# NETWORK INTRUSION DETECTION SYSTEM USING FUZZY LOGIC

**R. Shanmugavadivu**
Assistant professor, Department of Computer Science
PSG College of Arts & Science, Coimbatore-14

**Dr.N.Nagarajan**
Principal, Coimbatore Institute of Engineering and Information Technology, Coimbatore.

**ABSTRACT**

IDS which are increasingly a key part of system defense are used to identify abnormal activities in a computer system. In general, the traditional intrusion detection relies on the extensive knowledge of security experts, in particular, on their familiarity with the computer system to be protected. To reduce this dependence, various data-mining and machine learning techniques have been used in the literature. In the proposed system, we have designed fuzzy logic-based system for effectively identifying the intrusion activities within a network. The proposed fuzzy logic-based system can be able to detect an intrusion behavior of the networks since the rule base contains a better set of rules. Here, we have used automated strategy for generation of fuzzy rules, which are obtained from the definite rules using frequent items. The experiments and evaluations of the proposed intrusion detection system are performed with the KDD Cup 99 intrusion detection dataset. The experimental results clearly show that the proposed system achieved higher precision in identifying whether the records are normal or attack one.

***Keywords:*** Intrusion Detection System (IDS), Anomaly based intrusion detection, Fuzzy logic, Rule learning, KDD Cup 99 dataset.

## 1. INTRODUCTION

Intrusion incidents to computer systems are increasing because of the commercialization of the Internet and local networks [1]. Computer systems are turning out to be more and more susceptible to attack, due to its extended network connectivity. The usual objective of the aforesaid attacks is to undermine the conventional security processes on the systems and perform actions in excess of the attacker's permissions. These actions could encompass reading secure or confidential data or just doing vicious destruction to the system or user files [2]. A system security operator can detect possibly malicious behaviors as they take place by setting up intricate tools, which incessantly monitors and informs activities [22]. Intrusion detection systems are turning out to be progressively significant in maintaining adequate network protection [1, 3, 4, 5]. An intrusion detection system (IDS) watches networked devices and searches for anomalous or malicious behaviors in the patterns of activity in the audit stream [6]. Capability of discriminating between standard and anomalous user behaviors should be present in a good intrusion detection system [7]. This would comprise of any event, state, content, or behavior that is regarded as abnormal by a pre-defined criterion [8].

Intrusion detection has emerged as a significant field of research, because it is not theoretically possible to set up a system with no vulnerabilities [9]. One main confrontation in intrusion detection is that we have to find out the concealed attacks from a large quantity of routine communication activities [10]. Several machine learning (ML) algorithms, for instance Neural Network [11], Support Vector Machine [12], Genetic Algorithm [13], Fuzzy Logic [14], and Data Mining [15] and more have been extensively employed to detect intrusion activities both known and unknown from large quantity of complex and dynamic datasets. Generating rules is vital for IDSs to differentiate standard behaviors from strange behavior by examining the dataset which is a list of tasks created by the operating system that are registered into a file in historical sorted order [16]. Various researches with data mining as the chief constituent has been carried to find out newly encountered intrusions [17]. The analysis of data to determine relationships and discover concealed patterns of data which otherwise would go unobserved is known as data mining. Many researchers have used data mining to focus into the subject of database intrusion detection in databases [18].

According to the detection strategy used, data mining-based intrusion detection systems can be classified into two main categories [23]. They are misuse detection which identifies intrusions using patterns of well known intrusions or weak spots of the system and anomaly detection, which attempts to find out if departure from the recognized standard usage patterns can be flagged as attacks [19]. (a) **Misuse Detection**: On the basis of the

impressions of known intrusions and known system weaknesses misuse detection tries to model abnormal activities. (b) **Anomaly Detection**: Both user and system behavior can be predicted using normal behavior patterns. Anomaly detectors identify possible attack attempts by constructing profiles representing normal usage and then comparing it with current behavior data to find out a likely mismatch [20]. For specified, well-known intrusion excellent detection results are achieved by signature-based methods. But, they cannot find out unfamiliar intrusions though constructed as a least alteration of previously known attacks. Conversely, the capability of discovering intrusion events which are previously unobserved is the main advantage of anomaly-based detection techniques [21].

In the proposed system, we have designed anomaly based intrusion detection using fuzzy logic. The input to the proposed system is KDD Cup 1999 dataset, which is divided into two subsets such as, training dataset and testing dataset. At first, the training dataset is classified into five subsets so that, four types of attacks (DoS (Denial of Service), R2L (Remote to Local), U2R (User to Root), Probe) and normal data are separated. After that, we simply mine the 1-length frequent items from attack data as well as normal data. These mined frequent items are used to find the important attributes of the input dataset and the identified effective attributes are used to generate a set of definite and indefinite rules using deviation method. Then, we generate fuzzy rule in accordance with the definite rule by fuzzifying it in such a way, we obtain a set of fuzzy if-then rules with consequent parts that represent whether it is a normal data or an abnormal data. These rules are given to the fuzzy rule base to effectively learn the fuzzy system. In the testing phase, the test data is matched with fuzzy rules to detect whether the test data is an abnormal data or a normal data.

The rest of the paper is organized as follows: section 2 describes the detailed analysis of the KDD cup 99 dataset. The proposed intrusion detection system using fuzzy logic is given in section 3. Experimentation and performance analysis of the proposed system is discussed in section 4. Finally, the conclusion is given in section 5.

## 2. KDD CUP 99 DATASET

In 1998, DARPA in concert with Lincoln Laboratory at MIT launched the DARPA 1998 dataset for evaluating IDS [36]. The DARPA 1998 dataset contains seven weeks of training and also two weeks of testing data. In total, there are 38 attacks in training data as well as in testing data. The refined version of DARPA dataset which contains only network data (i.e. Tcpdump data) is termed as KDD dataset [37]. The Third International Knowledge Discovery and Data Mining Tools Competition were held in colligation with KDD-99, the Fifth International Conference on Knowledge Discovery and Data Mining. KDD dataset is a dataset employed for this Third International Knowledge Discovery and Data Mining Tools Competition. KDD training dataset consists of relatively 4,900,000 single connection vectors where each single connection vectors consists of 41 features and is marked as either normal or an attack, with exactly one particular attack type [38]. These features had all forms of continuous and symbolic with extensively varying ranges falling in four categories:

- In a connection, the first category consists of the ***intrinsic*** features which comprises of the fundamental features of each individual TCP connections. Some of the features for each individual TCP connections are duration of the connection, the type of the protocol (TCP, UDP, etc.) and network service (http, telnet, etc.).
- The ***content*** features suggested by domain knowledge are used to assess the payload of the original TCP packets, such as the number of failed login attempts.
- Within a connection, the ***same host*** features observe the recognized connections that have the same destination host as present connection in past two seconds and the statistics related to the protocol behavior, service, etc are estimated.
- The ***similar same service*** features scrutinize the connections that have the same service as the current connection in past two seconds.

A variety of attacks incorporated in the dataset fall into following four major categories: **Denial of Service Attacks:** A denial of service attack is an attack where the attacker constructs some computing or memory resource fully occupied or unavailable to manage legitimate requirements, or reject legitimate users right to use a machine. **User to Root Attacks:** User to Root exploits are a category of exploits where the attacker initiate by accessing a normal user account on the system (possibly achieved by tracking down the passwords, a dictionary attack, or social engineering) and take advantage of some susceptibility to achieve root access to the system. **Remote to User Attacks:** A Remote to User attack takes place when an attacker who has the capability to send packets to a machine over a network but does not have an account on that machine, makes use of some vulnerability to achieve local access as a user of that machine. **Probes:** Probing is a category of attacks where an attacker examines a network to collect information or discover well-known vulnerabilities. These network

investigations are reasonably valuable for an attacker who is staging an attack in future. An attacker who has a record, of which machines and services are accessible on a given network, can make use of this information to look for fragile points.

Table 1 illustrates a number of attacks falling into four major categories and table 2 presents a complete listing of a set of features characterized for the connection records.

| | |
|---|---|
| Denial of Service Attacks | Back, land, neptune, pod, smurf, teardrop |
| User to Root Attacks | Buffer_overflow, loadmodule, perl, rootkit, |
| Remote to Local Attacks | Ftp_write, guess_passwd, imap, multihop, phf, spy, warezclient, warezmaster |
| Probes | Satan, ipsweep, nmap, portsweep |

Table 1. Various types of attacks described in four major categories

| **Feature index** | **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. | symbolic |
| 3 | service | network service on the destination, e.g., http, telnet, etc. | symbolic |
| 4 | flag | normal or error status of the connection | symbolic |
| 5 | src_bytes | number of data bytes from source to destination | continuous |
| 6 | dst_bytes | number of data bytes from destination to source | continuous |
| 7 | Land | 1 if connection is from/to the same host/port; 0 otherwise | symbolic |
| 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_logins | number of failed login attempts | Continuous |
| 12 | logged_in | 1 if successfully logged in; 0 otherwise | Symbolic |
| 13 | num_compromised | number of ``compromised'' conditions | Continuous |
| 14 | root_shell | 1 if root shell is obtained; 0 otherwise | Continuous |
| 15 | su_attempted | 1 if ``su root'' command attempted; 0 otherwise | Continuous |
| 16 | num_root | number of ``root'' accesses | Continuous |
| 17 | num_file_creations | number of file creation operations | Continuous |
| 18 | num_shells | number of shell prompts | Continuous |
| 19 | num_access_files | 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 | Symbolic |
| 22 | is_guest_login | 1 if the login is a ``guest'' login; 0 otherwise | Symbolic |
| 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_rerror_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 | dst_host_count | count for destination host | continuous |
| 33 | dst_host_srv_count | srv_count for destination host | continuous |

| | | | |
|---|---|---|---|
| 34 | dst_host_same_srv_rate | same_srv_rate for destination host | continuous |
| 35 | dst_host_diff_srv_rate | diff_srv_rate for destination host | continuous |
| 36 | dst_host_same_src_port_rate | same_src_port_rate for destination host | continuous |
| 37 | dst_host_srv_diff_host_rate | diff_host_rate for destination host | continuous |
| 38 | dst_host_serror_rate | serror_rate for destination host | continuous |
| 39 | dst_host_srv_serror_rate | srv_serror_rate for destination host | continuous |
| 40 | dst_host_rerror_rate | rerror_rate for destination host | continuous |
| 41 | dst_host_srv_rerror_rate | srv_serror_rate for destination host | continuous |

Table 2. A complete list of features given in KDD cup 99 dataset

## 3. NETWORK INTRUSION DETECTION SYSTEM USING FUZZY LOGIC

Presently, it is unfeasible for several computer systems to affirm security to network intrusions with computers increasingly getting connected to public accessible networks (e.g., the Internet). In view of the fact that there is no ideal solution to avoid intrusions from event, it is very significant to detect them at the initial moment of happening and take necessary actions for reducing the likely damage [32]. One approach to handle suspicious behaviors inside a network is an intrusion detection system (IDS). For intrusion detection, a wide variety of techniques have been applied specifically, data mining techniques, artificial intelligence technique and soft computing techniques. Most of the data mining techniques like association rule mining, clustering and classification have been applied on intrusion detection, where classification and pattern mining is an important technique. Similar way, AI techniques such as decision trees, neural networks and fuzzy logic are applied for detecting suspicious activities in a network, in which fuzzy based system provides significant advantages over other AI techniques.

Recently, several researchers focused on fuzzy rule learning for effective intrusion detection using data mining techniques. By taking into consideration these motivational thoughts, we have developed a fuzzy rule based system in detecting the attacks. This system, anomaly-based intrusion detection makes use of effective rules identified in accordance with the designed strategy, which is obtained by mining the data effectively. The fuzzy rules generated from the proposed strategy can be able to provide better classification rate in detecting the intrusion behavior. Even though signature-based systems provide good detection results for specified and familiar attacks, the foremost advantage of anomaly-based detection techniques is their ability to detect formerly unseen and unfamiliar intrusion occurrences. On the other hand and in spite of the expected erroneousness in recognized signature specifications, the rate of false positives in anomaly-based systems is generally higher than in signature based ones [21]. The different steps involved in the proposed system for anomaly-based intrusion detection (shown in figure 1) are described as follows:

(1) Classification of training data
(2) Strategy for generation of fuzzy rules
(3) Fuzzy decision module
(4) Finding an appropriate classification for a test input

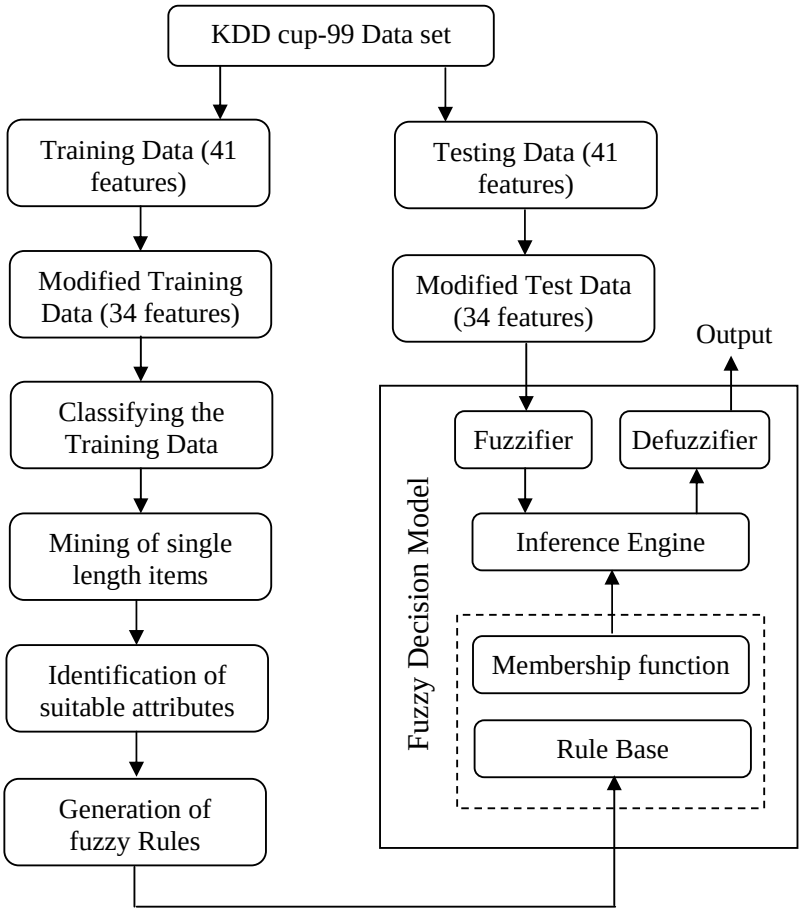


Fig.1. The overall steps of the proposed intrusion detection system

**(3.1)CLASSIFICATION OF TRAINING DATA**

The first component of the proposed system is of classifying the input data into multiple classes by taking in mind the different attacks involved in the intrusion detection dataset. The dataset we have taken for analyzing the intrusion detection behavior using the proposed system is KDD-Cup 1999 data. The detailed analysis of KDD-Cup 1999 data is given in section 3. Based on the analysis, the KDD-Cup 1999 data contains four types of attacks and normal behavior data with 41 attributes that have both continuous and symbolic attributes. The proposed system is designed only for the continuous attributes because the major attributes in KDD-Cup 1999 data are continuous in nature. Therefore, we have taken only the continuous attributes for instance, 34 attributes from the input dataset by removing discrete attributes. Then, the dataset ($D$) is divided into five subsets of classes based on the class label prescribed in the dataset $D = \{D_i ; 1 \le i \le 5\}$. The class label describes several attacks, which comes under four major attacks (Denial of Service, Remote to Local, U2R and Probe) along with normal data. The five subsets of data are then used for generating a better set of fuzzy rules automatically so that the fuzzy system can learn the rules effectively.

**(3.2)STRATEGY FOR GENERATION OF FUZZY RULES**

This section describes the designed strategy for automatic generation of fuzzy rules to provide effective learning. In general, the fuzzy rules given to the fuzzy system is done manually or by experts, who are given the rules by analyzing intrusion behavior. But, in our case, it is very difficult to generate fuzzy rules manually due to the fact that the input data is huge and also having more attributes. But, a few of researches are available in the literature for automatically identifying of fuzzy rules in recent times. Motivated by this fact, we make use of mining methods to identify a better set of rules. Here, definite rules obtained from the single length frequent

items are used to provide the proper learning of fuzzy system. The process of fuzzy generation is given in the following sub-section.

***(a) Mining of single length frequent items***

At first, frequent items (attributes) are discovered from both classes of input data and by using these frequent items, the significant attributes are identified for the input KDD-cup 99 dataset. In general, frequent itemset are mined using various conventional mining algorithms, such as Apriori [35] and FP-Growth [40]. These algorithms are suitable to mine frequent itemset with varying length only for the binary database, which contains only the binary values. But, the input dataset (KDD cup-99) contains continuous variable for each attributes so that, the conventional algorithm is not suitable for mining frequent items. By considering this property, we simply find the 1-length items from each attributes by finding the frequency of the continuous variable present in each attribute and then, the frequent items are discovered by inputting the minimum support. These frequent items are identified for both class namely, normal and attack (combining four types of attacks).

***(b) Identification of suitable attributes for rule generation***

In this step, we have chosen only the most suitable attributes for identifying the classification whether the record is normal or attack. The reason behind this step is that the input data contain 34 attribute, in which all the attributes are not so effective in detecting the intrusion detection. For identifying the suitable attribute, we have used deviation method, where mined 1-length frequent items are used. At first, the mined l-length items from each attribute are stored in a vector so that 34 vectors are obtained for each class (class 1 and class 2), represented as, $C_i = \lfloor V_1, V_2, \cdots, V_j, \cdots, V_{34} \rfloor$ where, $i = 1(\text{refer to normal}), 2(\text{refer to attack})$. Each vector ($V_j$) contains frequent items, whose frequency is greater than minimum support. $V_j = \{f_i ; 1 \le i \le m\}$ ; $support(f_i) \ge min\_supp$. Then, for each attribute, deviation range of frequent items is identified by comparing the frequent items present within a vector such a way, the deviation range {max, min} is obtained for every vector.

$$D_{v(j)} = \{f_{\max}, f_{\min}\}; \text{ where, } \quad f_{\max} = Max(f_j); \quad f_{\min} = Min(f_j)$$

Then, one-to-one comparison is performed in between both class of respective vector to identify the effective attribute. The attributes that not contain identical {max, min} range for both class is chosen as effective attribute, which will give significant detection rate rather than utilizing the all attribute for identifying the classification. The effective attributes chosen for rule generation process is represented as, $C_i = \lfloor V^{(1)}, V^{(2)}, \cdots, V^{(j)}, \cdots, V^{(k)} \rfloor$, Where, $k \le 34$.

***(c) Rule generation***

The effective attributes chosen from the previous step is utilized to generate rules that is derived from the {max, min} deviation. By comparing the deviation range of effective attributes in between the normal and attack data, the intersection points are identified for the effective attributes. By making use of these two intersection points, the *definite* and *indefinite rules* are generated. For example, {max, min} deviation range for normal data related to attribute1 is {1, 5} and {max, min} deviation for attack data corresponding to attribute1 is {2, 8}. Then, the rule is designed like, *"IF attribute1 is greater than 5, THEN the data is attack, "IF attribute1 is in between 2 and 5, THEN the data is normal OR attack"* and *"IF attribute1 is less than 2, THEN the data is normal"*. In addition to that, some of the data contains only one intersection point, which provides only two rules.

***(d) Rule filtering***

In order to learn the fuzzy rules efficiently and design a compact and interpretable classification system, we should concentrate in these two criteria given in [33, 34]: (1) The number of fuzzy rules should be decreased as much as possible, (2) The IF part of fuzzy rules should be short. By concentrating on these two criteria, we have filtered the rules such a way that, we take only the short and less number of rules. The rules that are generated from the previous step contain definite and indefinite rules. The *definite rules* are the rules that contain only one classified label in the THEN part and *indefinite rule* contain two classification label data in the THEN part. The proposed rule filtering technique filters the indefinite rule and selects only the definite rules for learning the fuzzy system.

***(e) Generating fuzzy rules***

In general, fuzzy rules are defined within the fuzzy system manually or the rules are obtained from the domain expert. But, in the proposed system, we automatically find the fuzzy rules based on the mined 1-length frequent items. The fuzzy rules are generated from the definite rules, where the IF part of the rule is a numerical variable and THEN part is a class label related to attack name or normal. But, the fuzzy rule should contain only the linguistic variable. So, in order to make the fuzzy rules from the definite rules, we should fuzzify the numerical variable of the *definite rules* and THEN part of the fuzzy rule is same as the consequent part of the *definite rules*. For example, *"IF attribute1 is H, THEN the data is attack* and *"IF attribute1 is VL, THEN the data is normal"*. These fuzzy rules are used to learn the fuzzy system so that the effectiveness of the proposed system will be improved rather than simply using the fuzzy rules without any proper techniques.

**(3.3)FUZZY DECISION MODULE**

This section describes the designing of fuzzy logic system for finding the suitable class label of the test dataset. Zadeh in the late 1960s [39] introduced Fuzzy logic and is known as the rediscovery of multivalued logic designed by Lukasiewicz. The designed fuzzy system shown in figure 2 contains 34 inputs and one output, where inputs are related to the 34 attributes and output is related to the class label (attack data or normal data). Here, thirty four-input, single-output of Mamdani fuzzy inference system with centroid of area defuzzification strategy was used for this purpose. Here, each input fuzzy set defined in the fuzzy system includes four membership functions (VL, L, M and H) and an output fuzzy set contains two membership functions (L and H). Each membership function used triangular function for fuzzification strategy. The fuzzy rules obtained from sub-section 4.2 are fed to the fuzzy rule base for learning the system.

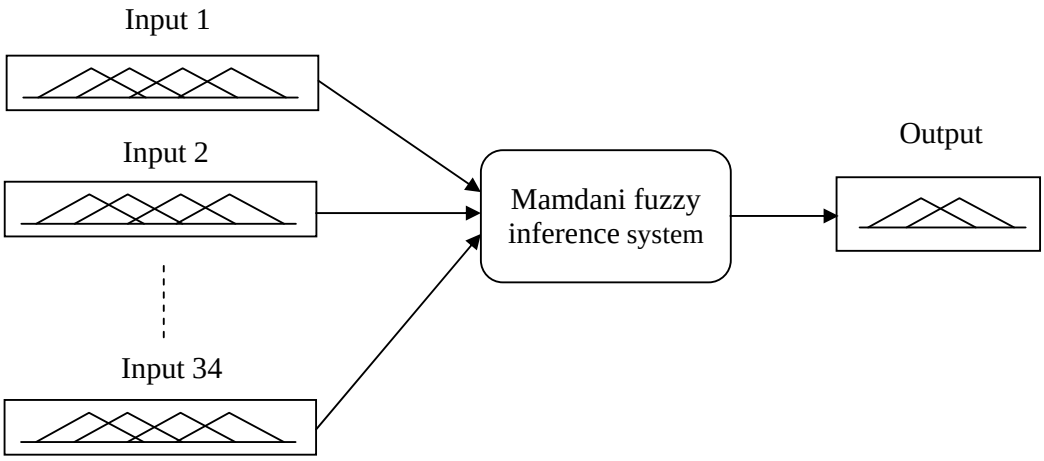


Fig.2. The designed Fuzzy system

**(3.4) FINDING AN APPROPRIATE CLASSIFICATION FOR A TEST INPUT**

For testing phase, a test data from the KDD-cup 99 dataset is given to the designed fuzzy logic system discussed in sub-section 4.3 for finding the fuzzy score. At first, the test input data containing 34 attributes is applied to *fuzzifier*, which converts 34 attributes (numerical variable) into linguistic variable using the triangular membership function. The output of the fuzzifier is fed to the *inference engine* which in turn compares that particular input with the rule base. *Rule base* is a knowledge base which contains a set of rules obtained from the *definite rules*. The output of inference engine is one of the linguistic values from the following set {Low and High} and then, it is converted by the *defuzzifier* as crisp values. The crisp value obtained from the fuzzy inference engine is varied in between 0 to 2, where '0' denotes that the data is completely normal and '1' specifies the completely attacked data.

## 4. EXPERIMENTATION

This section describes the experimental results and performance evaluation of the proposed system. The proposed system is implemented in MATLAB (7.8) and the performance of the system is evaluated using Precision, recall and F-measure. For experimental evaluation, we have taken KDD cup 99 dataset [37], which is mostly used for evaluating the performance of the intrusion detection system. For evaluating the performance, it is very difficult to execute the proposed system on the KDD cup 99 dataset since it is a large scale. Here, we have used subset of 10% of KDD Cup 99 dataset for training and testing. The number of records taken for testing and training phase is given in table 3 and table 4.

| Training Dataset | |
|---|---|
| Normal | 25,000 |
| DOS | 25,000 |
| Probe | 4107 |
| RLA | 77 |
| URA | 42 |

Table 3. Training dataset taken for experimentation

| Testing Dataset | |
|---|---|
| Normal | 26,000 |
| DOS | 26,000 |
| Probe | 4107 |
| RLA | 77 |
| URA | 42 |

Table 4. Training dataset taken for evaluation

### 4.1 EXPERIMENTAL RESULTS AND PERFORMANCE ANALYSIS

The training dataset contains normal data as well as four types of attacks, which are given to the proposed system for identifying the suitable attributes. The selected attribute for rule generation process is given in table 5. Then, using the fuzzy rule learning strategy, the system generates definite and indefinite rules and finally, fuzzy rules are generated from the definite rules.

| Attribute Index | Selected Attributes |
|---|---|
| 1 | duration |
| 5 | src_bytes |
| 6 | dst_bytes |
| 8 | wrong_fragment |
| 9 | urgent |
| 10 | hot |
| 11 | num_failed_logins |
| 13 | num_compromised |
| 16 | num_root |
| 17 | num_file_creations |
| 18 | num_shells |
| 19 | num_access_files |
| 23 | count |
| 24 | srv_count |

Table 5. Selected attributes for rule generation

In the testing phase, the testing dataset is given to the proposed system, which classifies the input as a normal or attack. The obtained result is then used to compute overall accuracy of the proposed system. The overall accuracy of the proposed system is computed based on the definitions, namely precision, recall and F-measure which are normally used to estimate the rare class prediction. It is advantageous to accomplish a high recall devoid of loss of precision. F-measure is a weighted harmonic mean which evaluates the trade-off between them.

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F - measure = \frac{(\beta^2 + 1)(Precision \cdot Recall)}{\beta^2 \cdot Precision + Recall} \text{ where, } \beta = 1$$

$$Overall\ accuracy = \frac{TP + TN}{TP + TN + FN + FP}$$

Where, $TP$ → True positive
$TN$ →True negative
$FN$ → False negative
$FP$ → False positive
These are computed using the confusion matrix in Table 6, and defined as follows:

| | | **Predicted class** | |
|---|---|---|---|
| | | **Positive class** | **Negative class** |
| **Actual** | Positive class | True positive (TP) | False negative (FN) |
| **Class** | Negative class | False positive (FP) | True negative (TN) |

Table 6. Confusion matrix

The evaluation metrics are computed for both training and testing dataset in the testing phase and the obtained result for all attacks and normal data are given in table 7, which is the overall classification performance of the proposed system on KDD cup 99 dataset. By analyzing the result, the overall performance of the proposed system is improved significantly and it achieves more than 90% accuracy for all types of attacks.

| | ***Metric*** | **Proposed System** | |
|---|---|---|---|
| | | **Training** | **Testing** |
| **PROBE** | Precision | 0.912522 | 0.912522 |
| | Recall | 0.37083 | 0.37083 |
| | F-measure | 0.52735457 | 0.52735457 |
| | Accuracy | 0.906208 | 0.909323 |
| **DOS** | Precision | 0.993563 | 0.993828 |
| | Recall | 0.90144 | 0.904154 |
| | F-measure | 0.94526236 | 0.94687236 |
| | Accuracy | 0.9478 | 0.949269 |
| **U2R** | Precision | 0.051948 | 0.051948 |
| | Recall | 0.190476 | 0.190476 |
| | F-measure | 0.08163265 | 0.08163265 |
| | Accuracy | 0.992812 | 0.993088 |
| **R2L** | Precision | 0.075949 | 0.075949 |
| | Recall | 0.155844 | 0.155844 |
| | F-measure | 0.10212766 | 0.10212766 |
| | Accuracy | 0.991586 | 0.991909 |
| **NORMAL** | Precision | 0.828439 | 0.829318 |
| | Recall | 0.99416 | 0.994385 |
| | F-measure | 0.90376539 | 0.90438129 |
| | Accuracy | 0.910852 | 0.903019 |

Table 7. The classification performance of the proposed intrusion detection system

## 5. CONCLUSION

We have developed an anomaly based intrusion detection system in detecting the intrusion behavior within a network. A fuzzy decision-making module was designed to build the system more accurate for attack detection, using the fuzzy inference approach. An effective set of fuzzy rules for inference approach were identified automatically by making use of the fuzzy rule learning strategy, which are more effective for detecting intrusion in a computer network. At first, the definite rules were generated by mining the single length frequent items from attack data as well as normal data. Then, fuzzy rules were identified by fuzzifying the definite rules and these rules were given to fuzzy system, which classify the test data. We have used KDD cup 99 dataset for evaluating the performance of the proposed system and experimentation results showed that the proposed method is effective in detecting various intrusions in computer networks.

## REFERENCES


[1] Yao, J. T., S.L. Zhao, and L.V. Saxton, "A Study On Fuzzy Intrusion Detection", In Proceedings of the Data Mining, Intrusion Detection, Information Assurance, And Data Networks Security, SPIE, Vol. 5812, pp. 23-30, Orlando, Florida, USA, 2005.

[2] Nivedita Naidu and Dr.R.V.Dharaskar, "An Effective Approach to Network Intrusion Detection System using Genetic Algorithm", International Journal of Computer Applications, Vol.1, No.3, pp.26–32, February 2010.

[3] J. Allen, A. Christie, and W. Fithen, "State Of the Practice of Intrusion Detection Technologies", Technical Report, CMU/SEI-99-TR-028, 2000.

[4] B.V. Dasarathy, "Intrusion Detection", Information Fusion, Vol.4, No.4, pp.243-245, 2003.

[5] R.G.Bace, "Intrusion Detection", Macmillan Technical Publishing, Indianapolis, USA, 2000.

[6] Marcos M. Campos, Boriana L. Milenova, "Creation and Deployment of Data Mining-Based Intrusion Detection Systems in Oracle Database 10g", in Proceedings of the Fourth International Conference on Machine Learning and Applications, 2005.

[7] Anazida Zainal, Mohd Aizaini Maarof and Siti Maryam Shamsudin , "Research Issues in Adaptive Intrusion Detection", in Proceedings of the 2nd Postgraduate Annual Research Seminar (PARS'06), Faculty of Computer Science & Information Systems, Universiti Teknologi Malaysia, 24 – 25 May, 2006.

[8] Dr. Fengmin Gong, "Deciphering Detection Techniques: Part II Anomaly-Based Intrusion Detection", White Paper from McAfee Network Security Technologies Group, 2003.

[9] Susan M. Bridges and Rayford B.Vaughn, "Fuzzy Data Mining And Genetic Algorithms Applied To Intrusion Detection", In Proceedings of the National Information Systems Security Conference (NISSC), Baltimore, MD, pp.16-19, October 2000.

[10] Jian Pei, Upadhyaya, S.J., Farooq, F., Govindaraju, V, "Data mining for intrusion detection: techniques, applications and systems ", in Proceedings of the 20th International Conference on Data Engineering, pp: 877 - 87, 2004.

[11] Cannady J, "Artificial Neural Networks for Misuse Detection", in Proceedings of the '98 National Information System Security Conference (NISSC'98), pp. 443-456, 1998.

[12] Shon T, Seo J, and Moon J, "SVM Approach with A Genetic Algorithm for Network Intrusion Detection", Lecture Notes in Computer Science, Springer Berlin / Heidelberg, Vol. 3733, pp. 224-233, 2005.

[13] Yu Y, and Huang Hao, "An Ensemble Approach to Intrusion Detection Based on Improved Multi-Objective Genetic Algorithm", Journal of Software, Vol.18, No.6, pp.1369-1378, June 2007.

[14] J. Luo, and S. M. Bridges, "Mining fuzzy association rules and fuzzy frequency episodes for intrusion detection", International Journal of Intelligent Systems, Vol. 15, No. 8, pp. 687-704, 2000.

[15] W. Lee, S. Stolfo, and K. Mok, "A Data Mining Framework for Building Intrusion Detection Model", In Proceedings of the IEEE Symposium on Security and Privacy, Oakland, CA, pp. 120-132, 1999.

[16] Dewan Md. Farid and Mohammad Zahidur Rahman, "Anomaly Network Intrusion Detection Based on Improved Self Adaptive Bayesian Algorithm", Journal of Computers, Vol.5, No.1, January, 2010.

[17] K.Yoshida, "Entropy Based Intrusion Detection", in Proceedings of the IEEE Pacific Rim Conference on Communications, Computers and signal Processing, Vol. 2, pp. 840 – 843, Aug 28- 30, 2003.

[18] Sujaa Rani Mohan, E.K. Park, Yijie Han, "An Adaptive Intrusion Detection System Using A Data Mining Approach", White paper from University of Missouri, Kansas City, October 2005.

[19] Rasha G. Mohammed Helali, "Data Mining Based Network Intrusion Detection System: A Survey", In Novel Algorithms and Techniques in Telecommunications and Networking, pp. 501-505, 2010.

[20] Pakkurthi Srinivasu, P.S. Avadhani, Vishal Korimilli, Prudhvi Ravipati, "Approaches and Data Processing Techniques for Intrusion Detection Systems", Vol. 9, No. 12, pp. 181-186, 2009.

[21] G. Macia Fernandez and E. Vazquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges", Computers & Security, Vol. 28, No. 1-2, pp. 18-28, February-March 2009.

[22] Mark Crosbie and Gene Spa Ord, "Defending a Computer System using Autonomous Agents", Technical report, 1995.

[23] Honig, A., Howard, A., Eskin, E., and Stolfo, S. J., "Adaptive Model Generation: An Architecture for the Deployment of Data Mining-Based Intrusion Detection Systems, Applications of Data Mining in Computer Security, Kluwer Academic Publishers, Boston, MA, pp. 154-191, 2002.

[24] Stephen F. Owens, Reuven R. Levary, "An adaptive expert system approach for intrusion detection", International Journal of Security and Networks, Vol: 1, No: 3/4, pp: 206-217, 2006.

[25] Alok Sharma, Arun K. Pujari, Kuldip K. Paliwal, "Intrusion detection using text processing techniques with a kernel based similarity measure", Computers & Security, Vol: 26, No: 7-8, pp: 488-495, 2007.
[26] Shi-Jinn Horng, Ming-Yang Su, Yuan-Hsin Chen, Tzong-Wann Kao, Rong-Jian Chen, Jui-Lin Lai, Citra Dwi Perkasa, "A novel intrusion detection system based on hierarchical clustering and support vector machines", Expert Systems with Applications, Vol: 38, No: 1, pp: 306-313, 2011.
[27] Abadeh, M.S., Habibi, J., "Computer Intrusion Detection Using an Iterative Fuzzy Rule Learning Approach", in Proceedings of the IEEE International Conference on Fuzzy Systems, pp: 1-6, London, 2007.
[28] Bharanidharan Shanmugam, Norbik Bashah Idris, "Improved Intrusion Detection System Using Fuzzy Logic for Detecting Anamoly and Misuse Type of Attacks", in Proceedings of the International Conference of Soft Computing and Pattern Recognition, pp: 212-217, 2009.
[29] O. Adetunmbi Adebayo, Zhiwei Shi, Zhongzhi Shi, Olumide S. Adewale, "Network Anomalous Intrusion Detection using Fuzzy-Bayes", IFIP International Federation for Information Processing, Vol: 228, pp: 525-530, 2007.
[30] Arman Tajbakhsh, Mohammad Rahmati, Abdolreza Mirzaei, "Intrusion detection using fuzzy association rules", Applied Soft Computing, Vol: 9, No: 2, pp: 462-469, 2009.
[31] Zhenwei Yu, Tsai, J.J.P., Weigert, T., "An Automatically Tuning Intrusion Detection System", IEEE Transactions on Systems, Man, and Cybernetics, Vol: 37, No: 2, pp: 373 - 384, 2007.
[32] Qiang Wang and Vasileios Megalooikonomou, "A clustering algorithm for intrusion detection", in Proceedings of the conference on Data Mining, Intrusion Detection, Information Assurance, and Data Networks Security, vol. 5812, pp. 31-38, March 2005.
[33] Cordon O, Gomide F, Herrera F, Hoffmann F, Magdalena L, “Ten years of genetic fuzzy systems: current framework and new trends”, Fuzzy Sets and Systems, vol.141, no.1, pp. 5–31, 2004.
[34] M. Saniee Abadeh, J. Habib and C. Lucas, “Intrusion detection using a fuzzy genetics-based learning algorithm”, Journal of Network and Computer Applications, vol.30, no.1, pp. 414–428, 2007.
[35] R. Agrawal, T. Imielinski, A., Swami, “Mining association rules between sets of items in large databases”, in Proceedings of 1993 ACM SIGMOD Intl. Conf. on Management of Data, Washington, DC, pp. 207–216, 1993.
[36] http://www.ll.mit.edu/mission/communications/ist/corpora/ideval/data/1998data.html
[37] http://www.sigkdd.org/kddcup/index.php?section=1999&method=data
[38] Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu and Ali A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set", in Proceedings of the Second IEEE international conference on Computational intelligence for security and defense applications, pp. 53-58, Ottawa, Ontario, Canada, 2009.
[39] Zadeh, L.A., “Fuzzy sets”, Information and control, vol.8, pp. 338-353, 1965.
[40] Jiawei Han, Jian Pei, Yiwen Yin, Runying Mao, "Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach", Data Mining and Knowledge Discovery, Vol: 8, No: 1, pp: 53 - 87, 2004.