed the model in the sensor when BPN has completed the training.

In this research, a three-layer BPN is adopted for our misuse detection model that includes an input layer, a hidden layer and an output layer. We use the abnormal packets, which were determined by anomaly detection model, as the input vector. The number of processing units in input layer is determined through the selected features for packet. And the number of processing units in hidden layer is designed through the mean method, which is input layer units adds output layer units divided by 2. After analysis, we know that eight common attacks exist in WSN, including: Spoofed, Altered, or Replayed Routing Information, Select Forward, Sinkhole, Sybil, Wormholes, Denial of Service, Hello Floods and Acknowledgment Spoofing. Nine processing units in the output layer represent eight different attacks and one normal behavior, to determine whether the inputted packet is an intrusion, and make a classification.

We collect the packet's historical records, which pass through CH in CWSN, as the sample data for training. Most of packets are normal in WSN. This results in an unbalanced training data. In other words, when normal packets are too large, the BPN neglects the part which occupies a low rate data. In addition, to avoid this problem, we filter the training data through the anomaly detection model, and leave the abnormal packets for training.

Before inputting the training data to BPN, we normalize the training data, and change it into a form, recognizable by BPN. In other words, we convert the packet records into binary values through preprocessing, and then input to BPN. The established flow chart of misuse detection model is shown in Figure 4. First, we set up the network parameters (we often get a better convergence when the learning rate is set to 0.5 or between 0.1 and 1.0 [13]). The actual learning rate is determined through simulation. Additionally, we assign values between 0 and 1 as the weights and biases randomly. We then feed the training data into BPN, computing the actual output through the method of feed forward. And calculate the error and correction of output and hidden layers through the method of back propagation, to update the weights and biases of network, until all training data have been done to stop, and it is called one epoch. We could learn training data repeatedly, and tune the weights between layers continuously, through many epochs, until the output of network is similar to the target value, and the training is complete.

All abnormal packets, which were determined by the anomaly detection model, are subjected to the misuse detection model. First, we convert the abnormal packets into binary value in a preprocessing step, and input the misuse

detection model to calculate the output. We finally deliver to the decision making model to integrate.

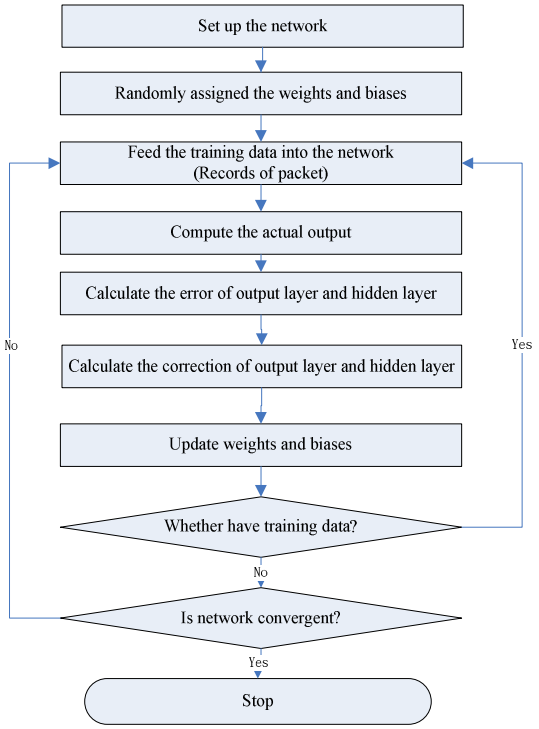

Figure 4. The established flow chart of misuse detection model

C. Decision Making Model

The decision making model is used to combine the outputs of the anomaly detection and misuse detection models. It determines whether or not an output is an intrusion, and the category of attack. It then has to report the results to the administrator to help them handle the state of the system and make further corrections. We adopt a rule-based method to establish the decision making model, using the rules to combine the outputs of two detection models, and its main advantages are that it is very simple and fast. The rules of the decision making model are shown in Table 2.

Table 2. The rules of decision making model

Rules
If anomaly detection model detects an attack **and** misuse detection model does not detect attack **then** it is not an attack and it is erroneous classification.
If anomaly detection model detects an attack **and** misuse detection model detects attack **then** it is an attack **and** determine the class of attack.

IV. Experiment

In this section, the proposed architecture is evaluated through simulation. Because the rules in the anomaly detection model are defined by experts, we cannot verify its performance through the simulation. As a result, the experiment in this research would evaluate the performance of misuse detection model, adopted by BPN.

A. Data Collection

Due to the real sample cannot be gotten in WSN for intrusion detection, the KDDCup'99 dataset [17] is used as the sample to verify the performance of the misuse detection model. The KDDCup'99 dataset, referred by Columbia University, was arranged from intrusions simulated in a military network environment at the DARPA in 1998. It was performed in the MIT Lincoln Labs, and then announced on the UCI KDD Cup 1999 Archive.

The features consist of 34 types of numerical features and 7 types of symbolic features, according to different properties of attack. Additionally, the KDDCup'99 dataset includes many attack behaviors, classified into four groups: Probe, Dos, U2R, and R2L. It also includes a kind of normal communication. Therefore, we would use these five behaviors for the classification of IDS in the experiment.

The attacks Spoofed, Altered, or Replayed Routing Information, Sinkhole, Sybil, Wormholes, and Acknowledgment Spoofing, need to make a probe step before they begin to attack, so they would be classified as Probe attacks. Select Forward, which uses illegitimate data forwarding to make attack, is known as a Dos attack. Sinkhole, Wormholes, and Hello Floods are caused by inner attacks, and are therefore classified as U2R. Spoofed, Altered, or Replayed Routing Information, Sinkhole, Sybil, Wormholes, Hello Floods, and Acknowledgment Spoofing, which through the weakness in the system to make attack, so they would be classified as classified as R2L.

In this research, we use the kddcup.data_10_percent.gz as our sample of training and testing dataset in experiments. This includes 10% data in the KDDCup'99 dataset, and the total number of communication records is 494021. It randomly samples 30000 records as training data, and 15000 records as testing data. However, because the sample number of Probe, U2R, and R2L is less, we sample their whole records, among these, putting two-thirds data as training data, and one-third data as testing data. While other sample numbers are sampled according to their ratio from kddcup.data_10_percent.gz, they are classified to Normal and DoS type separately. The Normal accounts for about 20%, the Dos accounts for about 80%. The data sampling number and ratio are shown in Table 3.

Table 3. Amount and ratio of data sampling

Data / Category	10% dataset		Training data		Testing data	
	Amount of total data	Ratio	Amount of training data	Ratio	Amount of testing data	Ratio
Normal	97278	19.69%	5295	17.65%	2648	17.65%
Probe	4107	0.83%	2738	9.13%	1369	9.13%
DoS	391458	79.24%	21181	70.60%	10591	70.61%
U2R	52	0.01%	35	0.12%	17	0.11%
R2L	1126	0.23%	751	2.50%	375	2.50%
Total	494021	100%	30000	100%	15000	100%

B. The Simulation Design of BPN

In this subsection, we present the flow of experiment, feature selection, data preprocessing and the training BPN model.

1) The Flow of Experiment

In this research, we first sample the training and testing data from the KDDCup'99 dataset, and filter some unimportant and noise features, to decrease the data dimension. We then normalize the data through the preprocessing step, and use the data to train the BPN model.

2) Feature Selection

Not every feature has decisive effects on the output of classification. Some features even make classification errors. Therefore, feature selection is an important factor to affect the performance of IDS. In this research, the feature selection method proposed by Jong *et al*. [5] is adopted. Therefore, the data dimensions and the complications are reduced; features, which are unimportant, and noise are filtered, BPN is used to verify the results of selection.

3) Data Preprocessing

Before training BPN model, it must be normalized for the training data, and letting it be a data type which recognizable by BPN. However, the original state is normalized for the training data, and 24 types of features are chosen [5]. To achieve normalization, these 24 features are converted into binary value. We design a corresponding binary value to transfer the original value for the symbolic data. In addition, we use formula (1) to transfer the values into fall between 0 and 1, for the numerical data, and then divide them into several blocks, finally use binary value to replace them.

$$v' = \frac{v - \min}{\max - \min} \qquad (1)$$

Additionally, the corresponding target value is classified into 5 groups: Normal, Probe, Dos, U2R, and R2L, which translates to 00001, 00010, 00100, 01000 and 10000, respectively.

4) Training BPN model

BPN is a network model of supervised learning, inputting training data which has target values to make training, learning the training data repeatedly, tuning the weights between layers continuously, until the output of network is similar to the target value, and training is completed. In the training process, original weights and biases are assigned from 0 to 1 randomly. Through the error back propagation to find out the correction, and it would stop until the network gets a convergence. The allocation of each layer in the three-layer BPN is shown below:

(1) Input layer: According the 24 types of features, chosen by Jong [5]. We transfer the features into 95 binary values, and produce 95 neurons.
(2) Output layer: The outputs have 5 types, including Normal, Probe, Dos, U2R and R2L. We produce 5 neurons.
(3) Hidden layer: According the mean method, adding the number with input units and output units, and dividing it by 2, to get the number of hidden layer unit. We produce 50 neurons.

In our research, two methods for training in the simulation exist, and observe the change in learning affects the performance. When the BPN training is complete, we input 15000 testing data to make classifications, so as to evaluate its performance and observe its classification accuracy. Two groups of parameters exist in this experiment that represents two different experiments. The design of experimental parameters is shown in Table 4.

Table 4. The design of experimental parameters

	Learning rate	Epoch
Experiment 1	0.5	5000
Experiment 2	0.1	5000

C. Simulation Results and Discussion

The adopted system in this research is the AMD Athlon(tm) 64 X2 Dual Core Processor 5000+ 2.59 GHz PC, with 2048MB ram, Windows XP Professional version OS, and using the NNtool which is built-in the MATLAB 7.1 to train the BPN model.

We evaluate the performance of two experiments by the detection rate (DR), the false positive rate (FP) and the accuracy, according to the formulas (2), (3) and (4).

$$Detection\ Rate = \frac{Number\ of\ detected\ attacks}{Number\ of\ attacks} \times 100\% \qquad (2)$$

$$False\ Positive\ Rate = \frac{Number\ of\ misclassified\ connections}{Number\ of\ normal\ connections} \times 100\% \qquad (3)$$

$$Accuracy = \frac{Number\ of\ correct\ classified\ connections}{Number\ of\ connections} \times 100\% \qquad (4)$$

We find the performance in different learning rate, using Table 5. Experiment 1 has a learning rate of 0.5, and we see that the DR amounts to 99.81%. Its FP is merely 0.57%, while the whole accuracy amounts to 99.75%. In the part of experiment 2, we set the learning rate to 0.1, and to verify whether or not the lower learning rate could get a better convergence on BPN, we could see the DR, FP and accuracy are all the same with experiment 1, using the results of the experiment. Therefore, we know that by setting the learning rate to 0.5, the network gains a better convergence, so as to achieve better performance.

Table 5. The performance evaluation of IDS

	DR	FP	Accuracy
Experiment 1	99.81%	0.57%	99.75%
Experiment 2	99.81%	0.57%	99.75%

As we analyze each class of attacks in Table 6 and 7, to observe each individual performance, we see that the detection performance of the U2R is worst. This is because the training data of U2R are too less, and result in the low detection performance.

Table 6. Experiment 1—The table of detailed classification

Category of attacks	Amount of correct detection /Amount of sample	DR
Normal	2633/2648	99.43%
Probe	1358/1369	99.20%
DoS	10590/10591	99.99%
U2R	10/17	58.82%
R2L	366/375	97.60%

Table 7. Experiment 2－The table of detailed classification

Category of attacks	Amount of correct detection /Amount of sample	DR
Normal	2633/2648	99.43%
Probe	1358/1369	99.20%
DoS	10590/10591	99.99%
U2R	10/17	58.82%
R2L	366/375	97.60%

V. Conclusion and Future Work

In our research, we proposed an architecture of HIDS that apply to CWSN, to detect intrusion by CH. The proposed HIDS consists of an anomaly detection model and a misuse detection model. It filters a large number of packet records, using the anomaly detection model, and performs a second detection with the misuse detection model, when the packet is determined to intrusion. Therefore, it efficiently detects intrusion, and avoids the resource waste. Finally, it integrates the outputs of the anomaly detection and misuse detection models with a decision making model. This determines the presence of an intrusion, and classifies the type of attack. The output of the decision making model is then reported to an administrator for follow-up work. This method not only decreases the threat of attack in the system, but also helps the user handle and correct the system further with hybrid detection.

In this paper, we evaluate the performance of the misuse detection model, which is implemented by BPN though experiment. The simulation results present the performance of this method: the detection rate is 99.81%, the false positive rate is only 0.57% and its accuracy achieves 99.75%. We also find that the individual detection rate is very low when the training sample is not substantial. Therefore, the training samples must be a specific amount for the BPN to ensure the accuracy of classification.

The method of feature selection is an important factor, which affects the performance of IDS. We adopt the proposed method of feature selection by Jong now, but we can use other methods to select features in the future, such as data mining, to find identifiable features, instead of relying on the viewpoint of experts. Additionally, our rule-based method is also defined by the experiences and observations of experts. We can use a method, which has learning ability, and collocate with the selected features to provide our anomaly detection model with better performance and flexibility.

Acknowledgments

This work was supported in part by the Taiwan National Science Council under Grants NSC95-2221-E-324-019-MY3, NSC96-2221-E-324-021 and NSC97-2221-E-324-007-MY3.

References

[1] O. Depren, M. Topallar, E.narim and M.K. Ciliz, “An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks,” *Expert Systems with Applications*, 29(4), 2005, pp. 713-722.

[2] X. Du and F. Lin, “Improving routing in sensor networks with heterogeneous sensor nodes,” *Proceedings of IEEE 61st Vehicular Technology Conference*, 4, 2005, pp. 2528-2532.

[3] W.R. Heinzelman, J. Kulik and H. Balakrishnan, “Adaptive protocols for information dissemination in wireless sensor networks,” *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, 1999, pp. 174-185.

[4] W.R. Heinzelman, A. Chandrakasan and H. Balakrishnan, “Energy-efficient communication protocols for wireless microsensor networks,” Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, 2000, pp. 1-10.

[5] K. Jong, E. Marchiori, M. Sebag and A. van der Vaart, “Feature selection in proteomic pattern data with support vector machines,” *Proceedings of the Computational Intelligence in Bioinformatics and Computational Biology* (CIBCB’04), 2004, pp. 41-48.

[6] C. Karlof and D. Wagner, “Secure routing in wireless sensor networks: attacks and countermeasures,” *Ad Hoc Networks*, 1(2-3), 2003, pp. 293-315.

[7] R.A. Kemmerer and G. Vigna, “Intrusion detection a brief history and overview,” *Computer*, 35(4), 2002, pp. 27-30.

[8] S. Lindsey and C.S. Raghavendra, “PEGASIS: Power-efficient gathering in sensor information systems,” *IEEE Aerospace Conference Proceedings*, 3, 20025, pp. 3-1125.

[9] A. Manjeshwar and D.P. Agrawal, “TEEN: a routing protocol for enhanced efficiency in wireless sensor networks,” *Proceedings of 15th International Parallel and Distributed Processing Symposium*, 2007, pp. 2009-2015.

[10] A. Manjeshwar and D.P. Agrawal, “APTEEN: a hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks,” *Proceedings of the International Parallel and Distributed Processing Symposium*, 2002, pp. 195-202.

[11] A. Murali and M. Rao, “A survey on intrusion detection approaches,” *Proceedings of the First International Conference on Information and Communication Technologies*, 2005, pp. 233-240.

[12] Y. Qiao and X. Weixin, “A network IDS with low false positive rate,” *Proceedings of the 2002 Congress on Evolutionary Computation*, 2, 2002, pp. 1121-1126.

[13] D.E. Philippe, *Neural network models: theory and projects*, London ; New York : Springer, 1997

[14] W.T. Su, K.M. Chang and Y.H. Kuo, “eHIP: An energy-efficient hybrid intrusion prohibition system for cluster-based wireless sensor networks,” *Computer Networks*, 51(4), 2007, pp. 1151-1168.

[15] Y. Wang, G. Attebury and B. Ramamurthy, “A survey of security issues in wireless sensor networks,” I*EEE Communications Surveys & Tutorials*, 8(2), 2006, pp. 2-23.

[16] A. D. Wood and J. A. Stankovic, “Denial of service in sensor networks,” *Computer*, 35(10), 2002, pp. 54-62.

[17] http://kdd.ics.uci.edu/databases/kddcup99/kddcup