**1. INTRODUCTION**

The widespread use of computers and networking has given rise to many security vulnerabilities. Intrusions are a commonplace today in any network. Therefore, there is a need to detect and prevent such intrusions which can jeopardize the privacy, confidentiality, integrity of network data.

An intrusion detection system (IDS) is a device or [software application](http://en.wikipedia.org/wiki/Software_application) that monitors network or system activities for malicious events or policy violations and produces reports to a Management Station. There are two types of IDSs.

1. Network-based IDS: Its data is mainly collected network generic stream going through network segments, such as Internet packets.
2. Host-based IDS: Its data come from the records of various host activities, such as audit record of operation system, system logs, application programs information.

There are two types of detection techniques possible.

1. Anomaly Detection: It is the identification of malicious traffic based on deviations from established normal network traffic patterns.
2. Misuse Detection: It is the identification of intrusions based on a known pattern for the malicious activity.

Data mining is a branch of [computer science](http://en.wikipedia.org/wiki/Computer_science) and [statistics](http://en.wikipedia.org/wiki/Statistics) that attempts to discover patterns in large [data sets](http://en.wikipedia.org/wiki/Data_set). It utilizes methods at the intersection of [artificial intelligence](http://en.wikipedia.org/wiki/Artificial_intelligence), [machine learning](http://en.wikipedia.org/wiki/Machine_learning), [statistics](http://en.wikipedia.org/wiki/Statistics), and [database systems](http://en.wikipedia.org/wiki/Database_system).

Fuzzy logic is a form of [many-valued logic](http://en.wikipedia.org/wiki/Many-valued_logic) or [probabilistic logic](http://en.wikipedia.org/wiki/Probabilistic_logic), dealing with [reasoning](http://en.wikipedia.org/wiki/Reasoning) that is approximate rather than fixed and exact. Fuzzy logic variables may have a [truth value](http://en.wikipedia.org/wiki/Truth_value) that ranges in degree between 0 and 1. An evolutionary algorithm (EA) is a [subset](http://en.wikipedia.org/wiki/Subset) of [evolutionary computation](http://en.wikipedia.org/wiki/Evolutionary_computation), a generic population based [meta heuristic](http://en.wikipedia.org/wiki/Metaheuristic) [optimization](http://en.wikipedia.org/wiki/Optimization_(mathematics)) [algorithm](http://en.wikipedia.org/wiki/Algorithm) which uses mechanisms like [reproduction](http://en.wikipedia.org/wiki/Reproduction), [mutation](http://en.wikipedia.org/wiki/Mutation), [recombination](http://en.wikipedia.org/wiki/Genetic_recombination), and [selection](http://en.wikipedia.org/wiki/Natural_selection), inspired by [biological evolution](http://en.wikipedia.org/wiki/Biological_evolution).

In an intrusion detection system (IDS) developed using data mining techniques and fuzzy logic, fuzzy rules have to be mined from the data records. To account for the new attacks, the fuzzy rule set is evolved with an evolutionary algorithm. To better the effectiveness of the classification by the fuzzy inference system, different modules are built to achieve optimum values of the accuracy parameters, True Positivity, True Negativity, False Positivity and False Negativity.

* **True Positive Rate:** Number of legitimate attacks detected.
* **False Positive Rate:** Number of events detected wrongly as attacks, i.e. when no attack has taken place.
* **False Negative Rate:** Number of legitimate attacks undetected.
* **True Negative Rate:** Number of events which are not attacks that are rightly identified as non-attacks.

Feature selection is done to minimize the number of features and select the most optimum features from the data record set. Feature selection, also known as variable selection, feature reduction, attribute selection or variable subset selection, is the technique of selecting a subset of relevant features for building robust learning models.  Feature selection helps improve the performance of learning models by:

* Alleviating the effect of the [curse of dimensionality](http://en.wikipedia.org/wiki/Curse_of_dimensionality).
* Enhancing generalization capability.
* Speeding up learning process.
* Improving model interpretability.

In the preprocessing step, quantization is done to give numeric values to the class types, the last attribute of the KDD data set used. This is useful while training and test the data. Scaling is the process of changing the values of the attributes to fit into the range 0-1. This is done to eliminate noise and reduce the time complexity in the further steps.

Membership functions (clusters) -low, medium and high are defined for each attribute selected from the feature selection stage. To calculate the membership values of each data point in each cluster, fuzzy c-means clustering is used. Fuzzy clustering is a class of [algorithms](http://en.wikipedia.org/wiki/Algorithm) for [cluster analysis](http://en.wikipedia.org/wiki/Cluster_analysis) in which the allocation of data points to clusters is not hard (all-or-nothing) but fuzzy in the same sense as [fuzzy logic](http://en.wikipedia.org/wiki/Fuzzy_logic). In fuzzy clustering, each point has a degree of belonging to clusters, as in [fuzzy logic](http://en.wikipedia.org/wiki/Fuzzy_logic), rather than belonging completely to just one cluster. Thus, points on the edge of a cluster, may be in the cluster to a lesser degree than points in the center of cluster.

In the next stage of our project, fuzzy rules are mined using Kuok’s Algorithm, the fuzzy version of Apriori Algorithm. Apriori is a classic algorithm for [learning association rules](http://en.wikipedia.org/wiki/Association_rule_learning). Apriori is designed to operate on [databases](http://en.wikipedia.org/wiki/Database) containing transactions. The purpose of the Apriori Algorithm is to find associations between different sets of data. The output of Apriori is sets of rules that tell us how often items are contained in sets of data. Apriori can work well on only binary data. To account for the quantitative attributes in our dataset, Kuok’s algorithm is used. Once the rules are obtained, Gfuzzy python library is used for classification. Gfuzzy is a python library used for creating a [fuzzy-logic](http://en.wikipedia.org/wiki/Fuzzy_logic) inference engine. It uses the [protocol buffer](http://code.google.com/p/protobuf/) library for its configuration.

**2. LITERATURE SURVEY**

**2.1 Study and analysis of different Intrusion Detection Systems**

A description of different techniques employed in data mining like machine learning, classification tree and support vector machines is provided and four types of intrusions namely denial of service, probing, unauthorized access to a resource and unauthorized access from a remote machine have been detected and classified [1].

* Classification tree is to learn from class-labeled training tuples for predicting classes of new or previously unseen data.
* Support vector machine is to build a hyperplane or set of hyperplanes to separate the tuples which belong to two classes -1 and +1.
* Relative performances of different algorithms are measured and false positive, false negative, true positive and true negative rates are recorded for each.
* It is concluded that classification tree technique has better performance than SVM.

The advantages of using data mining for intrusion detection like improved variants detection, controlled false alarms, reduced false dismissals and improved efficiency are cited. Effectiveness of different data mining techniques is discussed and a classification is made [2].

* Knowledge discovery in databases is described to consist of the following steps- understanding the application domain, data integration and selection, data mining, pattern evaluation and knowledge representation.
* A survey of Feature Selection and Machine Learning techniques like Inductive Rule Generation, Genetic Algorithms, Fuzzy Logic, Neural Networks, Support Vector Machine, Clustering techniques, Statistical techniques and Hidden Markov Models has been summarized.

A description of a prototype intelligent intrusion detection system (IIDS) to demonstrate the effectiveness of data mining techniques that utilize fuzzy logic and genetic algorithms is given [3].

* The development of misuse detection components using fuzzy data mining techniques and tuning of fuzzy membership functions to select an appropriate set of features is described. Fuzzy rule set is obtained for both normal and anomalous behavior.
* Fuzzy association rules are computed using Borgelt’s prefix trees.
* To compute the similarity between the two sets of fuzzy rules genetic algorithm is used.

**2.2 Feature Selection**

F-score feature selection strategy has been explained for optimal feature selection in machine learning models [4].

* F-score algorithm associates a score with every feature in the data set considered.
* Positive instances which lead to classification of a data record as belonging to a particular class and negative instances which help in not identifying the non-membership of a record for a class are used to calculate f-score for each feature using relevant formula.
* Based on the scores, ranking is assigned to features and the features with the best f-score values are the most crucial in correct classification of data records.
* Disadvantage of this method is it does not reveal mutual information about the attributes

**2.3 Fuzzy C-Means Clustering**

An overview of Fuzzy C-Means Clustering which is a commonly used algorithm in pattern recognition is provided. A modified FCM called psFCM (partition selection FCM) is proposed which is four times faster than FCM [7].

* Partitioning into different clusters is done based on fuzzy C-partition with the initial cluster centers being randomly generated from the dataset.
* Since the convergence of FCM depends upon obtaining good cluster centers as the result which in turn depends upon the initial cluster centers chosen, the proposed algorithm aims to find them.
* In the first phase of psFCM, the dataset is partitioned into small block cells using k-d tree method and reduced to a simplified dataset with unit blocks, the patterns in which are replaced by centroids.
* In the second phase, the method used in original FCM is used to obtain the cluster centers.

**2.4 Apriori Algorithm**

Two algorithms Apriori and AprioriTid have been proposed to discover association rules from large databases [8].

* Obtaining large itemsets from candidate itemsets, starting with the initial large itemsets is described.
* Algorithm to generate candidate itemsets from the previous large itemsets is described.
* Performance of Apriori and AprioriTid is better when compared to earlier algorithms like SETM and Association Rule Mining Algorithm.

**2.5 Kuok’s Algorithm**

An extension of Apriori Algorithm called Kuok’s Algorithm to mine fuzzy association rules is proposed [9].

* The suitability of Kuok’s Algorithm for quantitative data is demonstrated.
* Generation of large datasets is described.
* The methods to calculate significance factor (minimum support) and certainty factor (minimum confidence) are explained with examples.

**2.6 Mining fuzzy rules for intrusion detection**

An overview of an intelligent intrusion detection system which uses a data miner that integrates Apriori and Kuok’s algorithms is provided [10].

* Preprocessing is done to reduce the amount of data to be used in the further stages by employing a decision tree technique and configuration files are generated using the captured tcpdump data.
* Rules are mined using a combination of Apriori and Kuok’s algorithms.
* FuzzyJess, a fuzzy inference engine is used to evaluate the rules and do the classification.
* The methods to calculate minimum support and minimum confidence for the rules is described.

**3. PROBLEM DEFINITION AND OBJECTIVES**

Our aim is to develop an Anomaly-based Intrusion Detection System (ABIDS) tool with low false positive rate, to detect network anomalies by mining network data built using the combination of techniques of fuzzy logic and evolutionary algorithms.

Our objectives are:

* Preprocessing and Feature Selection
  + Preprocessing-quantization and scaling of the data.
  + Implement f-score algorithm for selecting the most crucial features for our Anomaly-based Intrusion Detection System.
* Mine fuzzy rule set using the selected attributes and do the classification
* Define fuzzy membership functions for each attribute – low, medium and high.
* Implement fuzzy c-means algorithm using the training set to find the centroid for each cluster representing a membership function for each important attribute obtained in Feature Selection stage.
* Calculate fuzzy membership values for the membership functions of each attribute for the testing test.
* Implement Kuok’s Algorithm to mine fuzzy rules from categorical and quantitative attributes.
* Improve the performance of the fuzzy logic system by optimizing it with evolutionary algorithms through the reduction of its rule set.
* Develop a second tier of classifiers for each type of attack to reduce the false positivity of the fuzzy classification, thereby improving the efficiency of classification.

**4. WORK DONE**

**4.1 Study and analysis**

Various IDSs were studied to understand their implementation and working. The importance of preprocessing and feature selection was understood. An analysis of IDSs developed using different techniques was done to understand their relative performance.

**4.1.1 Neural Networks:**

Neural networks provide a solution to the problem of modeling the users' behavior in anomaly detection because they do not require any explicit user model. In particular, the typical sequence of commands executed by each user is learned.

Advantages:

* The capability of learning by example allows the system to detect new types of intrusion.
* With learning by example approaches, ‘attack signatures’ can be extracted automatically from labeled traffic data which eliminates the subjectivity and other problems introduced by the presence of the human factor.

Observation:

* Neural nets perform with an accuracy of 89.09%.

**4.1.2 Support Vector Machines:**

Support vector machines (SVMs) are a set of related supervised learning methods used for classification and regression. They belong to a family of generalized linear classifiers. SVMs attempt to separate data into multiple classes though the use of a hyperplane. SVMs can be used to classify network traffic into 5 classes, one to identify normal traffic, and four to identify each of the four types of malicious activity in dataset.

Observation:

* SVM performs with better than 95% accuracy.
* SVMs are superior to neural nets in both accuracy and speed.

**4.1.3 Fuzzy Classifier:**

A fuzzy classifier classifies data into a normal class and m other classes called abnormal classes.

The data set used by the learning algorithms consists of a set of objects, each object with n+1 attributes with the first n attributes define the object characteristics and the last attribute defines the class that the object belongs to. Accordingly, fuzzy classifier system for solving intrusion detection problem should have a set of m+1 rules, one for the normal class and m for the abnormal classes, where the condition part is defined by the monitored parameters and the consequent part is an atomic expression for the classification attribute.

Advantages:

* Easy interpretation of rules is possible.
* The detection is faster when compared to all other techniques

**4.1.4 Artificial Immune System- Negative Selection Algorithm:**



**Figure 1: Negative Selection Algorithm for Intrusion Detection**

The negative selection, due to its ability to distinguish the difference between self and non-self can be used for intrusion detection. The general AIS based IDS can be divided into two parts, i.e. detector set generation and non-self-detection. To form the detection set, negative selection algorithm is applied. Initially, the immature detectors (parameter pattern) are randomly generated. Then, these immature detectors are compared with the normal network parameter patterns. If a random generated pattern matches a normal pattern, the immature detector will be rejected and deleted. Those which do not match any normal network parameter patterns will be saved as mature detectors. In the live detection stage, a monitored network parameter pattern is compared with detectors in the detector set. If it is matched with any detector, then a network intrusion is detected.

Advantages:

* Performs with the best accuracy, greater than 99%.
* High true positive rate and low false positive and false negative rates.

**4.2 Preprocessing - Quantization and Scaling.**

KDD data set of network intrusions released by Lincoln Laboratory, Massachusetts Institute of Technology, which has around 41 lakh data records was used. 5 thousand data records were randomly chosen from these records for feature selection.

The ‘class’ feature of all the data records is quantified, i.e. given numeric values to different classes so as to enable proper training of the data set. Scaling was performed to make all the feature values fall in the range from 0-1. Without proper scaling, erroneous values can result. Scaling was done using the following formula:

**scaled\_feature = ( feature\_val - low\_val ) / ( high\_val - low\_val )**

where

scaled\_feature\_val is the scaled value of the feature under consideration,

feature\_val is the original feature value,

low\_val is the lowest value of the feature appearing in the data set, and

high\_val is the highest value of the feature appearing in the data set.

**4.3 Feature Selection using F-score algorithm.**

Feature selection was done to select the most important features by employing F-score algorithms which gives a score to the attributes depending on their contribution in the classification. F-score reduces overall complexity of the further stages.

F-score algorithm was implemented to associate a score with every scaled feature in the data set considered. Positive instances of a feature lead to classification of a data record containing he feature as belonging to a particular class and negative instances of a feature help in identifying the non-membership of a record containing that feature for a class. F-score is calculated for each feature using the following formula:



where  are the average of the whole, positive and negative data sets respectively;  is the ith feature of the kth positive instance and is the ith feature of the kth negative instance. The numerator indicates the discrimination between the positive and negative sets, and the denominator indicates the one within each of the two sets. Based on the scores, ranking was assigned to features and the features with the best f-score values are the most crucial in correct classification of data records.

**4.4** **Python Implementation of F-Score Algorithm**

def cal\_Fscore(labels,samples):

data\_num=float(len(samples))

p\_num = {} #key: label; value: data num

sum\_f = [] #index: feat\_idx; value: sum

sum\_l\_f = {} #dict of lists. key1: label; index2: feat\_idx; value: sum

sumq\_l\_f = {} #dict of lists. key1: label; index2: feat\_idx; value: sum of square

F={} #key: feat\_idx; valud: fscore

max\_idx = -1

### pass 1: check number of each class and max index of features

for p in range(len(samples)): # for every data point

label=labels[p]

point=samples[p]

if label in p\_num: p\_num[label] += 1

else: p\_num[label] = 1

for f in point.keys(): # for every feature

if f>max\_idx: max\_idx=f

### now p\_num and max\_idx are set

### initialize variables

sum\_f = [0 for i in range(max\_idx)]

for la in p\_num.keys():

sum\_l\_f[la] = [0 for i in range(max\_idx)]

sumq\_l\_f[la] = [0 for i in range(max\_idx)]

### pass 2: calculate some stats of data

for p in range(len(samples)): # for every data point

point=samples[p]

label=labels[p]

for tuple in point.items(): # for every feature

f = tuple[0]-1 # feat index

v = tuple[1] # feat value

sum\_f[f] += v

sum\_l\_f[label][f] += v

sumq\_l\_f[label][f] += v\*\*2

### now sum\_f, sum\_l\_f, sumq\_l\_f are done

### for each feature, calculate f-score

eps = 1e-12

for f in range(max\_idx):

SB = 0

for la in p\_num.keys():

SB += (p\_num[la] \* (sum\_l\_f[la][f]/p\_num[la] - sum\_f[f]/data\_num)\*\*2 )

SW = eps

for la in p\_num.keys():

SW += (sumq\_l\_f[la][f] - (sum\_l\_f[la][f]\*\*2)/p\_num[la])

F[f+1] = SB / SW

return F

**4.5 Measuring the effectiveness of feature selection**

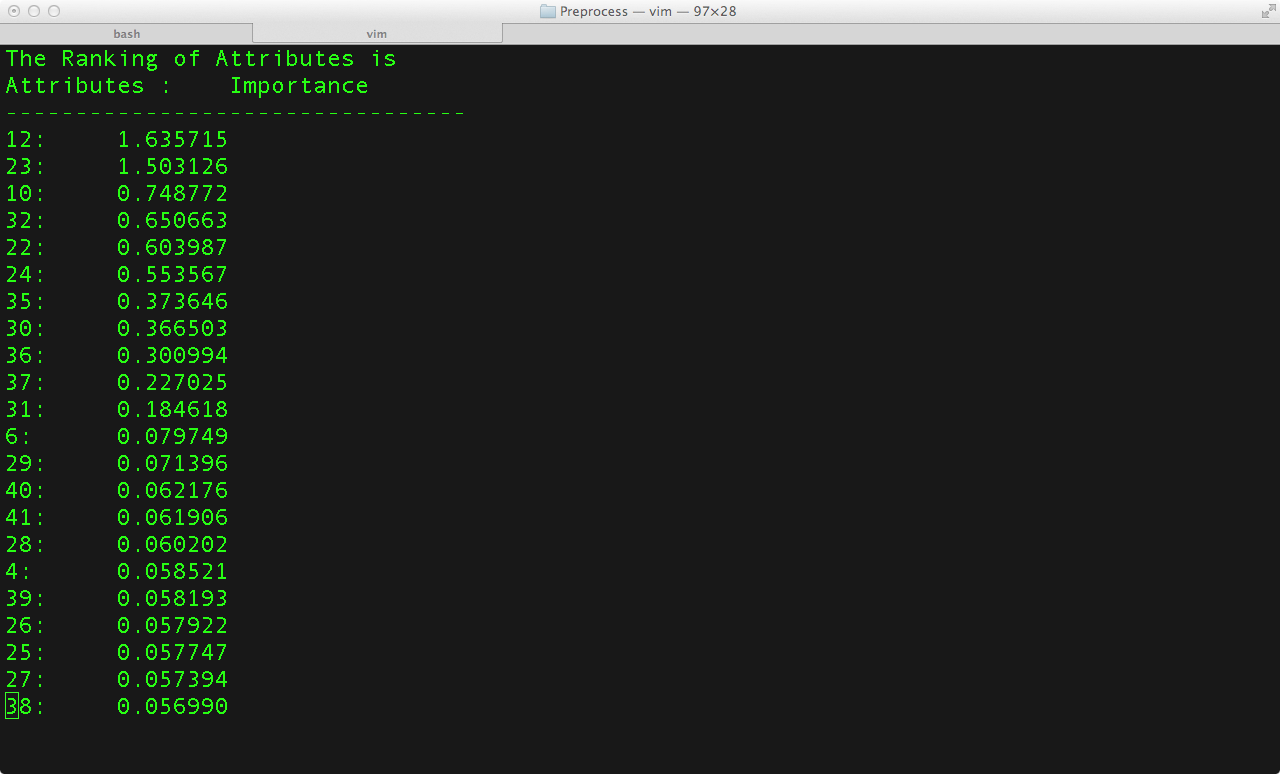
The accuracy of the reduced feature set obtained was verified by comparing it with the accuracy of the whole feature set. Accuracy of the feature set was calculated by calculating the classification accuracy resulting from the use of a particular feature set. After the reduced feature set was obtained, only those features were considered for training and testing the data records using LIBSVM, a library for SVM classifier which can be used for classifying data. Also, the whole feature set was considered for classification. Reduced feature set yielded as high results as the normal feature set.

|  |  |
| --- | --- |
| **Number of Features** | **Accuracy Obtained** |
| 29 | 99.60000 |
| 14 | 99.42000 |
| 7 | 99.18000 |
| 3 | 98.04000 |

**Figure 1: Accuracy obtained with different number of features with F-Score algorithm.**

This is the result of analysis of considering different number of top attributes obtained through F-Score algorithm for classification. This figure clearly indicates that major classification is dependent on a small set of features, But in the need of very high accuracy we have taken 29 attributes out of 41 for our future work hence reducing the computation required and removing the noise.

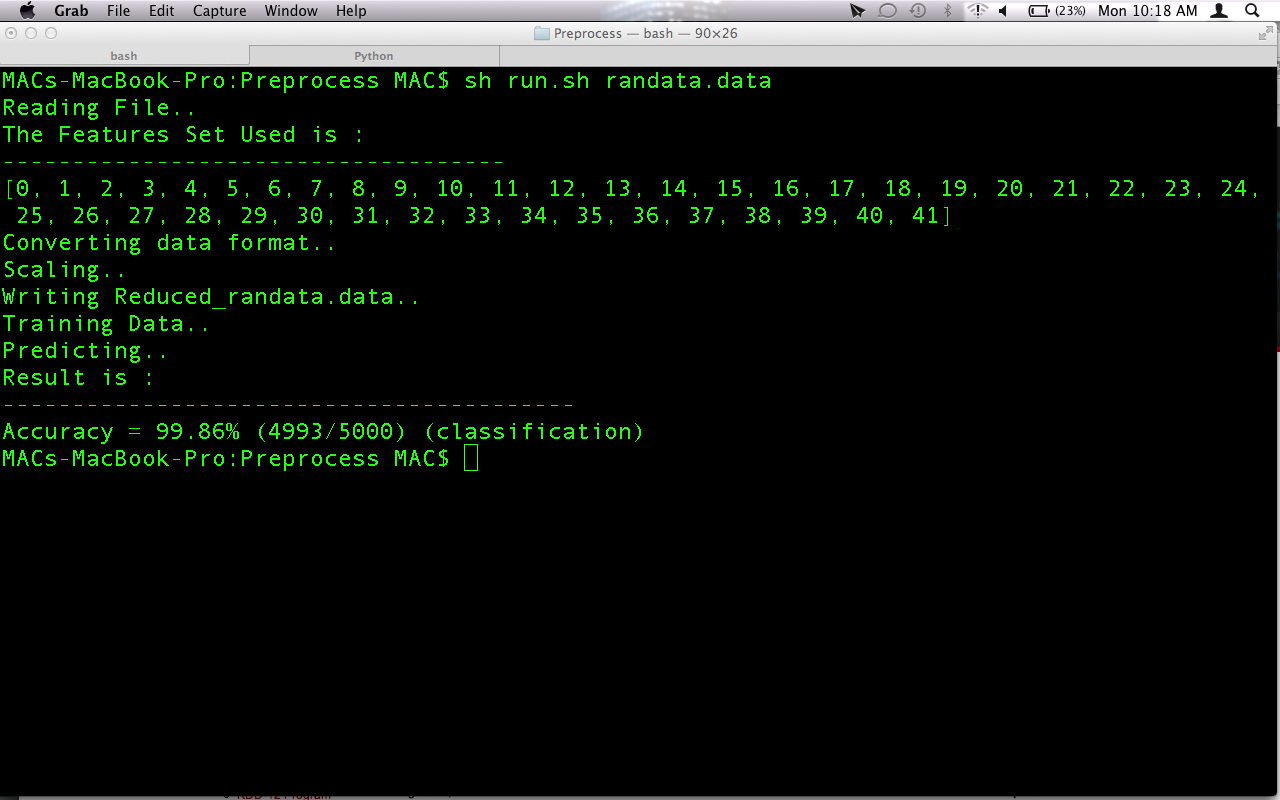
**5. RESULTS AND ANALYSIS**



**Figure 2: Results of F-score Algorithm**

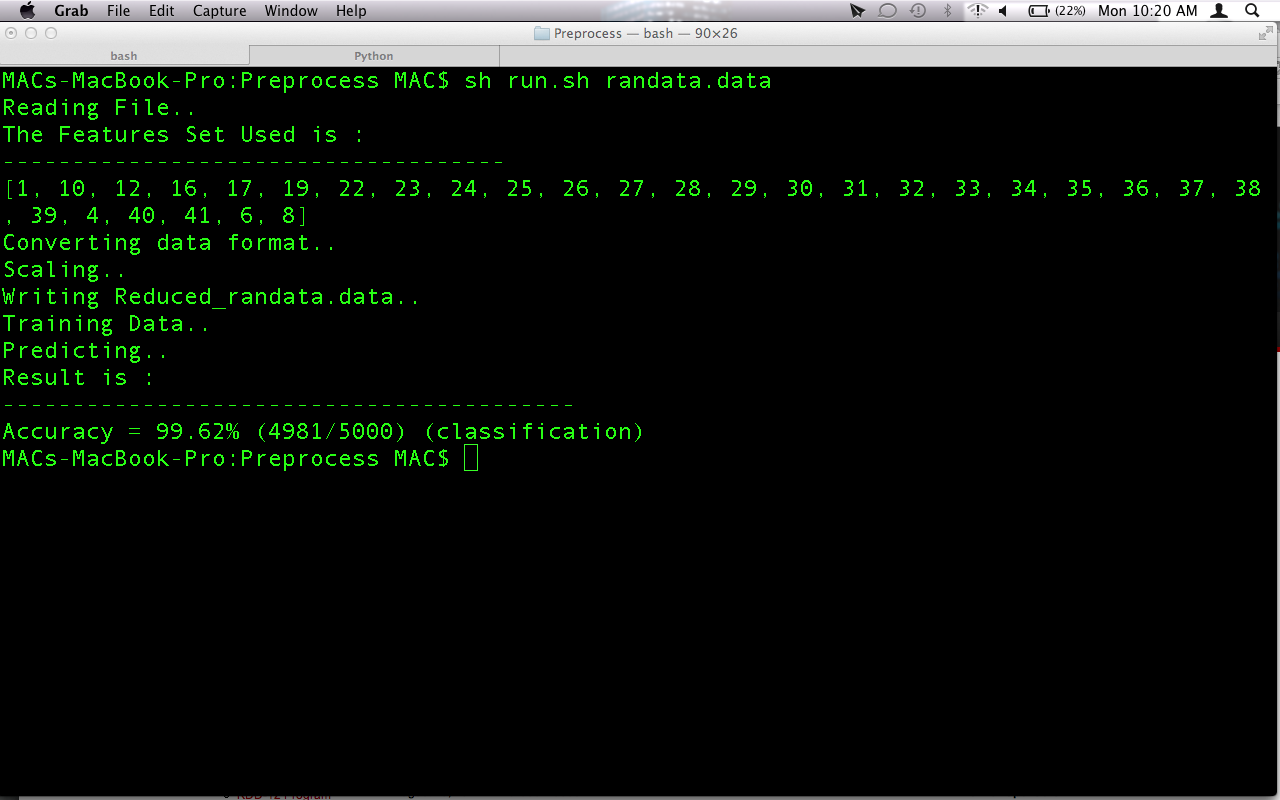
**Analysis:** Attributes in the decreasing order of their contribution to the classification.

We have used initially quantized the data by giving each non- numeric data a numeric value and then later the quantized data is scaled by the formula **(X-MIN)/(MAX- MIN**). Hence this preprocessed data is passed through F-Score (Appendices) to obtain the Importance of each feature in classification.

**Figure 3: Output indicating the Accuracy of SVM Classifier using all the features**

**Analysis:** Accuracy of the SVM classifier using all the features is 99.86 %.

An SVM library is used to classify the data in to normal and attacks here all the features are used to do classification and Accuracy is obtained by :( **NO. OF CORRECT PREDICTION)**/ (**TOTAL NUMBER OF PREDICTION**) is **99.86%**



**Figure 4: Output indicating the Accuracy of SVM Classifier using the reduced feature set and the feature set**

**Analysis:** Accuracy of the SVM classifier using reduced feature set is 99.62 %.

An SVM library is used to classify the data in to normal and attacks. Here only the selected features are used to do classification. The selected features are 1, 4, 6, 8, 10, 12, 16, 17, 19, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41 and Accuracy is obtained by :( **NO. OF CORRECT PREDICTION)**/ (**TOTAL NUMBER OF PREDICTION**) is **99.62%**.

**6. CONCLUSIONS AND FUTURE WORK**

After completing the first two phase of our project, we arrived at the following conclusions:

* Feature selection can be done to select only the most crucial attributes needed for the classification of anomalies without compromising on the accuracy of classification on account of a reduced feature set.
* Data points which are quantitative in nature can be clustered using fuzzy c-means clustering so that similar data points form a cluster and dissimilar points lie in different clusters.
* Kuok’s Algorithm can be used to mine fuzzy rules which are used for classification of attacks.

Future work:

* Fine tune the fuzzy rule set to eliminate rules which do not contribute towards the classification and to make the fuzzy logic system adaptable to classifying new types of attacks.
* Increase the efficiency of the classification using a unique classification technique for each type of attack to reduce the false positivity of fuzzy classification and to get optimum values for each accuracy parameter-TP, TN, FP and FN.

**7. REFERENCES**

[1]. Mohammadreza Ektefa, Sara Memar, Fatimah Sidi, Lilly Suriani Affendey. “Intrusion Detection Using Data Mining Techniques”. IEEE-International Conference On Information Retrieval and Knowledge Management March 17-18, 2010.

[2]. M. Moorthy, Dr. S. Sathiyabama. “A Study of Intrusion Detection using Data Mining”. IEEE-International Conference On Advances In Engineering, Science And Management (ICAESM -2012) March 30-31, 2012.

[3]. German Florez, Susan M. Bridges, Rayford B. Vaughn. “An Improved Algorithm for Fuzzy Data Mining for Intrusion Detection”. Annual Meet of the North American Fuzzy Information Processing Society Proceedings, 2002.

[4]. Andrew H. Sung, Srinivas Mukkamala. “The Feature Selection and Intrusion Detection Problems” lecture notes.

[5]. Andrew H. Sung, Srinivas Mukkamala. “Identifying Significant Features for Network Forensic Analysis Using Artificial Intelligent Techniques”. International Journal of Digital Evidence, 2003.

[6]. Yui-Wei Chen, Chih Jen-Lin. “Combining SVM with various Feature Selection techniques”. LIBSVM Library.

[7]. Ming-Chuan Hung, Don-Lin Yang. “An Efficient Fuzzy C-Means Clustering Algorithm”. Proceedings IEEE International Conference on Data Mining, 2001.

[8]. Rakesh Agrawal, Ramakrishnan Srikant. “Fast Algorithm for Mining Association Rules”. Proceedings of the 20th VLDB Conference, Santiago, Chile, 1994.

[9]. Chan Man Kuok, Ada Fu, Man Hon Wong. “Mining Fuzzy Association Rules in Databases”. Newsletter ACM Sigmoid Record, Volume 27 Issue 1, March 1998.

[10]. Aly El–Semary, Janica Edmonds, Jes´us Gonz´alez-Pino, Mauricio Papa. “Applying Data Mining of Fuzzy Association Rules to Network Intrusion Detection”. Proceedings of the 2006 IEEE Workshop on Information Assurance United States Military Academy, West Point, NY.

[11]. Jianjiang Lu, Baowen Xu, Hongji Yang. “A Classification Method of Fuzzy Association Rules”. IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications 8-10 September 2003, Lviv, Ukraine.

[12]. <http://en.wikipedia.org/wiki/Fuzzy_clustering>

[13]. <http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/cmeans.html>

[14]. <http://en.wikipedia.org/wiki/Apriori_algorithm>

**8. APPENDICES**

**Apriori Algorithm:**

Apriori Algorithm is used to mine association rules from databases. Given a set of itemsets , the algorithm attempts to find subsets which are common to at least a minimum number C of the itemsets. Apriori uses a bottom up approach, where frequent subsets are extended one item at a time (candidate generation step), and groups of candidates are tested against the data. The algorithm terminates when no further successful extensions are found.

Let I = {i1; i2; . . .;im} be a set of literals, called items. Let D be a set of transactions, where each transaction T is a set of items such that T CI. Associated with each transaction is a unique identifier, called its TID. A transaction T contains X, a set of some items in I, if X CT. An association rule is an implication of the form X =>Y , where X CI, Y CI, and X^Y = Ф. The rule X =>Y holds in the transaction set D with confidence c if c% of transactions in D that contain X also contain Y. The rule X => Y has support s in the transaction set D if s% of transactions in D contain XUY.

The first pass of the algorithm counts item occurrences to determine the large 1-itemsets, L1. A subsequent pass, say k, consists of two phases. First, the large itemsets Lk-1 found in the (k-1)th pass are used to generate the candidate itemsets Ck, using the apriori- gen function. Next, the database is scanned and the support of candidates in Ck is counted. For fast counting, we need to efficiently determine the candidates in Ck that are contained in a given transaction t.

Apriori Algorithm:

1) L1 = {large 1-itemsets};

2) for ( k = 2; Lk-1  Ф; k++ ) do begin

3) Ck = apriori-gen(Lk-1 ); // New candidates

4) forall transactions t 2 D do begin

5) Ct = subset(Ck , t); // Candidates contained in t

6) forall candidates c E Ct do

7) c.count++;

8) end

9) Lk = {c E Ck |c:count > minsup}

10) end

11) Answer = Uk Lk;

Apriori Candidate Generation:

The apriori-gen function takes as argument L k-1, the set of all large (k -1) itemsets. It returns a superset of the set of all large k-itemsets. The function works as follows. 1 First, in the join step, Lk is joined with L k-1. Next, in the prune step, all itemsets c Є Ck such that some (k-1)-subset of c is not in L k-1 are deleted.

insert into Ck select p.item1, p.item2, ..., p.item k-1, q.itemk-1 from L k-1 p, L k-1 q where p.item1 = q.item1, . . ., p.item k-2 = q.item k-2, p.item k-1 < q.item k-1;

1. forall itemsets c Є Ck do

2. forall (k-1) subsets s of c do

3. if (s ₡ L k-1) then

4. delete c from Ck;

**F-score Algorithm:**

F-score is a simple technique which measures the discrimination of two sets of real numbers. Given training vectors xk, k = 1,... ,m, if the number of positive and negative instances are n+ and n−, respectively, then the F-score of the ith feature is deﬁned as:



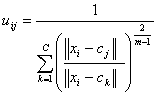
where  are the average of the whole, positive and negative data sets respectively;  is the ith feature of the kth positive instance and is the ith feature of the kth negative instance. The numerator indicates the discrimination between the positive and negative sets, and the denominator indicates the one within each of the two sets. The larger the F-score is, the more likely this feature is more discriminative. Therefore, this score is used as a feature selection criterion.

**Fuzzy C-Means Clustering:**

Fuzzy clustering is a class of [algorithms](http://en.wikipedia.org/wiki/Algorithm) for [cluster analysis](http://en.wikipedia.org/wiki/Cluster_analysis) in which the allocation of data points to clusters is not hard but fuzzy in the same sense as [fuzzy logic](http://en.wikipedia.org/wiki/Fuzzy_logic). Each point has a degree of belonging to clusters, in fuzzy clustering, rather than belonging completely to just one cluster. It is based on minimization of the following objective function:

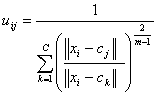
http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/images/image019.gif,     http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/images/image021.gif

where m is any real number greater than 1, uij is the degree of membership of xi in the cluster j, xi is the ith of d-dimensional measured data, cj is the d-dimension center of the cluster, and ||\*|| is any norm expressing the similarity between any measured data and the center.  
Fuzzy partitioning is carried out through an iterative optimization of the objective function shown above, with the update of membership uij and the cluster centers cj by:

,     

This iteration will stop when http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/images/image027.gif, where http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/images/image002.gif is a termination criterion between 0 and 1, whereas k are the iteration steps. This procedure converges to a local minimum or a saddle point of Jm.

Steps:

1. Initialize U=[uij] matrix, U(0)
2. At k-step: calculate the centers vectors C(k)=[cj] with U(k)
3. Update U(k) , U(k+1)
4. If || U(k+1) - U(k)||<http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/images/image002.gif then STOP; otherwise return to step 2.

**Fuzzy Logic:**

Fuzzy logic is a form of [many-valued logic](http://en.wikipedia.org/wiki/Many-valued_logic) or [probabilistic logic](http://en.wikipedia.org/wiki/Probabilistic_logic); it deals with [reasoning](http://en.wikipedia.org/wiki/Reasoning) that is approximate rather than fixed and exact. In contrast with traditional logic which can have varying values, where [binary](http://en.wiktionary.org/wiki/binary) sets have [two-valued logic](http://en.wikipedia.org/wiki/Two-valued_logic), true or false, fuzzy logic variables may have a [truth value](http://en.wikipedia.org/wiki/Truth_value) that ranges in degree between 0 and 1.