Volume 5, Issue 5, May 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

An Efficient Malicious Email Detection Using Multi Naive Bayes Classifier

Mansi Goyal, Ankita Sharma
CSE& Kurukshetra University
Haryana, India

Abstract — The growing use of e-mail has led to the emergence and further growth of the difficulties caused by unnecessary massive e-mails called as Spam. Anti-spam legal measures are gradually being used, but they have had very limited effect till now. Machine learning (ML) is the most effective approach for the classification of text and this approach is successfully being applied to combat spam. In this paper, an efficient Multi Naive Bayes Classifier with variable length sequence for the classification of malicious mails has been proposed. Multi-Naïve Bayes is an enhanced version of Naïve bayes algorithm. This algorithm consists of multiple Naïve Bayes algorithms that are responsible for on the whole classification of a data set. The comparison of other robust classifiers like Bayes Net, Naive Bayes and Naive Bayes Multinomal with Multi Naive Bayes classifier in terms of correctly and incorrectly classified instances have been done. Various measurements of performance like False Positive rate, Precision and Recall of various Classifiers have been computed.

Keywords— Naive Bayes, Multi Naive Bayes, Classifier, Spam mails, Machine learning

I. INTRODUCTION

EMAIL AND SPAM PROBLEM

E-mail or electronic mail is an electronic messaging scheme which transmits messages across the networks of computers. It has a very simple interface in which users simply have to type in their message and the E-Mail ID of receiver and then click on the send button to send a message. E-mail provides a low cost communication interface in addition to an efficient mail delivery scheme. The various features of email like its ease of use, reliability and availability of a broad collection of free e-mail services make it most accepted and a favoured means of communication. Due to this, businesses and individual users rely on this communication medium for sharing information and knowledge to a huge area.

The growing use of e-mail has led to the emergence and further growth of the difficulties caused by unnecessary massive e-mails called as Spam. It is evolving from a minor trouble to major problems. Spam has started to fade the trustworthiness of e-mail because of the great volume and offensive content of some of these emails.

The effects of spam are becoming severe day by day. Network bandwidth is wasted receiving these unwanted messages and the time spent by users in differentiating between Spam and ham messages is also wasted. Due to this, personal users and companies are greatly affected by Spam. It floods e-mail servers with these messages to the point where it crashes. It poses larger risks to network security and personal privacies. Spam also raises the danger of malwares and viruses. The everyday rising threats of spam certainly have need of strong control measures.

HEURISTIC TECHNIQUES

Traditionally there are many approaches available to control spam such as using sender domain check, content check, open relay prohibition and checking the IP address or domain names. However, spammers easily overcome these simple measures with more sophisticated variants of spam to evade detection.

One of the primary problems faced by the virus community is to devise methods for detecting new malicious programs that have not yet been analysed. Eight to ten malicious programs are created every day and most cannot be accurately detected until signatures have been developed for them. During this time span, systems protected by signature-based algorithms are vulnerable to attacks.

Current virus scanner technology has two parts: a heuristic classifier and a signature-based detector that detects new viruses. Signature-based detection algorithm relies on signatures of known malicious executable codes to generate detection models. Signature-based method creates a unique tag for each malicious program so that future examples of it can be correctly classified with a small error rate. Signature-based methods don't generalize well to detect new malicious binaries because they are created to give a false positive rate as close to zero if possible. Whenever a detection method generalizes to new instance, the trade off is for a higher false positive rate.

Another approach used to control spam is heuristics. The heuristic approach examines the e-mail's content and compares it against thousands of pre-defined rules. These rules are assigned a numerical score that weight the probability of the message being a spam. Each received message is verified against the heuristic filtering rules. The verification result when compared with a pre-defined threshold decides whether the message is spam or not. To filter the e-mails, the score of the

weight is then shared among all the users. Conversely, spammers use obfuscation to fool the rules to avoid detection and modifying heuristic tests to cope with new attack vectors devised by spammers which can be complicated.

Anti-spam legal measures are gradually being used, but they have had very limited effect till now. Anti-spam filters are the software tools that attempt to block automatically spam messages. Aside from blacklisting of frequent spammers and lists of trusted users, which can be incorporated into any anti-spam approach, these filters have until now relied on physically produced patterns of keywords. To be most efficient and to avoid the risk of accidentally removing non-spam messages, also known as legitimate messages, these patterns require to be manually tuned to the incoming e-mail of every user. Fine-tuning of patterns, however, requires expertise and time that are not always present. Even worse, the features of spam messages change with time, requiring that the keyword patterns to be updated from time to time. It is, therefore, required to develop anti-spam filters that should learn *automatically* how to block spam messages using previously received spam and legitimate messages.

MACHINE LEARNING TECHNIQUES (ML TECHNIQUES)

Nowadays, machine learning techniques are used for filtering spam intended to differentiate spam and benign messages. These techniques provide an automated and adaptive approach for spam detection. They do not depend on rules coded by hand, which prove to be ineffective due to the continuously varying signatures of spam messages usage by the spammers. Machine learning techniques works in two steps: firstly they gather information from the given set of messages, and after that use that acquired information to classify freshly received messages. Machine learning is a scientific discipline which is associated with the design & development of algorithms that permit computers to adapt their behavior based on data. Machine learning techniques are trained to identify complex patterns automatically and derive conclusions from that data.

Features

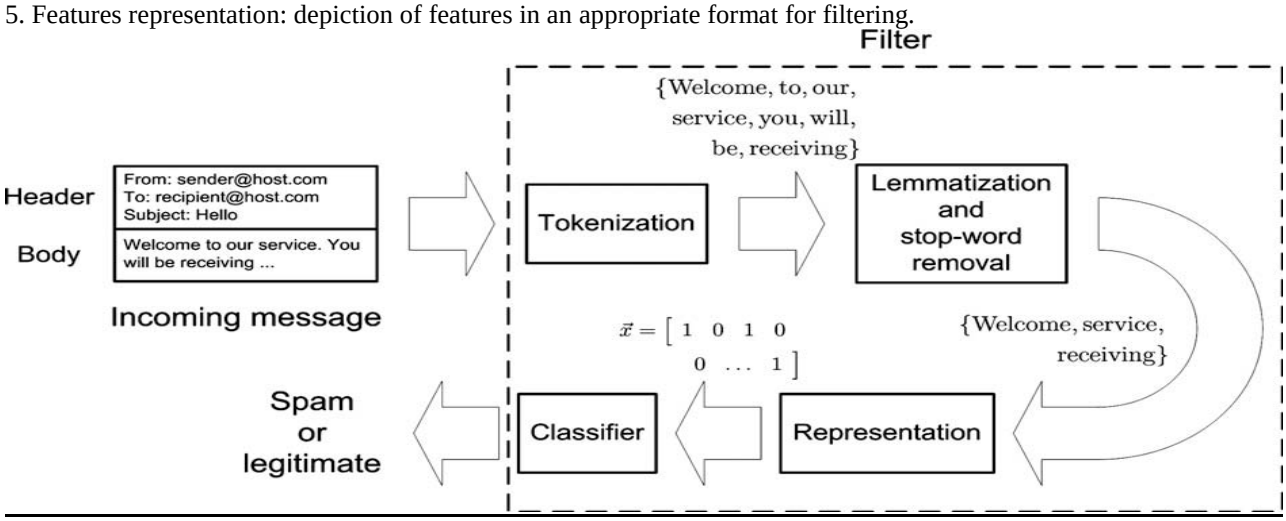
- Able to automatically construct a classifier for a domain by noticing the features of a set of documents that are physically classified by an expert of that domain.
- It is more suitable and easy to classify a set of documents automatically than to construct and code a set of rules.
- Machine learning (ML) is the most effective approach for the classification of text and this approach is successfully being applied to combat spam.
- It alleviates human involvement as machines are used to separate e-mail into spam and non-spam messages, so the cost of examining spam have also been reduced.

PREPROCESSING

The two sections header section and a body section constitute an email. All the general information is contained in the header section like email address of sender and recipient, subject of email as well as the route information whereas the actual message to be transmitted is contained in the body section. Preprocessing is a method by which this information is extracted prior to applying a filter process. The idea behind preprocessing is to convert messages in mail into a standardized format that is comprehensible to the machine learning algorithm.

The sequence of steps that constitutes the process of preprocessing is given below:

1. Feature extraction: drawing out features from e-mail like header or e-mail body into a vector space.
2. Feature selection: Dimensionality reduction that is reduction of the vector of features.
3. Stop word removal: deletion of non-useful words.
4. Noisy removal: elimination of unintelligible symbols or text from features
5. Features representation: depiction of features in an appropriate format for filtering.



NAÏVE BAYES TECHNIQUE

One of the classification techniques that makes use of statistical approach and is based on the conditional probabilities for the problems of patterns recognition is Naïve Bayes algorithm. For learning to classify the text documents, this algorithm is well known. It is also very scalable and quick for reference scoring and building model. Naive Bayes algorithm uses

the concept of Bayes' theorem also called **Bayes' law**. Given the probability of an event that has already taken place, Bayes' theorem computes the probability of another event that is taking place. It is a proficient and practical inductive learning algorithm in the field of machine learning. Naive Bayes algorithm is the most straightforward to apply in comparison to all data mining techniques.

On the basis of features that are contained in the program, the Naive Bayes classifier calculates the probability whether a program is malicious or not. Both string and byte sequence data are used in this method to compute the probability of the maliciousness of a binary. The chief supposition in this approach is that the executables have alike features for example signatures, binary instructions etc. In this technique, all features of each executable are treated as a text document and are classified based on these text documents..

Specifically, we required to calculate the class of a program given that the program consists of a set of features F. We define C to be random variable over the set of classes: *benign* and *malicious* executables classes. That is, here we want to compute P(C/F), the probability that a program is in certain class given the program containing set of features F. Here, we apply Bayes rule and calculate the probability as:

$$P(C|F) = \frac{P(F|C) * P(C)}{P(F)}$$

To use the naive Bayes rule we suppose that the features occur independently from one another. If features of program F include the features F1, F2. F3, F4,...Fn, then above equation becomes:

$$P(C|F) = \frac{\prod_{i=1}^{n} P(F_i|C) * P(C)}{\prod_{j=1}^{n} P(F_j)}$$

Each P(Fi/C) is the frequency that string Fi occurs in a program of class C. P(C) is proportion of class C in the complete set of programs. The output of the classifier is highest probability class for a provided set of strings. Since the denominator is the same for all the classes we take the maximum over all classes C of the probability of each class calculated to get:

$$\text{Most Likely Class} = \max_{C} \left(P(C) \prod_{i=1}^{n} P(F_i|C) \right)$$

Max C denotes a function that returns the class with the highest probability. Most Likely the class in C with the maximum probability and hence the most likely classification of the example having features F. To train classifier, we have recorded how many programs in each class containing unique feature. Here, we used this information to classify new program into a suitable class. Firstly, we used feature extraction to determine the features present in the program. Then we applied equation to compute the most expected class for the program. We used the Naive Bayes algorithm and calculated the most expected class for byte sequences and strings.

MULTI-NAIVE BAYES

Multi-Naïve Bayes is an enhanced version of Naïve bayes algorithm. This algorithm consists of multiple Naïve Bayes algorithms that are responsible for on the whole classification of a data set. Each Naive Bayes algorithm classifies the words in the data set used for testing as benign or malicious and output of each one is considered as a vote. Multi-Naive Bayes algorithm combines these votes of individual classifiers to output a complete classification for all the Naive Bayes. This method was devised because the size of the binary data stored into memory was too large to fit even after using a machine with one gigabyte of RAM. The Naive Bayes algorithm needs a complete set of all strings or bytes to compute its probabilities and produce results.

To rectify the problem with the Naïve bayes method, the Multi Naïve bayes algorithm first of all divides the large problem space into smaller parts that can be easily fitted into the memory and then train a Naive Bayes classifier over each of these sub problems. The data set is divided into a number of small data sets consistently by placing every Ith line in the binary into (i mod n)th data set where n is the number of sub sets. A Naive Bayes classifier is trained for each and every subset. The product of the prediction of these n classifiers is used for the whole prediction for the binary. The Multi-Naive Bayes method takes a vote of confidence among all of the basic Naive Bayes classifiers. Given a set of bytes F, each classifier gives the probability of class C. Multi-Naive Bayes uses these individual probabilities to compute the probability of class C over all the classifiers.

Provided the set of bytes F and the number of probabilities learned by each Naive Bayes classifier, the likelihood of a class C can be calculated. This equation shows the likelihood, L_{NB} (C/F), of class C over a set of bytes F.

$$L_{NB}(C|F) = \prod_{i=1}^{|NB|} P_{NB_i}(C|F) / P_{NB_i}(C)$$

NB_i = Naive Bayes classifier

NB = collective set of all Naive Bayes classifiers.

P_{NB} (C/F) = probability for class C calculated by the Naïve Bayes classifier

P_{NB} (C) = class C probability as computed by Naive Bayes.
Each P_{NBi} (C/F) is divided by P_{NBi} (C) to eliminate the reoccurring probabilities. After that, all the terms are multiplied to compute L_{NB}(C/F), the absolute likelihood of C given F. The result of Multi Naïve bayes classifier is the class having maximum probability over other classes given L_{NB}(C/F) and P_{NB}(C) the earlier probability of a given class.

MEASUREMENT OF PERFORMANCE

To determine the performance of a classifier, following measures are used:
1. **False Positives (FP)** are the count of incorrectly classified legitimate messages.
2. **False Negatives (FN)** are the count of incorrectly classified malicious codes as benign binaries.
3. **True Positives (TP)** are the count of correctly classified malicious executables.
4. **True Negatives (TN)** are the count of benign binary classified as benign.
5. **The detection rate** is ratio of malicious executables that appear to the total number of malicious executables.
6. **Recall** specifies the number of accurately classified spam against spam which is wrongly classified as benign and the spam identified as spam.
7. **Precision** corresponds to the ratio of the numbers of correctly classified spam to the number of messages noticed as spam.
8. **Accuracy** signifies the ratio of the number of correctly classified spam and legitimate mails to the total e-mails employed for testing.

$$Accuracy\ (A) = \frac{TP+TN}{TP+TN+FP+FN}$$

$$Precision(p) = \frac{TP}{TP+FP}$$

$$Recall(r) = \frac{TP}{TP+FN}$$

$$F_i = \frac{2*r*p}{r+p} * 100\%$$

II. RELATED STUDY

A. OVERVIEW OF VARIOUS TEXTUAL ANTI-SPAM FILTERING TECHNIQUES

[1] Thamarai Subramaniam and Hamid A. Jalab in October 2010 had given an outline of a variety of textual anti-spam filtering techniques. The authors argued that machine learning offers improved mechanisms of protection that are capable to combat spam. Every machine learning algorithm has its own benefits and drawbacks. To enhance the efficiency of any machine learning algorithm, proper pre-processing has to be done. The authors have summarized the most general techniques exploited for anti-spam filtering by examining the content of an email. They also took a glance into machine learning algorithms such as Naïve Bayesian Algorithm, support vector machine and neural network that had been accepted to detect and control spam.

B. REVIEW OF MACHINE LEARNING ALGORITHMS USING BOTH TEXTUAL AND IMAGE BASED APPROACHES

[2] Thiago S. Guzella and Walmir M. Caminhas in 2009 presented a complete analysis of the progress in the machine learning algorithm's application to filtering of spam. The authors have paid attention to both textual- and image-based approaches. Despite of regarding Spam filtering as a standard problem of classification, the authors highlighted the significance of considering explicit features of the problem in devising new filters. Two significantly key aspects were also discussed: the difficulties in updating a classifier based on the bag-of-words representation and a major distinction between two early naïve Bayes models.

C. CLASSIFICATION OF MALICIOUS EMAILS USING NAÏVE BAYES CLASSIFIER

[3] Ion Androutsopoulos, John Koutsias, Konstantinos, Constantine D. Spyropoulos and V. Chandrinos in Aug, 2000 proposed a method in which a Naïve Bayesian classifier is automatically trained to identify spam messages. The author tested this method on a large set of personal e-mail messages, which are available in encrypted form publicly. The author presented proper cost-sensitive measures. The author also examined the influence of training data size, lemmatization, size of attribute set, stop lists and the issues that had not been explored till then. Lastly, the Naive Bayesian classifier is compared to a filter makes use of keyword patterns to find its effectiveness.

D. DATA-MINING FRAMEWORK THAT DETECTS NEW, PREVIOUSLY UNSEEN MALICIOUS EXECUTABLES

[4] Matthew G. Schultz and Eleazar Eskin presented a framework of data mining that correctly and automatically identify novel, previously undetected malicious executables. The data-mining framework uses patterns to discover a set

of new malicious binaries which it automatically finds in a data set. This proposed technique results more than doubled the detection rates for novel malicious executables when compared their detection methods with a conventional signature based method.

E. DATA MINING TECHNIQUES TO DETECT AND CLASSIFY ZERO-DAY MALWARE EFFICIENTLY

[5] Mamoun Alazab in 2011 have proposed and evaluated a novel method of employing several data mining techniques to detect and classify zero-day malware with high levels of accuracy and efficiency based on the frequency of Windows API calls. The author also described the methodology employed for the collection of large data sets to train the classifiers, and analysed the performance results of the various data mining algorithms. The data mining framework employed in this research learns through analysing the behaviour of existing malicious and benign codes in large datasets. The author had evaluated the performance of strong classifiers, namely Naïve Bayes (NB) Algorithm, k−Nearest Neighbour (kNN) Algorithm, Back-propagation Neural Networks Algorithm, Sequential Minimal Optimization (SMO) Algorithm and J48 decision tree.

III. PROPOSED WORK

MOTIVATIONS

The recent growth in network usage has motivated the creation of new malicious code for various purposes, including economic and other malicious purposes. Currently, dozens of new malicious codes are created every day and this number is expected to increase in the coming years. Today's signature-based anti-viruses and heuristic-based methods are accurate, but cannot detect new malicious code. Classification algorithms can be used successfully for the detection of malicious code.

RESEARCH METHODOLOGY

Multi-Naive Bayes is a collection of Naïve Bayes algorithms that voted on an overall classification for a given dataset. Each Naive Bayes algorithm classifies the dataset as malicious and this is known as a vote. The votes are combined by the Multi-Naive Bayes algorithm to output a final classification for all the Naive Bayes. This method is required because even using a machine with one gigabyte of RAM, the size of the binary data is too large to fit into computer's memory. The Naive Bayes algorithm requires a table of all strings or bytes to compute its probabilities.

To correct this problem we divided the problem into smaller pieces that would fit in memory and trained a Naive Bayes algorithm over each of the sub problems. We split the data evenly into several sets. For each set we trained a Naive Bayes classifier. Final prediction for a binary is the product of the predictions of the n classifiers. In our experiments we will use 8 classifiers (n = 8).More formally, the Multi-Naive Bayes promotes a vote of confidence between all of the underlying Naive Bayes classifiers. Each Naive Bayes classifier gives a probability of a class C given a set of bytes F which the Multi-Naive Bayes uses to generate a probability for class C given F over all the classifiers.

IV. RESULTS

The proposed classifier result is compared with other classifiers for determining its efficiency. The robust classifiers namely Bayes Net, Naive Bayes, Naive Bayes Multinomal and Naive Bayes Multinomal Updateable are also tested with the same dataset of spam emails and their results obtained are summarized in the form of tables for comparison with the proposed classifier.

The following table shows the comparison of classification efficiency of the various classifiers. It shows the no of correctly and incorrectly classified instances by various classifiers out of the total 4601 no of instances. These values are obtained by applying all the classifiers one by one on the same dataset of spam emails in Weka.

Table 5.1: Comparison of efficiency of various classifiers

Classifier	Total Instances	Incorrectly Classified	Correctly classified
Bayes Net	4601	440	4161
Naïve Bayes	4601	942	3659
Naïve Bayes Multinomal	4601	955	3646
Naïve Bayes Multinomal Updateable	4601	916	3685
Multi Naïve Bayes	4601	942	3659

The comparison of the parameters of performance like TP Rate, FP Rate, Precision, Recall, F-Measure and the time taken to build the model is shown in the tables below. These parameters determine the detailed accuracy of the classifiers and our proposed classifier is compared with the existing classifiers in terms of these parameters to conclude its performance.

Table 5.2: Comparison of accuracy of the classifiers

Classifier	Time taken	TP Rate	FP Rate
Bayes Net	0.23	0.904	0.117
Naïve Bayes	0.56	0.795	0.148
Naïve Bayes Multinomal	0.03	0.792	0.232
Naïve Bayes Multinomal Updateable	0.06	0.801	0.196
Multi Naïve Bayes	0.16	0.795	0.148

Table 5.3: Performance Comparison of various Classifiers

Classifier	Precision	Recall	F-Measure
Bayes Net	0.905	0.904	0.904
Naïve Bayes	0.845	0.795	0.797
Naïve Bayes Multinomal	0.791	0.792	0.792
Naïve Bayes Multinomal Updateable	0.808	0.801	0.803
Multi Naïve Bayes	0.845	0.795	0.797

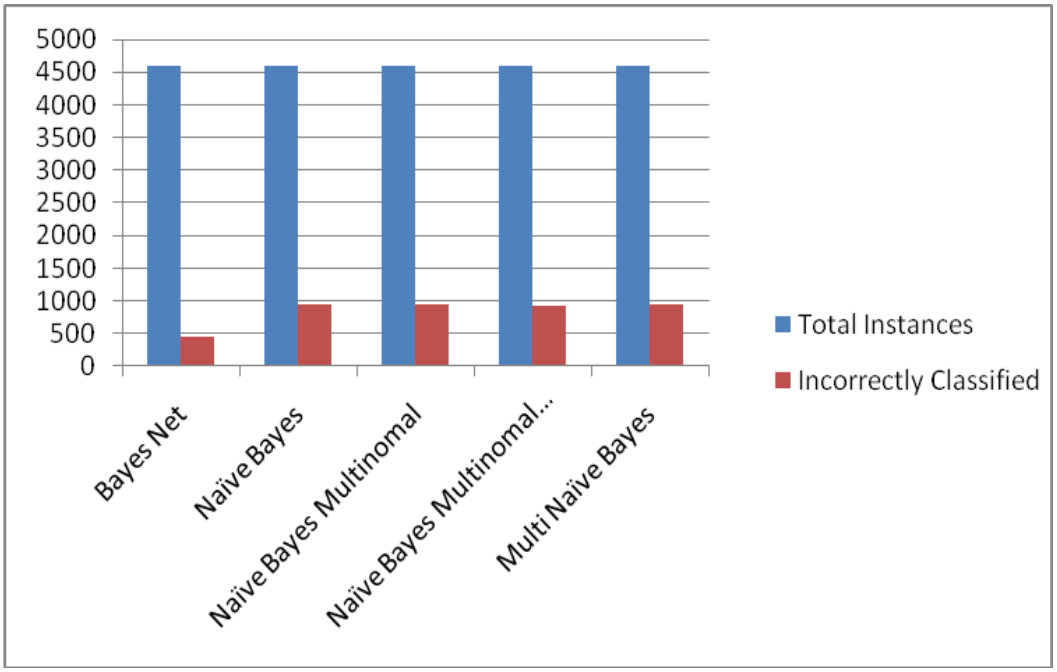


Fig 1:No of correctly and incorrectly classified instances by various classifiers

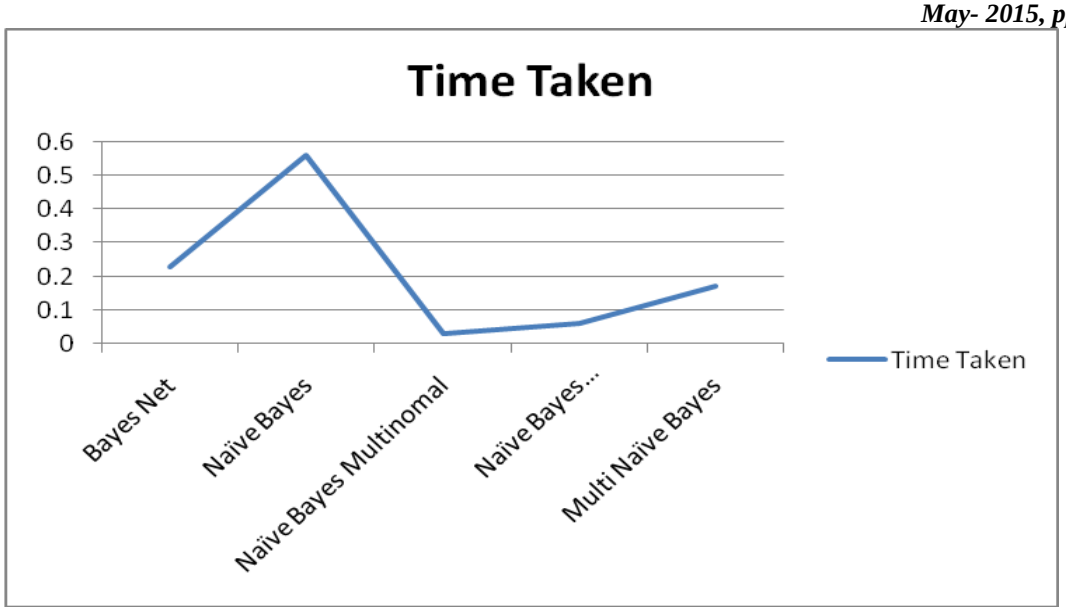


Fig 2: Time taken by various classifiers

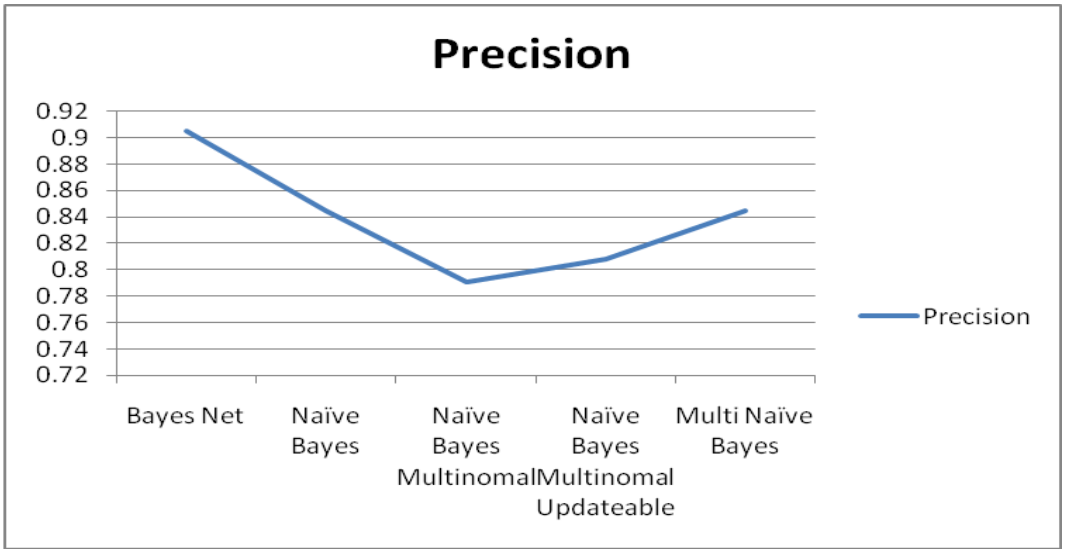


Fig 3: Precision shown by variuos classifiers

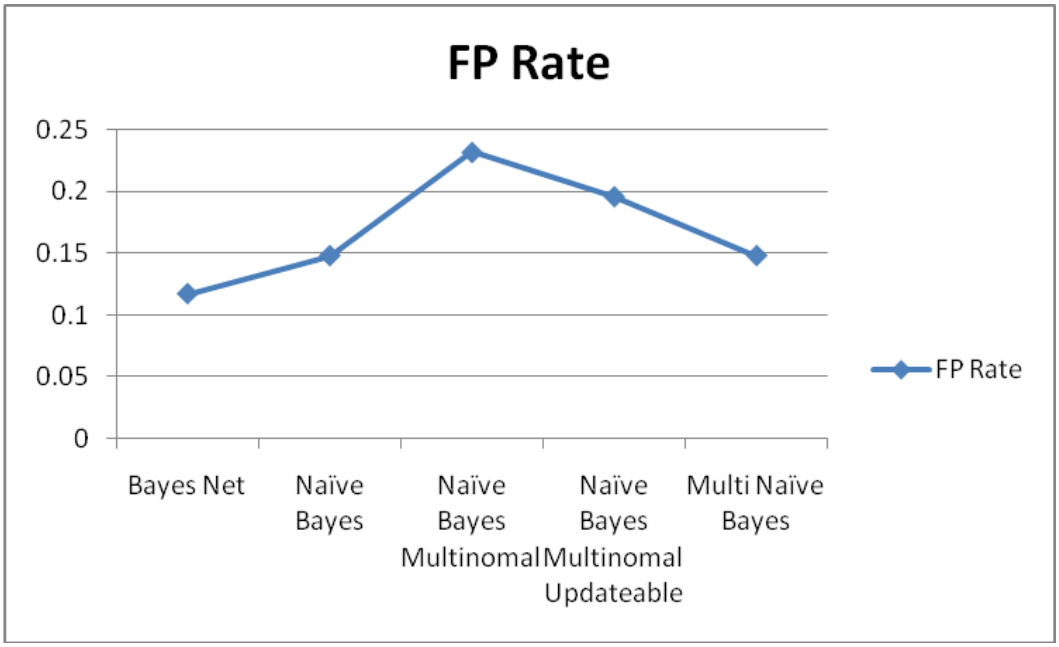


Fig 4: False Positive Rate of classifiers

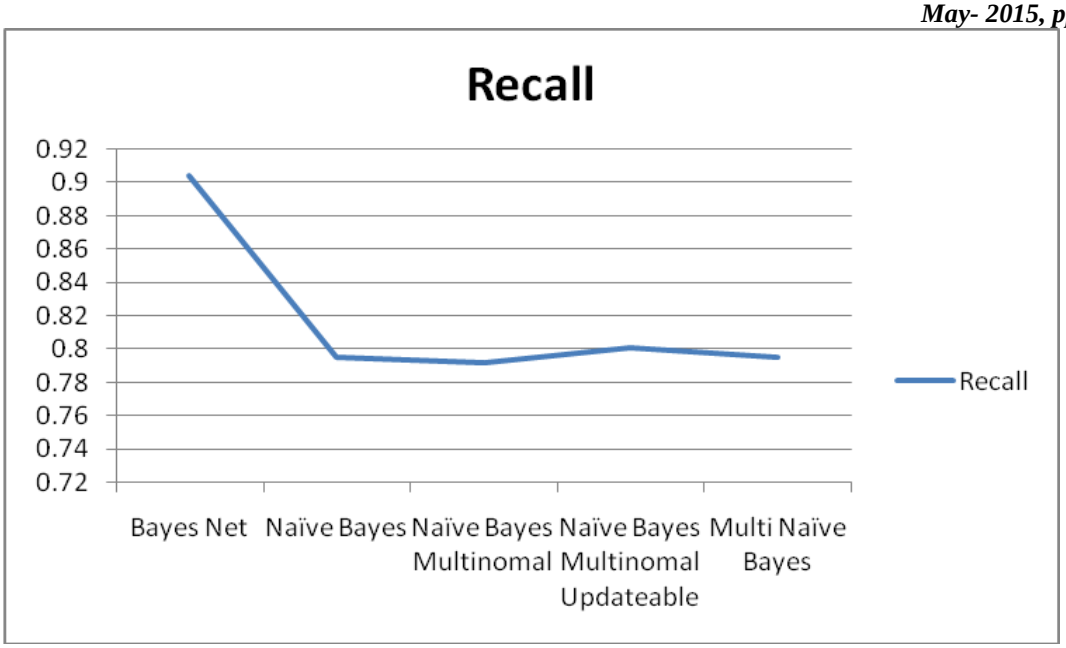


Fig 5: Recall given by classifiers

V. CONCLUSION

A performance comparison of Multi Naïve Bayes Classifier with other Classifiers has been done. The various metrics used for comparison have been shown in the form of line graphs. In terms of correctly and incorrectly classified instances, Naïve Bayes and Multi Naïve Bayes gives better performance but Multi Naïve Bayes Classifier takes less time to classify so it is more efficient than Naïve Bayes Classifier. However, the Naïve Bayes Multinomal Updateable Classifier gives the best Classification.

REFERENCES

[1] Thamarai Subramaniam, Hamid A. Jalab and Alaa Y. Taqa, "Overview of textual anti-spam filtering techniques", International Journal of the Physical Sciences Vol. 5(12), pp. 1869-1882, 4 October, 2010.

[2] Thiago S. Guzella and Walmir M. Caminhas, "A review of machine learning approaches to Spam filtering", Expert Systems with Applications 36 (2009) 10206–10222.

[3] Ion Androutsopoulos, John Koutsias, Konstantinos V. Chandrinos, and Constantine D. Spyropoulos. " An experimental comparison of naive Bayesian and keyword-based anti-spam filtering with personal e-mail messages". In Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval, Aug 2000.

[4] Matthew G. Schultz, EleazarEskin, ErezZadok and Salvatore J. Stolfo "Data Mining Methods for Detection of New Malicious Executables".

[5] Mamoun Alazab, Sitalakshmi Venkatraman, Paul Watters, and Moutaz Alazab, "Zero-day Malware Detection based on Supervised Learning Algorithms of API call Signatures", Conferences in Research and Practice in Information Technology (CRPIT), Vol. 121, 2011.

[6] B.V.R.R.Nagarjuna, V. Sujatha. "An Innovative Approach for Detecting Targeted malicious E-mail", International Journal of Application or Innovation in Engineering & Management (IJAIEM), Volume 2, Issue 7, July 2013.

[7] Anshul Goyal and Rajni Mehta, "Performance Comparison of Naïve Bayes and J48 Classification Algorithms", International Journal of Applied Engineering Research, ISSN 0973-4562 Vol.7 No.11 (2012).

[8] Shubair Abdulla, Sureswaran Ramadass, Altyeb Altaher and Amer Al-Nassiri, "Employing Machine Learning Algorithms to Detect Unknown Scanning and Email Worms", the International Arab Journal of Information Technology, Vol. 11, No. 2, March 2014.

[9] Mehran Sahami, Susan Dumais, David Heckerman and Eric Horvitz, "A Bayesian Approach to Filtering Junk E-Mail", AAAI Technical Report WS-98-05, 1998.

[10] Milan Jain, "Malicious Detection Using Multiple Classification Algorithms & Their Comparison Using Different Clustering Techniques", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 4, April 2013.

[11] J. Han and M. Kamber, "Data Mining: Concepts and Techniques", Morgan Kaufmann, San Francisco, 2001.

[12] Rossen Dimov, "WEKA: Practical Machine Learning Tools and Techniques in Java", 2006/07.

[13] Tretyakov K., "Machine Learning Techniques in Spam Filtering", Data Mining Problem oriented Seminar, MTAT.03.177, pp. 60-79, 2004.

[14] J. Zico Kolter and Marcus A. Maloof, "Learning to Detect and Classify Malicious Executables in the Wild", Journal of Machine Learning Research 7 (2006) 2721-2744.

[15] Ashokkumar Vijaysinh Solanki, "Data Mining Techniques Using WEKA classification for Sickle Cell Disease", International Journal of Computer Science and Information Technologies (IJCSIT), Vol. 5 (4) , 2014, 5857-5860.

[16] Uffe B. Kjærulff and Anders L. Madsen, "Probabilistic Networks —An Introduction to Bayesian Networks and Influence Diagrams", Department of Computer Science, Aalborg University, 10 May 2005.

[17] Zhijun Zhan, LiWu Chang and Stan Matwin, "Privacy-Preserving Naive Bayesian Classification".

[18] Dharmesh Kumar Babubhai Patel and Sahjanand Harshadbhai Bhatt, "Implementnig Data Mining for Detection of Malware from Code", COMPUSOFT, An international journal of advanced computer technology, 3 (4), April-2014 (Volume-III, Issue-IV).

[19] Irena Koprinska , Josiah Poon, James Clark and Jason Chan, "Learning to classify e-mail", Information Sciences 177 (2007) 2167–2187.

[20] V S Kumar and Ravi kumar, "An Efficient Model of Detection and Filtering Technique over Malicious and Spam E-Mails", International Journal of Engineering Trends and Technology (IJETT) – Volume 5 number 1 - Nov 2013.

