Detection of e-mail concerning criminal activities using association rule-based decision tree

S. Appavu alias Balamurugan*

Department of Information Technology,
Thiagarajar College of Engineering,
Madurai, India
E-mail: sbit@tce.edu
***Corresponding author**

R. Rajaram

Department of Computer Science and Information Technology,
Thiagarajar College of Engineering,
Madurai, India
E-mail: rrajaram@tce.edu

Abstract: Detection of e-mails about criminal activities using association rule-based decision tree is studied here. Instead of using words, word-relation that is, association rules from these words, is used for building decision tree. In our experiments, we first preprocess data. We then find out association relations among these words using Rakesh Agrawal et al.'s Apriori algorithm applying objective interestingness measures. These rules are used for training and testing the decision tree-based classification system. A discussion of the result obtained is also given.

Keywords: data mining; decision tree; association rules; Apriori algorithm.

Reference to this paper should be made as follows: Appavu, S. and Rajaram, R. (2007) 'Detection of e-mail concerning criminal activities using association rule-based decision tree', *Int. J. Electronic Security and Digital Forensics*, Vol. 1, No. 2, pp.131–145.

Biographical notes: S. Appavu alias Balamurugan received his ME in Computer Science from University of Madras in 2003 and is pursuing PhD in Information and Communication Engineering at Anna University, Chennai. Currently, he is a Lecturer at the Department of Information Technology, Thiagarajar College of Engineering, Madurai, India. His current research interest includes data mining and cyber security.

R. Rajaram is the Dean of Computer Science at Thiagarajar College of Engineering, Madurai, India. He received his PhD in Electrical Engineering from Madurai Kamaraj University, India. His current research focuses on text mining and information security.

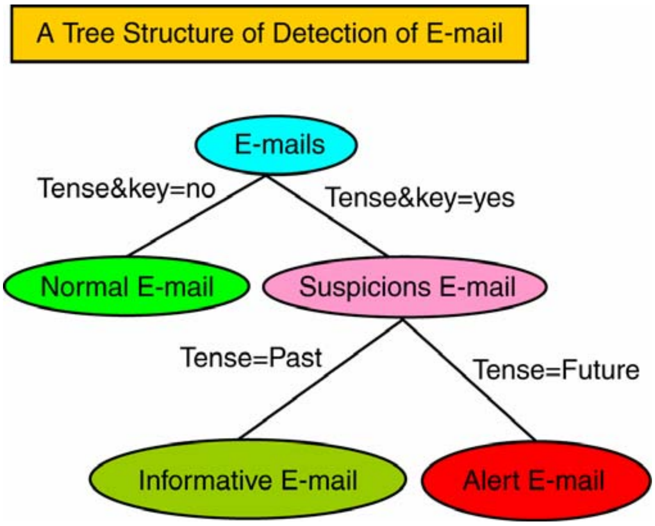
1 Introduction

Data mining has recently involved considerable attention from database practitioners and researchers because of its applicability in many areas such as decision support, market strategy, financial forecasts, etc. Combining techniques from the fields like statistics, machine learning, databases, etc., data mining helps in extracting useful and invaluable information from database. E-mail has become one of today's standard means of communication. E-mail data is also growing rapidly, creating needs for automated analysis. So, to detect crime, a spectrum of techniques should be applied to discover and identify patterns and make predictions. Data mining has emerged to address problems of understanding ever-growing volumes of information for structured data, finding patterns within data that are used to develop useful knowledge. As individuals increase their usage of electronic communication, there has been research into detecting deception in these new forms of communication. Models of deception assume that deception leaves a footprint. Work done by various researches suggests that deceptive writing is characterised by reduced frequency of first-person pronouns and exclusive words and elevated frequency of negative emotion words and action verbs. We apply, this model of deception and also the novel rich features to the set of e-mail dataset and preprocess the e-mail body, and to train the system, we used decision tree algorithm that categorises the e-mail as deceptive or not.

2 Problem statement and related works

The tragedy of 11 September is immeasurable and it has a permanent effect on the USA and the rest of the entire world. In order to avoid such disasters in future, effective security system needs to be established. Moreover, it is also being identified that the terrorists have used the e-mail as the medium for transferring information among them. In order to prevent such a tragedy in future, there is a need to identify the possible meaning of the information, which is exchanged among the terrorists. Thus, the problem is to find a system that identifies the deception in communication through e-mails (Figure 1). Dash and Liu (1997) proposed the feature selection algorithm for classification. David and Lewis (1992) focus on feature selection and feature extraction for text categorization. A recursive partitioning decision rule for nonparametric classification was discussed by Friedman (1977). An overview of various Machine learning algorithm was reported by Witten and Frank (2006). A various concepts and techniques of Data mining was described by Han and Kamber (2005). Tang et al (2005) provide concepts in Email Data Cleaning. Mingers (1989) presented a empirical comparison of selection measures for decision Tree Induction.Quilan (1986) proposed a first Decision Tree learning algorithm –ID3 for Text categorization. The concept of C4.5: Programs for Machine Learning was described by Quinlan (1993). Yang (1999) provided an evaluation of statistical approaches to text categorization. Yang and Pedersen (1997) have done a comparative study on feature selection in text categorization.

Figure 1 Categorisation of e-mails



The informative e-mails give detail about the past historic criminal activities by enhancing some common sense in us such as in the example shown above. We come to know that these types of e-mail will never have any consequences in future. The alert e-mails are identified using the deceptive theory and the present/future tense verbs, by which the security enforcing methods can be strengthened. Also, we can prevent the occurrences of future attacks.

Suspicious e-mail	*Normal e-mail*
Today there will be bomb blast in parliament house and the US consulates in India at 11.46 am. Stop it if you could. Cut relations with the USA. Long live Osama Finladen Asadullah Alkalfi	Hope your fine! How are u and family Members?

Alert e-mail	*Informative e-mail*
Today there will be bomb blast in parliament house and the US consulates in India at 11.46 am. Stop it if you could. Cut relations with the USA. Long live Osama Finladen Asadullah Alkalfi	The World Trade Center was attacked on 9 November 2001 by Osama BinLaden and his followers

Association rules have received much attention in the past. Rakesh Agrawal, Usama M. Fayyad, T. Imielinski, J.M. Bugajski, Ramakrishnan Srikant, H. Toivonen, H. Mannila, T. Zhang, C. Silverstein and many other scintillating researchers have worked here. There are two fundamental problems in the study of association rules: association rules and mining association rules. Agrawal et al. (1996) have applied association rule mining to the problem of classification. ARCS and associative classification use association rules for classification. CAEP mines 'emerging patterns' that consider the concept of support used in mining associations. An alternative classifier, called the JEP-classifier, was proposed based on Jumping Emerging Patterns (JEPs). In this work, ID3 is used for the analysis of e-mail categorisation system based on decision tree using association relations rather than using individual word as feature, which is a fully statistical approach.

One of the first decision tree algorithms, ID3 has applied the Chi-square statistic to the null hypothesis about the irrelevance of a test attribute. However, most other methods of decision tree learning like CART and C4.5 have adopted the post pruning approach for the sake of exploring a large set of potentially valid patterns. Keila and Skillicorn (2005) developed a method based on Singular Value Decomposition to detect Unusual and Deceptive Communication in e-mail. The problem with this approaches is that it does not deal with incomplete data in an efficient and elegant way and cannot incorporate new data incrementally without having to reprocess the entire matrix. List of existing approaches for classification and detection of threaten e-mail is only limited. Appavu and Rajaram (2007a–d) applied a Decision Tree (ID3) algorithm for threaten e-mail detection, the feature selection method applied in this paper cannot use Tense used in the e-mail as one of the features, as a result, it fails to differentiate Alert e-mail from normal. Appavu and Rajaram (2007a–d) presented association rule mining for suspicious e-mail detection and problem with approach is that it requires exponentially more computational effort. Appavu and Rajaram (2007a–d) compared a cross experiment between 4 classification methods including Decision Tree, Naive Bayes, SVM and NN for the classification of e-mail in to Threaten or Normal and ID3 performed well than other classifiers.

3 Mining association rules

Association rules are patterns discovered in data that includes the concept of transaction, basket or group. A common example of a transaction is the set of items someone buys during a supermarket trip. Association rules are typically found via the process of data mining which searches the database for patterns. These patterns can lead to concrete business decisions.

Several objective measures of association rule interestingness exist based on simplicity, certainty, utility and novelty. We will use certainty measure confidence and utility function support. Association rules that satisfy both a user-specified minimum confidence threshold and user-specified minimum support threshold are strong association rules and are considered interesting. Rules below the threshold likely reflect noise, rare or exceptional cases or minority cases and are excluded. Itemset satisfying minimum support is a frequent itemset. Association rule mining is a two-step process:

1 find all frequent itemsets

2 generate strong association rules from the frequent itemsets.

Additional interestingness measures can be applied, if desired. The overall performance of mining association rules is determined by the first step. Apriori is an influential algorithm for mining frequent itemsets using candidate generation for Boolean association rules. In order to use the Apriori property, all nonempty Subsets of a frequent itemset must also be frequent. Once the frequent itemsets from transactions in a database have been found, we generate strong association rules from them using the following equation for

$$\text{Support } (A \times B) = P(A \times B)$$

$$\text{Confidence } (A \times B) = P(B \mid A) = P\frac{(A \cup B)}{P(A)}$$

where support count ($A \times B$) is the number of transactions containing the itemsets $A \times B$, and support_count (A) is the number of transactions containing the itemset A.

4 Preparing e-mail for categorisation

E-mail categorisation is the automated assigning of natural language texts to predefined categories based on their content. E-mail stored in most text databases are semi structured data in that they are neither completely unstructured nor completely structured. During the first stage, the full text of an e-mail to be classified must be parsed to produce a list of potential features that could serve as a basis for categorisation. Incomplete, noisy and inconsistent e-mail data are commonplace properties of large real-world databases and data warehouses. In our experiments, we do some sort of data preprocessing for quality decisions. We have the list of common words in a database file. If words of this list are found in any input file, they are removed. Word stemming can either be performed by a morphological algorithm, which requires a lexicon and the morphological rules for the language can be approximated. We do not follow the former way; hence we have to replace manually similar words by the stem word to compensate this limitation. We then find out association relations among these words. For using more linguistic knowledge, one may extract phrases from the document text. Some users may like to find associations between pairs of keywords or terms from a given set of keywords or phrases, whereas others may wish to find the maximal set of terms occurring together. Therefore, based on user mining requirements, standard association mining or max-pattern mining algorithms may be evoked.

We have collected over 5000 e-mails through a brainstorming session, some of them are as follows and the first example is a real example (Table 1).

> Today there will be bomb blast in parliament house and the US consulates in India at 11.46 am. Stop it if you could. Cut relations with the U.S.A. Long live Osama Finladen Asadullah Alkalfi.

Minimising the number of features or attributes in a classification model is important for several reasons, from increasing the learning speed of a classification algorithm to dealing with the; 'curse of dimensionality' problem in parameter estimation. The feature filter model assumes selecting the feature before applying an induction algorithm, while the wrapper model uses the prediction accuracy of the Induction algorithm itself to evaluate the features. The wrapper approach is usually associated with a considerable computational effort since it requires the rerunning of an induction algorithm multiple times. The filter methods, on the other hand, are computationally cheaper, but there is a danger that the features selected by a filter method will not allow a classification algorithm to fully exploit its potential. The association rule mining presented in this paper implements automated feature selection 'Associations between pairs of terms and phrases' as an integral part of learning process.

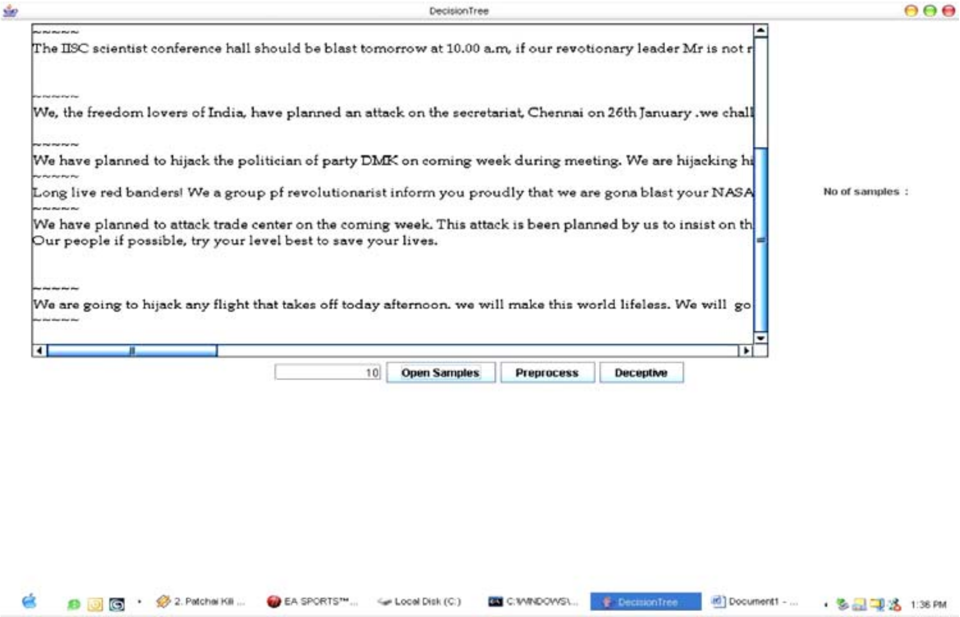
Table 1 Description of e-mails used in the experiment

E-mail ID	*Message*
E1	The IISC scientist conference hall should be blast tomorrow wat 10.00 am
E2	Imamali group has planned to cause a blast in meenakshi temple in the month of February.
E3	Indian airlines Boeing-727 from Chennai to Delhi wiLL be hijacked tomorrow
E4	There is going to be terrorist attack in Chennai airport
E5	Beware! Tomorrow, thene will be a bomb blast at CM' office

In this paper, training dataset is preprocessed with the help of java tokeniser class. We develop a preprocessor module which eliminates all the special characters, unwanted characters and also helps to search the particular keywords (Figures 2 and 3).

To illustrate training examples for the target concept, threaten e-mail detection, consider the learning task represented by training examples of Table 2. Here, the target attribute (also called class attribute) which can have values Alert/Informative/Normal, is to be predicted based on other attributes.

Figure 2 E-mail message before preprocessing



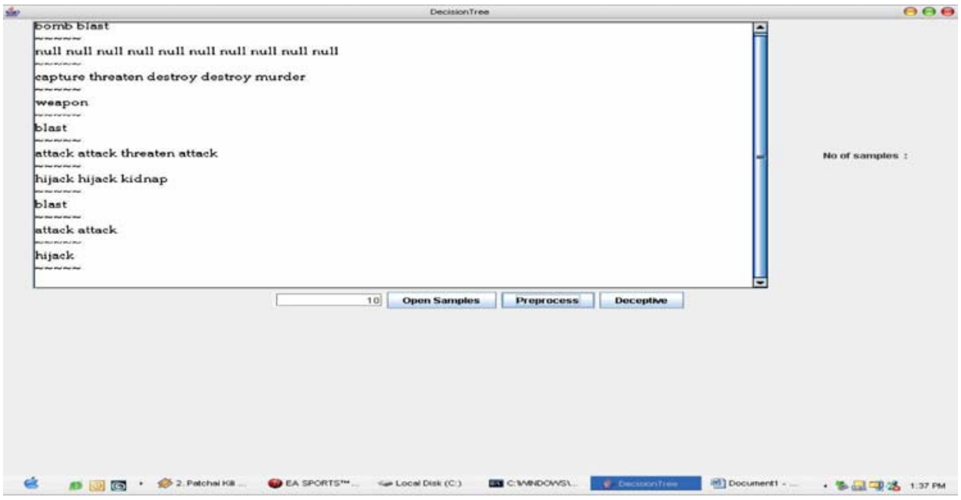
~~~~~~
The IISC scientist conference hall should be blast tomorrow at 10.00 a.m, if our revotionary leader Mr is not r

~~~~~~
We, the freedom lovers of India, have planned an attack on the secretariat, Chennai on 26th January .we chall

~~~~~~
We have planned to hijack the politician of party DMK on coming week during meeting. We are hijacking hi
~~~~~~
Long live red banders! We a group pf revolutionarist inform you proudly that we are gona blast your NASA
~~~~~~
We have planned to attack trade center on the coming week. This attack is been planned by us to insist on th
Our people if possible, try your level best to save your lives.

~~~~~
We are going to hijack any flight that takes off today afternoon. we will make this world lifeless. We will go
~~~~~
~~~~~~

Figure 3 E-mail after preprocessing



```
bomb blast
~~~~~~
null null null null null null null null null null
~~~~~~
capture threaten destroy destroy murder
~~~~~~
weapon
~~~~~~
blast
~~~~~~
attack attack threaten attack
~~~~~~
hijack hijack kidnap
~~~~~~
blast
~~~~~~
attack attack
~~~~~~
hijack
~~~~~~
```

Table 2 A sample training dataset used in experiment

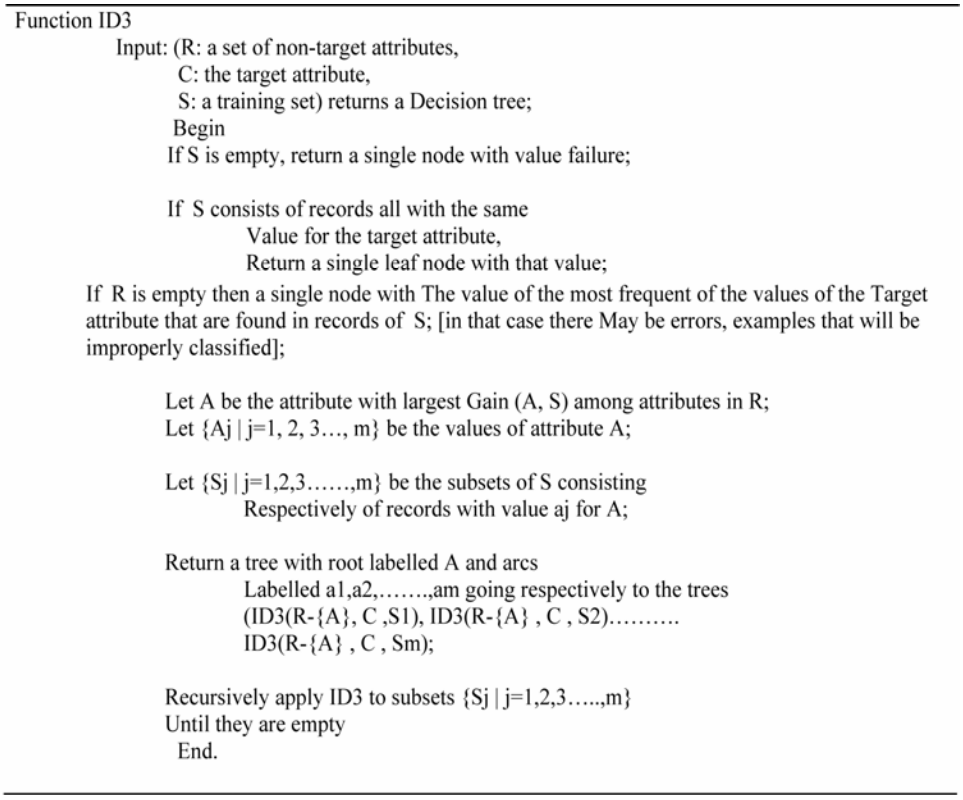
E-mail	*Tense*	*Bomb*	*Blast*	*Terrorist*	*Attack*	*Threaten*	*Class*
1	Past	y	y	y	y	n	Informative
2	Past	n	n	y	y	y	Informative
3	Present	y	y	y	y	n	Alert
4	Future	n	y	n	y	y	Alert
5	Past	n	n	n	n	n	Norma1
6	Present	y	y	y	n	n	Alert
7	Past	n	n	n	n	y	Informative
8	Past	y	y	y	y	y	Informative
9	Future	n	y	n	y	y	Alert
10	Future	y	n	y	n	y	Alert

5 Classification using decision tree

Data classification model may be represented in various forms, such as classification (IF-THEN) rules, mathematical formulae, neural networks or decision trees. A decision tree is a flow-chart-like tree structure, where each node denotes a test on an attribute value, each branch represents an outcome of the test and tree leaves represent classes or class distributions. When a decision tree is built, many of the branches will reflect anomalies in the training data due to noise or outliers. Tree pruning methods address this problem of overfitting the data.

ID3 algorithm is a set of computer programs that construct classification models by discovering and analysing patterns. In the software ID3, a decision tree is generated from a set of training cases. The tree is validated through a set of test (unseen) cases. Our experiment uses ID3, the decision tree generator program for the classification analysis. The fundamental file provides names for classes, attributes and attribute values. We separate and produce training and test sets randomly. The ID3 program produces unpruned and pruned trees analyses and predicts about the data. ID3 uses the training samples to estimate the accuracy of each rule. Since this would result in an optimistic estimate of rule accuracy, ID3 employs a pessimistic estimate to compensate for the bias (Figure 4).

Figure 4 ID3 algorithms for classification



```
Function ID3
        Input: (R: a set of non-target attributes,
              C: the target attribute,
              S: a training set) returns a Decision tree;
             Begin
            If S is empty, return a single node with value failure;

            If  S consists of records all with the same
                    Value for the target attribute,
                    Return a single leaf node with that value;
      If  R is empty then a single node with The value of the most frequent of the values of the Target
      attribute that are found in records of  S; [in that case there May be errors, examples that will be
      improperly classified];

            Let A be the attribute with largest Gain (A, S) among attributes in R;
            Let {Aj | j=1, 2, 3…, m} be the values of attribute A;

            Let {Sj | j=1,2,3……,m} be the subsets of S consisting
                    Respectively of records with value aj for A;

            Return a tree with root labelled A and arcs
                    Labelled a1,a2,…….,am going respectively to the trees
                    (ID3(R-{A}, C ,S1), ID3(R-{A} , C , S2)………
                    ID3(R-{A} , C , Sm);

            Recursively apply ID3 to subsets {Sj | j=1,2,3…..,m}
            Until they are empty
              End.
```

6 Experimental results

After preprocessing, we invoke Apriori algorithm to generate frequent candidate itemsets that are used to generate association rules satisfying minimum support (20%). These rules are added with corresponding class. The ID3 software (decision tree generator) is then invoked to generate decision tree. We select the support and confidence level at 0.50 and 0.60, respectively, because higher confidence level produces fewer association rules to discriminate texts and lower confidence level produces too many association rules to work with. Again, lower support level produces enormous rules to be used for attributes for decision tree generator. In the way to detect e-mail about criminal activities, we have chosen 2500 e-mails for training decision tree and the rest 2500 e-mails are for testing that tree. The e-mails are of three categories: Normal, Informative and Alert. The following Table 3 shows the distribution (Figures 5 and 6):

The input to the association rule generating program gets the values in numerical order. Hence, we are assigning the values to the attribute as given in Table 4.

Table 3 Description of dataset used in the experiment

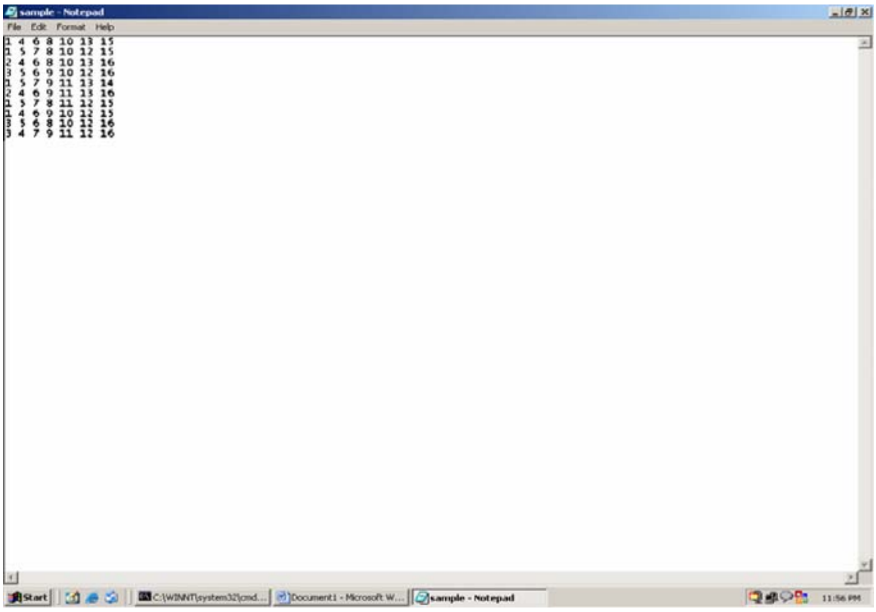
Class types	*Number of training sets*	*Number of test sets*
Normal	1000	1000
Informative	500	500
Alert	1000	1000
Total	2500	2500

Table 4 Assigned value for each attribute

Attribute	*Value 1*	*Value 2*	*Value 3*
Tense	Past = l	Present = 2	Future = 3
Bomb	Yes = 4	No = 5	–
Blast	Yes = 6	No = 7	–
Terrorist	Yes = 8	No = 9	–
Attack	Yes = l0	No = 11	–
Threaten	Yes = l2	No = 13	–
Class	Norma1 = 14	Informative = l 5	Alert = l 6

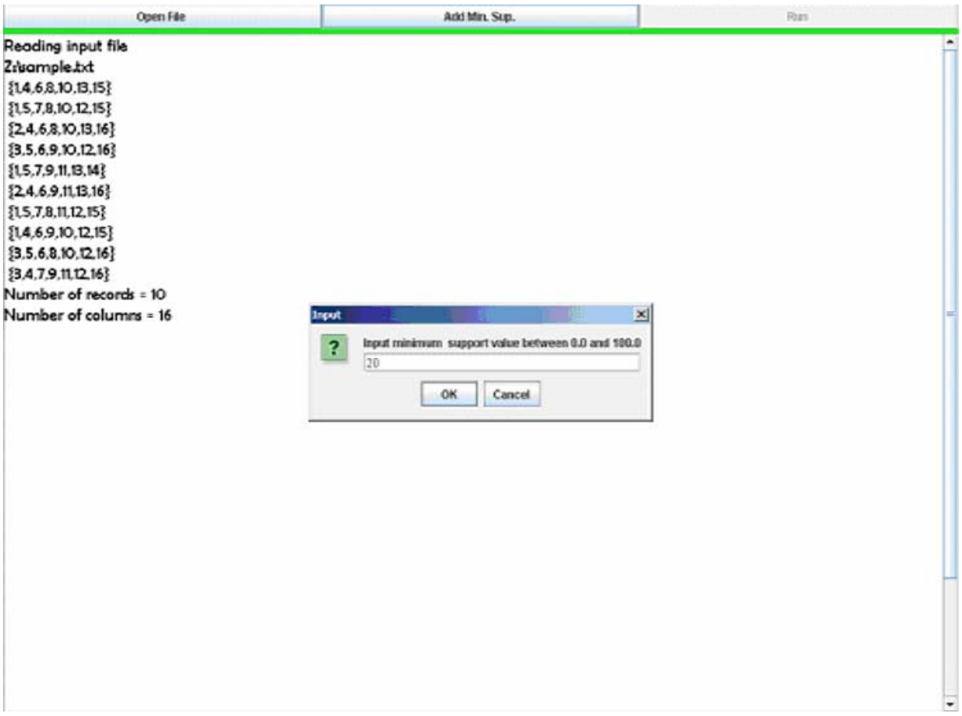
A collection of one or more attributes is called Item sets. In the first iteration of the algorithm, each item is a member of the set of candidate 1-itemsets, C_1. The algorithm simply scans all of the e-mails in order to count the number of occurrence of each attribute as given in Figure 7.

Figure 5 Description of input file used in the experiment



```
1 4 6 8 10 13 15
1 5 7 8 10 12 15
2 4 6 8 10 13 16
3 5 6 9 10 12 16
1 5 7 9 11 13 14
2 4 6 9 11 13 16
1 5 7 8 11 12 15
1 4 6 9 10 12 15
3 5 6 8 10 12 16
3 4 7 9 11 12 16
```

Figure 6 Description of reading input file

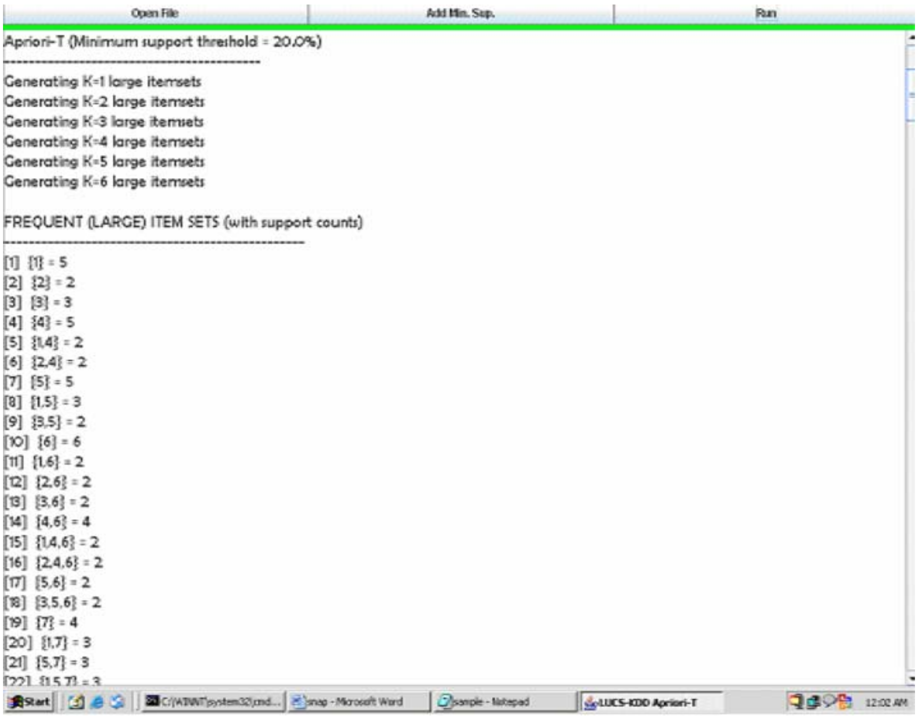


```
Reading input file
Z:\sample.txt
 {1,4,6,8,10,13,15}
 {1,5,7,8,10,12,15}
 {2,4,6,8,10,13,16}
 {3,5,6,9,10,12,16}
 {1,5,7,9,11,13,14}
 {2,4,6,9,11,13,16}
 {1,5,7,8,11,12,15}
 {1,4,6,9,10,12,15}
 {3,5,6,8,10,12,16}
 {3,4,7,9,11,12,16}
Number of records = 10
Number of columns = 16
```

Input minimum support value between 0.0 and 100.0

20

Figure 7 Frequent itemset generation using Apriori algorithm



```
Apriori-T (Minimum support threshold = 20.0%)
---------------------------------------------
Generating K=1 large itemsets
Generating K=2 large itemsets
Generating K=3 large itemsets
Generating K=4 large itemsets
Generating K=5 large itemsets
Generating K=6 large itemsets

FREQUENT (LARGE) ITEM SETS (with support counts)
------------------------------------------------
[1]  {1} = 5
[2]  {2} = 2
[3]  {3} = 3
[4]  {4} = 5
[5]  {1,4} = 2
[6]  {2,4} = 2
[7]  {5} = 5
[8]  {1,5} = 3
[9]  {3,5} = 2
[10]  {6} = 6
[11]  {1,6} = 2
[12]  {2,6} = 2
[13]  {3,6} = 2
[14]  {4,6} = 4
[15]  {1,4,6} = 2
[16]  {2,4,6} = 2
[17]  {5,6} = 2
[18]  {3,5,6} = 2
[19]  {7} = 4
[20]  {1,7} = 3
[21]  {5,7} = 3
[22]  {1,5,7} = 3
```

Based on frequent itemset L, find all non-empty subsets $f \subset L$ such that $f \rightarrow L - f$ satisfies the minimum support requirement given in Table 5.

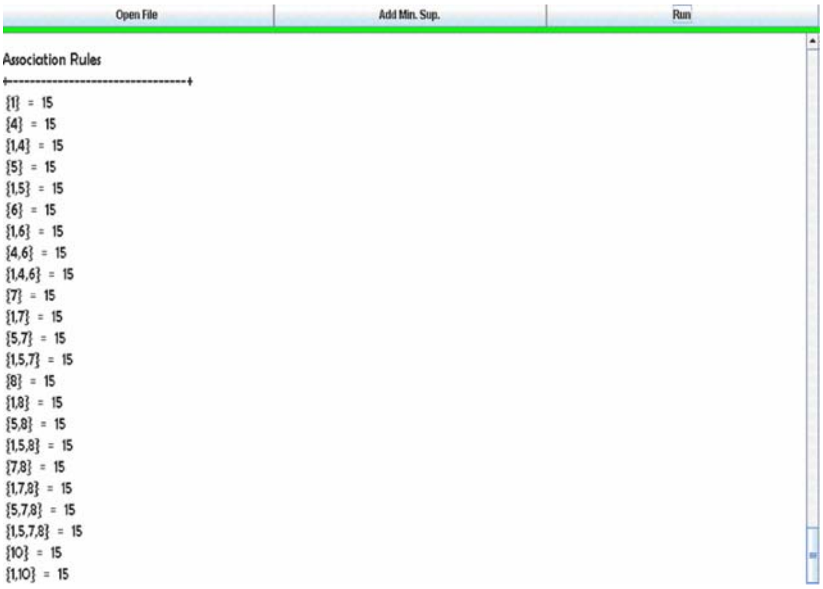
Table 5 Frequent large itemsets generation with minimum support = 20%

{Kill}	E6, E12
{Bomb}	E1, E5, E7, E12
{Bomb, Blast, Attack}	E1, E5, E12
{Bomb, Blast}	E1, E5, E7, E12, E15
{Threaten, Murder}	E1, E8, E14
{Weapons}	E4, E6, E9, E10
{Blast}	E1, E5, E7, E12, E15
{Attack, Threaten}	E4, E9, E11
{Hijack}	E3, E7, E13
{Attack}	E1, E5, E12
{Bomb, Attack, Terrorist}	E2, E3, E8, E13
{Attack, Weapon, Destroy}	E4, E6, E9, E10
{Hijack, Murder}	E3, E7, E13
{Bomb, Destroy}	E9, E11
{Bomb, Attack}	E1, E5, E7, E12, E15

Note: Where E1-E-mail 1, E2-E-mail 2,… E15-E-mail 15, etc.

Once the frequent itemsets from e-mails in a database D have been found, it is straightforward to generate strong association rules from them (where strong association rules satisfy both minimum support and minimum confidence (Figure 8).

Figure 8 Association rule generation through Apriori algorithm



```
Association Rules
+--------------------------------+
{1} = 15
{4} = 15
{1,4} = 15
{5} = 15
{1,5} = 15
{6} = 15
{1,6} = 15
{4,6} = 15
{1,4,6} = 15
{7} = 15
{1,7} = 15
{5,7} = 15
{1,5,7} = 15
{8} = 15
{1,8} = 15
{5,8} = 15
{1,5,8} = 15
{7,8} = 15
{1,7,8} = 15
{5,7,8} = 15
{1,5,7,8} = 15
{10} = 15
{1,10} = 15
```

Table 5 specifies all the frequent *k*-itemsets with their supports. For example, the support of the frequent item Bomb is 20% because 1/5 of the e-mails in the e-mail set contain the item 'Bomb'. Based on the frequent itemsets, which are generated through Association rule mining which are given as input to the decision tree algorithm, using the tabulated values as shown in Table 2, the information gain and entropy of each attribute is calculated. The attribute which has the highest information gain becomes the root node of the tree. This process goes on until all the attributes are mapped in to the tree based on the sorted information gain. Following each individual path in the tree, the rules are generated. The output of this module is decision tree and rules (Figure 9).

Rule 1: If Terrorist = yes AND Attack = yes AND Tense = past then Informative.

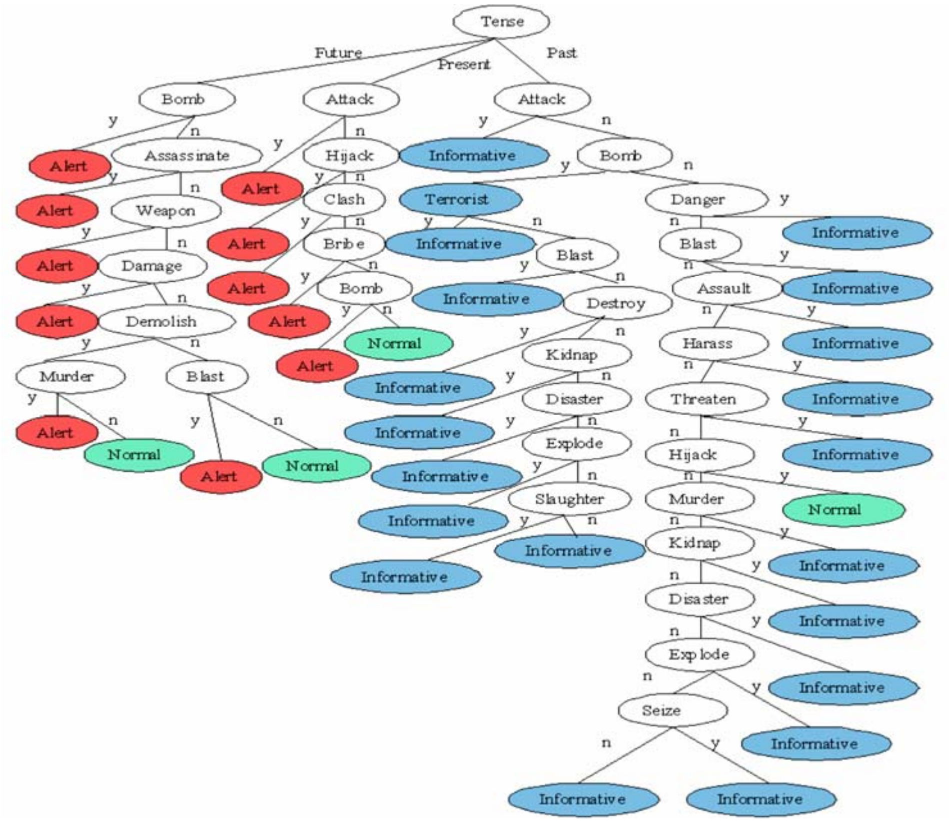
Rule 2: If Threaten = yes AND Tense = future then Alert.

Rule 3: If Tense = future AND Terrorist = yes AND Bomb = yes then Alert.

Rule 4: If Tense = future AND Attack = yes AND Blast = yes then Alert.

Rule 5: If Tense = past AND Terrorist = No AND Attack = No then Normal.

Figure 9 Proposed decision tree for criminal activity e-mail detection



7 Performance evaluation

To evaluate the classifiers on testing dataset, we defined an accuracy measure as follows. Accuracy (%) = correctly_classified_e-mails/Total _e-mails × 100.

An experiment measuring the performance against the size of dataset was conducted using dataset of different sizes. For example, in case of 2500 dataset, accuracy was 97.76% using association rule-based decision tree. Also, precision and recall were used as the metrics for evaluating the performance of each e-mail classification approach (Tables 6–9, Figure 10).

Table 6 Classification accuracy based on data size

No. of record	*NB*	*SVM*	*NN*	*J48*	*ID3*	*Proposed association rule- based decision tree*
500	97.6	98.4	98.6	98.4	98. 8	98.9
1000	90.2	92. 8	91.90	92.9	93.7	95.80
1500	92.26	93.54	93.46	93.9	94.66	96.20
2000	91.45	92.7	92.65	93.06	93.7	96.82
2500	94.04	87.88	95.44	93.36	95.76	97.76

Table 7 Normal e-mail Precision/Recall/TP Rate/FP Rate based on data size of 2500

	NB	*NN*	*SVM*	*J48*	*ID3*	*Proposed association rule-based decision tree*
Prec	0.976	0.964	0	0.788	0.97	0.98
Rec	0.698	0.804	0	0.791	0.83	0.89
TP rate	0.698	0.804	0	0.791	0.83	0.93
FP rate	0.02	0.003	0	0.022	0.003	0.001

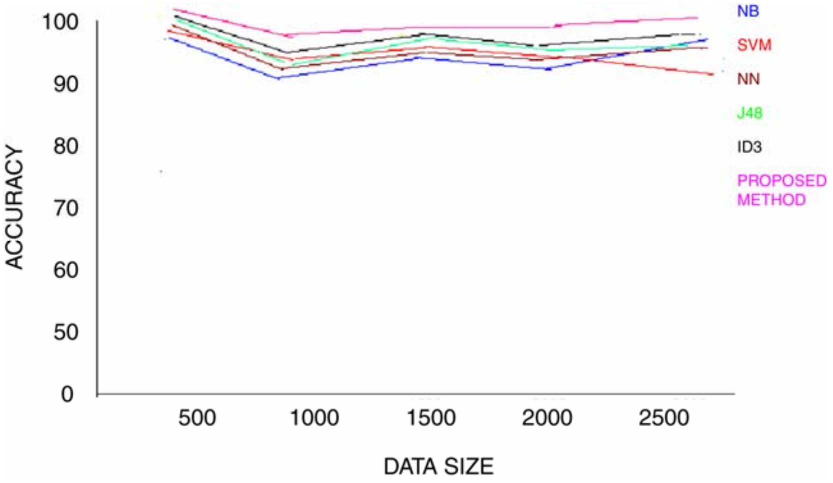
Table 8 Alert e-mail Precision/Recall/TP Rate/FP Rate based on data size of 2500

	NB	*NN*	*SVM*	*J48*	*ID3*	*Proposed association rule-based decision tree*
Prec	0.953	0.979	0.831	0.978	0.984	0.990
Rec	0.950	0.953	0.953	0.907	0.955	0.989
TP rate	0.950	0.953	0.953	0.907	0.955	0.992
FP rate	0.036	0.015	0.147	0.015	0.012	0.007

Table 9 Informative e-mail Precision/Recall/TP Rate/FP Rate based on data size of 2500

	NB	*NN*	*SVM*	*J48*	*ID3*	*Proposed association rule-based decision tree*
Prec	0.925	0.932	0.926	0.926	0.934	0.954
Rec	0.98	0.986	0.986	0.986	0.986	0.991
TP rate	0.98	0.986	0.986	0.986	0.986	0.992
FP rate	0.072	0.065	0.072	0.072	0.063	0.047

Figure 10 Performance of classifiers



8 Conclusion and future work

E-mail is an important vehicle for communication. It is one possible source of data from which potential problems can be detected. In this paper, we have employed decision tree-based classification approach to detect e-mails in relation to criminal activities. All the e-mails were classified as Alert/Informative/Normal. From this experiment, we can find that a simple association rule-based decision tree can provide better classification result for suspicious e-mail detection. In the near future, we plan to incorporate other techniques like different ways of feature selection and classification using other methods. One major advantage of the association rule-based decision tree classifier is that it does not assume that terms are independent and its training is relatively fast. Furthermore, the rules are humanly understandable and easy to be maintained. The proposed work will be helpful for identifying the deceptive e-mail and also assist the investigators to get the information in time to take effective actions to reduce the criminal activities.

A problem we faced when trying to test out new ideas dealing with e-mail systems was an inherent limitation of the available data. Because we only have access to our own data, our results and experiments no doubt reflect some bias. Much of the work published in the e-mail classification domain also suffers from the fact that it tries to reach a general conclusion using very small datasets collected on a local scale.

References

Appavu, S. and Rajaram, R. (2007a) 'A novel data mining approach to detect deceptive communication in email text', *Proceedings of the National Conference on Advanced Computing*, MIT, Chennai, India, pp.179–188.

Appavu, S. and Rajaram, R. (2007b) 'Association rule mining for suspicious email detection: a data mining approach', *Proceedings of the IEEE International Conference on Intelligence and Security Informatics*, New Jersey, USA, pp.316–323.

Appavu, S. and Rajaram, R. (2007c) 'Data mining techniques for suspicious email detection: a comparative study', *Proceedings of the European Conference on Data mining*, Portugal.

Appavu, S. and Rajaram, R. (2007d) 'Suspicious email detection via decision tree: a data mining approach', *Journal of Computing and Information Technology –CIT 15*, pp.161–169.

Agrawal, R., Mannila, H., Srikant, R., Toivonan, H. and Verkamo (1996) 'A fast discovery of association rules, *Advances in Knowledge Discovery and Data Mining*, pp.307–328.

Dash, M. and Liu, H. (1997) 'Feature selection for classification', *Intelligent Data Analysis*, Vol. 1, No. 3.

David, D. and Lewis (1992) 'Feature selection and feature extraction for text categorization. Speech and natural language', *Proceedings of a Workshop held at Harriman*, New York, 23–26 February, San Mateo, CA: Morgan Kaufmann, pp.212–217.

Friedman, J.H. (1977) 'A recursive partitioning decision rule for nonparametric classification', *IEEE Transactions on Computers*, p.404408.

Witten, I.H. and Frank, E. (2006) *Data Mining, Practical Machine Learning Tools and Techniques*.

Han, J. and Kamber, M. (2005) *Data Mining Concepts and Techniques*, Morgan Kaufmann Publishers.

Tang, J., Li, H., Cao, Y. and Tang, Z. (2005) *Email Data Cleaning*, KDD'05, Chicago, USA.

Keila, P.S. and Skillicorn, D.B. (2005) 'Detecting unusual and deceptive communication in email', *Technical Report*.

Mingers, J. (1989) 'An empirical comparison of selection measures for decision-tree induction', *Machine Learning*, Vol. 3, No. 4, pp.319–342.

Quinlan, J.R. (1986) 'Induction of decision trees', *Machine Learning*, Vol. 1, pp.1–106.

Quinlan, J.R. (1993) *C4.5: Programs for Machine Learning*, San Mateo, CA: Morgan Kaufmann.

Yang, Y. (1999) 'An evaluation of statistical approaches to text categorization', *Journal of Information Retrieval*, Vol. 1, Nos. 1/2, pp.67–88.

Yang, Y. and Pedersen, J. (1997) 'A comparative study on feature selection in text categorization', *In ICML*, pp.412–420.