Journal of Theoretical and Applied Information Technology
20th October 2015. Vol.80. No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

IMPROVEMENT OF PERFORMANCE INTRUSION DETECTION SYSTEM (IDS) USING ARTIFICIAL NEURAL NETWORK ENSEMBLE

[1]WIHARTO, [2]ABDUL AZIZ, [3]UDHI PERMANA
[1,2,3] Department of Informatic, Sebelas Maret University, Surakarta, Indonesia
E-mail: [1]wi_harto@yahoo.com,[2]aaziz@staff.uns.ac.id. [3]udhi.permana@gmail.com,

ABSTRACT

The main focus from one of the problems in computer networks is computer security systems because of the high threat of attack from the internet in recent years. Therefore, an Intrusion Detection System (IDS) that monitors the traffic of computer networks and oversight of suspicious activities in a computer network is required. Research on intrusion detection system have been carried out. Several researches have used artificial neural networks combined with a fuzzy clustering method to detect attacks. However, there is an issue that arise from the use of such algorithms. A single artificial neural network can produce overfitting on intrusion detection system output. This research used two methods of artificial neural networks, namely Lavenberg-Marquardt and Quasi-Newton to overcome that issue. Both algorithms are used to detect computer networks from attack. In addition, the use Possibilistic Fuzzy C-Means (PFCM) before going into the neural network ensemble with simple average. Then on the output, Naive Bayesian classification method is used. Dataset used in the research were NSL-KDD dataset which is an improvement of KDD Cup'99. KDDTrain+ used for training data and KDDTest+ for testing data. Evaluation results show good precision in detection of DoS (89.82%), R2L (75.78%), normal (72.25%) and Probe (70.70%). However, U2R just get 14.62%. At recall, good results achieved by normal state (91.44%), Probe (87.11%) and DoS (83.31%). Low results occurred in U2R (9.50%) and R2L (6.14%). Meanwhile, lowest accuracy on normal category (81.18%) and highest in U2R (98.70%). The results showed that the neural network ensemble method produces a better average accuracy than previous researches, amounting to 90.85%.

Keywords: *Accuracy, Anomaly Based, Intrusion Detection System, Neural Network Ensembles, NSL-KDD*

1. INTRODUCTION

Computer security systems in recent years has been the main focus from one of the problems in computer networks because of the high threat of attack from the internet. Types of attacks are often carried out on a system is a Denial of Service (DoS), Remote to user Attacks (R2L), User to Root Attacks (U2R) and Probing (PRB) [1,3]. Based on survey data of CSI (Computer Security Institute) in 2010/ 2011 shows that many companies are using the security system, in which 62.4% of companies employ advanced IDS (Intrusion Detection System) [2,4].

IDS is a system that monitors the traffic of computer networks and oversight of suspicious activities in a computer network. There are several approaches used in IDS, the signature-based IDS and anomaly-based IDS. Signature-based IDS using attack pattern database to detect attacks. The disadvantage is it must be update to detect new attack patterns. It would be very inconvenient for network traffic speed that so fast, especially on website traffic. Thus required a system capable of learning from past patterns of attack.

Anomaly-based IDS cover the weaknesses of signature-based IDS. This type of IDS detect by checking the traffic on the network and then compare it to the normal state of the system that had previously been defined by the administrator. ANN methods (Artificial Neural Network) is one of the Anomaly-based IDS systems. It able to learn from the past attack patterns to be able to detect new attack patterns.

In recent years there has been much research on using the IDS that include intelligent systems. Faraoun and Boukelif uses k-means clustering and neural network backpropagation gradient descent with momentum. The conclusion is the DoS detection rate can reach 97.23%, 96.63% PRB, R2L U2R 30.97% and 87.71%. R2L detection rate is very low and there is no precision [5]. R2L

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

result is low might be due to the use of k-means clustering. It can not handle outlier data [6].

Next research come from Gang Wang and colleagues. In that research, it used fuzzy c-means clustering and neural network gradient descent with momentum and final output using fuzzy aggregation. The results show that the highest precision at DoS, 99.91% but there is a lowest precision at PRB, 48.12% [7]. PRB result is low because the use of fuzzy c-means clustering. It can not handle outlier data [8]. Moreover, fuzzy aggregation has problem with subjective membership function [9].

All of researches above use KDD Cup'99 dataset. Tavallaee and colleagues have analyzed the KDD Cup'99 dataset. The research shows that there are some weaknesses in the KDD Cup'99 dataset. First, a lot of redundant data (78% and 75% on train and test). Second, the dataset must be random to generated a good evaluation [10].

Tavallaee and colleagues also generates NSL-KDD dataset which is a refinement of the KDD Cup'99 dataset. They use 7 methods with the highest accuracy is NB Tree, 82.02% [10]. However, the results of the evaluation without precision and recall.

Based on Tavallaee and colleagues's research, Panda and colleagues used the refinement of Naive Bayes method. It called Discriminative Multinomial Naive Bayes combined with Random Projection. The results showed accuracy of 81.47% [11]. However, the research only classify normal and attack regardless of the type of attack.

Other research from Najafi and Afsharchi [12] which uses Tree Augmented Naive Bayesian method with NSL-KDD dataset. Good precision and recall results with an average of 90% more. However, the results are very bad from U2R categories, that is 0%.

Research from Govindarajan and colleagues [13] using a combined method of SVM and RBF and using NSL-KDD dataset. The results showed accuracy of 85.19%. Meanwhile, precision and recall are not evaluated.

In order to achieve better IDS, researchers must solve "how to overcome overfitting problem and outlier data?" In this paper, researchers want to know if "the use of neural network ensemble (using Lavenberg-Marquardt and Quasi-Newton), Possibilistic Fuzzy C-Means and Naive Bayesian can overcome overfitting problem and outlier data".

Based on the above explanation, this research proposed a better model of IDS. Researchers attempt to make new framework with a new method as well to address weaknesses in the system that have been made previously. It combines methods from previous research based on refinement of Gang Wang and his colleagues's research. The proposed IDS using possibilistic fuzzy c-means clustering [8] which is an extension of the fuzzy c-means clustering, neural network ensemble using a simple average to overcome the overfitting problem [14]. Then, the Naive Bayesian method is used to determine the final outcome [15].

In this paper, researchers build an intrusion detection system with neural network ensemble (using Lavenberg-Marquardt and Quasi-Newton), Possibilistic Fuzzy C-Means and Naive Bayesian; training and testing IDS using NSL-KDD dataset; and evaluate using precission, recall and overall accuracy. The research doesn't include testing from real network traffic.

2. BASIC THEORY

2.1 Intrusion Detection System (IDS)

In 1987, Dorothy E. Dening proposed intrusion detection as an approach to computer network attacks. In general, intruder is defined as a system, program or people who try to and perhaps made to get in a computer system or perform illegal actions on the computer. IDS is a tool that can monitor network activity and traffic on the computer then analyzes the signs of security threats. There are 3 types of IDS, namely misuse/ signature-based detection, anomaly-based detection, and hybrid.

Misuse/ signature-based detect by matching a known pattern. Anomaly-based detect by identify abnormalities of the normal system. Meanwhile, hybrid detection combine the two.

IDS can also be divided into two types based on part of a computer network are identified. Host-based IDS monitors the activities associated with the host. Meanwhile, network-based IDS monitors network traffic [16].

2.2 Data Normalization

Normalization of data needed in a system, including IDS. This is done so that the data input and the target used in the system is in a certain range. Normalization method used is boxcox. The statistician George Box and David Cox developed a procedure to identify the appropriate exponent (lambda = λ) used to transform data into "normal form". Lambda value indicates the exponent applied to each data. For that reason, lambda = -5 to +5 so we get the best value [17]. Equations 1 and 2 are used to obtain normal data.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

If $\lambda \neq 0$, then : $data(\lambda) = \frac{data^{\lambda}-1}{\lambda}$ (1)

If $\lambda = 0$, then : $data(\lambda) = \log(data)$ (2)

2.3 Fuzzy Cluster Validity Index

Fuzzy cluster validity index is a way to determine the ideal number of clusters (groups) of all existing data will be used in the calculation. This method is very useful when the number of clusters to be generated is not yet known [18].

2.4 Artificial Neural Network Ensemble

Artificial Neural Networks Ensemble or Neural Network Ensemble (NNE) was first introduced by Hansen and Salamon (1990) where NNE is an integration of some of Neural Network (NN). NNE process divide the training data into several sub-samples based on n-input [14]. Basically the neural network ensemble is a merger of several single neural network by using certain methods in order to improve the performance of the system [15].

2.5 Levenberg-Mardquart (LM) Algorithm

LM algorithm is one kind of backpropagation ANN training algorithms with two kinds of computation, forward computation and backward computation. Levenberg-Marquardt algorithm is designed by using the second derivative approach without having to compute the Hessian matrix. If the feed forward neural network performance using the sum of square function, then the Hessian matrix can be approximated by the equation 15 :

$$H = J^{t} * J \quad (15)$$

the gradient can be calculated as (equation 16) :

$$gW = J^{t} * e \quad (16)$$

J is the Jacobian matrix that contains first derivatives of the network errors on the weights, and e is a vector that contains a network error. Jacobian matrix can be computed with standard backpropagation technique, which is much simpler than calculating the Hessian matrix.

Using the Levenberg-Marquardt algorithm to compute the Hessian matrix approach, through the improvement of Newton method (equation 17) :

$$W_{k+1} = W_k - [J^{t} * J + m * I]^{-1} * J^{t} * e \quad (17)$$

If *m* equal to 0, then this approach will be the same as Newton's method. However, if *m* is too large, then this approach will be the same as the gradient descent with a very small learning rate. Newton's method is very fast and accurate to obtain minimum error. Therefore, the algorithm is expected as soon as possible to change the value of *m* to be equal to 0. To that end, after several iterations, this algorithm will reduce the value of *m*, the increase in the value of m is only done if needed a move (temporarily) to degrade the performance of the functions [19].

2.6 Quasi Newton (QN) Algorithm

QN algorithm is one kind of backpropagation ANN training algorithms with two kinds of computation, forward computation and backward computation. This method such as LM method, differing only in the Hessian matrix calculation approach. Equation 18 shows the formula approach Quasi Newton :

$$W_{k+1} = W_k + \alpha_k d_k \quad (18)$$

W is the weight, $\alpha_k d_k$ is approach calculation. α is the step length that minimizes the function 19.

$$f(\alpha) = f(W_k + \alpha_k d_k) \quad (19)$$

While *d* is the search direction [20].

2.7 Simple Average

Simple average was one method of merging the output of neural networks in Neural Network Ensemble (NNE). After training each sub-sample, combine training results from each single sub-sample. After weighting from training phase of each sub-sample obtained, the weighted value used to process the entire training data and generate the appropriate output [14]. The output of the entire training data combined with the simple average method with equation 20 as follows :

$$\hat{y}_t = \frac{\sum_{i=1}^{n} \hat{y}_{it}}{n} \quad (20)$$

The $\hat{y}_t$ is output predictor for NNE, $\hat{y}_{it}$ is the network output and *n* are number of network.

2.8 Naïve Bayesian

Naive Bayesian is classifier based on probability method and the Bayesian theorem, assuming that each variable *X* is free (independence). In other words, naive Bayesian classifier that assumes that the presence of an attribute (variable) has nothing to do with the existence of another attributes (variables) [21].

Because the assumption of unrelated attributes (conditionally independent), then (equation 21) :

$$P(X|C_i) = \prod_{k=1}^{n} P(x_k|C_i) \quad (21)$$

If $P(X|C_i)$ can be determined through the calculation above, then class (label) of the data sample *X* is a class (label) that have maximum $P(X|C_i)*P(C_i)$.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Naive Bayesian method takes two steps in the process of classification, the training phase and the classification phase. At training phase, it process the sample data as much as possible so it can be a representation of the data. Next is the determination of prior probabilities for each category based on sample data. In the classification stage, category of data based on terms that emerged in the data that are classified [22].

2.9 Precission, Recall and Overall Accuracy

In the testing phase, the dataset used in testing the IDS, which produces output in the form of normal activity or an certain attack. Output results are used for IDS evaluation. The evaluation used precision, recall and overall accuracy [23].

Precision (equation 22) to calculate the amount of success with the proper IDS detects an attack on the overall results of detection which produces output in the form of an attack. Recall (equation 23) to calculate the amount of success IDS detects an attack by the right of all existing attack from training data and testing. Overall accuracy (equation 24) to calculate the success of the IDS to detect attacks and normal state with the right of the overall results of the IDS output. Here is the formula of each method of evaluation :

$$Precision = TP/(TP + FP) \quad (22)$$

$$Recall = TP/(TP + FN) \quad (23)$$

$$Overall\ Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (24)$$

Legend :

TP : True Positive

TN : True Negative

FP : False Positive

FN : False Negative

3. METHODOLOGY

3.1 Collecting Data

3.1.1 Collecting Secondary Data

The data used in this research is the NSL-KDD dataset which can be downloaded at http://nsl.cs.unb.ca/NSL-KDD/. NSL-KDD dataset is a dataset refinement of the KDD Cup'99 dataset. Tavallaee and his colleagues have examined the KDD Cup'99 dataset. There are two major drawbacks of KDD'99, that is a lot of redundant data (78% and 75% on train and test). The next issue is the need for random data on the train and test dataset to generated a good evaluation. The research also produced NSL-KDD dataset [10].

3.1.2 Study Literature

Study literature was conducted through books, articles and journals relevant to the object being studied in order to obtain precision step in doing research. Acquired literacy is related to the material issues, design, and implementation of the system, among which the concept of IDS, IDS that uses artificial neural networks and genetic algorithms, methods of possibilistic fuzzy c-means clustering, neural network ensemble, Lavenberg-Marquardt method, quasi Newton and Naive Bayesian method.

3.2 Analysis and Design

3.2.1 Data Analysis

NSL-KDD dataset consists of KDDTest+ for data testing and KDDTrain+ for data training. Each dataset has 41 features (Table 1) to recognize four types of attacks (DoS, U2R, R2L, Probe) and 1 normal state (Table 2) that occurs on a computer network. Each dataset has also been given the category label attacks as mentioned above/ normal and the number of IDS that can detect the attack data based on research from Tavalee and colleagues. Table 1 shows the features of the dataset. Index feature 1-9 is basic features of TCP connections, index feature 10-22 is content features from domain knowledge and indexes feature 23-41 is traffic features.

This research uses 5% KDDTrain+ to get the proper cluster (6300 data), then 100% KDDTrain+ (22544 data) for training data and 100% KDDTest+ (125973 data) for testing data. Dataset composed of normal data and 4 categories of attacks, namely DoS, U2R, R2L and Probe. Table 2 shows the composition of the training and testing datasets :

Table 2: Number of KDDTrain+ and KDDTest+

Data	Number of Each Category				
	Normal	DoS	U2R	R2L	Probe
KDDTrain+	67343	45927	52	995	11656
KDDTest+	9711	7458	200	2754	2421

3.2.2 Design of Algorithm Steps

Making scheme to facilitate the research steps. Starting from the early design stages of data processing to final output and testing. IDS placement schemes on computer networks shown in figure 1. System design scheme of intrusion detection system can be seen in Figure 2.

Journal of Theoretical and Applied Information Technology
20th October 2015. Vol.80. No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

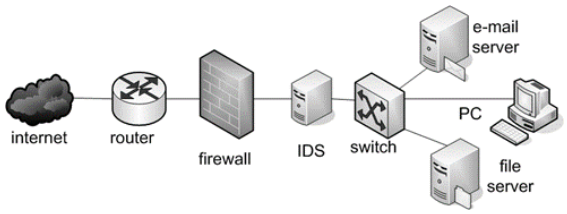

Figure 1: IDS Location in Computer Networking

3.3 Implementation

3.3.1 Preprocessing

This research used 125973 data from KDDTrain+ for training and 22544 data KDDTest+ for testing. Existing datasets which contains symbolic data should be codified for index feature 2nd, 3rd, 4th and targets in order to be processed by the system. Table 3 below shows the features and targets are codified :

Table 3: Codification of KDDTest+ and KDDTrain+

Feature Index	Feature Name	Codification
2	protocol_type	icmp = 1
		tcp = 2
		udp = 3
3	service	Aol = 1
		Z39_50 = 70
4	flag	OTH = 1
		SH = 11
42	attack_type	DoS = 1
		U2R = 2
		R2L = 3
		Probe = 4
		Normal = 5

'Service' Feature codified into numbers 1 through 70 with a sequence of ascending name as well as 'flag' features from 1 to 11.

3.3.2 Data Normalization

Training data that have been codified get into the next step, the step of data normalization using boxcox. At this step, training data that are not well distributed will be normalized so that the distribution of the data become normal. The goal is to achieved stability of data distribution and useful for adjusting data value with a range of activation functions used in the network.

3.3.3 Number of Clusters

The next process is to determine the exact number of clusters from training data using fuzzy validity index. Dataset will be clustered into 2-14 clusters then look for the most appropriate number of clusters by calculating and comparing the graphics fuzzy validity index Partition Coefficient (PC), Classification Entropy (CE), Partition Index (SC), Separation Index (S), Xie and Beni's Index (XB), Dunn's Index (DI) and Alternative Dunn Index (ADI).

3.3.4 Clustering Training Data

Training data with the results of normalization boxcox methods enter clustering step using possibilistic fuzzy c-means (PFCM). Training data is divided into several clusters according calculation from fuzzy validity index. Table 4 shows the parameters used by PFCM :

Table 4: PFCM Parameters

Parameter	Value
number of cluster (c)	*'according to fuzzy validity index result'*
exponent matrix U (*m*)	2
exponent matrix T (η)	2
smallest error (Ɛ)	0,00001
initial iteration (t)	1
maximum iteration (MaxIter)	100
user-defined constant (γ)	0,08
fuzzy constant (a)	1
probability constant (b)	4
initial objective function (P_0)	0

3.3.5 Training for Neural Network Ensemble

Training data that has been clustered with PFCM go into the next step, the step of training the neural network ensemble. Each cluster will be trained using two neural networks, Levenberg-Marquardt and Quasi Newton. The number of clusters obtained from fuzzy validity index to determine the total amount of artificial neural networks.

The number of layers are 2 layers. Activation function used in neural network hidden layer and output layer is sigmoid value in the range -1 to 1. Number of neurons in the hidden layer is 11 and 3 at the output layer. Number of hidden layer is taken from the calculation by equation 25 [24].

$$\sqrt{N_o * N_i} \quad (25)$$

N_o is the number of output neurons and N_i is the number of input neurons. Number of input neurons correspond to the number of variables/ features in the dataset is 41 and the number of output neurons are 3 as a representation of the number of categories of attacks and normal state. Figure 3 shows the structure used for each neural network.

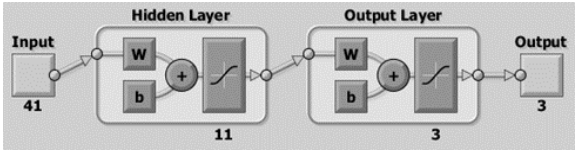

Figure 3: Artificial Neural Network Structure

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Levenberg-Marquardt ANN parameters used are shown in table 5 and Quasi Newton ANN in table 6 below :

Table 5: Levenberg-Marquardt ANN Parameters

Parameter	Value
maximum epoch	1000
LM parameter (μ)	0,001
beta factor (β)	10
maximum LM parameter (μ_{max})	10000000000
initial epoch	0
Goal	0
minimum gradien	0,00001
maximum fail	6

Table 6: Quasi Newton ANN Parameters

Parameter	Value
maximum epoch	1000
initial epoch	0
goal	0
minimum gradien	0,000001
maximum fail	6
alfa (α)	0,001
delta (d)	0,01

After the training of each cluster, single NN will be combine using simple average method. Weights in the training phase for each cluster is obtained, it is used to process the entire training data to generate the appropriate output. Equation 26 used to calculate simple average :

$$\hat{y}_t = \frac{\sum_{i=1}^{n} \hat{y}_{it}}{n} \quad (26)$$

The $\hat{y}_t$ is output predictor for NNE, $\hat{y}_{it}$ is the network output and *n* are number of network.

3.3.6 Final Output

Output results of training artificial neural network ensemble get into the next step, namely the final output using naive Bayesian method. Naive Bayesian algorithm consists of two phases, training and testing phases. The output of neural network ensemble will be the training data used to train the model with naive Bayesian classification. After that the model can be used to classify the testing of data that will be discussed at a later step. Here is the algorithm of the naive Bayesian classifier model training :

a. Form prediction of IDS results
b. For each category v_j, count :
   - ✓ Determine v_hsl_j (IDS result set that includes category v_j)
   - ✓ Calculate $P(v_j)$ by equation 27 :

$$P(v_j) = \frac{|v\_hsl_j|}{|hasilIDS|} \quad (27)$$

c. Calculate $P(w_k|v_j)$ with equation 28 for each prediction value from prediction of IDS results :

$$P(pred_i|v_j) = \frac{|pred_i|}{|v\_hsl_j|} \quad (28)$$

3.4 Testing

At this step, we used KDDTest+ for testing dataset. Testing data is entered into the neural network ensemble. The output of neural network then entered into the model to get the naive Bayesian prediction of intrusion detection system. Here is the algorithm of testing data using naive Bayesian models :

a. Calculate equation 29 :

$$P(v_j) \prod_i P(a_i|v_j) \quad (29)$$

for each category v_j

b. Determine the maximum value of above calculation as a result of output

After obtained the predicted results of data testing, then evaluation system done by using the calculation precision, recall and overall accuracy.

3.5 Documentation

Report writing step is to do the documentation on all matters relating to research, which includes the introductory chapter, the basic theory, research methodology, discussion, and cover.

4. RESULTS

4.1 Fuzzy Validity Index Calculation Results

In the implementation phase, fuzzy validity index is after preprocessing and data normalization. Fuzzy validity index is done to determine the exact number of clusters for the training data that subsequently used in clustering methods possibilistic fuzzy c-means. Fuzzy validity index calculations using data from a sample of 5% KDDTrain+. It is random data from each category.

Index Partition Coefficient (PC) and index Classification Entropy (CE) yields c = 3 (c value indicates the number of clusters). Partition Index (SC) and Separation Index (S) yields c = 11. Graph Xie and Beni's Index (XB) produces c = 8. Dunn's Index (DI) yields c = 10. Alternative Dunn's Index (ADI) yields c = 3.

Optimal cluster of training data is obtained by comparing seven indices mentioned above [17]. Number c = 3 appear at most in the index above so it can be concluded that the appropriate number of clusters for the training dataset is 3. This is evident from the results of the Partition Coefficient (PC),

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Classification Entropy (CE) and Alternative Dunn's Index (ADI).

4.2 PFCM Clustering Results

Data normalization using boxcox method enter to clustering step using possibilistic fuzzy c-means (PFCM). The smallest error (Ɛ) used was 0.00001, maximum iterations (maxiter) = 100 and the number of clusters = 3 (corresponding fuzzy validity index calculation in the previous step). Other parameters used in PFCM can be seen in the previous table, table 4. PFCM clustering results produced 3 clusters. The first cluster has 37101 data, the second cluster has 40183 data and the third cluster has 48689 data. Table 7 shows the composition of each cluster.

Table 7: PFCM Result

Cluster	Category	Number
Cluster 1	Normal	597
	DoS	34345
	U2R	0
	R2L	11
	Probe	2148
Cluster 2	Normal	23242
	DoS	11001
	U2R	17
	R2L	430
	Probe	5493
Cluster 3	Normal	43504
	DoS	581
	U2R	35
	R2L	554
	Probe	4015

The composition of the first cluster has 34345 number of DoS category data, which is the highest number compared to other categories. However, the first cluster does not have data from U2R category. The second cluster has majority data in the normal category with the number 23242. This is in contrast to the amount of data in U2R category, it is only 17 data. The third cluster has 43,504 normal category data, which is the highest number and least data in U2R category, it is 35 data. That is because the number of U2R attacks are fewer than other attacks.

4.3 Result of Training and Testing Neural Network Ensemble

Training data that has been clustered into 3 clusters enter to the neural network ensemble training. The training data processed by 2 method of artificial neural networks, namely Levenberg-Marquardt and Quasi-Newton, so the total artificial neural networks are used are 6. Calculation of data cluster fixing weights of each neural network.

Furthermore, all of the training data (KDDTrain+) get into artificial neural networks that have been trained. The output applied as testing data and it is used as input of Naive Bayesian method. Testing results of neural network ensemble with KDDTrain+ combined together using simple average method.

4.4 Naive Bayesian Classification Results

Simple average calculation results get into classification step using Naive Bayesian algorithms. This algorithm inputs from 3 simple average of each neural network ensemble and the target of KDDTrain+. After that the algorithm was tested with input from KDDTrain+.

4.5 Results of Testing IDS with KDDTest+

Once the training is done to determine the weights of neural network ensemble and training in naive Bayesian classification to determine the final output of the intrusion detection system, the next step is testing the system by using KDDTest+. Testing is done by entering KDDTest+ data into neural network ensemble.

Testing the data into six neural networks (Levenberg-Marquardt and Quasi-Newton). Testing data is processed using neural network with new weight that has been fixed. Then, the output of the artificial neural network combined with the simple average method. Furthermore, the neural network outputs are classified with Naive Bayesian method that has been trained. The results of this process are the specific type of attacks of each KDDTest+ data. Code of the type of attack can be seen in the previous table, Table 3.

KDDTest+ test results are evaluated using precision, recall and overall accuracy. Evaluation aims to determine how well the intrusion detection system that has been created. IDS detection results compared with actual data KDDTest+ targets. The full results of the detection can be seen in Table 8.

In Table 8 shows that the highest precision in the detection of DoS, from overall result in DoS detection, 89.82% is exactly DoS attacks. Then followed with R2L category (75.78%), normal (72.25%) and Probe/ PRB (70.70%). However, detection of U2R only reaches 14.62%.

Highest recall in the normal category, for 91.44% of actual normal data KDDTest+ can be detected precisely as normal data. Then the probe (87.11%), DoS (83.31%) and U2R (9.50%). R2L lowest results by 6.14%. Highest overall accuracy at U2R (98.70%). Next is Probe (94.74%), DoS (91.35%) and R2L (88.29%). Lowest result in the normal category (81.18%).

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Evaluation results of this research compared with previous researches using NSL-KDD dataset. Research from Tavalee and his colleagues [10] using 7 methods with the lowest accuracy results on SVM method and the highest at NB Tree. Panda and his colleagues [11] using the method of development of Naive Bayes with 25192 training data. These research only classify normal and attack regardless of the type of attack. Research from Govindarajan and his colleagues [13] using the method of SVM and RBF. Comparison of system accuracy in this research with the previous system can be seen in Table 9.

Table 9: Comparison of IDS Accuration

Metode	Accuracy
Multinomial Naive Bayes+N2B [11]	38,89 %
Multinomial Naive Bayes updateable+N2B [11]	38,94 %
SVM [10]	69,52 %
Naive Bayes [10]	76,56 %
Multi-layer Perceptron [10]	77,41 %
Random Forest [10]	80,67 %
J48 [10]	81,05 %
Discriminative Multinomial Naive Bayes+RP [11]	81,47 %
Random Tree [10]	81,59 %
NB Tree [10]	82,02 %
RBF [13]	83,57 %
RBF-SVM [13]	85,19 %
NN Ensemble (LM-QN)+Naive Bayes (proposed)	**90,85 %**

Other research from Najafi and Afsharchi [12], which employed Tree Augmented Naive Bayesian, Decision Tree, and SVM with NSL-KDD dataset. Good precision and recall results with an average of 90% more, but very bad for U2R categories, it was 0%. Based on Table 8, U2R precision was 14.62% and recall was 9.50% indicating that the IDS results of this research were able to improve the precision and recall from Najafi and Afsharchi.

5. CONCLUSIONS AND FUTURE DIRECTIONS

This research resulted in a new model of intrusion detection system that uses neural network ensemble with 2 algorithm, namely Levenberg-Marquardt and Quasi-Newton and using NSL-KDD dataset which is an improvement of the KDD'99 dataset. Highest precision in the detection of DoS (89.82%). Meanwhile, a low precision in U2R (14.62%). Recall is the highest in the normal category (91.44%), but a low recall results occurred in U2R category (9.50%) and R2L (6.14%). Low precision and recall due to false negatives and false positives of U2R and R2L is too big. Overall Accuracy of all categories can be said to be good with the lowest normal results (81.18%). The results also show that the neural network ensemble method produces better accuracy than previous research, that is 90.85%.

The research's results indicates that the new framework can overcome overfitting problem and outlier data using neural network ensemble (using Lavenberg-Marquardt and Quasi-Newton), Possibilistic Fuzzy C-Means and Naive Bayesian. This research also use NSL-KDD dataset so the evaluation get more precise results. It provides more detail evaluation using precision, recall and overall accuracy. However, it doesn't tested in real network traffic.

For further research can be carried out experiments on the number of clusters of training data. Simple average method at combining step in neural network ensemble can also be replaced by a different method. In addition, the changes in number of hidden layer neural network Levenberg-Marquardt and Quasi Newton can also be done.

REFRENCES :

[1] T.S. Bhatti, R.C. Bansal, and D.P. Kothari, "Reactive Power Control of Isolated Hybrid Power Systems", *Proceedings of International Conference on Computer Application in Electrical Engineering Recent Advances (CERA)*, Indian Institute of Technology Roorkee (India), February 21-23, 2002, pp. 626-632.

[2] B.N. Singh, Bhim Singh, Ambrish Chandra, and Kamal Al-Haddad, "Digital Implementation of an Advanced Static VAR Compensator for Voltage Profile Improvement, Power Factor Correction and Balancing of Unbalanced Reactive Loads", *Electric Power Energy Research*, Vol. 54, No. 2, 2000, pp. 101-111.

[3] Hoque, M. S., Mukit, M. A., & Bikas, M. A. (2012). An Implementation Of Intrusion Detection System Using Genetic Alagorithm. International Journal of Network Security & Its Applications (IJNSA), 109-120.

[4] Richardson, R. (2010). 2010/ 2011 CSI Computer Crime and Security Survey. Computer Security Institute.

[5] Faraoun, K. M., & Boukelif, A. (2007). Neural Networks Learning Improvement using the K-Means Clustering Algorithm to Detect Network Intrusions. World Academy of Science, 549-556.

[6] Hautam¨aki, Ville, Cherednichenko, Svetlana, K¨arkk¨ainen, Ismo, Kinnunen, Tomi, &

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Fr¨anti, Pasi. (2005). Improving K-Means by Outlier Removal. SCIA 2005, LNCS 3540, pp. 978–987.

[7] Wang, G., Hao, J., Ma, J., & Huang, L. (2010). A new approach to intrusion detection using Artificial Neural Networks and fuzzy clustering. Expert Systems with Applications, 1-8.

[8] Pal, N. R., Pal, K., Keller, J. M., & Bezdek, J. C. (2005). A Possibilistic Fuzzy c-Means Clustering Algorithm. IEEE Transaction On Fuzzy Systems, 517-530.

[9] Khairudin. (2004). Fuzzy Nonlinear Program. Fakultas Teknik Sipil dan Perencanaan Universitas Bung Hatta, Padang.

[10] Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009). A Detailed Analysis of the KDD CUP 99 Data Set. Proceedings of the 2009 IEEE Symposium on CISDA.

[11] Panda, M., Abraham, A., & Patra, M. R. (2010). Discriminative Multinomial Naive Bayes for Network Intrusion Detection. Sixth International Conference on Information Assurance and Security (IAS), 5-10.

[12] Najafi, R. & Afsharchi, Mohsen. (2012). Network Intrusion Detection Using Tree Augmented Naive-Bayes. CICIS'12, IASBS, Zanjan, Iran.

[13] Govindarajan, M. & Chandrasekaran, RM. (2012). Intrusion Detection using an Ensemble of Classification Methods. Proceedings of the World Congress on Engineering and Computer Science 2012 Vol I.

[14] Dewi, N. S., Sulandari, W., & Wibowo, S. (2012). Modeling Dollar Exchange Rate Against Rupiah Using Neural Network Ensembles (NNE). Prosiding Seminar Nasional Matematika, 317-322.

[15] Cho, S.-B., & Won, H.-H. (2007). Cancer Classification Using Ensemble Of Neural Networks. Springer Science+Business Media, LLC, 243-250.

[16] Paliwal, S., & Gupta, R. (2012). Denial-of-Service, Probing & Remote to User (R2L) Attack Detection using Genetic Algorithm. International Journal of Computer Applications, 57-62.

[17] Buthmann, A. (2010). Making Data Normal Using Box-Cox Power Transformation. Diambil dari iSixSigma: http://www.isixsigma.com/tools-templates/normality/making-data-normal-using-box-cox-power-transformation/.

[18] Balasko, B., Abonyi, J., & Feil, B. (n.d.). Fuzzy Clustering and Data Analysis Toolbox For Use with Matlab. Diambil dari http://www.sunfinedata.com/.

[19] Kusumadewi, S. (2004). Neural networks build Using MATLAB & EXCELLINK. Yogyakarta: Graha Ilmu.

[20] Soelaiman, R., & Chasiani, N. (2011). Application of Chaos Optimization Method And BFGS (Broyden, Fletcher, Goldfarb, And Shanno) On completion of Nonlinear Equations System Problems. Fakultas Teknologi Informasi Institut Teknologi Sepuluh Nopember, 1-8.

[21] Olson, D. L., & Delen, D. (2008). Advanced Data Mining Techniques. Verlag Berlin Heidelberg: Springer.

[22] Hamzah, A. (2012). Text classification with Naive Bayes classifier (NBC) For Grouping News And Abstrac Academic Texts. Prosiding Seminar Nasional Aplikasi Sains & Teknologi (SNAST) Periode III, 269-277.

[23] Shanmugavadivu, R., & Dr. Nagarajan, N. (2011). Network Intrusion Detection System using Fuzzy Logic. Indian Journal of Computer Science and Engineering (IJCSE), 101-111.

[24] Atiliani, A. (2013). Training Multilayer Perceptron Neural Network using Genetic Algorithm Levenberg Marquardt. Tesis: Informatics UNS, 1-7.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 1: NSL-KDD Features

Feature Index	Feature Name	Feature Index	Feature Name
1	Duration	21	Is_hot_login
2	Protocol_type	22	Is_guest_login
3	Service	23	Count
4	Flag	24	Srv_count
5	Src_bytes	25	Serror_rate
6	Dst_bytes	26	Srv_serror_rate
7	Land	27	Rerror_rate
8	Wrong_fragment	28	Srv_rerror_rate
9	Urgent	29	Same_srv_rate
10	Hot	30	Diff_srv_rate
11	Num_failed_logins	31	Srv_diff_host_rate
12	Logged_in	32	Dst_host_count
13	Num_compromised	33	Dst_host_srv_count
14	Root_shell	34	Dst_host_same_srv_rate
15	Su_attempted	35	Dst_host_diff_srv_rate
16	Num_root	36	Dst_host_same_src_port_rate
17	Num_file_creations	37	Dst_host_srv_diff_host_rate
18	Num_shells	38	Dst_host_serror_rate
19	Num_access_files	39	Dst_host_srv_serror_rate
20	Num_outbound_cmds	40	Dst_host_rerror_rate
		41	Dst_host_srv_rerror_rate

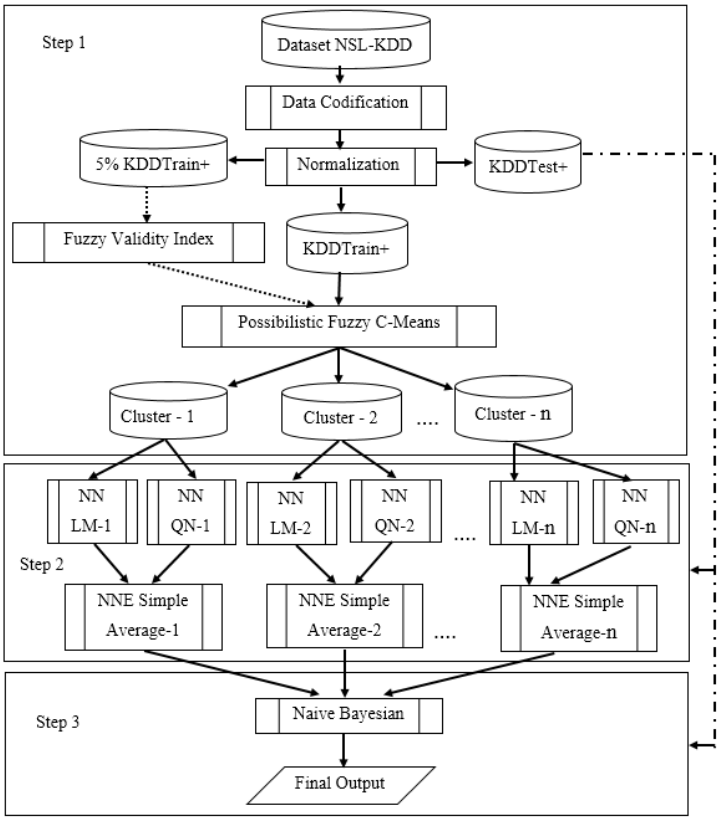

Figure 2: IDS Scheme

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 8: IDS Test Results

	TP	TN	FP	FN	Precision (%)	Recall (%)	Overall Accuracy (%)
DoS	6.213	14.382	704	1.245	89,82	83,31	**91,35**
U2R	19	22.233	111	181	14,62	9,50	**98,70**
R2L	169	19.736	54	2.585	75,78	6,14	**88,29**
Probe	2.109	19.249	874	312	70,70	87,11	**94,74**
Normal	8.880	9.422	3.411	831	72,25	91,44	**81,18**
Mean					64,63	55,50	**90,85**