A Modified Naïve Bayesian-based Spam Filter using Support Vector Machine

Md. Sabir Hossain
Department of Computer Science and Engineering
Chittagong University of Engineering and Technology (CUET), Chittagong-4349, Bangladesh
sabir.cse@cuet.ac.bd

Md. Zubair
Department of Computer Science and Engineering
Chittagong University of Engineering and Technology (CUET), Chittagong-4349, Bangladesh
zubairhossain773@gmail.com

Mohammad Obaidur Rahman
Department of Computer Science and Engineering
Chittagong University of Engineering and Technology (CUET), Chittagong-4349, Bangladesh
obaidur_91@gmail.com

Muhammad Kamrul Hossain Patwary
Institute of Information and Communication Technology
Chittagong University of Engineering and Technology (CUET), Chittagong-4349, Bangladesh
muhammadkamrulhossain@gmail.com

Md. Golam Sarwar Rajib
Department of Computer Science and Engineering
Chittagong University of Engineering and Technology (CUET), Chittagong-4349, Bangladesh
gsr.rajib@gmail.com

Abstract*— The ever-growing problem which is threatening the current mailing system is spam. Spam is nothing but an unsolicited bulk e-mail frequently sent in a financial nature which generates the need for creating an anti-spam filter. Amongst many spam filtering techniques, the most advanced method "Naïve Bayesian filtering" using the Support Vector Machine (SVM) have been implemented. Spammers are very careful about the filtering techniques. For that very reason, dynamic filtering is needed and the proposed method meets the demand. The algorithm splits the received email into tokens and uses Bayes' theorem of probability to calculate the probability of spam for each token to determine the total spam probability of the mail. Implementation of SVM instead of corpora is one of the added features of the algorithm. The most challenging feature was to take the words as well as whole sentences as input in the SVM as tokens and feature vectors. The inclusion of sentences in the dataset training has increased the accuracy of detecting spam and ham. Natural Language Tool Kit (NLTK) has been used as a useful language processing tool to tokenize the sentences and also to understand the meaning of the same types of sentences to some extent. As a test mail is being compared by word to word and also sentence to sentence from the training datasets to determine if the mail is spam or not, it will improve the performance of the filter. With some simple modifications, the filter can be used in both server and client end. The efficiency increases gradually with the increased number of email it processes.

Keywords- Spam, Bayesian Approach, SVM, Tokenization, Spamicity, Dataset.

I. INTRODUCTION

The internet and E-mail become major communication media in every aspect of our life, they have revolutionized the way of doing business and socializing. The Internet has made information gathering and publishing easy and provided the convenience of online shopping and financial management. However, these new technologies have created new issues and problems at the same time. Among the issues, the so-called 'spam' is the major puzzle.

The word "Spam" as applied to email means Unsolicited Bulk Email (UBE). It can be said in another way that the recipient has not any granted way to detect the sender of mail containing a large collection of messages. Some people may include the e-mail produced by mass-mailing viruses or Trojan horses as spam.

The manifestation of spam has become a fundamental problem for internet user in general. The recipients of bulk emails get puzzled while differentiating the important mail from spam. It also reduces the efficiency of a company and somewhat more. Apart from direct losses, indirect losses are made as well as consumption of storage, internet bandwidth and so on. Some folks even predict that the arrival of spam will bring about the end of e-mail entirely.

Dealing with the issues, numbers of spam detection and filtering methods have been developed. These techniques are effective in some degrees but not limitation free. Signature-based filtering is accurate and fast but applicable for particular emails. On the contrary, the Learning-based system is slower. Above techniques are based on pattern matching rules. Most often it requires tuning for each user's messages which is time-consuming and difficult. Furthermore, the spammers are very aware of the spam detection techniques and try to overlook the techniques variously. An automated learning system that would able to separate the legitimate message is highly suggested.

This paper examines and implements the cutting edge anti-spam technique 'Naïve Bayesian filtering' using Support Vector Machine (SVM) which is learning based dynamic solution. It can adapt to the new techniques of spammers by enriching the trained dataset.

II. RELATED WORK

The naïve Bayesian spam filter is a member of the Learning-based spam filter family. At the very beginning,

Pantel and Lin developed a spam filtering technique based on text classification [1]. Afterward, many scientists and researchers worked on Bayesian spam filtering [2]. Paul Graham proposed a modified algorithm with a large dataset in 2003 and be able to extend the accuracy rate and minimize the probability of false positive significantly [3]. The experiment of Pantel and Lin showed their success rate is 92% with 1.16% false positive. On the other hand, Paul's algorithm was able to detect 99.5% spam with 0.03% false positive. Paul claimed that the main reason behind it was working with a large number of dataset.

Biju Issac and Wendy J. Jap developed a system using Porter's stemmer algorithm along with Bayes theorem. They used porter's stemmer algorithm to split every keyword to its stem which improved the efficiency. Because it reduced the keywords to be searched and developed the margin of accuracy. [4]

Lin Li & Chi Li also implemented a system based on the Naïve Bayesian spam filter. They use the TF-IDF method for more accuracy [5].

Spam detection with K-means clustering algorithm can also be used to detect spams. It also provides a promising result [6].

Naïve Bayes is a strong tool for spam filtering. There are several forms of Naïve Bayes which gives different accuracy [7].

In some cases, unsupervised learning provides more accurate result and it is best suited [8]. Another most recent technique is deep learning. Researchers are also trying to deploy a technique based on deep learning in spam detection [9].

Although all of the solutions discussed and implemented above are effective to some degree, each of the processes has limitations too. None of the solutions can provide 100% accuracy, and there's always a percentage of false positive. No single method is enough to detect and filter spam because spammers are very much aware of the modern spam detection techniques and they are smart enough to circumvent a spam filter. For this reason, a "cocktail" approach is used to detect spam successfully. Combination of the strength of each system may provide a robust solution in case of spam detection.

III. PRELIMINARIES

A. Various Spam Filtering Techniques

Spammers are very conscious of spam filtering techniques. They try their best to overlook existing spam filtering techniques. For meeting up the demand, over the years many filtering techniques have been introduced for filtering spam emails. Signature-based, rule-based and learning-based filtering are some of them.

- In the signature-based filtering, the hash code of known spam is stored in the database. If an incoming email's signature is matched with the stored signature, then the incoming mail is said to be spam. The process is extremely accurate, but it can't detect a new technique of spam.
- The rule-based filter uses a list of keywords; these keywords are the spam keywords. If an incoming mail contains these keywords, then it is considered as spam. The keywords are stored in the database. These filters have high false positive and negative rates. It's because of specific words.
- In learning-based filtering, the filter is trained by itself from time to time according to its input. This approach is a dynamic one. As the spammers are being more conscious of the conventional spam filtering techniques, learning based filtering is the best solution for it.

Through the experiment Bayesian spam filter is used which is one of the learning based spam filters. This system tokenizes the received mail and compares them with the tokens stored in the database. It requires an initial training period. Throughout the training period, the Bayesian filter supervises the incoming and outgoing email and started to create a dataset for the mail flow. System administrators may also tweak the database to better train the Bayesian engine.

B. Bayes' Theorem in Spam Filtering

Computing conditional probabilities, Bayes' formula is one of the famous formulae. Generalized Bayes' formula:

$$P(B_1|A) = \frac{P(B_1 \cap A)}{P(A)} = \frac{P(B_1)P(A|B_1)}{\sum_{i=1}^{n} P(B_i)P(A|B_i)} \quad (1)$$

Where,

- $P(B_1|A)$ is a conditional probability that indicates occurring $P(B_1)$ given that P(A) is true.
- $P(A|B_1)$is also a conditional probability that indicates the probability of occurring P(A)given that $P(B_1)$ is true.

B. Naïve Bayes' Formula in Spam Filtering

Bayesian email filters are based on Bayes' theorem (1). The theorem is most often used in the field of spam detection.

Naïve Bayes' theorem is given below which is used to calculate the spam probability of the system.

$$P_r(S|W) = \frac{P_r(W|S).P_r(S)}{P_r(W|S).P_r(S) + P_r(W|H).P_r(H)} \quad (2)$$

Where,

- Pr(S|W) denotes the probability of spam message when the spam detecting word is in it.
- Pr(S) denotes the overall probability of spam message.
- Pr(W|S) denotes the probability of spam word in the spam messages.
- Pr(H) denotes the overall probability of spam message
- Pr(W|H) denotes the probability of spam word in the ham messages.

Initially when the mail received it is assumed that the mail is neither spam nor ham i.e., the probability of the mail to be spam or ham is equal, so Pr(S) = Pr(H) = 0.5.

a. Calculating individual probability

The overall probability of an email being spam, regarding all of the tokens or a set of tokens of the mail can be calculated using the following formula**:**

$$P = \frac{P_1 P_2 \ldots\ldots\ldots P_N}{P_1 P_2 \ldots\ldots\ldots P_N + (1-P_1)(1-P_2)\ldots(1-P_N)} \quad (3)$$

Here,

- P is the probability spam of a fishy message.
- P_1 is equal to the probability of $P_r(S|W_1)$.
- P_N is the probability $P_r(S|W_n)$

The output of the formula p is usually compared to a defined point for deciding a spam or ham message. If the probability is greater than the specified value, the message is considered spam. Otherwise, it is not a spam message.

b. Overall View of Naïve Bayes' Formula:

Naïve Bayes is easy, but effective classifier.it is totally based on the Bayes theorem. This feature defines feasibility related to a class based on the probability of each attribute value. Abstractly, NB is a conditional probability model: given a problem instance to be classified to any class {Cj }1<j<n, represented by a vector of features (independent variables) X € {x1…..xn} it can be formally shown as:

$$C_{MAP} = argmax_{c \epsilon C} P(x_1, \ldots\ldots\ldots, x_n | c) P(c) \quad (4)$$

$$C_{NB} = argmax_{c \epsilon C} P(c_j) \prod_{x \in X} P(x|c) \quad (5)$$

Reducing the number of parameters required for modeling the important advantages of the NB algorithm in technical concerns. Exemplary, due to the independence assumption, we need 2n parameters to model P (X|Y) instead of the original 2(2n – 1). These quality procedures guarantee simplicity and speed.

C. Support Vector Machine (SVM)

SVM is associated with statistical learning theory. In 1992(Boser, Guyon & Vapnik), it was first introduced properly [10]. SVM algorithm became popular after getting success in handwriting digit recognition.

Training data consists of input and output functionality is known as supervised learning in machine learning. The support vector machine (SVM) is one of the supervised machine learning algorithms [11]. It is mostly used for solving two-group classification problems. In this system, it is used for differentiating between spam and ham.

An SVM training algorithm builds a model that enrolls new examples to one of the two categories which makes the non-probabilistic binary linear classifier. If the new example which will be classified falls into one side of the hyperplane, it is defined in the category according to the class of that side.

Beside linear classification, different types of method are introduced like kernel trick, etc. for non-linear classification in SVM.

D. Natural Language Tool Kit (NLTK)

As the first challenge to create a spam filter was to build an SVM that will take input an email and extract the words as well as sentences of the message body. So that there will be word to word and sentence to sentence comparison simultaneously, to calculate spam probability. For this approach, the Natural Language Tool Kit (NLTK) is used in python programming.

The NLTK is introduced for helping the system with Natural Language Processing (NLP) technique. The NLKT helps the system by splitting sentences from paragraphs, splitting up words, recognizing the part of speech of those words and so on. However, the primary purpose of using NLTK is the tokenization of sentences and words.

IV. PROPOSED METHODOLOGY

A. Algorithm for Building the filtering system

The total process is given below sequentially.

1) Retrieving spam and ham email messages

A lot of sample ham and spam messages are needed to train the SVM. Almost 16,000 ham messages and 3,000 spam messages have been collected from the machine learning repository of Enron dataset [12]. The system retrieves all the messages one by one and takes it into the machine.

2) Tokenization

Tokenization is done into two steps. Each email is split into two parts, words and sentences are separately taken. Words are directly listed into a database which is called data dictionary. Using NLTK, the sentences are converted into a vector and then stored to the data dictionary.

3) Feature Extraction

Feature extraction measures the frequency of spam and ham emails. Generally, each of the words is labeled 0 if it came from the ham email and labeled 1 if it is from spam email. In this method, the vectors generated from sentences are also labeled 0 and 1 in accordance with spam and ham messages. In this way, SVM calculates the frequency of the tokens.

4) Shuffle the data dictionary

After labeling, all the tokens are stored in the data dictionary in the format of the 2D array in a random approach. The spam and ham email messages are stored randomly in the data dictionary.

5) Generate. sav file

After completion of all the steps above, SVM generates a .sav file which contains the training datasets of the data we have collected to train the system.

6) *SVM. Seeding database*

The data dictionary has been seeded with spam and ham messages as many as possible. The more data added, the more accuracy of a mail being ham or spam will be achieved.

7) *Test method (Input message)*

After training the SVM with enough datasets of spam and ham emails, the testing session has been taken place. A sample email body has been taken to test.

8) *Tokenization of test message*

The whole message is split into two portions. One portion takes the word and keeps them in an array and another portion converts every sentence into a feature vector and keeps it in an array.

9) *Frequency calculation*

After tokenizing the test message, the frequency of each token and vectors in the token list is calculated.

10) *Retrieve Spam and Ham frequency*

In this step, the spam and ham frequency is retrieved from the training datasets of our SVM model.

11) *Retrieve Spam and Ham count*

Spam and ham count are done from the datasets.

12) *Calculate Spam probability*

Calculate the spam probability of each token that is found in the database using the following formula:

Spam probability, P_s

$$P_S = \frac{\text{spam frequency in database}}{\text{number of spam mail in database}} \quad (6)$$

If the probability is greater than 1, set it to 1.

13) *Calculate Ham Probability:*

Calculate the spam probability of each token that is found in the database using the following formula:

Ham probability P_H,

$$P_H = \frac{Ham\ frequency\ in\ database}{number\ of\ Ham\ mail\ in\ database} \quad (7)$$

If the probability is greater than 1, set it to 1.

14) *Calculate Spamicity*

Calculate the spamicity of each token that is found in the database output of "(6)" & "(7)" using the following formula:

$$\text{Spamicity} = \frac{P_S}{P_S + P_H} \quad (8)$$

15) *Total spam probability:*

Calculate the total spam probability using the output of "(8)". It can be found as:

$$\text{Total probability} = (S_1 * S_2 * .... * S_n) + ((1-S_1) * \ldots * (1-S_n)) \quad (9)$$

Where,

- S_1 is the spamicity of 1st token.
- S_2 is the spamicity of 2nd token.
- S_n is the spamicity of nth token.

16) *Decision*

We examined the proposed system with different threshold values. Which shows the results in Table I.

TABLE I EXPERIMENTAL ACCURACY WITH DIFFERENT THRESHOLD VALUES

Value	.2	.3	.4	.5	.6	.7	.8
Accuracy(%)	39.11	60	80	90	85	70.21	43.55

When the threshold value is less than 0.5, the system considers some spam emails as ham. On the other hand, some ham email is considered as spam with the threshold value greater than 0.5. So, we considered 0.5 as an optimal threshold value.

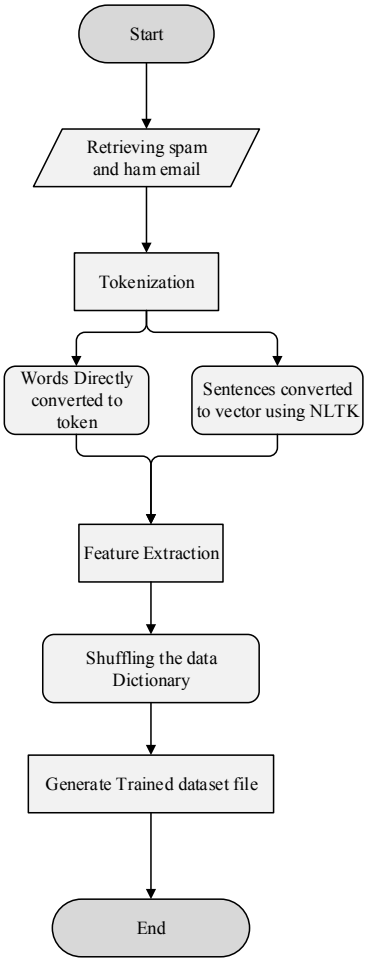

Fig.1 Representation for building Support Vector Machine (SVM)

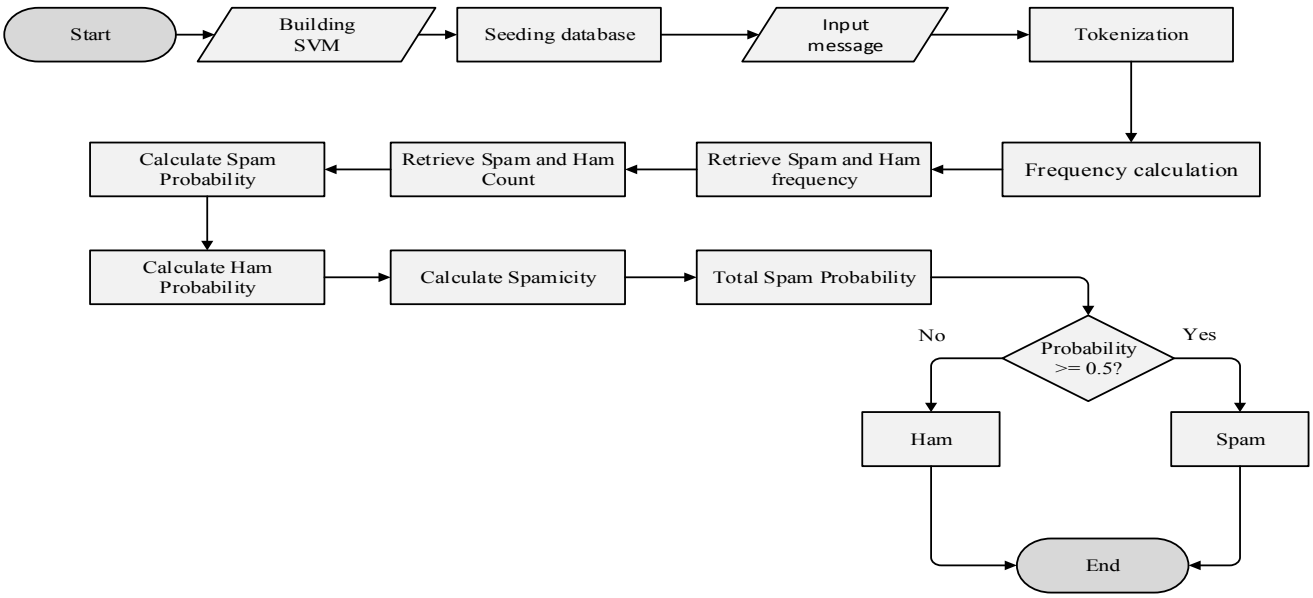

Fig. 2 Representation of Naïve Bayesian Spam Filter using SVM

B. Flowchart of the Naïve Bayesian spam filter using SVM

1) Building SVM

First of all, we have built an SVM for training datasets.

The flowchart of building SVM is in fig.1.

2) The complete procedure of Spam Filter

After successfully building the SVM, we prepared the spam filter for testing new mail. The whole process is shown in fig. 2.

V. Implementation of BayesiaN Filter

A. Sample spam message

The result of the algorithm after receiving a spam email and processing it with the system is shown using a well-known "Enron datasets" [12] spam message. The message is given below:

From: BUMA SARO WIWA <bsarowiwa@incamail.com>
To: imukaviva@incamail.com
Subject: Dobmeos with hgh my energy level has gone up! stukum introducing doctor formulated hgh.

-increased muscle strength. - loss in body fat. - increased bone density. - lower blood pressure. - quickens wound healing. - reduces cellulite. - improved vision. - wrinkle disappearance. -increased skin thickness texture.	- increased energy levels. - improved sleep and emotional stability. - improved memory and mental alertness. - increased sexual potency. - resistance to common illness. - strengthened heart muscle. - controlled cholesterol. - controlled mood swings. -new hair growth and color restore.

It is referred to in medical science as the master hormone. it is very plentiful when we are young, but near the age of twenty - one our bodies begin to produce less of it. by the time we are forty nearly everyone is deficient in hgh , and at eighty our production has normally diminished at least 90 - 95 % advantages of hgh :

Read more at this website www.hmboe.com.

Subscribe today.

B. Processing steps

1) Tokenization: After receiving the mail, the Bayesian spam filter first split the message into tokens. For the concern message tokens are shown in "table II". At the same time, sentences are separated too by counting the comma (,), full stop (.), exclamatory sign (!), interrogative sign (?), etc. punctuation marks. It is shown in "table III".

2) *Frequency Calculation*: Using the stop words list, we eliminate the unnecessary words that are not effective in spam detection. Then the frequency of single tokens is calculated. After that, with the help of NLTK, the sentences are converted into feature vectors, and then their frequency is also calculated.

TABLE II TOKEN LIST (WORDS)

Dobmeos	with	hgh	My	energy
level	has	Gone	up	stukm
human	growth	hormone	also	Called
hgh	Is	Referred	To	in
medical	Master	hormone	It	Is
very	plentiful	When	we	are
Young	but	near	The	Age
Of	Twenty	one	our	Bodies
Begin	to	Produce	less	of
It	by	the	Time	we
are	Forty	nearly	everyone	Is
deficient	of	In	hgh	and
At	eighty	our	Production	Has
normally	Diminished	at	least	90-95%
advantages	of	Hgh	Increased	muscle
Strength	loss	In	Body	fat
increased	Bone	density	lower	Blood
Pressure	quickens	Wound	healing	reduces
Cellulite	improved	vision	Wrinkle	disappearance
increased	Skin	thickness	texture	Increased
energy	levels			

3) Retrieve Spam and Ham Frequency: Retrieve Spam and Ham Frequency: From the SVM, the data dictionary has retrieved where the spam and ham frequency is stored.

TABLE III TOKEN LIST (SENTENCES)

Sentence
Dobmeos with hgh my energy level has gone up
Introducing doctor – formulated hgh
human growth hormone - also called hgh
It is referred to in medical science as the master hormone
it is very plentiful when we are young
but near the age of twenty - one our bodies begin to produce less of it
by the time we are forty nearly everyone is deficient in hgh
and at eighty our production has normally diminished at least 90 - 95 %. advantages of hgh
increased muscle strength
loss in body fat
increased bone density
lower blood pressure
quickens wound healing
reduces cellulite
improved vision
wrinkle disappearance
increased skin thickness texture
increased energy levels
improved sleep and emotional stability
improved memory and mental alertness
increased sexual potency
resistance to common illness
strengthened heart muscle
controlled cholesterol
controlled mood swings
improved memory and mental alertness
Read more at this website www
Subscribe today

4) Retrieve spam and ham count: After getting the spam and ham frequency, we now go through the spam and ham count. Here numbers of spam and ham messages from the data dictionary are found.

5) Calculate spam and ham probability: In this step, the spam and ham probability of each token, and vectors generated from sentences are calculated using the formula discussed above. If either spam or ham probability is found greater than 1, and then it is set to 1.

6) Spamicity Calculation: The spamicity of each token is calculated by using spam and ham probability using the formula discussed above.

7) Total spam probability: In our test mail, there are 122 tokens for words, and 27 vectors generated from sentences. It is really a lengthy calculation, so it would take some time in the compiler to show the result.

Using Naïve Bayes formula, the total probability of the message to be a spam can be calculated as:

Term 1: (0.99911) (0.654) (0.756) (0.889) (0.776) (0.445) (0.667) (0.334) (0.554) (0.233) (0.556)……. (0.912) (0.814) (0.823)

Term 2: (1-0.99911) (1-0.654) (1-0.756) (1-0.889) (1-0.776) (1-0.445) (1-0.667) (1-0.334) (10.554) (1-0.233) (1-0.556) ……. (1-0.912) (1-0.814) (1-0.823)

$$Total\ spam\ probability = \frac{Term\ 1}{Term\ 1 + term\ 2} = 0.74 \quad (10)$$

We can see that the final probability of the message to be spam is 74%. In our methodology, we have already stated that our probability measurement threshold value is 0.5. If the probability is greater or equal to 0.5, the message will be considered spam, and if less, we will consider it ham.

VI. ANALYSIS OF THE PROPOSED SPAM FILTER

A. Experimental Result

We have tested the performance of the spam filter using the corpus of the SpamAssassin project of Apache Foundation [13]. For some sample input email, the output of the Naïve Bayesian spam filter is given in "table IV" & "table V".

TABLE IV EXPERIMENTAL RESULT OF NÄIVE BAYESIAN FILTER

Steps	Total no. of Email (input)	No. of Spam (input)	No. of Ham (input)	No. of Spam (output)	No. of Ham (output)
initially	120	45	75	-	-
1	30	10	20	9	18
2	60	30	30	28	27
3	90	38	52	34	45
4	120	45	75	39	68

Experimental result of false positive, false negative and accuracy of "table IV" is shown in "table V".

TABLE V EXPERIMENTAL RESULT OF NÄIVE BAYESIAN FILTER

Steps	False Positive (Output)	False Negative (Output)	Accuracy (%)
initially	-	-	-
1	1	2	90
2	2	3	91.67
3	4	7	88
4	6	7	87

We considered a total of 120 emails among them 45 spam and 75 ham/nonspam. The input emails are provided in four steps. At step-1 and step-2, step-3 and step-4 total 30, 60, 90, 120 emails are used respectively.

During the experiment for step-1 and step-2, we used one spam and one ham alternatively. This is done so that there is a balance between the spam and ham database. For the next two steps, no such order was maintained.

B. Comparative Analysis

Naïve Bayesian spam filters are adequate to detect spam than other types of filters such as K-means clustering [6], deep learning [9]. But supervised learning is better than unsupervised learning in this case, and deep learning is a complex process to classify spam and ham. So, we can say that our filtering technique using a Naïve Bayesian spam filter is legitimate.

Another reason for the great performance of Naïve Bayesian spam filters is that most of them that are available and personally trained by individual users to better filter their emails using own set of the token. But other filters don't support this feature such as signature-based and list-

based filters are used in the server side and rule-based filters support client-side filtering but in small scale. Comparative analysis is shown in "table VI".

The learning system is slower than other approaches because it works with lots of tokens and database. It takes time to retrieve, insert and update the database each time. But other systems do a little business with the database than learning-based systems. Learning systems are reliable because it works with the content of the email rather than the sender or sending network of the message. As a result, the spammer cannot circumvent the filter by sending it from a new address or network. This increases the reliability of learning systems while rule-base and list-based are less reliable.

TABLE VI COMPARATIVE ANALYSIS OF FILTERING SYSTEM

Filter Type	Fast	Reliable	Dynamic	Application Area
Signature-based	Yes	Yes	No	Server
Rule-based	Yes	Sometimes	No	Both server and client
Our proposed filter	No	Yes	Yes	Both server and client
List based	Yes	Sometimes	No	Server

It is also worth mentioning that the Naïve Bayesian spam filter is not our new invention. We added some more techniques to enhance reliability, accuracy, and dynamicity. Fig. 3 depicts the comparison between the conventional Naïve Bayesian spam filter [14] and our proposed model. We calculate average accuracy, recall, precision, F1-measure.

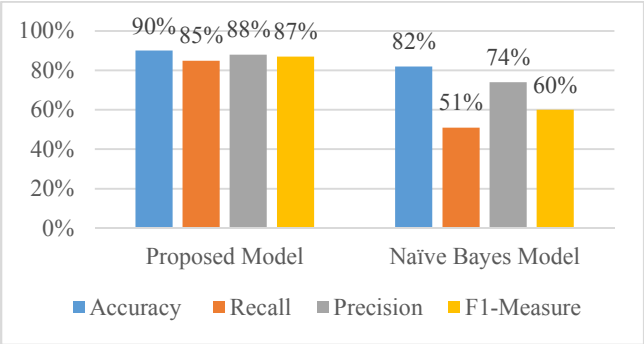

Fig.3 Comparison with the Naïve Bayes Model

The comparison clearly says that our proposed model is best suited in all the purposes.

VII. Conclusion

No spam filter can detect spam with 100% accuracy. So, the proposed spam filter is not above all weakness but we can certainly improve the performance of the filter by incorporating various features into the filter. The accuracy of the filter is shown based on some real-time data during the experiment phase of the filter, and we have gained the accuracy of spam detecting at a standard level of 88%. The accuracy can be increased by tokenizing the incoming email more efficiently and suitably building the spam and ham seeding database with proper tokens. The accuracy also depends on the order at which emails are received during the training period of the database. Lastly, to defeat spam, it is more important to analyze the content of the mail than other techniques, because the content is everything for the spammer to deliver to the people. Spammers are getting smarter day by day. But with proper analysis of content and a large amount of data training, it is possible to develop an efficient spam filter. Hence, there is a lot of scope for future improvement. Modern email system also contains HTML data, images, and other languages besides English and so on. None of these things are considered in the system. One can work with these fields for detecting spam email more effectively.

References

[1] Pantel, Patrick, and Dekang Lin. "Spamcop: A spam classification & organization program," in *Proceedings of AAAI-98 Workshop on Learning for Text Categorization*, 1998.

[2] Androutsopoulos, Ion, et al. "An evaluation of naive bayesian anti-spam filtering." arXiv preprint cs/0006013 (2000).

[3] P. Graham, "Better Bayesian Filtering", *Paulgraham.com*, 2019. [Online]. Available: http://www.paulgraham.com/better.html. [Accessed: 14- Apr- 2019].

[4] Issac, Biju, and Wendy J. Jap. "Implementing spam detection using Bayesian and Porter Stemmer keyword stripping approaches." *TENCON 2009-2009 IEEE Region 10 Conference*. IEEE, 2009.

[5] Li, Lin, and Chi Li. "Research and improvement of a spam filter based on Naive Bayes." *7th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), Vol. 2*. IEEE, 2015.

[6] N. Omer Fadl Elssied and O. Ibrahim, "K-Means Clustering Scheme for Enhanced Spam Detection", *Research Journal of Applied Sciences, Engineering and Technology*, pp. 1940-1952, 2014.

[7] V. Metsis, I. Androutsopoulos and G. Paliouras, "Spam Filtering with Naive Bayes – Which Naive Bayes?", *The Third Conference on Email and Anti-Spam*, 2016.

[8] E. Tan, L. Guo, S. Chen, X. Zhang and Y. Zhao, "UNIK: unsupervised social network spam detection", *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management - CIKM '13*, 2013. Available: 10.1145/2505515.2505581

[9] Y. Ren and D. Ji, "Neural networks for deceptive opinion spam detection: An empirical study", *Information Sciences*, vol. 385-386, pp. 213-224, 2017. Available: 10.1016/j.ins.2017.01.015

[10] Boser, Bernhard E., Isabelle M. Guyon, and Vladimir N. Vapnik. "A training algorithm for optimal margin classifiers." In *Proceedings of the fifth annual workshop on Computational learning theory*., ACM, 1992.

[11] Cristianini, Nello, and John Shawe-Taylor. Cambridge University Press, 2000.

[12]Enron-Spam-dataset [Online] Available: http://nlp.cs.aueb.gr/software_and_datasets/Enron-Spam/index.html [Accessed: 14- Apr- 2019].

[13]"Apache SpamAssassin: Welcome", *Spamassassin.apache.org*, 2019. [Online]. Available: https://spamassassin.apache.org/. [Accessed: 14- Apr- 2019].

[14] N. Fitriah Rusland, N. Wahid, S. Kasim and H. Hafit, "Analysis of Naive Bayes Algorithm for Email Spam Filtering across Multiple Datasets", in *International Research and Innovation Summit (IRIS2017)*, 2017.