# A Network Intrusion Detection Model Using Fuzzy C4.5 Decision Tree

Thuzar Hlaing, May Aye Khine
*University of Computer Studies, Yangon*
*thuzarhlaing85@gmail.com, maya.khine@gmail.com*

## Abstract

*With the growing rate of inter-connections among computer systems, reliable network communication is becoming a major challenge. Intrusion detection has emerged as a significant field of research, because it is not theoretically possible to set up a system with no vulnerabilities. This paper purposes the use of fuzzy logic to generate decision tree to classify the intrusion data. Further, the fuzzy decision tree is then converted to fuzzy rules. The fuzzy decision tree (C4.5) method is used the minimize measure of classification ambiguity for different attributes. This method overcomes the sharp boundary problems; provide good accuracy dealing with continuous attributes and prediction problems. The experimental result is carried out by using 10% KDD Cup 99 benchmark network intrusion detection dataset.*

Keywords: Fuzzy Logic, Fuzzy C4.5, Fuzzy Rules, Intrusion Detection

## 1. Introduction

Intrusion Detection System (IDS) is the science of detection of malicious activity on a computer network and the basic driver for network security. Network security compromises the three security tokens such as confidentiality, integrity, availability. Confidentiality is the task of preventing unauthorized disclosure of information. Integrity is the task of preventing unauthorized or accidental modification, creation or deletion of information. Availability is the task of providing access to information and services when access is needed.

In general, IDSs can be divided into two techniques: misuse detection and anomaly detection [1, 2]. Misuse detection refers to detection of intrusions that follow well-defined intrusion patterns. It is very useful in detection known attack patterns. Anomaly detection refers to detection performed by detecting changes in the patterns of utilization or behavior of the system. It can be used to detect known and unknown attack. The anomaly detection techniques have the advantage of detecting unknown attacks over the misuse detection technique [3]. Anomaly based intrusion detection using data mining algorithms such as decision tree (DT), naïve Bayesian classifier (NB), neural network (NN), support vector machine (SVM), k-nearest neighbors (KNN), fuzzy logic model, and genetic algorithm have been widely used by researchers to improve the performance of IDS [4][5].

This paper purposes fuzzy C4.5 which is used to estimate the accuracy and to detect the cyber attacks. The fuzzy decision tree method generates the rules which give the better understanding of the relationship between the parameters and prediction.

The rest of the paper is organized as follows: Section 2 reviews related work. Section 3 introduces about KDD dataset. Section 4 demonstrates a detailed description of the fuzzy C4.5 based on network intrusion detection system and in section 5, experimental study is presented. Finally, the paper is concluded with section 6.

## 2. Related Work

In recent times, intrusion detection has received a lot of interest among the researchers since it is widely applied for preserving the security within a network. Here, some of the techniques used for intrusion detection.

In 1980, the concept of intrusion detection began with Anderson's seminal paper [6]; he introduced a threat classification model that develops a security monitoring surveillance system based on detecting anomalies in user behavior. In 1986, Dr. Denning proposed several models for commercial IDS development based on statistics, Markov chains, time-series, etc [7]. In 2000, Valdes et al. [8] developed an anomaly based IDS that employed naïve Bayesian network to perform intrusion detecting on traffic bursts. In 2001, J.Gomez et al. [9] proposed a technique (genetic algorithm) to generate fuzzy rules (instead of manual design) that are able to detect anomalies.

In 2003, Kruegel et al. [10] proposed a multisensory fusion approach using Bayesian classifier for classification and suppression of false alarms that the outputs of different IDS sensors were aggregated to produce single alarm. In the same year, Shyu et al. [11] proposed an anomaly based intrusion detection scheme using principal components analysis (PCA), where PCA was applied to reduce the dimensionality of the audit data and arrive at a classifier that is a function of the principal components. In 2003, Yeung et al. [12] proposed an anomaly based intrusion detection using hidden Markov models that computes the sample likelihood of an observed sequence using the forward or backward algorithm for identifying anomalous. Dickerson et al. [13] developed the Fuzzy Intrusion Recognition Engine (FIRE) using fuzzy logic that process the network data and generate fuzzy sets for every observed feature and then the fuzzy sets are used to detect network attacks.

Huang, Pei and Goodman [14], where the general problem of GA optimized feature selection and extraction is addressed. In their paper, Huang, et al. applies a GA to optimize the feature weights of a KNN classifier and choose optimal subset of features for a Bayesian classifier and a linear regression classifier. Experiments in their paper show that the performance of all these three classifiers with feature weighing or selection by a GA is better than that of the same classifiers without a GA. They conclude that performance gain is completely dependent on what kind of classifier is used over what type of data set.

Srinivas and Sung [15] presented the use of support vector machine (SVM) to rank these extracted features, but this method needs many iterations and is very time-consuming. In the research of detection model generation, it is desirable that the detection model be explainable and have high detection rate, but the existing methods cannot achieve these two goals.

## 3. Introduction of KDD Dataset

The KDD Cup 1999 Intrusion Detection contest data KDD99 [16] is used in this experiments. This data was prepared by the 1998 DARPA Intrusion Detection Evaluation program by MIT Lincoln Labs. They acquired nine weeks of raw TCP dump data. For each TCP/IP connection, 41 various quantitative (continuous data type) and qualitative (discrete data type) features were extracted among the 41 features, 34 features are numeric and 7 features are symbolic. The data contains 22 attack types that could be classified into four main categories:

**1. Denial of Service (DOS**): In this type of attacks an attacker makes some computing or memory resources too busy or too full to handle legitimate requests, or denies legitimate users access to a machine.
**2. Remote to User (R2L):** In this type of attacks an attacker who does not have an account on a remote machine sends packets to that machine over a network and exploits some vulnerability to gain local access as a user of that machine.
**3. User to Root (U2R):** In this type of attacks an attacker starts out with access to a normal user account on the system and is able to exploit vulnerability to gain root access to the system.
**4. Probing:** In this type of attacks an attacker scans a network of computers to gather

information or find known vulnerabilities. An attacker with a map of machines and services that available on a network can use this information to look for exploits.

**Table 1. Input attributes in KDD 99 Dataset**

| No | Input Attribute | Type | No | Input Attribute | Type |
|---|---|---|---|---|---|
| 1 | duration | Con. | 22 | is_guest_login | Dis. |
| 2 | protocol_type | Dis. | 23 | count | Con. |
| 3 | service | Dis. | 24 | srv_count | Con. |
| 4 | flag | Dis. | 25 | serror_rate | Con. |
| 5 | src_bytes | Con. | 26 | srv_serror_rate | Con. |
| 6 | dst_bytes | Con. | 27 | rerror_rate | Con. |
| 7 | land | Dis. | 28 | srv_rerror_rate | Con. |
| 8 | wrong_fragment | Con. | 29 | same_srv_rate | Con. |
| 9 | urgent | Con. | 30 | diff_srv_rate | Con. |
| 10 | hot | Con. | 31 | srv_diff_host_rate | Con. |
| 11 | num_failed_logins | Con. | 32 | dst_host_count | Con. |
| 12 | logged_in | Dis. | 33 | dst_host_srv_count | Con. |
| 13 | num_compromised | Con. | 34 | dst_host_same_srv_rate | Con. |
| 14 | root_shell | Con. | 35 | dst_host_diff_srv_rate | Con. |
| 15 | su_attempted | Con. | 36 | dst_host_same_src_port_rate | Con. |
| 16 | num_root | Con. | 37 | dst_host_srv_diff_host_rate | Con. |
| 17 | num_file_creations | Con. | 38 | dst_host_serror_rate | Con. |
| 18 | num_shells | Con. | 39 | dst_host_srv_serror_rate | Con. |
| 19 | num_access_files | Con. | 40 | dst_host_rerror_rate | Con. |
| 20 | num_outbound_cmds | Con. | 41 | dst_host_srv_rerror_rate | Con. |
| 21 | is_hot_login | Dis. | - | - | - |

**Table 2. Classes in the 10% of the KDD Cup 99 data set**

| CLASS | SUB-CLASS | SAMPLES |
|---|---|---|
| Normal | | 97278 |
| Dos | Back, land, Neptune, pod, smurf, teardrop | 391458 |
| U2r | Buffer_overflow, perl, load_module, rootkit | 52 |
| R2l | ftp_write, imap, phf, guess_passwd,spy, multihop,warezclient, warezmaster | 1126 |
| Probing | Satan,ipsweep, nmap, portsweep | 4107 |
| | **TOTAL** | 494021 |

## 4. Fuzzy C4.5 Decision Tree for NIDS

Network intrusion detection systems (NIDSs) have become important and widely used tools for ensuring network security. Machine learning is a valuable tool for intrusion detection that offers a major opportunity to improve quality of IDs.

### 4.1 Proposed Framework

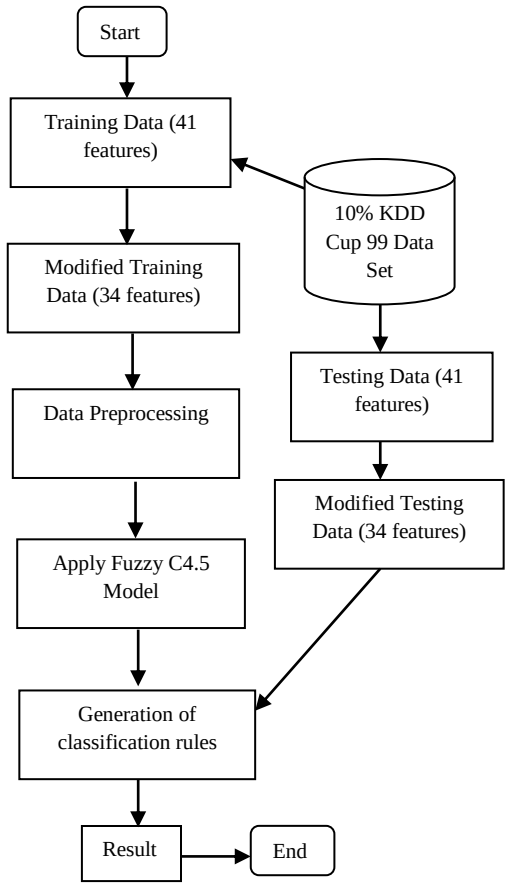



**Figure 1. Overview of the Proposed Framework**

The detailed analysis of KDD data set is given in section 3. Based on the analysis, the KDD data set contains four types of attacks and normal behavior data with 41 attributes that have both continuous and discrete attributes. The proposed system is designed only for the continuous attributes because the major attributes in KDD data set are continuous in nature. Therefore, the proposed system have taken only the continuous attributes for instance, 34 attributes from the input dataset by removing

discrete attributes. So, the proposed system is used the modified 34 features for training data and data preprocessing performs a transformation on the modified training data. And then, the training data are analyzed by fuzzy c4.5 decision tree algorithm and the classifier is represented in the form of classification rules. Test data are used to estimate the accuracy of the classification rules.

### 4.2 Fuzzy C4.5

begin
1. Start with examples set of entry, having the weights of the examples (in root node) equalto1.
2. At any node N still to be expanded, compute the number of examples of each class. The examples are distributed in part or in whole by branches. The distributed amount of each example to a branch is obtained as the product of its current weight and the membership degree to the node.
3. Compute the standard information content.
4. At each node search the set of remaining attributes to split the node.
    4.1. Select with any criteria, the candidate attributes set to split the node.
    4.2. Compute the standard information content to each child node obtained from each candidate attribute.
    4.3. Select the candidate attribute such that information gain is maximal.
5. Divide N in sub-nodes according to possible outputs of the attribute selected in the previous step.

6. Repeat steps 2-5 to stop criteria are satisfied in all nodes.
end.

**Figure 2. Fuzzy C4.5 Algorithm**

Decision tree induction has been widely used in extracting knowledge from feature-based examples for classification and decision making [17]. C4.5 is the algorithm proposed by R.Quinlanin 1993 [18] for building a decision tree. The C4.5 decision tree divides data items into subsets, based on attribute. If an attribute maximizes the gain ratio when dividing data into categories, it is considered useful for producing a decision tree.

Fuzzy logic is appropriate for the intrusion detection problem for two major reasons. First, many quantitative features are involved in intrusion detection. The second motivation for using fuzzy logic to address the intrusion detection problem is that security itself includes fuzziness. The use of fuzziness in representing these quantitative features helps to smooth the abrupt separation of normality and abnormality and provides a measure of the degree of normality or abnormality of a particular measure.

A common C4.5 algorithm classifies data into categories based on information gain ratio. An entropy info(S) of a set of data items, represented as S = {s1, s2,..., sx}, is given by (1).

$$info(s) = -\sum_{i=1}^{k}\left\{\frac{freq(C_i,S)}{|S|} * log_z \frac{freq(C_i,S)}{|S|}\right\} \quad (1)$$

The maximum value of the integer index x is the number of data items belonging to S. |S| is the sum of the sums of the possibilities for each class of all the examples included in the set S. The integer index k is the number of categories into which the classes divide the data. Here, freq ($C_i$, S) is the sum of the possibilities of belonging to class $C_i$ of all the examples and is given by (2).

$$freq(C_i,S) = \sum_{h=1}^{x} \mu\,(C_i, S_h) \quad (2)$$

The membership functions used to calculate the possibility of belonging to each "preference" data class. Here, the possibility of belonging to class Ci of the data item s is presented as μ ($C_i$, s).

An entropy info$X_p$ (T), where examples belonging to the set T are divided into some subset $T_j$ (j: 1 − n) by an attribute $X_p$, is given by (3). Entropy info ($T_j$) is given by (4).

$$info_{x_p}(T) = \sum_{j=1}^{n} \left\{ \frac{|T_j|}{|T|} * info(T_j) \right\} \quad (3)$$

$$info(T_j) = \sum_{i=1}^{k} \left\{ \frac{freq(C_i, T_j)}{|T_j|} * log_2 \frac{freq(C_i, T_j)}{|T_j|} \right\} \quad (4)$$

An attribute $X_p$ divides data into fuzzy sets $T_j$ (j :1−n) and gives a possibility grade $\mu(T_j, s_h)$ (h :1 − x).The sum of the possibilities belonging to class Ci for the examples belonging to the subset $T_j$ , represented as freq $(C_i,T_j)$, is given by (5).

$$freq(C_i, T_j) = \sum_{h=1}^{x} \{ \mu(C_i, S_h) * \mu(T_j, S_h) \} \quad (5)$$

The information gain of an attribute $X_p$ shown as gain $(X_p)$ is calculated by (6). This describes a reduction in information entropy where the set T is divided into subsets Tj (j :1−n) by attribute $X_p$.

$$gain(X_p) = info(T) - info_{x_p}(T) \quad (6)$$

The amount of split information splitinfo $(X_p)$ is calculated by (7).

$$splitinfo(X_p) = -\sum_{j=1}^{n} \frac{|T_j|}{|T|} * log_2 \left( \frac{|T_j|}{|T|} \right) \quad (7)$$

Here, a set T is divided into n subsets by an attribute $X_p$. The information gain ratio, gain ratio $(X_p)$is calculated by (8).

$$gain\,ratio(X_p) = gain(X_p) / splitinfo(X_p) \quad (8)$$

## 4.3 Data Preprocessing

The original 10% KDD Cup data set where each numerical value in the data set is normalized between 0.0 and 1.0 according to the following equation:

$$x = \frac{x - MIN}{MAX - MIN} \quad (9)$$

Where, x is the numerical value, MIN is the minimum value for the attribute that x belongs to and MAX is the maximum value.

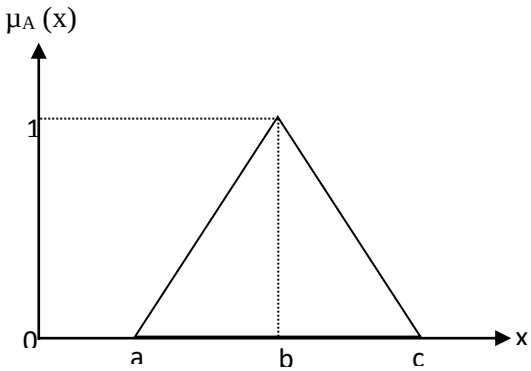
**Table 3. Normalization of the 10% KDD Cup 99**

| duration | Src_bytes | dst_bytes | count | Srv_count | Dst_host_count | Dst_host_srv_count |
|---|---|---|---|---|---|---|
| 184 | 260 | 1837 | 21 | 11 | 24 | 159 |
| 305 | 241 | 261 | 9 | 2 | 34 | 169 |
| 257 | 257 | 818 | 2 | 12 | 44 | 179 |

| duration | Src_bytes | Dst_bytes | count | Srv_count | Dst_host_count | Dst_host_srv_count |
|---|---|---|---|---|---|---|
| 0.003 | 0.260 | 0.184 | 0.041 | 0.022 | 0.094 | 0.623 |
| 0.005 | 0.241 | 0.026 | 0.018 | 0.004 | 0.133 | 0.663 |
| 0.004 | 0.257 | 0.082 | 0.004 | 0.023 | 0.173 | 0.702 |

## 4.4 Fuzzification of Numerical Numbers

Fuzzification is a process of fuzzifying numerical numbers into linguistic terms, which is often used to reduce information overload in human decision making process [19]. In this paper, triangular membership functions are used to represent fuzzy sets because of its simplicity, easy comprehension, and computational efficiency. Membership functions are usually predefined by experienced experts. The triangular membership function is denoted as $\mu_A$ (x) and is defined as:



$$\mu_A(x) = \begin{cases} 0 & if\ x \le a \\ \dfrac{x-a}{b-a} & if a \le x < b \\ \dfrac{c-x}{c-b} & if\ b \le x < c \\ 0 & if x \ge c \end{cases}$$

a, b and c represent the x coordinates of the three vertices of $\mu_A(x)$ in a fuzzy set A. In this paper, five fuzzy membership values (Low, Medium Low, Medium, Medium High, and High) are produced for each course score according to the predefined membership functions.

(0, 0.166, 0.333)----------Low
(0.166, 0.333, 0.5)-------Medium Low
(0.333, 0.5, 0.666)-------Medium
(0.5, 0.666, 0.833)-------Medium High
(0.666, 0.833, 1)----------High

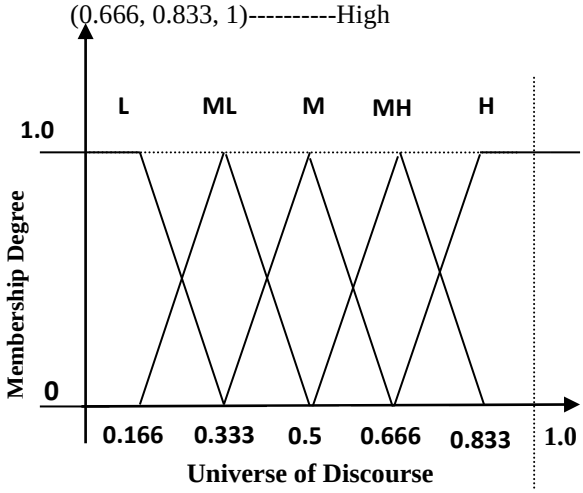



**Figure 3. Fuzzy space with five fuzzy sets**

If the input data value is 0.623, the degree of membership will calculate using triangular formula:

$$\mu_A(x) = \frac{c-x}{c-b} = \frac{0.666-0.623}{0.666-0.5} = \frac{0.043}{0.166} = 0.259$$

## 5. Experimental Result

All experiments were performed using a 2.20GHZ Dual-Core Processor and 2GB of RAM running windows 7. In the International Knowledge Discovery and Data Mining Tools Competition only "10% KDD" dataset is employed for the purpose of training. 10% KDD training dataset consists of relatively 494021 records.

Due to the huge number of audit data records in the 10% KDD99 data set, this paper is evaluated on a subset of 10% KDD dataset by random sampling 55,285 audit records for the training phase and 35,148 records for the testing phase.

**Table 4.Training and Testing Dataset Taken for Experimentation**

| Attack Types | Training Dataset | Testing Dataset |
|---|---|---|
| Normal | 25000 | 14863 |
| Dos | 25000 | 15000 |
| U2r | 52 | 52 |
| R2l | 1126 | 1126 |
| Probing | 4107 | 4107 |
| **Total** | 55,285 | 35,148 |

The proposed approach was able to generate simple classification rules. The following are some classification rules that were evolved in a sample run:

If su_attempted='low' and dst_host_srv_rerror_rate='low' and dst_host_srv_diff_host_rate='low' and dst_host_diff _srv_rate ='low' and hot='low' and num_shells='low' and dst_ host_count='low' and dst_host_srv_serror_rate='low' and root _shell='low' and wrong_fragment='low' and dst_ host_ same_src_ port_ rate= 'mediumHigh' and duration='low' and same_srv_rate='high' and dst_host_srv_count='low' and srv_ diff_host_rate='low' and dst_ host _same _srv _rate ='high' and src_bytes='mediumLow' and dest_bytes='low' and urgent='low' and num_failed_logins='low' and num_ compromised='low' and num_root='low' and num_file _ creations='low' and num_access_files='low' and count = 'low' and srv_count='low' and serror_rate='low' and srv_serror _rate='low' and rerror_rate='low' and srv_ rerror_rate='low' and diff_srv_rate='low' and dst_host_ serror_rate= 'mediumLow' Then Class is **probing**

If su_attempted='low' and dst_host_srv _ rerror_ rate='low' and dst_host _srv_ diff_ host_ rate='low' and dst_host_ diff_srv_rate= 'low' and hot='low' and num_shells='low' and dst_host_ count='low' and dst_host_srv_serror_rate='low' and root_shell = 'low' and wrong_fragment='low' and dst_ host _same_src _ port _rate='mediumHigh' and duration= 'low'

and same_srv_rate='high' and dst_ host_srv_count='low' and srv_diff_ host_rate = 'medium Low' Then Class is **normal**

If su_attempted='low' and dst_host_srv _rerror _ rate='low' and dst_host_srv_diff_host_rate='low' and dst_host_diff_ srv_rate='low' and hot='low' and num_shells='low' and dst_host_count='low' and dst_host_ srv_serror_rate='low' and root_shell='low' and wrong_fragment='low' and dst_host _same_ src_port _rate='high' and duration='low' and src_byte ='low' and same_srv_rate='high' and rerror_rate='low' and dst_host_srv_count='low' and dest_bytes ='medium' and urgent='low' and num_failed_logins='low' and num_ compromised='low' and num_root='low' and num_file_ creations='low' and num_access_files='low' and count='low' and srv_count='low' and serror _rate ='low' and srv_ serror_rate='low' and srv_rerror rate = 'low' and diff_ srv_rate='low' and srv_diff_ host_rate ='low' and dst_ host_ same_srv_rate='low' Then Class is **u2r**

If su_attempted='low' and dst_host_srv_ rerror _ rate='low' and dst_host_srv_diff_host_rate='low' and dst_host_ diff_ srv_rate='low' and hot='low' and num_ shells='low' and dst_ host_count='low' and dst_host_srv_serror_rate='low' and root_shell='low' and wrong_ fragment='low' and dst_host_ same_src_port_rate='high' and duration='low' and src_ bytes= 'low' and same_srv_rate='high' and rerror_rate='low' and dst_host_srv_count='low' and dest_bytes='high' and urgent= 'low' and num_failed_logins = 'low' and num_ compromised= 'low' and num_root='low' and num_file_ creations = 'low' and num_access_ files= 'low' and count='low' and srv_count= 'low' and serror_ rate='low' and srv_serror_rate='low' and srv_ rerror_rate='low' and diff_srv_rate='low' and srv_diff _host_rate= 'low' and dst_host_ same_srv_ rate= 'medium' Then Class is **r2l**

If su_attempted='low' and dst_host_ srv_rerror_ rate= 'low' and dst_host_srv_diff_host_rate='low' and dst_host_diff_ srv_rate= 'low' and hot='low' and num_ shells='low' and dst_host_count ='low' and dst_host_ srv_serror_rate='low' and root_shell='low' and wrong_ fragment='mediumLow' Then Class is **dos**

Besides, the classification accuracy is used to estimate the performance of IDS which is given as bellow:

$$Classification\ Rate = \frac{Number\ of\ classified\ patterns}{Total\ number\ of\ patterns} * 100\%$$

Table 5 compare the different algorithm performance, the total performance of proposed algorithm is better than other algorithm.

**Table 5. Different algorithms Performances**

| Algorithm | Accuracy |
|---|---|
| Fuzzy C4.5 | 97.886% |
| Fuzzy Logic[20] | 94.6% |
| C4.5[21] | 93.67% |

## 6. Conclusion

As security incidents become more numerous, IDS tools are becoming increasingly necessary. It is very likely that IDS capabilities will become core capabilities of network infrastructure (such as routers, bridges and switches) and operating systems. A fuzzy C4.5 Model was designed to build the system more accurate for attack detection, using fuzzy logic based on numeric numbers . By analyzing the result, the overall performance of the proposed system is improved significantly and it achieves more than 97% accuracy.

## References


[1] E. Biermann, E. CloeteandL.M.Venter,"A comparison of intrusion detection Systems", Computer and Security, vol.20, pp.676-683, 2001.

[2] T.Verwoerd and R.Hunt,"Intrusion detection techniques and approaches", Computer Communications, vol.25, pp.1356-1365, 2002

[3] E. Lundin and E. Jonsson, "Anomaly-based intrusion detection: privacy concerns and other problems", Computer Networks, vol.34, pp.623-640, 2002.

[4] Barbara, Daniel, Couto, Julia, Jajodia, Sushil, Popyack, Leonard, Wu, and Ningning, “ADAM: Detecting intrusion by data mining,” IEEE Workshop on Information Assurance and Security, West Point, New York, June 5-6, 2001.

[5] T. Shon, J. Seo, and J.Moon, “SVM approach with a genetic algorithm for network intrusion detection,” In Proc. of 20th International Symposium on Computer and Information Sciences (ISCIS 2005), Berlin: Springer-Verlag, 2005, pp. 224-233.

[6] James P. Anderson, “Computer security threat monitoring and surveillance,” Technical Report
98-17, James P. Anderson Co., Fort Washington, Pennsylvania, USA, April 1980.
[7] Dorothy E. Denning, “An intrusion detection model,” IEEE Transaction on Software Engineering, SE-13(2), 1987, pp. 222-232.
[8] A. Valdes, K. Skinner, “Adaptive model-based monitoring for cyber attack detection,” in Recent
Advances in Intrusion Detection Toulouse, France, 2000, pp. 80-92.
[9] J.Gomez and D.Dasgupta, "Evolving Fuzzy Classifiers for Intrusion Detection", Proceeding of the IEEE Workshop on Information Assurance, United States Military Academy, June 2001.
[10] C. Kruegel, D. Mutz, W. Robertson, F. Valeur, “Bayesian event classification for intrusion detection,” in Proc. of the 19th Annual Computer Security Applications Conference, Las Vegas, NV, 2003.
[11] M.L. Shyu, S.C. Chen, K. Sarinnapakorn, L. Chang, “A novel anomaly detection scheme based
on principal component classifier,” in Proc. of the IEEE Foundations and New Directions of Data Mining Workshop, Melbourne, FL, USA, 2003, pp. 172-179.
[12] D. Y. Yeung, and Y. X. Ding, “Host-based intrusion detection using dynamic and static behavioral models,” Pattern Recognition, 36, 2003, pp. 229-243.
[13] J.E. Dickerson, J.A. Dickerson, “Fuzzy network profiling for intrusion detection,” In Proc. of the
19th International Conference of the North American Fuzzy Information Processing Society (NAFIPS), Atlanta, GA, 2000, pp. 301-306.
[14] Huang, Z., Pei, M., Goodman, E., Huang, Y., and Li, G.Genetic algorithm optimized feature transformation: a comparison with different classifiers. In Proc. GECCO 2003, pp. 2121-2133.
[15] Srinivas, M., Sung, A., “Feature Ranking and Selection for Intrusion Detection”. Proceedings of the International Conference on Information and Knowledge Engineering, 2002.
[16]KDD CUP 1999 DATASET: http://kdd.ics.uci.edu/databases/kddcup99/
[17] T.M.Mitchell, Machine Learning, the McGraw-Hill Co., 414p, 1997.
[18]J.R.Quinlan.C4.5:ProgramforMachineLearning.MorganKaufmannPublishers, 1993.
[19] M. R. Civanlar and H. J. Trussell, “Constructing membership functions using statistical data,” Fuzzy Sets and Systems, vol. 18, 1986, pp. 1-14.
[20] R.Shanmugavadivu, N.Nagarajan,”An Anomaly-Based Network Intrusion Detection System Using Fuzzy Logic”. International Journal of Computer Science and Information Security, Vol.8, No.8, November 2010.
[21] K.Saravanan,” An Efficient Detection Mechanism for Intrusion Detection Systems Using Rule Learning Method”. International Journal of Computer and Electrical Engineering, Vol.1, No.4, October, 2009.