Real-Time Neural Network-Based Network Analyzer for Hotspot Area

Rahmadya Trias Handayanto, Haryono, and Jarot Prianggono
Laboratory of Software
Faculty of Engineering, Islam 45 University
Email: rahmadya_trias@yahoo.com, h4ryon0@yahoo.com, Jarotprianggono@yahoo.com

***Abstract*— We present a real-time neural network-based network analyzer system for hotspot area. There are many applications that available in the market today for provide us the network graph of our hotspot areas. These graphs will be analyzed by a network administrator. Because a hotspot area often runs 24 hours, an administrator has a difficulty to monitor the traffic all the time. Therefore we proposed the automatic system to help a network administrator in monitoring the network. This system will replace human skill in interpreting the graph with an Artificial Neural Network System. To minimize the number of input vector we use mean value of axis, so the micro-computer e.g. notebook, laptop, PDA, and other gadgets can handle this system. Testing result showed this system could classify between normal, high and un-normal traffic of network graph periodically.**

I. INTRODUCTION

In this paper, we present a real-time neural network-based algorithm for identifying network condition of hotspot .[1] The identification runs automatically that follow the network traffic graph generation from network sniffer and automatic capture application. This system will help network administrator to monitor network traffic by inserting this system into network.

A hotspot area is a region that offers Internet access (usually free) over a wireless local area network (wi-fi) through the use of router, modem, and access point. Hotspots typically use Wi-Fi technology that support not only computer but also other small devices such as PDA, cellular phone, tablet PC, etc [14,15].

Based on signal delivery, hotspot area can be included as the broadcast system. Its mean the signal from one node will spread to others nodes. The numbers of users who access the hotspot area affect the availability of the network. If one user continuously download or upload some files, it will be simply create a high traffic network. Many open source software for snipping the hotspot area are available free from the internet. Instead of using it for crime, we can use it for monitoring purpose [2].

Viruses and Denial of Service (DoS) are the main problem of every network administrators. They could easily down the network by sending a lot of packet regularly to the network or server. It is a network administrator's task to investigate and make a decision when a client sends and downloads a threat. Many softwares available in the market for finding un-normal traffic condition for example SNORT, TCPDUMP, Network Flight Recorder (NRF), etc [1]. We can compare kind of network sniffers like some research by e.g. Clincy and Abu-Halaweh [2]. Instead of text-based system, our proposed system based on artificial neural network (NNs) algorithm which is a kind of soft computing.

There are many machine learning systems for classification e.g. fuzzy inference system (FIS), adaptive neuro-fuzzy inference system (ANFIS), Support Vector Machine (SVM), etc. They all sometimes called nonparametric classifier [12]. We used NNs because this kind of machine learning is simple, especially the backpropagation learning algorithm. Although NNs need a lot of memories, in our propose system we made some modification that will be discussed in section IV. Moreover, some machine learning algorithms e.g. SVM, FCM, and so on, need a lot of data samples for good result in learning because it based on statistics [12, 13]. We refuse to use FIS because it needs creating rules. May be the ANFIS can be used, but we did not like our neurons force to be rules instead of transfer functions.

Some hotspot areas rely on a router (e.g. mikrotik) in making a bandwidth management for each client. But the DoS attack and some network viruses are big problems because of the broadcast characteristic of wireless LAN. By using a network sniffer we can do an IT forensic task after getting a note from our proposed system.

1. Demonstration of this system can be seen at http://www.youtube.com/watch?v=IyOn5RdctOk.

Rahmadya Trias Handayanto, Haryono, and Jarot Prianggono are with Department of Engineering, Islam 45 University, Cut Meutia Avenue, No. 83, Bekasi - Indonesia.

II. DESCRIPTION OF THE SYSTEM

Like other supervised learning machines, our system consists of three stages: Learning stage, identification stage, and forensic stage. Both of learning and identification stages have a digital image processing system to convert images to matrix that have size 20 x 60. The forensic stage is optional because it is only done if there was un-normal network condition; moreover, it only use network sniffer alone without our proposed system.

We used a network sniffer to capture our network graph. A lot of kinds of sniffing tools are available in the market and most of them are free to download because of open source software. Figure 1 shows the wireshark sniffer (after the old name ethereal) when capturing the network graph of our hotspot areas. Firstly, we must set the wireshark for consistency, e.g. a scale of ordinate both for learning and verifying. We can use scale 100 or 10 packet per second and avoid using "auto" scale.

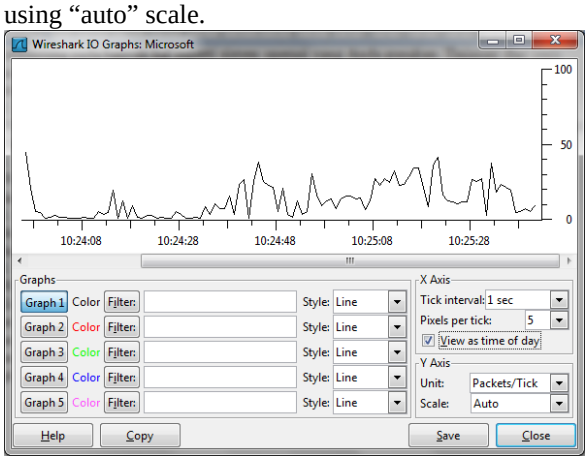


Fig. 1. Traffic captured by Wireshark Software

When the wireshark captures the network graph continuously, we capture periodically its image using an automatically capture software (Shown in Fig 2). We used Quick Screenshot Maker.

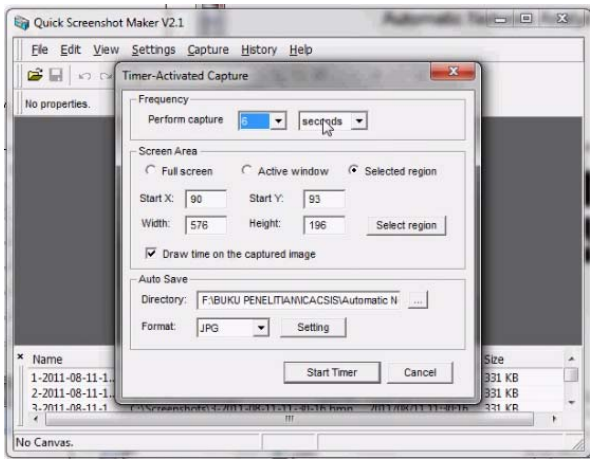


Fig. 2 Timer-Activated Capture of Network's Graph

After selecting the region of capture object, we start the timer and this application captures periodically the graph in that region. In implementation we may set timer longer than in testing e.g. one minute or two.

The traffic capture has been done with wireshark and some images has captured by automatic capture software, in this research we try to proposed the system for monitoring by interpreting of our network graph. We used Matlab even though many strong programming language available such as C++, Java, etc. For good interaction with user, Graphic User Interface (GUI) has been made (see figure 3). By clicking the start button, the system running periodically and will analyzing the network graph.

The system will stop automatically when there is no traffic captured in its current directory. We must save the image in the same directory of our application. Ideally we set time when capturing by automatic capture software the same time as when analyzing by Matlab application. If we think it is difficult, let the time when capturing an image quicker than when analyzing.

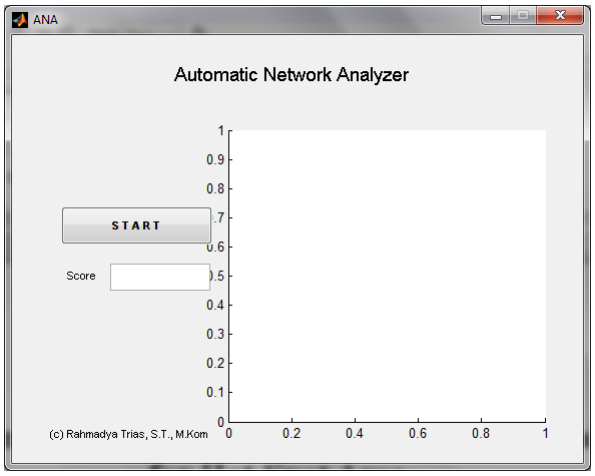


Fig. 3 GUI of Our Proposed System

After analyzing, this application send a message to a network administrator with sound in order to get good interaction. We also can add some action together with message, for example send a SMS to network administrators' cellular phone, reset a hotspot area, etc.

A. *Learning Stage*

Our proposed system use Artificial Neural Network algorithm with supervised learning system. The system has been trained by using backpropagation learning algorithm in order to make our system have knowledge to classify traffic into three categories: Normal, High Traffic, and un-normal condition. This learning algorithm, found by Hopfield, is a famous learning algorithm that combine forward and backward process [3].

Because we use network traffic graph for

analyzing, we must convert it into vector before send to the artificial neural network system. The graph resize to 100 by 300 after some process (gray, enhance, cleaning, block processing, and cropping). The final image processing has matrix 20 x 60 (see Figure 4).

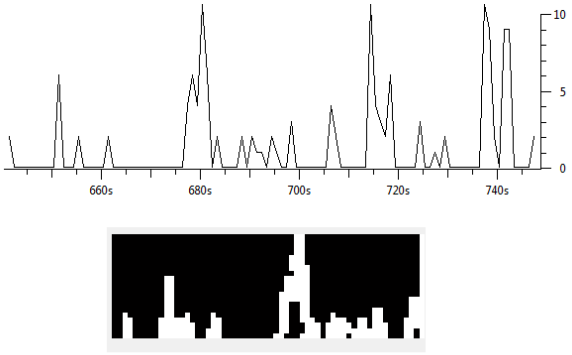


Fig. 4. Traffic before image processing (above) and after digital image processing (below).

In fig 4, we minimize the resolution of image in order to make learning process easier. But we still get feature extraction of that image. By this size, every computer, even a notebook, can be trained a backpropagation. We must fix our ordinate in 10 or 100 packet/s scale, and for our purpose we set 10 packet/s.

For throwing away the axis, we use crop function. So the system do not false in interpret a network graph. Figure 5 shows the crop area of image.

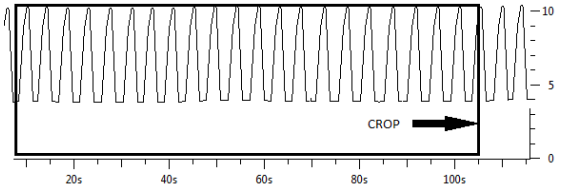


Fig. 5. Croping mechanism for throwing away noise (axis and ordinate of traffic's image).

Pattern recognition is one of artificial intelligent topic and many research papers can be found [4,5,6]. Our research, typically like signature identification and every image convert to vector as an input to artificial neural network system.

A lot of literatures can be found about digital image processing with matlab [7], and about its basic theory with pattern recognition [8]. If there are not any problems at this step, we continue to training step by backpropagation learning algorithm.

The learning stage is the heart of our system because it trains the system for getting an ability to identify kind of traffic being analyzed. Firstly, we choose the input of learning. This is a matrix contain a vector that represent of kind of traffic (normal, high traffic, or un-normal). Because our system used supervised learning method, we must choose a target vector that presents each kind of classification (normal, high, and un-normal condition) according to input matrix (see Figure 6).

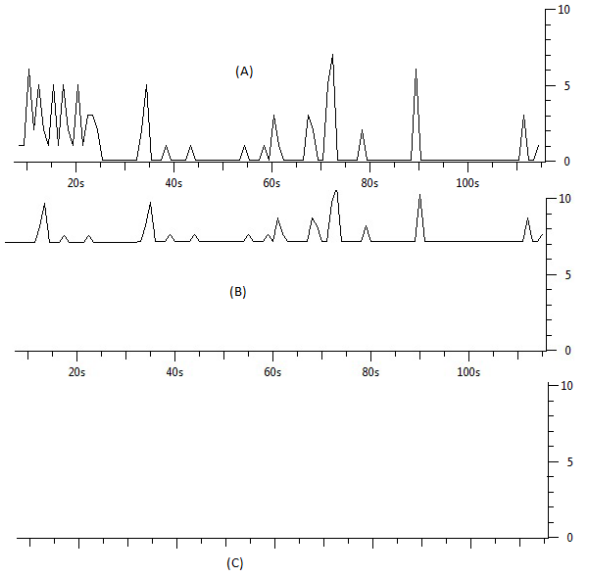
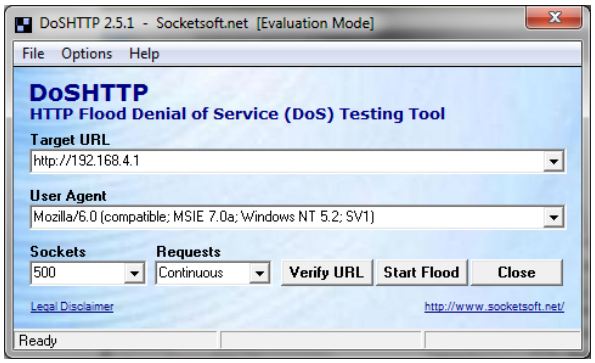


Fig 6 Kinds of Traffic Network for Training, a) Normal, b) high traffic, and c) Un-normal

Normal condition happen if we saw the traffic tend to lower bound of the graph. The high traffic condition if we saw traffic tends to upper bound. The un-normal condition happen if we do not see any traffic in our graph because of failure of hotspot area or the traffic is so high outside the range of graph.

There are many DoS attack software available in the market for testing purpose only (in USA this activity DoS activity is illegal). Figure 7 shows impact of DosHTTP, software for creating DoS attack (http://www.socketsoft.net).



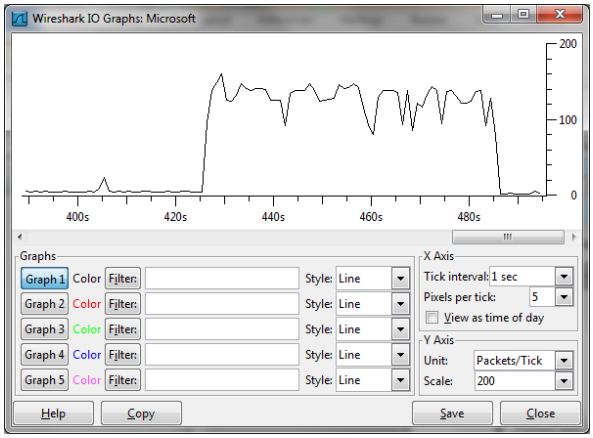


Fig 7 Software for creating DoS attack which set for 30 second (above) and the effect shown by wireshark (below).

Dos attack is very dangerous because it cause the network suddenly down. This kind of attack must be noticed by a network administrator. As figure 7 shows, there is an increase 15 times to normal condition of traffic after the DoS Attack.

We used five samples for each traffic condition (normal, high, and un-normal) and made them into single input matrix after image processing. It consisted of fifteen vectors because the image traffic for training were fifteen (five for each condition).

Then we must train it by supervised learning with target vector:

target = [1 1 1 1 1 5 5 5 5 5 10 10 10 10 10]. Its mean that column one to five in matrix input train to have answer one, column six to ten have the answer five, and column 11 to 15 have the answer five. Column in matrix target represent the condition of traffic. After training process we get an network for testing some input. If the result closes to one, the traffic closes to normal condition. If the answer is 5, the traffic closes to high traffic condition, and if the answer is 10, the traffic close to un-normal condition. After training process, the network has weights and biases that has an ability to classify the testing traffic weather normal, high traffic or un-normal.

For simply, we use Matlab toolbox of NNs (nntool) with parameters shown in Table 1.

TABLE I
NEURAL NETWORK PARAMETERS

No.	Parameter	Type
1	Network Type	Feed-forward Backpropagation
2	Training Function	levenberg marquardt
3	Transfer Function	Tangent Sigmoid
4	Number of Neuron	20
5	hidden layer	one

Those parameters are filled in neural network toolbox and then the network was trained. Figure 8 shows the neural network diagram was created by Matlab software. Another way is using command, but toolbox is easier than command; moreover, for programming purpose Matlab has an ability to convert toolbox into script easily.

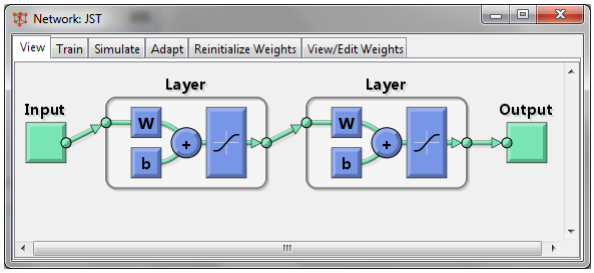


Fig. 8. Neural Network Diagram which Ready for Training

The training result can be seen at Figure 9 with parameters such as epochs, time, performance, gradient, etc. We can also see the graph of learning process, training state, and regression. The result of our neural network training is weights and biases.

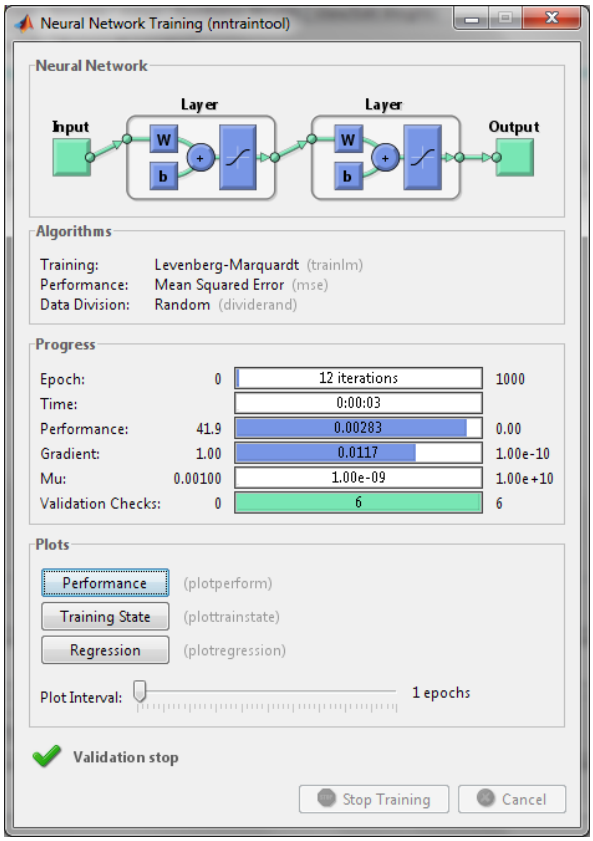


Fig. 9. Neural Network Training Results

The network which has been trained must be save in a file using the extension “mat”. This important file will be used when analyzing a traffic after image processing system process using function “sim” in Matlab.

B. *Identification Stage*

Different from verification with yes or no condition, in identification the system must choose what kind of network is. So the system must have ability to discriminate the sample being tested. The flowchart shown in Figure 8 shows flow of our proposed system.

After starting the wireshark, we open the network graph option in order ready to be captured by automatic capture software. Because the image is saved in the same directory as Matlab, when it open, the application could find the image and start to process it. Then, the vector feed to artificial neural network engine for classifying. The message send to user and the process back to getting file again before the user stop the process or there are not any images available anymore.

Image vector is tested using NNs algorithm (by sim function) for getting the result. There are three classes: near to one (normal), near to five (high traffic) or near to ten (un-normal condition). Figure 10 shows our proposed system flowchart.

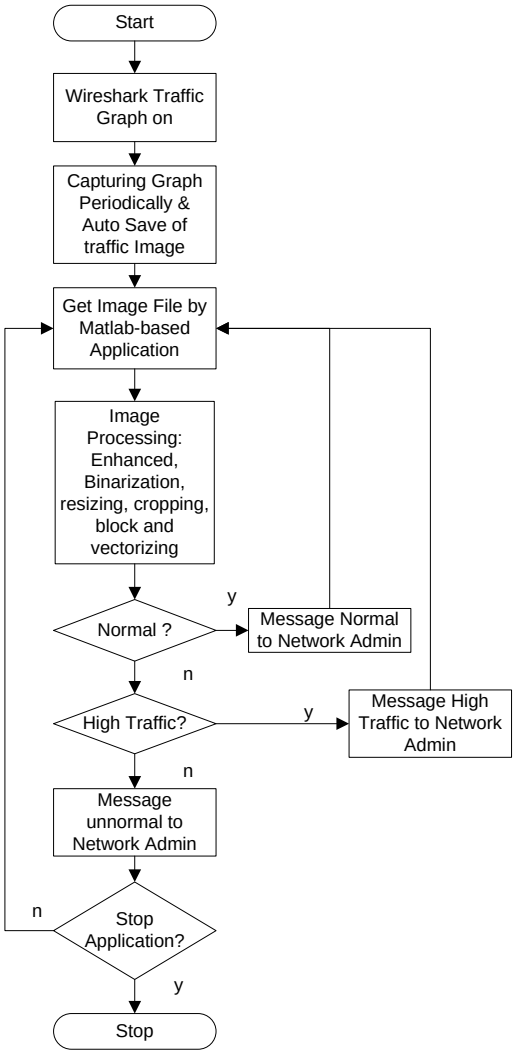


Fig. 10. Flowchart of Identification System. This flowchart only shows of identification process in brief manner.

When identifying process, the system must has ability to identify the traffic weather normal, high traffic, and un-normal or other classify acording to target in learning process.

In order the application runs periodically, we used timer script.

```
t = timer('TimerFcn', 'stat=false; disp("Timer!")',...
          'StartDelay',4);
start(t)
stat=true;
while(stat==true)
<script of image processing and neural network simulation is here>
End
```

We can set the period more instead of just 4 seconds. Additional script must be presented in the script, such as rename and delete image file after analyzing for efficiency of storage usage. Some software for converting a bath file to exe is available in the market even in freeware.

C. *Forensic Stage*

Forensic stage is the stage if there is an accident. The accident may be affect to vulnerability because attacker accessing ther root access [9]. In our system, forensic can be done by searching who are log in to our hotspot areas by wireshark software when our system notice high traffic and un-normal condition.

Figure 9 shows the ability of wireshark to sniff a hotspot area to trace the packet. May be this stage run only to know how much client that connect to our hotspot areas. If small number of client consume a lot of bandwidth, must be there were something wrong to our hotspot areas.

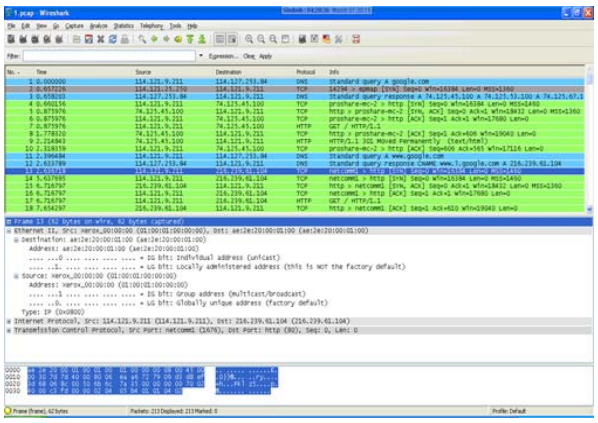


Fig. 9. Checking Packet by Wireshark

Forensic is not a daily task, it only run if there is an accident to our system. But with our proposed system we can warn the network administrator to check the network because there is a strange kind of network

traffic. Of course a network administrator could not monitor the network 24 hours, so our propose system can help him for monitoring the hotspot area.

The source of forensic is a log file. In wireshark every packets that send and receive can be traced back. After looking at a warning message, the network administrator saves the wireshark capture into a log file using the extension “pcap”. This is a network administrator’s task and out of our proposed system’s task. Sometimes the administrator just want to know the client who attacked the hotspot area or from where some viruses were spread.

.

III. Experimental Results

We use 10 samples for each kind of traffic that was not been used at training. We note the answer about kind of traffic by result of network’s simulation (from function “sim” in Matlab software). That number then compare to target for finding an error. For example if the result of testing is 5 for high traffic then the error is zero per cent. Table 1 shows the experimental results.

TABLE II
Experimental Results

No	Traffic	Result	Sim Result	Similarity
1	Normal	1,06	"Normal"	0,06
2	Normal	1,35	"Normal"	0,35
3	Normal	1,27	"Normal"	0,27
4	Normal	1,39	"Normal"	0,39
5	Normal	1,47	"Normal"	0,47
6	Normal	1,55	"Normal"	0,55
7	Normal	1,51	"Normal"	0,51
8	Normal	1,51	"Normal"	0,51
9	Normal	1,04	"Normal"	0,04
10	Normal	1,03	"Normal"	0,03
11	High	8,24	"Un-normal"	3,24
12	High	2,02	"High"	2,98
13	High	7,56	"High"	2,56
14	High	5,64	"High"	0,64
15	High	8,76	"High"	3,76
16	High	9,64	"High"	4,64
17	High	9,07	"Un-normal"	4,07
18	High	9,56	"Un-normal"	4,56
19	High	1,33	"Normal"	3,67
20	High	3,00	"High"	2,00
21	Un-normal	9,97	"Un-normal"	0,03
22	Un-normal	9,97	"Un-normal"	0,03
23	Un-normal	9,97	"Un-normal"	0,03
24	Un-normal	9,97	"Un-normal"	0,03
25	Un-normal	9,97	"Un-normal"	0,03
26	Un-normal	9,97	"Un-normal"	0,03
27	Un-normal	9,97	"Un-normal"	0,03
28	Un-normal	9,97	"Un-normal"	0,03
29	Un-normal	9,97	"Un-normal"	0,03
30	Un-normal	9,97	"Un-normal"	0,03

The supervised learning of this system classifies the target into three categories: normal, high, and un-normal condition or by number 1, 5, and 10 sequencely.

If the high traffic and un-normal condition are categorized in undesired condition (warning to network administrators), we have three false negatives from experiment result (see table II). The test number 17 and 18 are closed to 10 but actually the traffic seems to high traffic. Also test number 19 closed to 1 but actually the traffic seems to normal condition. Figure 10 shows Receiver Operating Characteristic (ROC) of our experiment.

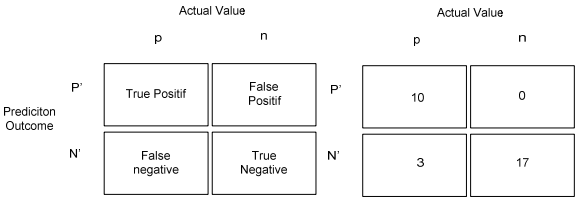


Fig. 10. Receiver Operating Characteristic (ROC) of System

By using rule-based expert system, we resolved it by using if-else condision and lower the range of high traffic:

x<1.49, for normal condition

1.5<x<7.9, for high traffic condition, and

x> 8, for un-normal condition.

Where variable x is a simulation result NNs. In Matlab we simply use a function “sim” that mean to process the input with weights and biases from our network model. And the result is number from 1 to 10 according to the number of our learning target.

Also we test this system using cross validation that separate learning data and testing data. We used three group of sample consisted of three condition. Some literatures have discussed this kind of testing [11]. When one group learned as a model, the last group tested the model. This continued until the last group learned as a model and the first group test it, so we have three couple of learning and testing process to be analyzed (see figure 11).

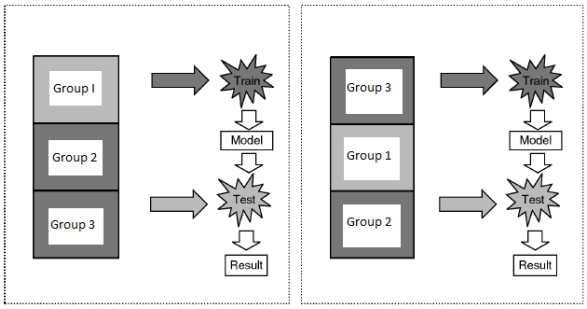


Figure 11 Cross Validation Ilustration

We used three samples each group that represent normal, high and un-normal, so we have 9 samples (each sample having size 10 by 1). By this method every samples always do both learning and testing. The result can be seen at table III.

TABLE III
CROSS VALIDATION RESULTS

No.	Learning	Testing	sim	Kelas	Summary
1	Group 1	Group 3			
		normal	1.89	normal	ok
		high	9.91	un-normal	no
		un-normal	9.94	un-normal	ok
2	Group 3	Group 2			
		normal	3.50	high	no
		high	5.32	high	ok
		un-normal	9.18	un-normal	ok
3	Group 2	Group 1			
		normal	7.54	high	no
		high	4.99	high	ok
		un-normal	9.99	un-normal	ok

Table III shows there are errors about 30 per cent for three models which were tested by other group. It happen because of limitation of our samples that only three for each group, but accuracy for up to 70 per cent is adequate. Of course if we want to get the best result, we must train with a lot of more samples than we already have. Simulation result can be seen at column sim which state 1, 5, and 10 for normal, high traffic, and un-normal consecutively.

IV. IMPROVING EFFICIENCY OF LEARNING

Random Access Memory (RAM) is an important stuff in a microcomputer such as a notebook or a gadget. Neural network, like other soft computing algorithms, need a lot of memory because it works on a parallel system.

Because our categories just identify our graph which tends to lower/middle or upper, we saw the graph from ordinate point of view. We can use average value of points along axis (by using function "mean" in Matlab). Therefore, a 20 x 60 matrix become a 20 x 1vector with size when using this method. Becouse the input vector with size 1200 reduced to 20, it helps a computer learn easily. For example, if we have an image 4x5:

a =

0.8147	0.6324	0.9575	0.9572	0.4218
0.9058	0.0975	0.9649	0.4854	0.9157
0.1270	0.2785	0.1576	0.8003	0.7922
0.9134	0.5469	0.9706	0.1419	0.9595

And we only want to know how the tendency whether below, middle or upper, so we count the average of row:

Average =

0.7567
0.6739
0.4311
0.7064

Of course this method does not run if we want to make a classification according to both axis and ordinate point of view e.g. pattern matching, signature image, etc. This can help some microcomputers (e.g. laptops, netbooks, PDA, etc.) for learning artificial neural network.

V. COMPARISON TO OTHER SYSTEMS

It is easy to compare this system to other common systems (SNORT, TCPDUMP, Network Flight Recorder (NRF), and so on) because we used soft computing algorithm for sensing the traffic. There are some research papers about intrusion detection using soft computing, but they used genetic algorithm instead of NNs[10]. They combine soft computing with rule-based expert system if-then like our proposed system. But that system did not use image processing. The system that they proposed could identify kinds of attack (DoS, Probe, U2R, and R2L) well and run in LAN with switch, instead of just broadcast system (hotspot area and hub/concentrator).

We only proposed system based on graph from network sniffer that may be does not enough to understand full network perform. Our system can be used to understand the intrusion of attack only in brief manner but it can be implemented in large scale, such as Multi Router Traffic Grapher (MRTG) for Wide Area Network that serve the graph of network daily or weekly. We just help network administrator from interpreting the graph of our network and let our system do it. Figure 12 illustrate the MRTG graph. It must be very interesting because we must take into a count about the color of graph.

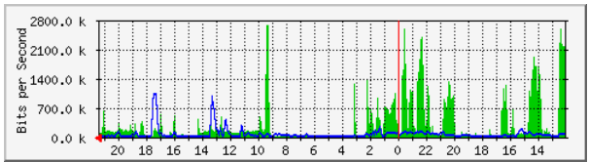


Fig 12 Example of MRTG (source: www.en.wikipedia.org)

VI. Conclusions and Future Research

Experimental results show that our system can be identify types of network traffic weather normal, high traffic, or un-normal, and become the first alarm of network administrators with accuracy about 70 to 90 per cent. By combining with rule-based expert systems this system can identify the type of network traffic. The network graph is captured by network sniffer, so network administrators can analyze deeply why their hotspot area traffic high or un-normal.

We used image of network graph as an input of our system. So, many network monitoring applications can be integrated with our system for larger implementation. In the next research we have to use soft computing in text-based area, because many log files of network sniffer are in text-based form. For a non-hotspot area network (using switch) we must use network sniffer application that running in switching-based network.

Acknowledgment

We would like to thanks other people who had helped this research. The wireshark (http://www.wireshark.org) gives the open source software for capturing the traffic of our hotspot areas area, Quick Screenshoot Maker (http://www.etrusoft.com) for it's trial version for capturing periodically the traffic's image and also the Matlab Software (http://www.mathworks.com) as a language programming for prototyping the system. The sounds of our application are recorded from http://www.translate.google.com, and finally we also thanks to Institute for Research and Community Services (LPPM) of Islam 45 University for funding this research.

References

[1] Roesch, Martin, *"SNORT – Lightwight Intrusion Detection For Network"*, in *Proceeding of LISA '99: 13th System Administration Conference*, Seatle, Washington 7-12th of November 1999, pp 229-237.

[2] Clincy, V.A., Nael Abu-Halaweh, *"A Taxonomy of Free Network Sniffers for Teaching and Research"*, Consortium for Computing Sciences in Collegees, Georgia, 2005, pp. 64-75.

[3] Fausett, Laurene V, *"Fundamentals of Neural Network: Architectures, Algorithms and Applications"*, New Jersey: Prentice Hall, 1993.

[4] Czajka, Adam & Andrzej Pacut, *"Neural networks for signature classification and identity verification secure"*, Materialy Konferencyjne, 2002: 1 – 7.

[5] Kalera, Meenakshi K., Srihari Sargur & Aihua Xu, *"Offline Signature Verification and Identification Using Distance Statistics"*, International Journal of Pattern Recognition and Artificial Intelligence, 2004: 18, 1339-1360.

[6] Sabourin, Robert, *"Off-line Signature Verification: Recent Advanced and Perspective"*, Journal of Laboratoire d'Imagerie de Vision et d'Intelligence Artificielle (LIVIA), 1997, 19, 976-988.

[7] Gonzalez, Rafael C., Richard E. Woods, Steven L. Eddins, *" Digital Image Processing Using Matlab, Second Edition"*, USA: Gatesmark Publishing, 2009.

[8] Leondes, Cornelius T. *"Image Processing and Pattern Recognition"*, USA: Academic Press, 1998.

[9] Albertson, Mikael & Peter Gunnarsson,"Computer Forensic", TDDCO3 Project, Spring, Sweden, 2004, pp. 1-6.

[10] Kandeeban, S. Selvakani and Rengan S. Rejash, *"Integrated Intrusion Detecting System Using Soft Computing"*, International Journal of Network Security, Vol 10, No.2, 2010.

[11] McQuarrie, Allan D. R., Chih-Ling Tsai, *"Regression and Time Series Model Selection"*, Singapore: World Scientific Publishing, Co. Pte. Ltd, 1998.

[12] Abe, Sigeo,*"Support Vector Machines for Pattern Classification"*, London: Springer-Verlag London Limited, 2010.

[13] Miyamoto, Sadaaki, Hidetomo Ichihashi and Katsuhiro Honda,*"Algorithms for Fuzzy Clustering – Methods in c-Means Clustering with Application"*, Germany: Springer-Verlag Berlin Heidelberg, 2008.

[14] Anoname, *"Encyclopedia - Definition of: Hotspot"*, from: http://www.pcmag.com/encyclopedia_term/0,2542,t=Wi-Fi+hotspot&i=61778,00.asp, The Computer Language Company Inc., 21November 2011.

[15] Anoname, *"Hotspot (Wi-Fi)"*, from: http://en.wikipedia.org/wiki/Hotspot_%28Wi-Fi%29, 21 November 2011.