To implement an effective Real-Time Network Intrusion Detection System to reduce False Alarm Rate and improve efficiency, using Machine Learning Techniques

Prof. S. K. Wagh, Mrinal S. Bhalerao, Anjum M. Pathan, Jyoti S. Bhole, Sampada L. Pingale
MES College Of Engineering, Pune.
IEEE Conference Publishing

Abstract- In today's world, almost everybody is affluent with computers and network based technology is growing by leaps and bounds. So, network security has become very important, rather an inevitable part of our computer system. An Intrusion Detection System(IDS) is designed to detect system attacks and classify system activities into normal and abnormal form. Signature based IDSs have been designed and implemented. So, anomaly based intrusion detection system, to detect all kinds of newer attacks will be implemented by us using Machine learning techniques which give high accuracy. A combination of k-NN, decision tree, Ripper rule, Back Propagation neural network will be used to build an efficient and accurate IDS giving least False Alarm Rate.

Model of IDS contains following elements: Audit collection, Storage, Processing, Configuration data, Reference data, active and Processing data, Alarm.

We intend to design an IDS which is more efficient than present IDSs, by increasing detection rate and reducing FAR.

***Keywords*: RT-IDS, machine learning**

I. INTRODUCTION

Nowadays, many organizations Internet services as their communication and marketplace to do business. Together with the growth of computer network activities, the growing rate of network attacks has been advancing. Thus, it is impacting the confidentiality and integrity of critical data. Therefore a network system must use one or more security tools such as firewall, antivirus, IDS to prevent important data from being misused. A network system using a firewall only; is not enough to prevent from all attack types.

The firewall cannot defend the network against intrusion attempts during port opening Intrusion Detection System (IDS) are systems that automate the process of monitoring and analyzing the events that occur in a computer network, to detect malicious or spurious activity. Hence a Real-Time Intrusion Detection System (RT-IDS),(shown in fig. 1) is a preventive mechanism that gives an alarm signal to the computer user or network administrator for antagonistic activity on the opening session, by inspecting hazardous network activities.

Since the severity of attacks occurring in the network has increased drastically, Intrusion detection system have become a necessary component of security infrastructure of most organizations. Intrusion detection allows organizations to protect their systems from the threats that are a cause of increasing network connectivity and reliance on other interconnected systems.

Typically, intrusion detection techniques fall into two categories: signature/misuse detection or anomaly detection. Signature detection helps to find well-known patterns of attacks and intrusions by searching for pre-classified signatures either in network traffic or data patterns.

Anomaly detection, which is designed to capture behavior that deviates from the normal, is the counterpart to signature detection. These systems "predict" anomalous behavior. Hence, they can detect new/unknown intrusions. However, they suffer from false alarms. Since the number of new attacks is increasing and variations of old attacks are more prevalent, next generation IDSs must employ anomaly detection.

Regardless of the algorithm to be used, the initial step in anomaly detection is the real-time feature extraction. Due to the complexity of gathering detailed information, existing techniques have limited effectiveness. Besides the packet payload, a single packet does not offer much information. But by processing a series of packets, we can mine for more precise characteristics of the data.

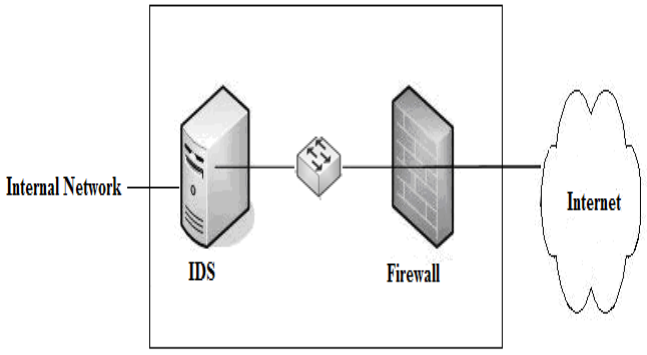


Fig 1. Real Time IDS

II. Previous Research Work

1. Off-line network intrusion detection

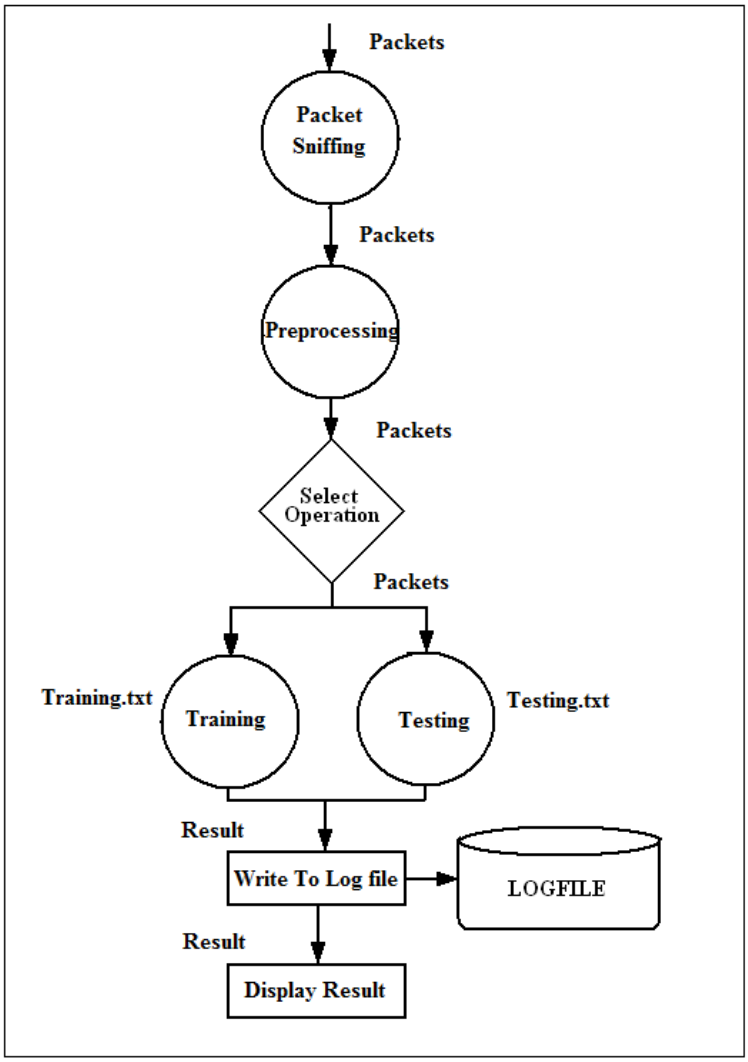
The artificial neural network is one of the most popular techniques for the IDS design. Jirapummin et al. proposed hybrid neural network using a combination of Self-Organizing Map (SOM) and Resilient Back-Propagation Neural Network (BPNN). For their approach, they used an available the preprocessed dataset which is KDD99.

Pan et al. designed a hybrid system by using a BPNN and a C4.5 Decision Tree using the KDD99 dataset. The results showed that using only a BPNN without C4.5 Decision Tree, their system could not detect certain network attack types such as User to Root (U2R) and Root to Local (R2L).

Moradi and Zulkernine used a Multi- Layer Perceptron (MLP) artificial neural network in off-line mode to classify normal network activity, Satan (Probe) attacks and Neptune attacks.

Ngamwitthayanon et al. designed a multi-state IDS system to classify normal data and each attack type using the KDD99 dataset. Their results showed higher detection rate in each classification category than when only a single state was used to classify all categories.

Fuzzy Sets is also a technique often used for IDS. This technique generally falls into two categories, fuzzy misuse detection and fuzzy anomaly detection. Abraham and Jain used three types of fuzzy rules to compare with linear generic programming (LPG), Decision Tree, and Support Vector Machines (SVM). Their result showed that one of their fuzzy rules gave the best detection rate using the 41 features of the DARPA 1998 dataset. Liao et al. used fuzzy logic and an expert system with the DARPA 2000 dataset and achieved more than 91.5% detection rate over all attack types, while reducing complexity of traditional techniques for ranking fuzzy numbers.



.

2.2. On-line (real-time) network intrusion detection

An IDS that can detect a network intrusion while an attack is occurring is called a real-time detection system. A real-time IDS captures present network traffic data which is on-line data. In our study, we found very few on-line (real-time) network IDS approaches that have been proposed previously.

Labib and Vemuri developed a real-time IDS using Self- Organizing Maps (SOM) to detect normal network activity and DoS attack. They preprocessed their dataset to have 10 features for each data record. Each record contained information of 50 packets. Their IDS was evaluated by human visualization for different Network intrusion detection system environment characteristics of normal data and DoS attack. No detection rate was reported.

Puttini et al. used a Bayesian classification model for anomaly detection to classify normal network activity and attack using a 3-month training dataset and a 1-month test dataset. They evaluated their approach by adjusting a penalty value to see how it affected the classification results. They also needed human expert to visualize the normal and abnormal network behaviors. No detection rate was reported. Amini et al. designed a real-time IDS using two unsupervised neural network algorithms which are Adaptive Resonance Theory (ART) and Self-Organizing Map (SOM). They classified two attack types plus normal data during a 4-day experiment with a 27-feature dataset, where each feature captures number of occurrences of an event in each time interval. The detection results showed that the ART-2 gave higher detection speed and detection rate than the SOM. The detection rate was reported to be over 97%, separating normal traffic data from network attacks. However, the attacks were not classified into types or categories.

Su et al. created a real-time network IDS using fuzzy association rules and conducted their experiments by using

four computers with 30 DoS attack types in WIN32. They could separate the normal network activity from network attacks but they did not identify the attack type. They preprocessed the network data to have 16 features. After testing, the results showed that the 30 DoS attack types have a similarity ratio of less than 0.4 while normal network activity gave a similarity ratio more than 0.75.

The similarity ratio represents how close or similar the data is to normal data, i.e. 1.0 means that they are perfectly matched. Li et al. developed a high-speed intrusion detection model using TCP/IP header information. However, their approach is limited to only one type of attack which is DoS.

A well-known intrusion detection tool called SNORT was studied in. SNORT has become a commercial tool. Its attack signature rules are available only to their registered customers. The signature rules or patches have to be frequently updated and installed in order to detect current attack types. In summary, most researchers proposed IDS classification algorithms based on machine learning techniques and used KDD 99 dataset to evaluate their IDS approaches in an off-line environment without considering real-time processing/detection. The KDD99 dataset, which was created about 10 years ago, is complex and lacks of many current attack types. While this approach is of theoretical interest, it provides only post hoc assistance to network administrators and thus is less useful for building practical tools. Although a few researchers have started investigating real-time intrusion detection systems using different techniques, most of the on-line IDS performance still needs further improvement. Some on-line IDSs require a human expert to visualize or identify the attacks, while the remaining systems have other limitations. For example, some IDS can separate normal network data from attack data, but cannot classify attack type. Some IDSs were designed to detect only one attack type. In this paper, we are interested in developing a practical realtime IDS approach which can be applied with well-known machine learning algorithms. The approach should be simple yet efficient in detecting network intrusions in a real-time environment.

III. Machine Learning Techiniques

Machine learning deals with the task of building programs that improve their performance through experience. Machine learning algorithms have proven to be of great value in a variety of domains. They are particularly useful for (a) poorly understood problem domains where little knowledge exists for the humans to develop effective algorithms; (b) domains where there are large databases containing valuable implicit regularities to be discovered; or (c) domains where programs must adapt to changing conditions.

Machine learning is basically used to give the high detection accuracy for real-time intrusion detection system and improvement of algorithms that are existing for the same.

Basically, input given to machine learning technique is empirical data and output of this technique is the patterns/features of the underlying mechanism that generated the data.

Many approaches for machine learning techniques are

(a) Decision tree
(b) Ripper Rule
(c) Back-Propagation Neural Network
(d) Bayesian Network
(e) Radial Basis Function Neural Network.

1. Decision tree.

Decision tree concept is basically used in data mining to efficiently classify the data.

It consists of non terminal and terminal nodes. Non terminal means a root and internal node, and the terminal nodes are (leaves). Initially, decision trees classify the known data and untrained data.in decision tree by identifying attribute and value that will be used input data at each internal node after training. The data decision tree traverse from the starting of root node to internal node.

Algorithms for decision tree are (a) C4.5 Algorithm and (b) ID3 Algorithm.

2. Ripper Rule.

Ripper is the (Repeated Incremental Pruning to Produce Error Reduction) and rule is the efficient rule based learning algorithm process the various noisy dataset. It should be noted that ripper is used to handle data sets with the target that take on more than two unique values. It consist of two stages (a) first initialize the rule condition. (b) Usages rule optimization technique.

Used to minimize the amount of error. Rule in the form of if-else statement. Form. Consisting of two loops:

(1)Outer loop and (2) inner loop.

Algorithm used for ripper rule is Ripper Down Rule.

3.Bayesian Network

A Bayesian Network involves both a graphical model and a probabilistic model representing random variables and condition dependence through a DAG. Nodes in the graph represent random variables, edges represent conditional dependencies. Thus nodes that are not connected represent variables that are conditionally independent on each other.

4.Back-Propagation Neural Network

Back-Propagation Neural Network (BPNN) model is a well known supervised learning model that can effectively classify many types of data.

BPNN is a feed-forward multi-layer network. Its input vectors and the corresponding target vectors are used in training the network until it can approximate a function,

IV. Mathematical Models

1. Information gain:

Information Gain can be chosen for feature selection. For this, we compute entropy value for every attribute of data.

This entropy is used for identifying and ranking features that affect data classification. If a feature has very low information gain then it can be neglected as it does not have a great influence on data classification. This feature can be ignored without affecting the detection accuracy. Definition of information gain is:

Let X and Y be discrete variables representing sample features (X1, X2, . . . , Xm), and class attributes (Y1,Y2, . . .,Yn), respectively.

The information gain of a given feature X with reference to the class attribute Y is the uncertainty reduction of the value of Y, when we know the value of X.

Entropy(Y) is the uncertainty of the value of Y. The Entropy (YIX) is the uncertainty of the value of Y when we know the value of X.

The information gain can be mathematically presented as:

InGain(Y;X)=Entropy(Y)- Entropy(YIX) ..(1)

Where: Entropy (Y) =$-\sum_{i=1}^{n} P(Y = yi)\, log2P(Y = yi)$

P(Y=y_i)=probability that the class attribute y_i occurs.

Entropy (YIX)=$\sum_{j=1}^{m} P(X = xj)Entropy(YIX = xj)$

The information gain can be also represented as:

InGain(Y;X)=Entropy(X)+Entropy(Y)- Entropy(X,Y) ..(2)

The Entropy(X,Y) is the joint entropy of X and Y, so

Entropy(X,Y)=$\sum_{ij} P(X = xj, Y = yj)log2P(X = xj, Y = yi)$ ..(3)

Here, X defines individual feature of inputs in classification, and Y defines attack class (Normal data, Probe attack and DoS attack).

We need to compute the Information Gain for a number of possible features. We must select a small number of features with high Information Gain to use in our network traffic classification.

2. K-Nearest Neighbor Classification:

The KNN technique is classification scheme based on the use of distance measures as. It assumes that the entire sampling set includes not only the data in the set, but also the desired classification for each item. When a classification is to be made for a newly arrived item, its distance to every item in the sampling set must be calculated. Only the k-closest entries in the sampling set are considered for operation. The new item is then classified according to findings which reveal the class that contains the most items from this set of k-closest items.

The distance between two instances represents their similarity; hence, ingredients of an instance denote features. Euclidean distance is usually adopted in the KNN.

For any two n-feature instances, say X = (x_1, x_2, . . . , x_n) and Y = (y_1, y_2, . . . , y_n), their Euclidean distance is computed as:

Dist(X,Y)=

$$\sqrt{(x1 - y1)^2 + (x2 - y2)^2 + \cdots + (xn - yn)\text{^}2}$$

$$=\sqrt{1.(x1 - y1)^2 + 1.(x2 - y2)^2 + \cdots + 1.(xn - yn)\text{^}2}$$

V. PROCEDURE

1. Pre-processing Phase

The system to be implemented is real time and thus we need faster capturing tools to be used. These can be:

- Wireshark (Ethereal)
- Capsa

These tools will give the packets to RT-NIDS system which will process them and give the results in real time.

In this phase packet capturing and extraction of packet features is done with the help of packet sniffing tools, which are used to capture the packet information like, IP/TCP/ICMP headers, from each of the packet. After that we are going to partition the packet header with source address, destination address etc.

In this phase we are going to use the mathematical models for selection of essential feature. And finding whether packet is normal or intrusion.

2. Classification

Depending on feature values the corresponding algorithms will classify the packet into similar groups. Information gain is calculated and mathematical model will be finally evaluated to get the results.

In classification phase we are going to utilize the data received from the previous phase for detecting whether normal packet or attack packet. It consists of two processes:

(a) Training data (b) Testing data

In training phase answer class is provided along with the packet features which will help to formulate rules deciding mapping domains. These rules may get changed replaced depending on further training.

Every algorithm has its own strategy of classification. Decision tree will assign a feature at each level and go on comparing the value at each level which results the packet to be sent at left or right side. Ripper rule will keep on making rules at each evaluation and updating these rules until it reach a level of correctness. K-NN is using reduced calculation by assigning new entry to its k nearest neighbor's group. Whereas, BPNN uses backtracking method to eliminate errors.

In the Testing Phase, untrained data is given to the system for sampling whether true answers are obtained or not. The system process is performed providing input as packets without specifying answer class

3. Post Processing

The result we get in preprocessing phase is evaluated against answer class and system performance is measured in combinations of correctness and false alarms. i.e. True Positive True Negative False Positive and False Negative.

Resulting table of TP, TN, FP and FN is current system knowledge and performance which has to be improved further.

CASE	Expected Answer	Answer Obtained	Outcome
True Positive	ALARM	ALARM	Desirable
True Negative	NORMAL	ALARM	Desirable
False Positive	ALARM	NORMAL	Not Desirable
False Negative	NORMAL	NORMAL	Not Desirable

Results of how effective and efficient our system is can be given by:

- Total Detection Rate (TDR) is the percentage of DoS attacks, Probe attacks, and normal network data that the RT-IDS can correctly detect.
- Normal Detection Rate (NDR) is the percentage of the normal class that the RT-IDS can correctly detect.
- Attack Detection Rate (ADR) is the percentage of all attack classes that the RT-IDS can correctly detect.
- DoS Detection Rate (DDR) is the percentage of the DOS attacks that the RT-IDS can correctly detect.
- Probe Detection Rate (PDR) is the percentage of the Probe attacks that the RT-IDS can correctly detect.

4. Reducing False Alarms

If the system is still giving some false alarms for all the four algorithms some more training is needed to be given. This is the machine learning mechanism i.e. the system will keep on learning on its own without human interference. And hence there is no updating required.

We might resort to using a majority voting algorithm for every five consecutive detection results for each pair of source and destination IP addresses. We can determine if the result is normal network activity or is an attack type by grouping the network data from the classification phase into groups of five records. In each group, if there are at least 3 out-of 5 records which are reported to be the same attack type, then this group of data is considered the attack. Else, the data is considered as normal.

This procedure can increase the detection accuracy and the user's confidence in the alarms provided by our IDS.

VI. CONCLUSION

So far, there have been very few or no tries at attempting to develop a real-time IDS. All prior systems focus on Offline traffic only. We have attempted to take traffics that is not stored on disk, but is online.

In our attempt we have examined various algorithms like Decision Tree, k-NN, BPNN and Ripper Rule and will try to implement them in practice.

Result of algorithms will then be scrutinized and techniques for reducing FAR and increasing the accuracy will be tried and tested. Hence, using best algorithms we can implement real time online network intrusion detection system.

Future scope for this research will be to take traffic that comes directly from the network and try to prevent the intrusions by making the False Alarm Rate low or nil.

REFERENCES

[1] C. Jirapummin, N. Wattanapongsakorn, J. Kanthamanon, Hybrid neural networks for intrusion detection system, in: The International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC),Thailand, 2002, pp. 928–931.

[2] W. Lee, S. Stolfo, K. Mok, Mining in a data-flow environment: experience in network intrusion detection, in: The 5th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '99), San Diego, 1999.

[3] Z. Pan, S. Chen, G. Hu, D. Zhang, Hybrid neural network and C4.5 for misuse detection, in: The 2nd International Conference on Machine Learning and Cybernetics, China, 2003, pp.2463–2467.

[4] M. Moradi, M. Zulkernine, A neural network based system for intrusion detection and classification of attacks, in: The IEEE International Conference on Advances in Intelligent Systems Theory and Applications, Luxembourg, 2004,pp. 148–153.

[5] N. Ngamwitthayanon, N. Wattanapongsakorn, C. Charnsripinyo, D.W.Coit, Multi-stage network-based intrusion detection system using back propagation neural networks, in: Asian International Workshop on Advanced Reliability Modeling (AIWARM), Taiwan, 2008, pp. 609–619.

[6] A. Abraham, R. Jain, Soft computing models for network intrusion detection systems, in: Knowledge Discovery, Computational Intelligence, vol. 4, Heidelberg, 2005, pp. 191–207.

[7] N. Liao, S. Tian, T. Wang, Network forensics based on fuzzy logic and expert system, Computer Communications 32 (2009) 1881–1892.

[8] N. Ngamwitthayanon, N. Wattanapongsakorn, D.W. Coit, Investigation of fuzzy adaptive resonance theory in network anomaly intrusion detection, in: The IEEE International Symposium on Neural Networks, China, 2009.
[9] A.N. Toosi, M. Kahani, A new approach to intrusion detection based on an evolutionary soft computing model using neuro-fuzzy classifiers, Computer Communications 20 (2007) 2201–2212.
[10] C-H. Tsang, S. Kwong, H. Wang, Genetic-fuzzy rule mining approach and evaluation of feature selection techniques for anomaly intrusion detection, Pattern Recognition 50 (2007) 2373–2391.
[11] S. Pukkawanna, V. Visoottiviseth, P. Pongpaibool, Lightweight detection of DoS attacks, in: The IEEE International Conference on Networks (ICON), Australia, 2007, pp 77–82.
[12] V. Katos, Network intrusion detection: evaluating cluster, discriminant, and logit analysis, Information Sciences: an International Journal 177 (15) (2007) 3060–3073.
[13] C-M. Chen, Y-L. Chen, H-C. Lin, An efficient network intrusion detection, Computer Communications 33 (2010) 477–484.
[14] K. Labib, R. Vemuri, NSOM: a real-time network-based intrusion detection system using self-organizing maps, Networks and Security (2002).
[15] R.S. Puttini, Z. Marrakchi, L. Me, A Bayesian classification model for real-time intrusion detection, in: The 22nd International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering. AIP Conference Proceedings, vol. 659, 2003, pp. 150–162.
[16] M. Amini, A. Jalili, H. Reza Shahriari, RT-UNNID: a practical solution to realtime network-based intrusion detection using unsupervised neural networks, Computer & Security 25 (2005) 459–468.
[17] M-Y. Su, G-J. Yu, C-Y. Lin, A real-time network intrusion detection system for large-scale attacks based on an incremental mining approach, Computers and Security 28 (2009) 301–309.
[18] Z. Li, Y. Gao, Y. Chen, HiFIND: a high-speed flow-level intrusion detection approach with DoS resiliency, Computer Networks 54 (2010) 1282–1299.
[19] S. Chakrabarti, M. Chakraborty, I. Mukhopadhyay, Study of snort-based IDS, in:International Conference and Workshop on Emerging Trends in Technology (ICWET), Mumbai, India, 2010, pp. 43–47.
[20] M. Panda, M.R. Patra, Semi-Naı¨ve Bayesian method for network intrusion detection system, Neural information processing, Lecture Notes in Computer Science (Springer Link) 5863 (2009) 614–621.
[21] P. Sangkatsanee, N. Wattanapongsakorn, C. Charnsripinyo, Network intrusion detection with artificial neural network, decision tree and rule based approaches, in: The International Joint Conference on Computer Science and Software Engineering, Thailand, 2009.
[22] P.N. Tan, M. Steinbach, V. Kumar, Introduction to Data Mining, Pearson Addison Wesley, 2005.
[23] Weka 3.6.0 tools [Online]. <http://www.cs.waikato.ac.nz/ml/weka/>.
[24] S.X. Wu, W. Banzhaf, The use of computational intelligence in intrusion detection system: a review, Applied Soft Computing 10 (2010) 1–35.
[25]Process Explorer tool [Online]. <http://technet.microsoft.com/en-us/sysinternals/bb896653.aspx>.
[26] Jpcap library [Online]. <http://jpcap.sourceforge.net/>.
[27] J.H. Lee, J.H. Lee, S.G. Sohn, J.H. Ryu, T.H. Chung, Effective value of decision tree with KDD 99 intrusion detection datasets for intrusion detection system, in: International Conference on Advanced Communication Technology (ICACT), Korea, 2008.