Design Network Intrusion Detection System using hybrid Fuzzy-Neural Network

Muna Mhammad T. Jawhar muna.taher@gmail.com
Faculty of Natural Science
Department of computer science
Jamia Millia Islamia
New Delhi, 110025, India

Monica Mehrotra drmehrotra2000@gmail.com
Faculty of Natural Science
Department of computer science
Jamia Millia Islamia
New Delhi, 110025, India

Abstract

As networks grow both in importance and size, there is an increasing need for effective security monitors such as Network Intrusion Detection System to prevent such illicit accesses. Intrusion Detection Systems technology is an effective approach in dealing with the problems of network security. In this paper, we present an intrusion detection model based on hybrid fuzzy logic and neural network. The key idea is to take advantage of different classification abilities of fuzzy logic and neural network for intrusion detection system. The new model has ability to recognize an attack, to differentiate one attack from another i.e. classifying attack, and the most important, to detect new attacks with high detection rate and low false negative. Training and testing data were obtained from the Defense Advanced Research Projects Agency (DARPA) intrusion detection evaluation data set.

Keywords: FCM clustering, Neural Network, Intrusion Detection.

1. INTRODUCTION

With the rapid growth of the internet, computer attacks are increasing at a fast pace and can easily cause millions of dollar in damage to an organization. Detection of these attacks is an important issue of computer security. Intrusion Detection Systems (IDS) technology is an effective approach in dealing with the problems of network security.

In general, the techniques for Intrusion Detection (ID) fall into two major categories depending on the modeling methods used: misuse detection and anomaly detection. Misuse detection compares the usage patterns for knowing the techniques of compromising computer security. Although misuse detection is effective against known intrusion types; it cannot detect new attacks that were not predefined. Anomaly detection, on the other hand, approaches the problem by attempting to find deviations from the established patterns of usage. Anomaly detection may be able to detect new attacks. However, it may also cause a significant number of false alarms because the normal behavior varies widely and obtaining complete description of normal behavior is often difficult. Architecturally, an intrusion detection system can be categorized into three types host based IDS, network based IDS and hybrid IDS [1][2]. A host based intrusion detection

system uses the audit trails of the operation system as a primary data source. A network based intrusion detection system, on the other hand, uses network traffic information as its main data source. Hybrid intrusion detection system uses both methods [3].

However, most available commercial IDS's use only misuse detection because most developed anomaly detector still cannot overcome the limitations (high false positive detection errors, the difficulty of handling gradual misbehavior and expensive computation[4]). This trend motivates many research efforts to build anomaly detectors for the purpose of ID [5].

The main problem is the difficulty of distinguishing between natural behavior and abnormal behavior in computer networks due to the significant overlap in monitoring data. This detection process generates false alarms resulting from the Intrusion Detection based on the anomaly Intrusion Detection System. The use of fuzzy clustering might reduce the amount of false alarm, where fuzzy clustering is usesd to separate this overlap between normal and abnormal behavior in computer networks.

This paper addresses the problem of generating application clusters from the KDD cup 1999 network intrusion detection dataset. The Neural Network and Fuzzy C-Mean (FCM) clustering algorithms were chosen to be used in building an efficient network intrusion detection model. We organize this paper as follows, section 2 review previous works, section 3 provides brief introduction about Neural Network, section 4 present fuzzy C-means clustering algorithm, section 5 explain the model designer and training Neural Network, section 6 discusses the experiments results followed by conclusion.

2. PREVIOUS WORK

In particular several Neural Networks based approaches were employed for Intrusion Detection. Tie and Li [6] used the BP network with GAs for enhance of BP, they used some types of attack with some features of KDD data. The detection rate for Satan, Guess-password, and Peral was 90.97, 85.60 and 90.79 consequently. The overall accuracy of detection rate is 91.61 with false alarm rate of 7.35. Jimmy and Heidar [7] used feed-forward Neural Networks with Back Propagation training algorithm, they used some feature from TCP Dump and the classification result is 25/25. Dima, Roman and Leon[8] used MLP and Radial Based Function (RBF) Neural Network for classification of 5 types of attacks, the accuracy rate of classifying attacks is 93.2 using RBF and 92.2 using MLP Neural Network, and the false alarm is 0.8%. Iftikhar, Sami and Sajjad [9] used Resilient Back propagation for detecting each type of attack along, the accurse detection rate was 95.93. Mukkamala, Andrew, and Ajith [10] used Back Propagation Neural Network with many types of learning algorithm. The performance of the network is 95.0. The overall accuracy of classification for RPBRO is 97.04 with false positive rate of 2.76% and false negative rate of 0.20. Jimmy and Heidar[11] used Neural Network for classification of the unknown attack and the result is 76% correct classification. Vallipuram and Robert [12] used back-propagation Neural Network, they used all features of KDD data, the classification rate for experiment result for normal traffic was 100%, known attacks were 80%, and for unknown attacks were 60%. Dima, Roman, and Leon used RBF and MLP Neural Network and KDD dataset for attacks classification and the result of accuracy of classification was 93.2% using RBF Neural Network and 92.2% using MLP Neural Network.

3. NEURAL NETWORK

Neural Networks (NNs) have attracted more attention compared to other techniques. That is mainly due to the strong discrimination and generalization abilities of Neural Networks that utilized for classification purposes [13]. Artificial Neural Network is a system simulation of the neurons in the human brain [14]. It is composed of a large number of highly interconnected processing elements (neurons) working with each other to solve specific problems. Each processing element is basically a summing element followed by an active function. The output of

each neuron (after applying the weight parameter associated with the connection) is fed as the input to all of the neurons in the next layer. The learning process is essentially an optimization process in which the parameters of the best set of connection coefficients (weights) for solving a problem are found [15].

An increasing amount of research in the last few years has investigated the application of Neural Networks to intrusion detection. If properly designed and implemented, Neural Networks have the potential to address many of the problems encountered by rule-based approaches. Neural Networks were specifically proposed to learn the typical characteristics of system's users and identify statistically significant variations from their established behavior. In order to apply this approach to Intrusion Detection, we would have to introduce data representing attacks and non-attacks to the Neural Network to adjust automatically coefficients of this Network during the training phase. In other words, it will be necessary to collect data representing normal and abnormal behavior and train the Neural Network on those data. After training is accomplished, a certain number of performance tests with real network traffic and attacks should be conducted [16]. Instead of processing program instruction sequentially, Neural Network based models on simultaneously explorer several hypotheses make the use of several computational interconnected elements (neurons); this parallel processing may imply time savings in malicious traffic analysis [17].

4. FUZZY C-MEANS CLUSTERING

The FCM based algorithms are the most widely used fuzzy clustering algorithms in practice. It is based on minimization of the following objective function [18], with respect to U, a fuzzy c-partition of the data set, and to V, a set of K prototypes:

$$J_m(U,V) = \sum_{j=1}^{n}\sum_{i=1}^{c} u_{ij}^{m} \|X_i - V_i\|_2 \quad , 1<m<\infty \qquad \ldots\ldots (1)$$

Where m is any real number greater than 1, Uij is is the degree of membership of Xj in the cluster I, Xj is jth of d-dimensional measured input data, Vi is the d-dimension center of the cluster, and $\|*\|$ is any norm expressed the similarity between any measured data and the center. Fuzzy partition is carried out through an iterative optimization of (1) with the update of membership Uij and the cluster centers Vi by:

$$U_{ij} = \frac{1}{\sum_{k=1}^{c}\left(\frac{d_{ij}}{d_{ik}}\right)^{\frac{2}{m-1}}} \qquad \ldots. (2)$$

$$v_i = \frac{\sum_{j=1}^{n} u_{ij}^{m} \cdot x_j}{\sum_{j=1}^{n} u_{ij}^{m}} \qquad \ldots. (3)$$

The criteria in this iteration will stop when maxij $|U_{ij} - \hat{U}_{ij}| < \varepsilon$, where ε is a termination criterion between 0 and 1, also the maximum number of iteration cycles can be used as a termination criterion [19].

5. EXPERIMENT DESIGN

The block diagram of the hybrid model is showed in the following figure (1)

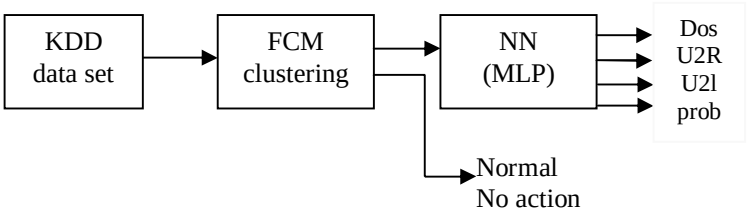

FIGURE 1: the block diagram of the model

5.1 KDD Data Set

KDD 99 data set are used as the input vectors for training and validation of the tested neural network. It was created based on the DARPA intrusion detection evaluation program. MIT Lincoln Lab that participates in this program has set up simulation of typical LAN network in order to acquire raw TCP dump data [20]. They simulated LAN operated as a normal environment, which was infected by various types of attacks. The raw data set was processed into connection records. For each connection, 41 various features were extracted. Each connection was labeled as normal or under specific type of attack. There are 39 attacker types that could be classified into four main categories of attacks:

- DOS (Denial of Service): an attacker tries to prevent legitimate users from using a service e.g. TCP SYN Flood, Smurf (229853 record).
- Probe: an attacker tries to find information about the target host. For example: scanning victims in order to get knowledge about available services, using Operating System (4166 record).
- U2R (User to Root): an attacker has local account on victim's host and tries to gain the root privileges (230 records).
- R2L (Remote to Local): an attacker does not have local account on the victim host and try to obtain it (16187 records).

The suggested model was trained with reduced feature set (35 out of 41 features as in appendix A). We get 25000 training data patterns from 10 percent training set and test data patterns from test set which has attack patterns that are not presented in the training data, we divided test data pattern into two sets.

5.2 FCM Algorithm

The first stage of the FCM algorithm is to initialize the input variable, the input vector consists of 35 features as mentioned previously, the number of cluster is 2 (1=attack and 2=normal), and the center of cluster is calculated by taking the means of all feature from random records in KDD dataset, and the parameter of the object function (m) is 2. After apply the FCM to two different datasets the result after iteration four is 99.99% classification of normal from attack records as seen in the following tables.

Input data	Iteration No.1	Iteration No. 2	Iteration No. 3	Iteration No. 4	Iteration No. 5	Iteration No. 6
Normal 998	1725	1049	1003	1001	1001	1001
Attack 21135	20408	21081	21130	21132	21132	21132

TABLE (1): the result of the first experiment of using FCM clustering

Iteration No.	1	2	3	4	5	6
Normal classification rate (%)	57.80	95.10	99.59	99.98	99.98	99.98
Attack classification rate (%)	96.50	99.74	99.97	99.98	99.98	99.98
False positive (%)	0.728	0.0541	0.00501	0.0030	0.0030	0.0030
False negative (%)	0.421	0.048	0.0049	0.0029	0.0029	0.0029

TABLE (2): the classification rate of the first experiment

Input data	Iteration No.1	Iteration No. 2	Iteration No. 3	Iteration No. 4	Iteration No. 5	Iteration No. 6
Normal 1018	1752	1062	1022	1019	1019	1019
Attack 9002	8277	8958	8998	9001	9001	9001

TABLE (3): the result of the second experiment of using FCM clustering

Iteration No.	1	2	3	4	5	6
Normal classification rate (%)	57.62	95.77	99.60	99.99	99.99	99.99
Attack classification rate (%)	91.90	99.57	99.95	99.99	99.99	99.99
False positive (%)	0.7121	0.0432	0.0039	0.0009	0.0009	0.0009
False negative (%)	0.418	0.0414	0.0039	0.0009	0.0009	0.0009

TABLE (4): the classification rate of the second experiment

As shown in table 1 the total input data is 22133 records, 998 records as normal and 21135 records as attacker. After applying FCM algorithm, the result after iteration one is 1725 record for normal and 20408 records for attack. After second iteration of FCM algorithm the result is 1049 records for normal and 2108 records for attack, after iteration three the result is 1003 records for normal and 21130 records for attack, the result after iteration four is 1001 records for normal and 21132 records for attack and the result after iteration five and six is the same and there is no change, therefore FCM algorithm is stopped.

As seen the final result of the first experiment in table 1 is 1001 records are normal and 21132 records are attack, the original input data is 998 records as normal and 21135 records as attack. Then we calculated the normal and attack classification rate by the following equation[3]:

$$\text{Classification rate} = \frac{\text{Number of classified patterns}}{\text{Total number of patterns}} * 100 \qquad \text{.....(4)}$$

False negative means if it is attack and detection system is normal, false positive means if it is normal and detect system is attack. The false positive alarm rate calculated as the total number of normal instances that were classified as intrusions divided by the total number of normal instances and the false negative alarm rate calculated as the total number of attack instances that were classified as normal divided by the total number of attack instances.

The same calculation is applied for the second experiment.

5.3 MLP Training Algorithm

The anomaly detection is to recognize different authorized system users and identify intruders from that knowledge. Thus intruders can be recognized from the distortion of normal behavior. Because the FCM clustering stages are classified normal from attack, the second stage of NN is used for classification of attacks type. Multi-layer feed forward networks (MLP) is used in this

work. The number of hidden layers, and the number of nodes in the hidden layers, was also determined based on the process of trial and error. We choose several initial values for the network weight and biases. Generally these chosen to be small random values. The Neural Network was trained with the training data which contains only attack records. When the generated output result doesn't satisfy the target output result, the error from the distortion of target output was adjusted. Retrain or stop training the network depending on this error value. Once the training was over, the weight value is stored to be used in recall stage. The result of the training stage of different network architectures with different training algorithms and different activation functions is shown in the following tables.

Function	No of Epochs	Accuracy (%)
Gradient descent	3500	61.70
Gradient descent with moment	3500	51.60
Resilient back propagation	67	98.04
Scaled conjugate gradient	351	80.87
BFGS quasi-Newton method	359	75.67
One step secant method	638	89.60
Levenberg- marquardt	50	79.34

TABLE (5): test performance of different Neural Network training functions

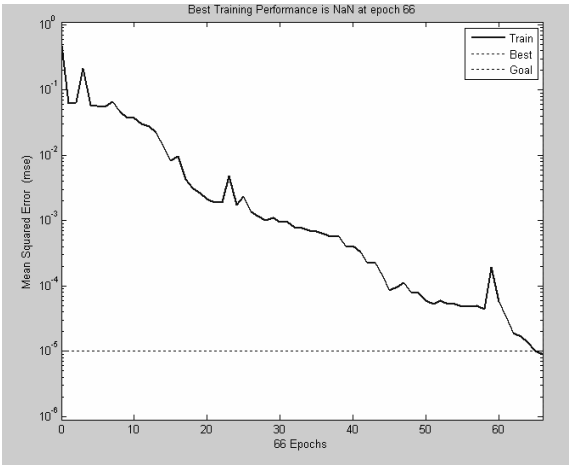

FIGURE (2) : the performance of Resilient back propagation

As seen from above table the best training algorithm is Resilient back propagation which takes less time, low no. of epoch, and high accuracy, the performance of the Resilient back propagation is shown in figure(2), therefore we used it in this paper. The architecture based on this program used one hidden layer, consisting of 12 neurons and 3 neurons in the output layer, the desired mean square error is 0.00001 and the No. of Epoch is 1000, the result of training is illustrated in table(6).

	Input	Output	Accuracy
Dos	23084	23084	100%
U2R	7	7	100%
U2L	608	608	100%
Prob	1301	1301	100%
MSE		0.00001	
Time		00:00:54	
Epoch		56	

TABLE (6): the training experiment of Resilient back propagation

6. TEST AND RESULTS

The model was designed to provide output values between 0.0 and 1.0 in the output nodes. The first stage of the model is FCM clustering, the classification rate is 99.99% which means that the false negative rate is 0.01% and the false positive rate is 0.01% as mentioned previously the manner of calculation them, is very low according to the previous researches. FCM algorithm separates the normal records from attack records, then the MLP stage is the classification of attack to four types. During the testing phase, the accuracy classification of each attack types was calculated, classification time of two different inputs of datasets, the result is shown in table (7).

Attack name	Input 1	Output	Accuracy	Input 2	Output	Accuracy
Dos	23088	23089	99.9%	20463	20463	100%
U2R	7	7	100%	2	2	100%
U2L	608	608	100%	5	2	40%
Prob	1301	1301	100%	665	666	99.8%
Unknown	18	17	94.4%	114	166	68.6%
Time(sec)	5.8292			4.6766		

TABLE (7): The result of testing phase

7. CONCLUSION

The main contribution of the present work is to achieve a classification model with high intrusion detection accuracy and mainly with low false negative; this was done through the design of a classification model for the problem using FCM with Neural Network for detection of various types of attacks. The first stage of the model is FCM clustering, the classification rate is 99.99% that is means the false negative rate is 0.01% and false positive rate is 0.01% which is very low according to the previous researches as illustrated in table (8) and figure(3). The second stage of the model is Neural Network. After many experiment on the Neural Network using different training algorithms and object functions, we observed that Resilient back propagation with sigmoid function was the best one for classification therefore we used it in this work. And we trail many architectures with one hidden layer and two hidden layers with different number of neurons to obtain the best performance of the Neural Network.

author name / properties	Mehdi 2004	Srinivas 2005	Dima 2006	Iftikar 2007	Pizeniyslaw 2008	Khattab 2009	Muna 2010
Classification rate	87%	97.07%	93%	95.93%	92%	97.0%	99.9%
False negative	-	2.76%	-	-	-	0.80%	0.01%
False positive	-	0.20%	0.8%	-	8.8%	2.76%	0.01%

TABLE (8): the comparison result with previous works

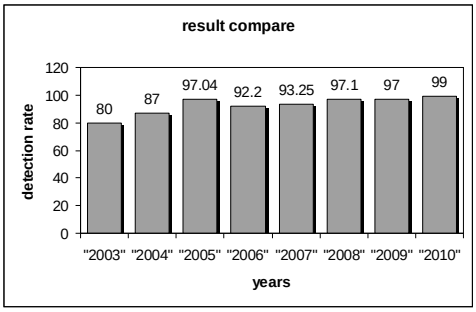

FIGURE (3): The result compare

8. REFERENCES

1. J., Muna. M. and Mehrotra M., "*Intrusion Detection System : A design perspective"*, 2rd International Conference On Data Management, IMT Ghaziabad, India. 2009.
2. M. Panda, and M. Patra, "*Building an efficient network intrusion detection model using Self Organizing Maps*", proceeding of world academy of science, engineering and technology, Vol. 38. 2009.
3. M. Khattab Ali, W. Venus, and M. Suleiman Al Rababaa, "*The Affect of Fuzzification on Neural Networks Intrusion Detection System"*, IEEE computer society.2009.
4. B. Mykerjee, L. Heberlein T., and K. Levitt N., *"Network Intrusion Detection"*, IEEE Networks, Vol. 8, No.3, PP.14-26. 1994.
5. W. Jung K., "*Integration Artificial Immune Algorithms for Intrusion Detection"*, dissertation in University of London, PP.1-5.2002.
6. T. Zhou and LI Yang, *"The Research of Intrusion Detection Based on Genetic Neural Network"*, Proceedings of the 2008 International Conference on Wavelet Analysis and Pattern Recognition, Hong Kong, IEEE.2008.
7. J. Shum and A. Heidar Malki, *"Network Intrusion Detection System Using Neural Networks"*, Fourth International Conference on Natural Computation, IEEE computer society.2008.
8. D. Novikov, V. Roman Yampolskiy, and L. Reznik, *"Anomaly Detection Based Intrusion Detection"*, IEEE computer society.2006.
9. I. Ahmad, S. Ullah Swati and S. Mohsin, "*Intrusions Detection Mechanism by Resilient Back Propagation (RPROP)"*, European Journal of Scientific Research ISSN 1450-216X Vol.17 No.4, pp.523-531.2007.
10. S. Mukkamala, H. Andrew Sung, and A. Abraham, *"Intrusion detection using an ensemble of intelligent paradigms"*, Journal of Network and Computer Applications 28. pp167–182.2005.
11. S. Jimmy and A. Heidar, "*Network Intrusion Detection System using Neural Networks"*, IEEE computer society.2008.
12. M. Vallipuram and B. Robert, *"An Intelligent Intrusion Detection System based on Neural Network",* IADIS International Conference Applied Computing.2004.
13. M. Al-Subaie, "*The power of sequential learning in anomaly intrusion detection",* degree master thesis, Queen University, Canada.2006.
14. P. Kukielka and Z. Kotulski, "*Analysis of different architectures of neural networks for application in intrusion detection systems*", proceeding of the international multiconference on computer science and information technology, pp. 807-811.2008.
15. M. Moradi and M. Zulkernine, "*A Neural Network based system for intrusion detection and classification of attacks"*, Queen University, Canada.2004.
16. D. Novikov, V. Roman Yampolskiy, and L. Reznik, *"Artificial Intelligence Approaches For Intrusion Detection"*, IEEE computer society.2006.

17. S. Lília de Sá, C. Adriana Ferrari dos Santos, S. Demisio da Silva, and A. Montes, *"A Neural Network Application for Attack Detection in Computer Networks",* Instituto Nacional de Pesquisas Espaciais – INPE, BRAZIL.2004.
18. J. Bezdek, C., *"pattern Recognition with Fuzzy Objective Function Algorithms"*. Plenum, New York.1981.
19. Y. John and R. Langari, "*Fuzzy Logic intelligence, control, and information"*, Publish by Dorling Kindersley, India, pp.379-383.2006.
20. P. Kukiełka and Z. Kotulski, "*Analysis of Different Architectures of Neural Networks for Application in Intrusion Detection Systems"*, Proceedings of the International Multiconference on Computer Science and Information Technology, IEEE, pp. 807– 811.2008.
21. KDD-cup dataset. http://kdd.ics.uci.edu/data base/ kddcupaa/kddcup.html
22. Loril D., *"Applying Soft Computing Techniques to intrusion Detection"*, Ph.D thesis, Dep. Of Computer Sce. University of Colorado at Colorado Spring, 2005.

APPENDIX -A-

The table (A1) describes the 41 features of each connection record in the DARPA KDD cup 1999[23]. The fields with blue color are features that have been considered in this research.

Table (A1): feature of KDD cup 1999 data

No.	Feature name	Description	Type
1	Duration	length (number of seconds) of the connection	Continuous
2	Protocol-type	type of the protocol, e.g. tcp, udp, etc.	Discrete
3	Service	network service on the destination, e.g., http, telnet, etc.	Discrete
4	Flag	normal or error status of the connection	discrete
5	Src-bytes	number of data bytes from source to destination	Continuous
6	Det-bytes	number of data bytes from destination to source	Continuous
7	Land	1 if connection is from/to the same host/port; 0 otherwise	Discrete
8	Wrong fragment	number of ``wrong'' fragments	Continuous
9	Urgent	number of urgent packets	Continuous
10	Hot	number of ``hot'' indicators	Continuous
11	Num-failed-logien	number of failed login attempts	Continuous
12	Logged-in	1 if successfully logged in; 0 otherwise	Discrete
13	Num-compromised	number of ``compromised'' conditions	continuous
14	Root-shell	1 if root shell is obtained; 0 otherwise	discrete
15	Su-attempted	1 if ``su root'' command attempted; 0 otherwise	discrete
16	Num-root	number of ``root'' accesses	discrete
17	Num-file-creation	number of file creation operations	continuous
18	Num-shells	number of shell prompts	continuous
19	Num-access-file	number of operations on access control files	continuous
20	Num-outbound-cmds	number of outbound commands in an ftp session	continuous
21	Is-hot-login	1 if the login belongs to the ``hot'' list; 0 otherwise	discrete
22	Is-guest-login	1 if the login is a ``guest''login; 0 otherwise	discrete
23	Count	number of connections to the same host as the current connection in the past two seconds	continuous
24	Srv-count	number of connections to the same service as the current connection in the past two seconds	continuous
25	Serror-rate	% of connections that have ``SYN'' errors	continuous
26	Srv-serror-rate	% of connections that have ``SYN'' errors	continuous
27	Rerror-rate	% of connections that have ``REJ'' errors	continuous
28	Srv-error-rate	% of connections that have ``REJ'' errors	continuous
29	Same-srv-rate	% of connections to the same service	Continuous
30	Diff-srv-rate	% of connections to different services	Continuous
31	Srv-diff-host-rate	% of connections to different hosts	Continuous
32	Det-host-count	Number of connection to the same host	Continuous
33	Dst-host-srv-co	Number of connection to the same serves for the host	Continuous

34	Dst-host-same-srv-rate	% of connections with the same service	Continuous
35	Dst-host-diff-srv-rate	% of connections different services	Continuous
36	Dst-host-same-srv-host-rate	% of connections using same source port	Continuous
37	Dst-host-diff-srv-host-rate	% of connections with same service but to different host	Continuous
38	Dst-host-serror-rate	% of connections that have "SYN" error	Continuous
39	Dst-host-srv-rate	% of connections with same service that have "SYN" errors	Continuous
40	Dst-host-error-rate	% of connections that have "REJ" error	Continuous
41	Dst-host-srv-rer-rate	% of connections with same service that have "REJ" errors	continuous