International Journal of Computer Science and Applications ©Technomathematics Research Foundation
Vol. 6, No. 3, pp 20 – 32, 2009

ATTACK CLASSIFICATION BASED ON DATA MINING TECHNIQUE AND ITS APPLICATION FOR RELIABLE MEDICAL SENSOR COMMUNICATION

Hayoung Oh

Dept. of Computer Science and Engineering,
Seoul National University, Seoul 151-744, Korea
hyoh@popeye.snu.ac.kr

Inshil Doh, Kijoon Chae

Dept. of Computer Science and Engineering,
Ewha Womans University, Seoul 120-750, Korea
isdoh@ewhain.net, kjchae@ewha.ac.kr

Detecting network intrusion has been not only important but also difficult in the network security research area. In Medical Sensor Network(MSN), network intrusion is critical because the data delivered through network is directly related to patients' lives. Traditional supervised learning techniques are not appropriate to detect anomalous behaviors and new attacks because of temporal changes in network intrusion patterns and characteristics in MSN. Therefore, unsupervised learning techniques such as SOM (Self-Organizing Map) are more appropriate for anomaly detection. In this paper, we propose a real-time intrusion detection system based on SOM that groups similar data and visualize their clusters. Our system labels the map produced by SOM using correlations between features. We experiments our system with KDD Cup 1999 data set because MSN data is not available yet. Our system yields the reasonable misclassification rates and takes 0.5 seconds to decide whether a behavior is normal or attack.

Keywords: *Real time Intrusion Detection System, Countermeasures, Supervised Learning, Unsupervised Learning, Correlations, Network Security, Medical Sensor Network*

1. Introduction

Computer network security has become increasingly important since intrusion attacks can easily cause huge damages. To protect networks, network intrusion detection systems are used for detecting intrusions and anomalous behavior [1]. Any kind of network intrusion should be monitored and detected for reliable communication.

Nowadays in our society, an increasing number of people have chronic medical conditions. If these people's health conditions could be monitored continuously and remotely, medical professionals could react to life-threatening situations much more quickly[17]. This is the reason that Medical Sensor Network(MSN) is an important issue in network communication. Mobile telemedicine allows patients to do their daily activities while they are monitored continuously anytime, anywhere[16]. Current monitoring solutions, however, are both cumbersome and costly[17]. Existing sensor network research has mainly focused on monitoring the physical environment. A humancentered sensor network has distinct features such as the sensitive nature of the data, the mobility of sensors, and the proximity to potential attackers, leading to various security challenges[17]. However, it is difficult to make an intrusion detection system since the amount of monitoring data generated is extensive, thus incurring large processing overheads. Also temporal changes in network intrusion patterns and characteristics often render existing classification-based intrusion detection data mining techniques ineffective [2][3][4][5][6].

Unsupervised learning techniques are more appropriate for anomalous behaviors and new attacks in a dynamic intrusion detection environment for accommodating the change in the characteristics of attacks especially in MSN. Unsupervised learning or clustering algorithms have been recently refocused on research areas [7]. The most important advantage of using clustering to detect attacks is the ability to find new attacks that have not been seen before. Clustering algorithms can group new data instances into coherent groups which can be used to increase the performance of existing classifiers. Grouping similar data together eases the task of labeling by experts.

SOM (Self-Organizing Map), one of the various clustering algorithms is a data visualization technique invented by Professor Teuvo Kohonen which reduces the dimensions of data through the use of self-organized neural networks. The way of SOM to reduce dimensions is by producing a map of usually 1 or 2 dimensions that plot the similarities of the data by grouping similar data items together [8].

Gonzales et al. compared the Neuro-Immune and SOM in terms of the classification rates. They showed their classification rates are similar and very correct [7]. Jirapummin et al. proposed the intrusion detection mechanism using SOM for clustering and RPROP (Resilient Propagating Neural Network) for labeling [9]. However, RPROP cannot accommodate a new untrained attack and its performance depends on the number of neurons in the map.

In this paper, we propose a real-time intrusion and anomaly detection system using SOM and our labeling method for reliable MSN. Our system labels the map produced by SOM using correlations between features. It classifies neurons as normal or attacks. In the case of attack neurons, it classifies them according to the types of attacks again. When a malicious behavior is caught, our system detects the type of the intrusion or a new untrained attack.

We describe our experimental data set in section 2, our intrusion detection system in section 3 and our experimental results in section 4. In section 5, we describe the MSN architecture where we can apply proposed data classification technique. Finally, the conclusion and future work are described in section 6.

2. The Experimental Data Set

We use a part of DARPA 1998 Intrusion Detection Evaluation data set used in the 3rd International Knowledge Discovery and Data Mining Tools Competition in 1999(KDD Cup 1999) [10] for our experiment because there is no MSN data set available so far. We think that if the amount of MSN data is sufficient, our mechanism will work well under the MSN environment. It is TCPdump data generated over nine weeks of simulated network traffic in a hypothetical military local area network. It includes some 7 million TCP connection records.

DARPA 1998 Intrusion Detection Evaluation data set consists of the labeled training data with about 5 million connections (KDD-TND) and the test data with 2 million connections (KDD-TD). The labels filtered during the unsupervised training are used to aid the interpretation and the test of the trained SOM. Each connection is detailed in terms of 41 features, categorized as follows: Basic TCP features, Content features, and Time- and Host-based Traffic features. Kayacik et al. used only the Basic features as an experimental data [11]. However, we employed the 9 Basic features (No.1 ~ 9 of Table 1) and 19 (Time and Host-based) Traffic features (No.10 ~ 28 of Table 1) since we estimated that results using Basic and Traffic features is better than those of Basic features and those of Traffic features through our simple experiments.

Table 1. Basic and Traffic Features

No.	Feature	Description
1	duration	length (number of seconds) of the connection
2	protocol	type of the protocol, e.g. tcp, udp, etc.
3	service	network service on the destination, e.g., http, telnet, etc.
4	flag	normal or error status of the connection
5	src_byte	number of data bytes from source to destination
6	dst_byte	number of data bytes from destination to source
7	land	1 if connection is from/to the same host/port; 0 otherwise
8	wrong_fragment	number of ``wrong" fragments
9	urgent	number of urgent packets
10	count	number of connections to the same host as the current connection in the past two seconds
11	srv_count	number of connections to the same service as the current connection in the past two seconds
12	serror_rate	% of connections that have ``SYN" errors
13	srv_serror_rate	% of connections that have ``SYN" errors
14	rerror_rate	% of connections that have ``REJ" errors
15	srv_rerror_rate	% of connections that have ``REJ" errors
16	same_srv_rate	% of connections to the same service
17	diff_srv_rate	% of connections to different services
18	srv_diff_host_rate	% of connections to different hosts
19	dst_host_count	number of connections to the destination host as the current connection in the past two seconds
20	dst_host_srv_count	number of connections to the destination service as the current connection in the past two seconds
21	dst_host_same_srv_rate	% of connections to the same service at destination host
22	dst_host_diff_srv_rate	% of connections to the different

		service at destination host
23	dst_host_same_src_port_rate	% of connections to the same source ports at destination host
24	dst_host_srv_diff_host_rate	% of connections to the different host at destination host
25	dst_host_serror_rate	% of connections that have ``SYN" errors at destination host
26	dst_host_srv_serror_rate	% of connections that have ``SYN" errors at destination host
27	dst_host_rerror_rate	% of connections that have ``REJ" errors at destination host
28	dst_host_srv_rerror_rate	% of connections that have ``REJ" errors at destination host

We also consider the 10 types of attacks. There are DoS attack such as Back, Neptune, Pod, Smurf and Teardrop and probing attack such as Ipsweep, Nmap, Portsweep and Satan.

3. The SOM-based intrusion detection mechanism

Figure 1 show our system with three steps as follows: Training, Labeling and Detection & Training. Training means creating the map based on the preprocessed and normalized data using SOM. Labeling means classifying clusters from the trained map based on the traffic feature correlations. Finally, Detection & Training means the real-time intrusion detection and the continuous training to adapt our system to new data.

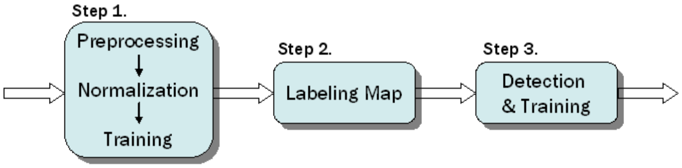


Figure 1. The SOM-based Intrusion Detection Mechanism

Training

The accurate training requires the modification of data because the values of traffic features have the various ranges. Our mechanism has the preprocessing stage and the normalization stage before training.

Preprocessing

It transforms the TCPdump data into the numeric data because SOM resolves only numerical data but the TCPdump data has some features whose data types are not numeric. A transform example shows in Table 2.

Table 2. Transformation of the 'Protocol' Field Data

Protocol	Numeric
ICMP	1
TCP	6
UDP	17

Normalization

SOM makes the maps for each feature respectively and then construct the U-matrix (unified matrix) based on the all feature maps. Every feature in Table 1 has such various ranges that the values of the feature *src_byte* are in the range between 0 and 2194619, the feature *duration*'s values are in the range between 0 and 42448 and some feature's values are in the range between 0 and 1. Some wide range features such as *src_byte* and *duration* affect the U-matrix construction much more than any other features. Thus normalization is needed to make U-matrix reflecting all features fairly.

Our normalization makes the minimum values of every feature 0 and their maximum values 1. All the normalized values of each feature are in the range of 0 and 1($0\leq N_{i(x)} \leq 1$). The normalization equation used in this paper is as follows:

$$N_{i(x)}=(i(x)-V_{min}(x))/(V_{max}(x)-V_{min}(x)),$$

where x means one of 28 features, $i(x)$ is the original data value of the feature x ($V_{min}(x) \leq i(x) \leq V_{max}(x)$), $V_{min}(x)$ is the minimum value of the feature x and $V_{max}(x)$ is the maximum value of the feature x.

Training

SOM is not only the clustering model using the neural network method but also the unsupervised learning model. Figure 2 shows the difference of the supervised learning and the unsupervised learning. The unsupervised learning in (b) of Figure 2 automatically categorizes the varieties of input presented during training and can then express how sell new inputs fit the patterns it has discerned while the supervised learning in (a) of Figure 2 has determined what output he desires for each of many possible input values.

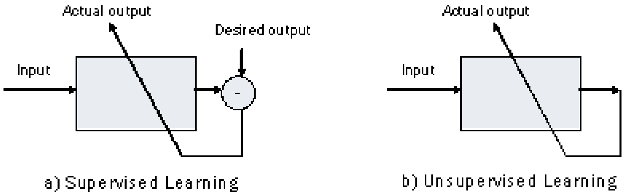
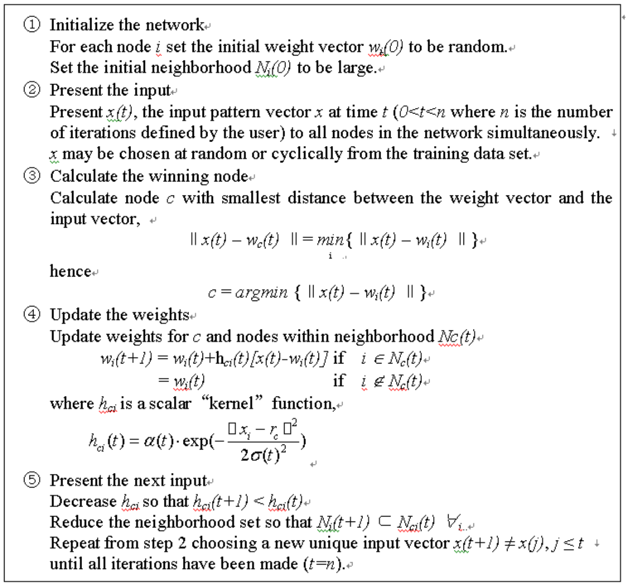


Figure 2. Supervised Learning vs. Unsupervised Learning

The 5-step algorithm of SOM [12][13][14] is described in Figure 3 and can build the U-matrix for KDD-TND in Figure 4. The U-matrix has 42 X 27 neurons and the lighter color neurons mean the borders dividing clusters.



① Initialize the network
For each node i set the initial weight vector $w_i(0)$ to be random.
Set the initial neighborhood $N_i(0)$ to be large.

② Present the input
Present $x(t)$, the input pattern vector x at time t ($0<t<n$ where n is the number of iterations defined by the user) to all nodes in the network simultaneously.
x may be chosen at random or cyclically from the training data set.

③ Calculate the winning node
Calculate node c with smallest distance between the weight vector and the input vector,

$$\| x(t) - w_c(t) \| = \min_i \{ \| x(t) - w_i(t) \| \}$$

hence

$$c = argmin \{ \| x(t) - w_i(t) \| \}$$

④ Update the weights
Update weights for c and nodes within neighborhood $Nc(t)$

$$w_i(t+1) = w_i(t) + h_{ci}(t)[x(t) - w_i(t)] \text{ if } i \in N_c(t)$$
$$= w_i(t) \text{ if } i \notin N_c(t)$$

where h_{ci} is a scalar "kernel" function,

$$h_{ci}(t) = \alpha(t) \cdot \exp(-\frac{\square x_i - r_c \square^2}{2\sigma(t)^2})$$

⑤ Present the next input
Decrease h_{ci} so that $h_{ci}(t+1) < h_{ci}(t)$
Reduce the neighborhood set so that $N_i(t+1) \subset N_{ci}(t) \ \forall_i$
Repeat from step 2 choosing a new unique input vector $x(t+1) \neq x(j), j \leq t$
until all iterations have been made ($t=n$).

Figure 3. SOM Algorithm to Construct Maps

Labeling

It is difficult deciding which SOM cluster is normal or abnormal and which attacks the abnormal input is because feature maps and U-matrix give no information about input data (refer Figure 4). To solve this problem, correlations between features on each attack are analyzed. Pearson correlation coefficient equation used to analyze them is as follows [15]:

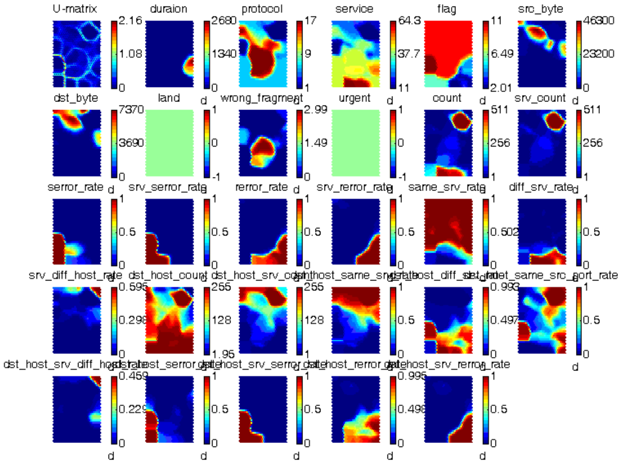


Figure 4. U-matrix and 28 Feature Maps

$$r_{xy} = \frac{\sum (x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum (x_i - \overline{x})^2}\sqrt{\sum (y_i - \overline{y})^2}}$$

where

$$\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i , \overline{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$$

and $(x_i, y_i) \in \{(x_1, y_1), \cdots, (x_n, y_n)\}$

Table 3 shows the correlation results between features on each attack. For instance, the attack Pod has the 3 feature-pair correlations such as (*21*+*23*) and {*3*-(*5*+*8*)}. (21+23) means the positive correlation between feature no. 21(the percentage of connections to the same service at destination host) and 23(the percentage of connections to the same source ports at destination host). {3-(5+8)} means the minus correlations between feature no. 3(the network service types on the destination) and 5(the number of data bytes from source to destination) and between 3 and 8(the number of "wrong" fragments) and the positive correlation between 5 and 8. Figure 5 shows U-matrix and Feature Maps only for the attack Pod where the color distribution and shape of the circled feature map for the feature no. 5 are very similar to those of 8 but are opposed to those of 3 and the maps for feature no. 21 and 23 are much alike in the color distribution and shape.

Table 3. Correlations between features on each Attack

Figure 1. Normal	Figure 2. =(*10*+*11*)∩(*12*+*13*)∩(*14*+*15*)∩(*16*-*17*)∩(*20*+*21*)
Figure 3. Neptune	Figure 4. =(*11*+16)
Figure 5. Smurf	Figure 6. =(*10*+*11*)∩(*19*+*20*)∩{(*21*+*23*)-*22*}
Figure 7. Teardrop	Figure 8. =(*20*+*21*+*23*)
Figure 9. Back	Figure 10. =(*12*+*13*)∩(*25*+*26*)∩(*27*+*28*)
Figure 11. Pod	Figure 12. =(*21*+*23*)∩{*3*-(*5*+*8*)}
Figure 13. Ipsweep	Figure 14. =(*3*+*19*+*22*)-(*21*+*23*)
Figure 15. Nmap	Figure 16. ={(*4*+*12*+*13*+*26*)-*21*}
Figure 17. Portsweep	Figure 18. =(*22*+*23*+*27*)∩(*15*+*28*)
Figure 19. Satan	Figure 20. =(*10*+*22*)∩(*14*-*23*)∩(*15*+*28*)

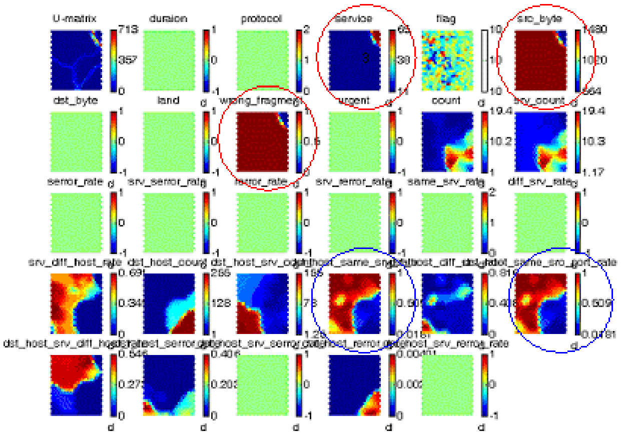


Figure 5. U-matrix and Feature Maps for Pod attack

Figure 6 shows results analyzing which cluster in U-matrix is abnormal or normal through the correlation results between features on each attack in Table 3. For instance, the most left and center cluster means the attack Pod. The largest part not circled means normal clusters [2].

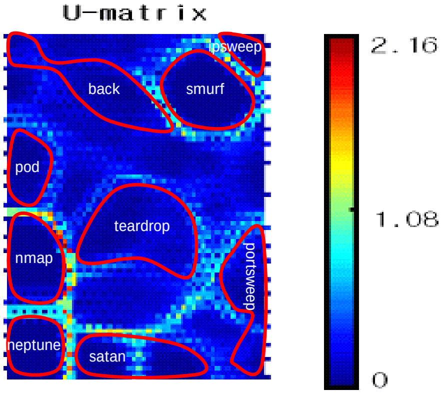
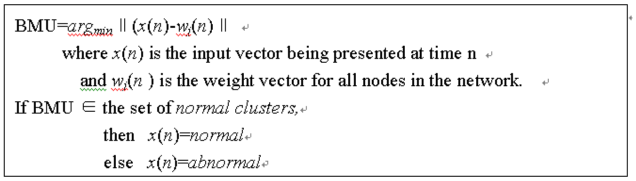


Figure 6. Correlation results between features on each attack in Table 3

Detection and Training

Real time intrusion detection is achieved by finding BMU (the best matching unit) with the smallest Euclidean Distance measured between input data and the map unit. The process of finding BMU is the process in the stage 3 of Figure 3. If BMU is a cluster in the normal cluster set, the input is normal. Otherwise, it is abnormal. Our detection process is described in Figure 7.



```
BMU=$arg_{min}$ ‖ ($x(n)$-$w_i(n)$) ‖
    where $x(n)$ is the input vector being presented at time n
      and $w_i(n$ ) is the weight vector for all nodes in the network.
If BMU ∈ the set of *normal clusters,*
          then  $x(n)$=*normal*
          else  $x(n)$=*abnormal*
```

Figure 7. Intrusion Detection Algorithm using Euclidean Distance

After the detection process, weights of BMU and its neighbors are updated based on the equation as follows:

$$w_i(n+1) = w_i(n)+\mathrm{h}_{ci}(n)[x(n)-w_i(n)]$$

This computation is the process in the stage 4 of Figure 3. U-matrix becomes updated as time goes.

Figure 8 shows the process of our intrusion detection mechanism. ○ ,1 in Figure 8 means the training step, ○ ,2 in Figure 8 means the labeling step and ○ ,3 in Figure 8 means the real-time detection and training.

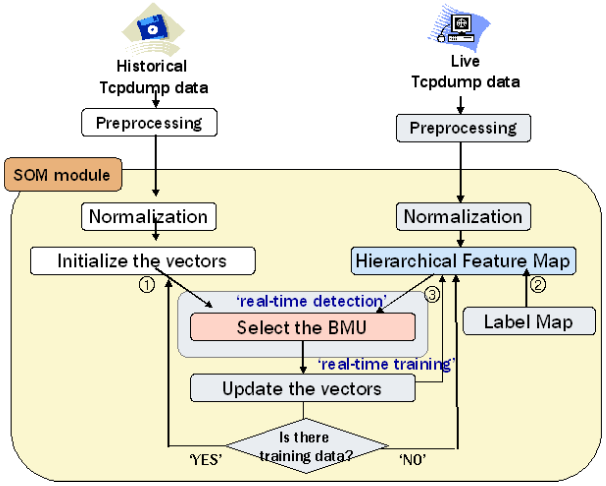


Figure 8. Process of Our Intrusion Detection System

4. Experimental Results

As described chapter 2, we use two subsets of KDD Cup 1999 data for training (KDD_TND) and test(KDD_TD). We consider Back, Neptune, Pod, Smurf, teardrop, Ipsweep, Nmap, Portsweep and Satan as attacks. 70% of KDD_TND is used for training and 30% of KDD-TND and KDD-TD are used for test of anomaly detection and misuse detection. Misuse detection trains the data including normal and all considering attacks and catches intrusions in terms of the characteristics of known attacks. We train the mixture of normal and attacks and build and label the U-matrix to evaluate the performance of misuse detection in our system. Then we collect the results of BMU for test data in our system. Table 4 shows the misclassification rates in terms of the misuse detection. Since the misclassification means that our system selects the wrong BMU, the lower rate is the better. The rate of 30% KDD_TND is lightly lower than that of KDD_TD because 30% of KDD_TND is familiar with 70% of KDD_TND and U-matrix is more adjustable to the characteristics of 30% of KDD_TND. If our system collects and train the data at the network installed, the result is similar to that of 30% of KDD_TND.

Table 4. Misuse Detection

Data	Misclassification rate
30% of KDD_TND	0.03
KDD_TD	0.05

Anomaly detection catches intrusions of the new attack not trained. Our system is trained for 70% of KDD_TND without only one attack among 9 attacks. In the case of the anomaly detection for the attack Back, our system is trained for the mixture of normal and attacks without Back and test whether our system decide that attack Back is abnormal. Table 5 shows the misclassification rates of 9 attacks respectively for 30% KDD_TND and those rates for KDD_TD. Since the anomaly detection of the attack Back is perfect, its characteristics are clear while those of the attack Smurf are a little dim.

Table 5. Anomaly Detection

Data		Misclassification rate
30% of KDD_TND	Back	0.000
	Neptune	0.050
	Pod	0.054
	Smurf	0.092
	Teardrop	0.045
	Ipsweep	0.057
	Nmap	0.035
	Portsweep	0.058
	Satan	0.049
KDD_TD	Back	0.002
	Neptune	0.068
	Pod	0.059
	Smurf	0.099
	Teardrop	0.071
	Ipsweep	0.067
	Nmap	0.049
	Portsweep	0.069
	Satan	0.058

The detection time means the average time taken to decide whether the given input is normal or abnormal after training. The detection time is 0.5 seconds in our whole experiments, which means our system can be called the real-time intrusions detection system.

5. Application to Medical Sensor Network

In MSN, a mobile patient can communicate with hospital data center and/or physician through wireless networks (e.g., cellular and sensor networks). And during the communication, a large amount of data should be delivered through the intermediate sensor nodes. As mentioned previously, medical sensor data is very important because it is the data about patients' health. When any kind of attack exists, it should be detected quickly and correctly. Otherwise, MSN will be collapsed.

Figure 9 shows the general architecture of MSN. Among the processes, Data Relay by Sensor nodes can be influenced by various attacks. Because of the basic characteristics of MSN it is not easy to supervise the nodes if they operate properly or not. Especially, in sensor network, nodes can be added or removed randomly. In this environment, proposed attack classification through unsupervised learning data mining mechanisms is proper to apply. When some attacks are detected, we can replace the nodes under attack or take proper measures to the nodes, and make the data communication reliable. We believe that because MSN allows patients to do their daily activities while they are monitored continuously anytime, anywhere, our proposed unsupervised learning mechanism for attack detection mechanism is well applied.

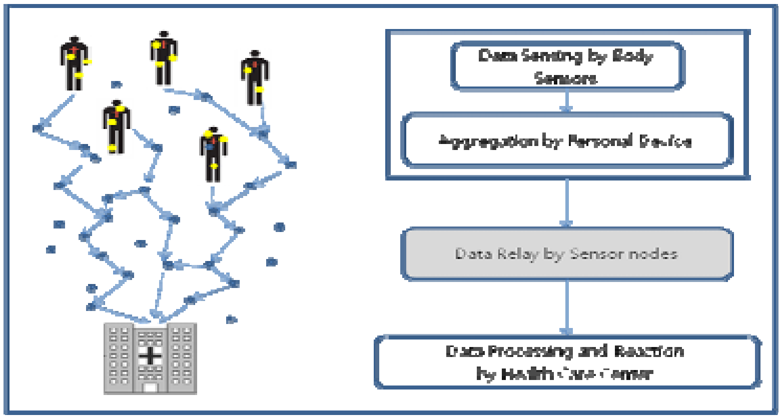

Data Sensing by Body Sensors
Aggregation by Personal Device
Data Relay by Sensor nodes
Data Processing and Reaction by Health Care Center

Figure 9. Medical Sensor Network architecture

6. Conclusion

As network technologies have evolved and the number of Internet users increase rapidly, security and detection techniques against attacks are required very urgently. For Medical Sensor Network, security and attack detection are much more important. Since these attacks are diffused increasingly using worm virus and the intellectual attack tool, their damage will be getting more serious. Therefore, it is important to detect those attacks as soon as possible. In MSN application, small, lightweight wireless medical sensors are attached to patients and deployed in large area between the patients and Health Care Center. And in these areas, the communication is especially vulnerable to various attacks. So we proposed attack detection mechanism applying data mining techniques.

We used KDD Cup 1999 Data for training and testing and used SOM, one of unsupervised learning data mining mechanisms for our proposed intrusion detection mechanism. It consists of 3 steps. U-matrix clustering traffics with similar characteristics is built through the initial learning process. Next, we labeled U-matrix for determining whether a cluster is normal or abnormal using correlations between features. Finally, our system decided whether an input matched attack clusters or not and learned it continuously. Our system yielded the reasonable misclassification rates through several experiments.

We got the characteristics of each attack whose unclassifiable features look to have no relations among themselves. Our feature correlation results can be used in other intrusion detection systems using other technologies even if a new attack happens because our correlation information is adjusted as time grows. We will analyze attacks with unclear characteristics such as Smurf, extract important features to reduce the process overhead and then make our system more accurate. We believe that our proposal can be used to detect various attacks in Medical Sensor Network and make the sensor communication reliable.

Acknowledgments

“This work was supported by the Korea Science and Engineering Foundation(KOSEF) grant funded by the Korea government(MEST) (NO. R01-2008-000-20062-0).”

References

[1] John E. Canavan, "Fundamentals of Network Security", Artech House, pp. 21-47, 261-278, 2001.

[2] Nong Ye, Xiangyang Li, "A Scalable Clustering Technique for Intrusion Signature Recognition", Proceedings of 2001 IEEE Workshop on Information Assurance and Security, 2001.

[3] Jack Marin, Daniel Ragsdale, John Shurdu, "A Hybrid Approach to the Profile Creation and Intrusion Detection", Proceedings of DARPA Information Survivability Conference and Exposition, IEEE, 2001.

[4] Christopher Kruegel, Darren Mutz, William Robertson, Fredrik Valeur, "Bayesian Event Classification for Intrusion Detection", Reliable Software Group University of California.

[5] A. A. Sebyala, T. Olukemi, L. Sacks, "Active Platform Security through Intrusion Detection Using Naive Bayesian Network for Anomaly Detection", In London Communications Symposium, 2002.

[6] Seong Soo Kim, A. L. Narasimha Reddy, Marina Vannucci, "Detecting Traffic Anomalies at the Source through aggregate analysis of packet header data", TAMU-ECE, 2003.

[7] Fabio Gonzalez, Dipanker Dasgupta, "Neuro-Immune and Self-Organizing Map

Approaches to Anomaly Detection: A comparison", ICARIS, 2002.
[8] Tom Germano, "Self Organizing Maps", Available in http://davis.wpi.edu/~matt/courses/soms/.
[9] Chaivat Jirapummin, Naruemon Wattanapongsakorn, Prasert Kanthamanon, "Hybrid Neural Networks for Intrusion Detection System", King Mongkut's University of Technology Thonburi, 2001.
[10] KDD Cup 1999 Data, Available in http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.
[11] H. Gunes Kayacik, A. Nur Zuncir-Heywood, Malcolm I. Heywood, "On the Capability of an SOM based Intrusion Detection System", International Joint Conference on Neural Networks, 2003.
[12] Binh Viet Nguyen, "Self-Organizing Map for anomaly detection", Available in http://www.cs.umd.edu/~bnguyen/papers/papers.html
[13] SOM Toolbox for Matlab, Available in http://www.cis.hut.fi/projects/somtoolbox/
[14] Juha Vesanto, John Himberg, Esa Alhoniemi, and Juha Parhankangas, "SOM Toolbox for Matlab 5", SOM Toolbox Team, Helsinki University of Technology, 2000.
[15] Pearson Correlation Coefficient, Available in http://www.indstate.edu/nurs/mary/N322/ pearsonr.html/
[16] Narasimha Challa, Hasan C¸ am, and Madhur Sikri, "Secure and Efficient Data Transmission over Body Sensor and Wireless Networks", EURASIP Journal on Wireless Communications and Networking Volume 2008, Article ID 291365.
[17] Kriangsiri Malasri, Lan Wang, "Addressing Security in Medical Sensor Networks", HealthNet’07 ACM, June, 2007.