مجلة
كلية التــراث الجامعة

مجلة علمية محكمة
متعددة التخصصات نصف سنوية
العدد الحادي والأربعون
30 نيسان 2025
ISSN 2074-5621

رقم الايداع في دار الكتب والوثائق 719 لسنة 2011

مجلة كلية التراث الجامعة معترف بها من قبل وزارة التعليم العالي والبحث العلمي بكتابها المرقم (ب 3059/4) والمؤرخ في (4/7 /2014)

Machine Learning Approach for SMS Spam Detection Based on TF-BNB

Mustafa Tareq
College of Computer Science, University of Technology, Baghdad, Iraq

ABSTRACT

Different machine learning (ML) algorithms have been explored to address the challenge of SMS spam classification. Many studies highlight the role of preprocessing SMS datasets using TF-IDF and applying the Naive Bayes classifier, particularly the multivariate BernoulliNB (BNB) model. Together, TF-IDF and BNB provide effective tools for message classification. Spam messages, often disguised as legitimate texts, remain a persistent problem as they consume valuable time during sorting and deletion, complicate the separation of personal and professional communication, and increase the risk of accidentally overlooking important information. To overcome this issue, our approach integrates ML, deep learning (DL), and natural language processing (NLP) methods, with NLTK as the primary toolkit for preprocessing. The proposed TF-BNB system achieves a classification accuracy of 99.75%, underscoring its effectiveness in spam detection.

1. Introduction

Spam is "unrestricted mass email," containing "info designed to be distributed to innumerable people, despite their desires. In the approach for mass mailing, spam images containing compelling or propellant content are distributed. Nevertheless, given the many media spam techniques employed, including email spam and SMS spam, such spam may be easily identifiable. Spammers overwhelm the SMS personnel and send a large volume of unrestricted SMS to the end users (Siddique, Zeeshan Bin, et al.,2021).

From a commercial standpoint, SMS users must put effort into eliminating received spam SMS, which undoubtedly causes the benefit decrease and may result in difficulties for associations. How to correctly and effectively identify SMS spam with high precision becomes a huge report from this point forward. Information mining will be employed in this evaluation to control AI by making use of various classifiers for testing and preparation, as well as channels for information pretreatment and highlight selection. It intends to evaluate various metrics, or more precisely, identify the best mix model. There are now many assessment studies that have been carried out using information digging techniques, such as information digging using planned methods. Overall, a lot of effort is focused on a single classifier. In any event, spam practice is adapting its tactics to avoid the spam zone (Yadav, K., et al.,2012). Therefore, in this investigation, we will focus on the whole framework for controlling SMS spam by using an information mining approach. Through testing, it will be possible to determine, for instance, if a cross-assortment model produces results with higher precision compared to a single classifier used to detect email spam. Today's generation utilizes phones for a variety of purposes, including preserving personal information in the form of papers, notes, and media, conducting banking transactions, purchasing, and more. Hacking phones is one of the biggest appeals to those with immoral motives since there is a vast range of information saved on devices, much of which is personal and critical. Because SMS is utilized by individuals of all ages and socioeconomic backgrounds, and they are unaware of the repercussions, it is a simple approach to target people.

When a phone gets hacked, hackers gain access to the phone and all of its data, and the users do not know at all. Critical data is lost as a result, which might be used for criminal activities. For the victims, it can be distressing, leading to emotional desolation and monetary losses. Not only are messages written in English, but also in other languages, and even terminology and abbreviations from these other languages may appear in those written in English. Since SMS does not contain a header like emails do, it is difficult to recognize spam SMS, as noted by Yadav et al. (2012).

The rapid growth in mobile phone usage has fueled the expansion of the multi-billion-dollar SMS industry. At the same time, the decreasing cost of messaging services has led to a significant rise in spam delivered to mobile devices. Unlike email, there are no well-established SMS databases, and the brevity of text messages—often written in informal language with limited details—makes it more challenging for filtering algorithms to classify them effectively. The increasing volume of unsolicited messages, many of which are deceptive, poses a serious problem in daily communication. Users waste valuable time sorting and deleting spam, which creates difficulties in distinguishing important personal or professional messages and increases the risk of overlooking or accidentally deleting critical information. However, to solve the problem of spam messages, we will classify messages using ML, DL and NLP, NLTK.

The contribution of this paper is as follows:

- To pre-process the random SMS dataset using Term Frequency-Inverse Document Frequency (TF-IDF).
- To apply the Naive Bayes classifier for multivariate BernoulliNB models (BNB). It is used as a tool in applications and programs to classify messages.
- To implement TF-BNB techniques based on a random SMS dataset.

2. Related Works

This chapter presents an overview of related works on SMS spam.

According to Shirani (2013), the project goal is to apply several ML algorithms to the issue of classifying spam SMS, compare their results to gather knowledge and investigate other issues, and develop a program based on one of these algorithms that can accurately filter spam SMS, (5,572) text messages from the UCI device are stored in a database. In 2012, he gathered learning repository data and employed SVM as a learning algorithm, producing an overall accuracy of 97.64%. The next-best worksheet in their collection uses naive Bayes and has an accuracy improvement of 97.50%.

The classifier has a total error reduction of more than 50% when compared to the output of the prior research. Learning curves and misclassified data were examined as contributing elements that helped this increase in results. Other significant aspects that were included were the length of the letters in the number of characters, with the inclusion of specific length restrictions, and learning curves.

Describe how Naive Bayes performs better for classifying unwanted SMS than the random forest method and the logistic regression approach (Sethi et al., 2017). Text may easily be classified as spam or not based on naive Bayes algorithms high accuracy of 98.445% using the information gain matrix. In Figure 1, we can see a comparative study of different ML algorithms.

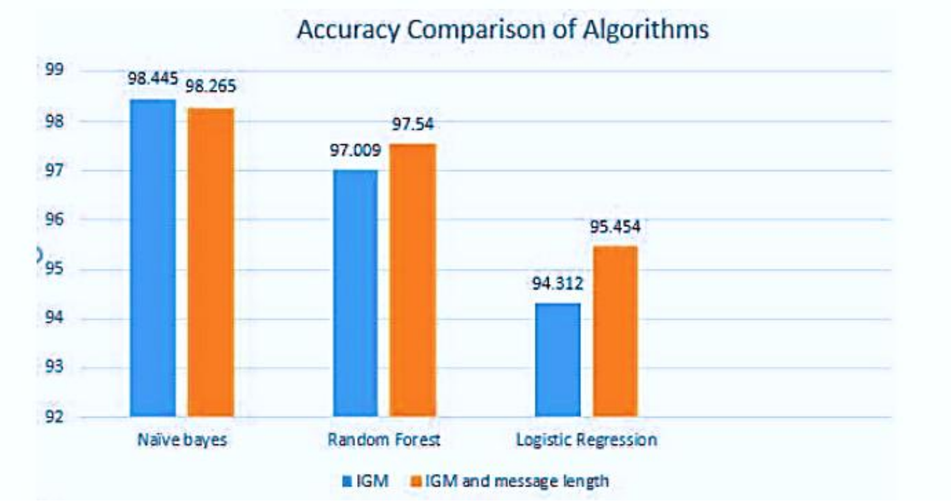


Figure 1. Comparative study of different ML algorithms

They examined and researched the relative merits of several machine learning methods for spam detection messages transmitted on mobile devices (Sethi et al., 2017). We were able to obtain data from a publicly available source and supplied two datasets for testing and confirmation, detection precision. The algorithms employed to rate these communications prioritized spam. The findings unequivocally demonstrate how different machine learning algorithms perform differently when given different features for spam categorization. Keywords: machine learning, spam detection, and Natural Language Processing (NLP).

On several data sets gathered from prior research, Gupta, Mehul, et al. (2018) analyze various classification approaches and grade them based on their accuracy, accuracy, recall, and CAP curve. She provided a comparison between deep learning approaches and conventional machine learning techniques. Where comparisons are made with eight various workbooks. According to the findings of his classifier evaluation, the convolutional neural network classifier achieves the greatest accuracy of 99.19% and 98.25%, as well as AR values of 0.9926 and 0.9994 for the two datasets.

Although being widely used in image-related data classification, CNN outperforms classical classifiers and achieves the highest accuracy among them for text-related data. CNN's achievement has significantly broadened the research side of its application over text-related classification challenges such as sentiment prediction and review categorization.SVM and NB, two conventional classifiers, perform admirably and, for both datasets, closely resemble CNN. This research may be applied in the real world for the identification of spam SMS because significant results have been gained from it.

Although the work may be used for both incoming and outgoing SMS, (Bosaeed et al.,2020) detection and classification system they provided a tool to identify spam from outgoing SMS messages. They specifically created a system that uses numerous ML-based classifiers that use Naive Bayes NB, Vector Machine SVM support, and NB multinomial NBM classification methods, as well as five pre-processing and feature extraction techniques. The method is designed to be employed in stratified cloud, fog, or edges and is evaluated using 15 datasets generated by four frequently used public SMS datasets. In response to user settings, such as rating accuracy, True Negatives TN, and computing resource requirements, the system recognizes spam messages and provides suggestions for spam filters and classifiers to utilize. They demonstrate that the PF5 filter and SVM achieve the greatest overall performance in SMS classification.

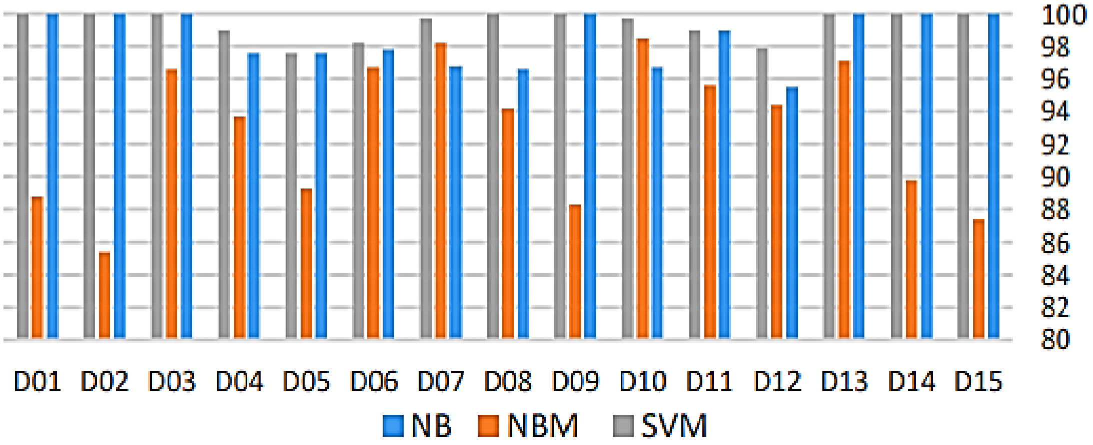


Figure 2 TNR using PF5

They devised this technique as a result of the enormous problem presented by popular social media networks, including Twitter, Facebook, and Quora are frequently targeted by spam accounts. These accounts are intended to prey on unsuspecting real users by fooling them into clicking on malicious links or producing repeating messages with bots. This might have a significant influence on how customers engage with these websites. Effective techniques for identifying certain types of spam have been the subject of extensive study. We can successfully address this issue by conducting sentiment analysis on these posts.

The major goal of this proposed effort is to create a system that can assess a tweet's sentiment and identify whether it is spam or junk. Different classifiers, including decision trees, logistic regression, polynomial naive Bayes, support vector machines, random forests, and Bernoulli naive Bayes for spam detection, are used to categorize the characteristics retrieved after pre-treating tweets. For sentiment analysis, DL methods include Stochastic Gradient Descent, Support Vector Machine, Logistic Regression, Random Forest, Naive Bayes, and Simple Recurrent Neural Network RNN Model, Long Term Memory Model LSTM, Bidirectional Long Term Memory BiLSTM, and Network Model 1D Convolutional Neural CNN.

Each classifier's performance is evaluated. According to the classification findings, It is feasible to consistently determine whether a given tweet is spam or not by utilizing the attributes extracted from it, as well as to develop a learning model that correlates tweets to certain moods.

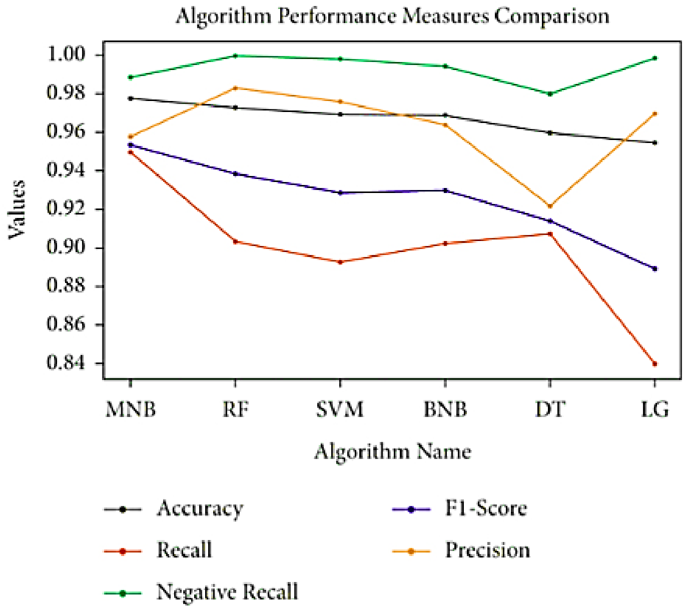


Figure 3 Algorithm performance measures comparison

The validation accuracy for classifying Twitter spam using the naïve polynomial Bayes classifier was 97.78%, whereas the validation accuracy for the deep learning model, the LSTM, was 98.74%. The classification technique demonstrates how information retrieved from tweets may be used to determine whether or not a tweet is spam. The results of the ratings showed that attributes collected from tweets may be used to properly judge sentiment in tweets. The LSTM deep learning model surpassed the SVM classifier in the analysis of Twitter sentiment, with a classification accuracy of 73.81%.

3. Methodology

This section describes the datasets utilized in the study and outlines the proposed methodology.

3.1 Datasets

Two datasets were used to evaluate the proposed method these datasets are :

A. Dataset Context

The SMS Spam Collection is a collection of SMS-tagged messages gathered for the SMS Spam study. It comprises one collection of 5,572 SMS messages in English that have been classified as ham (legal) or spam. (Harper, F. el al.,2015).

B. Dataset Content

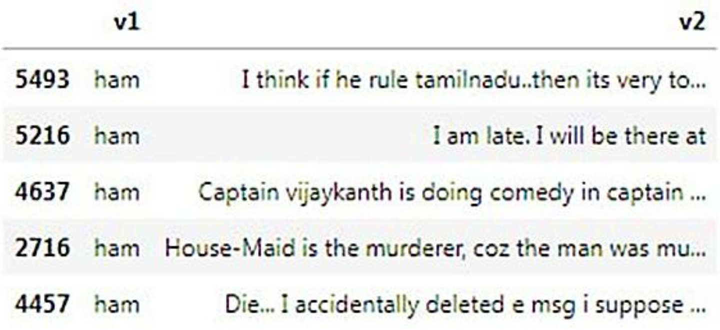
Each file has one message per line. Each line is made up of two columns: v1, which carries the label (ham or spam) and v2, which contains the raw text. This corpus was compiled from free or low-cost research sources on the Internet: (Berns, Fabian, et al.,2019)

- (425) SMS spam messages were meticulously gathered from the Grumble text website. This is a UK forum where mobile phone users make public claims about SMS spam messages, the majority of which do not contain the spam message itself. Detecting text or spam messages in claims is a complex and time-consuming task that necessitates methodically scanning hundreds of web pages.
- A subset of 3,375 SMS messages chosen at random from the NUS SMS Corpus (NSC), a corpus of about 10,000 legitimate messages obtained for research at the National University of Singapore's Department of Computer Science. The messages are largely from Singaporeans, mostly from University students. These remarks were compiled from participants who were informed that their contributions would be made public.
- Caroline Tag's doctoral study provided 450 SMS ham messages.
- Finally, the SMS Spam Corpus v.0.1 Big has been included. There are 1,002 ham SMS messages and 322 spam texts in it.

3.2 Proposed Method

The data consists of v1 and v2. V1 classifies messages, if they are random, write spam, and if not, write ham. V2 is for messages to be classified. NLP from NLTK was used to analyze the data, as shown in Table 1.

Table 1 Data Sample



	v1	v2
5493	ham	I think if he rule tamilnadu..then its very to...
5216	ham	I am late. I will be there at
4637	ham	Captain vijaykanth is doing comedy in captain ...
2716	ham	House-Maid is the murderer, coz the man was mu...
4457	ham	Die... I accidentally deleted e msg i suppose ...

This process takes place according to the following steps:

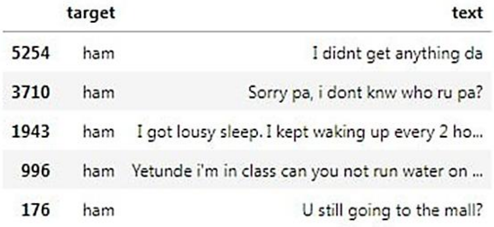
Step 1: Data Cleaning

Data cleaning entails correcting errors within your data set. Empty cells, data in the wrong format, incorrect data, and duplicates are all examples of bad data.

1. Change name of columns

v1,v2 has been changed and renamed as v1 represents target and v2 represents text. As shown in Table 2.

Table 2 Sample Data After Changing Names



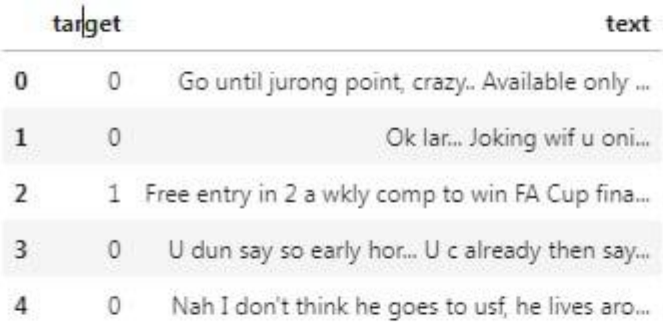
	target	text
5254	ham	I didnt get anything da
3710	ham	Sorry pa, i dont knw who ru pa?
1943	ham	I got lousy sleep. I kept waking up every 2 ho...
996	ham	Yetunde i'm in class can you not run water on ...
176	ham	U still going to the mall?

2. Digital Number

Then make a label encoder for the target because the algorithm does not read the symbols, but only the numbers (0 and 1) as shown in Table 3.

After the digital number process comes the duplicate process of deleting duplicate messages, amounting to 403 messages, so the total number of messages after this process becomes 5169 messages.

Table 3 Digital Number In Column Target



	target	text
0	0	Go until jurong point, crazy.. Available only ...
1	0	Ok lar... Joking wif u oni...
2	1	Free entry in 2 a wkly comp to win FA Cup fina...
3	0	U dun say so early hor... U c already then say...
4	0	Nah I don't think he goes to usf, he lives aro...

Denotes number 1 to spam and it represents 653 messages. Denotes number 0 to ham and it represents 4516 messages. The percentage of spam and ham is shown in Figure 4.

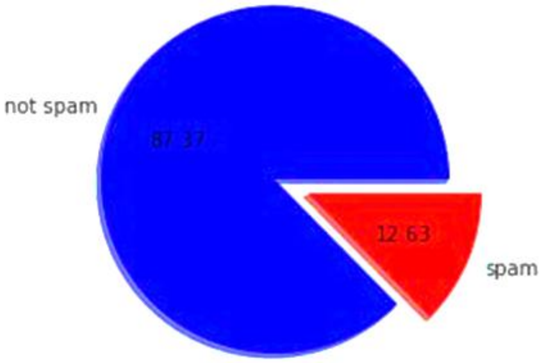
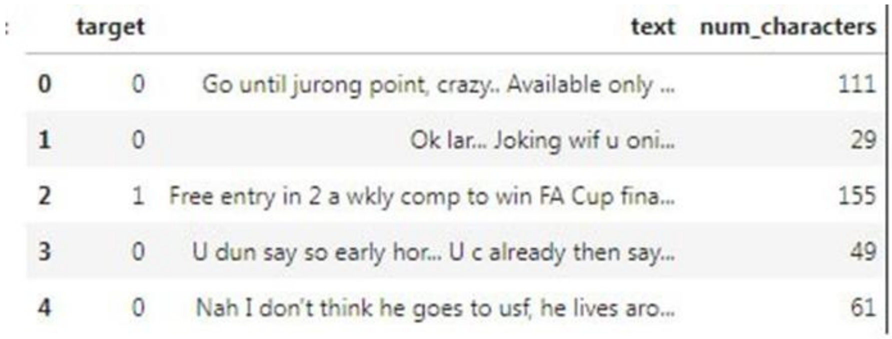


Figure 4 The percentage of spam and ham

3. Add a new column

Add a column named number characters that contains the number of characters in the message, the length of the message with the empty spaces, as shown in Table 4

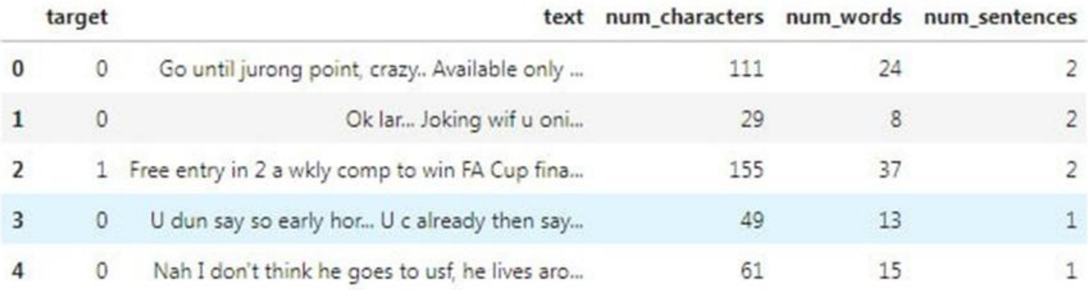
Table 4 Add a column named Number_Characters



	target	text	num_characters
0	0	Go until jurong point, crazy.. Available only ...	111
1	0	Ok lar... Joking wif u oni...	29
2	1	Free entry in 2 a wkly comp to win FA Cup fina...	155
3	0	U dun say so early hor... U c already then say...	49
4	0	Nah I don't think he goes to usf, he lives aro...	61

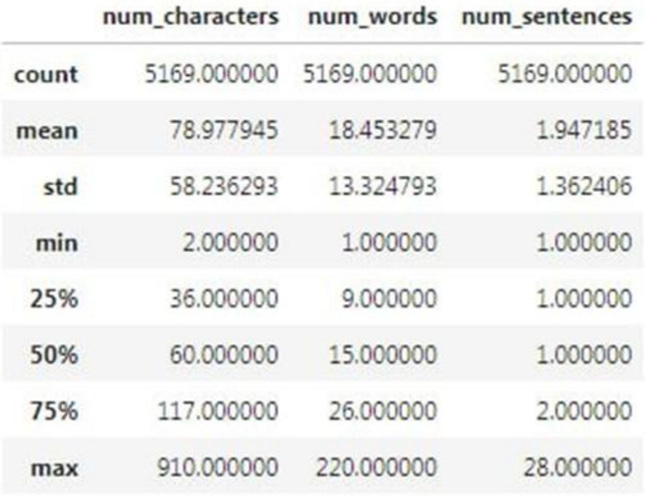
Then add two columns in terms of the quantity of words and phrases in the communications, as shown in Table 5.

Table 5 Add Two Columns for words and sentences



	target	text	num_characters	num_words	num_sentences
0	0	Go until jurong point, crazy.. Available only ...	111	24	2
1	0	Ok lar... Joking wif u oni...	29	8	2
2	1	Free entry in 2 a wkly comp to win FA Cup fina...	155	37	2
3	0	U dun say so early hor... U c already then say...	49	13	1
4	0	Nah I don't think he goes to usf, he lives aro...	61	15	1

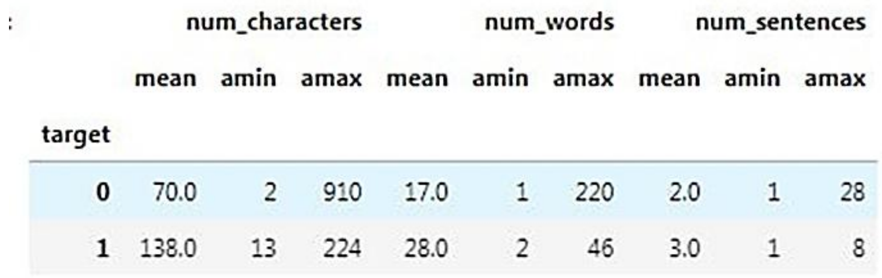
Table 6 Shows the ratios of the three added columns



	num_characters	num_words	num_sentences
count	5169.000000	5169.000000	5169.000000
mean	78.977945	18.453279	1.947185
std	58.236293	13.324793	1.362406
min	2.000000	1.000000	1.000000
25%	36.000000	9.000000	1.000000
50%	60.000000	15.000000	1.000000
75%	117.000000	26.000000	2.000000
max	910.000000	220.000000	28.000000

Table 6 explains, value can be seen as count in num_characters , num_word and num_sentences same equal 5169.000000 . Mean in num_characters equal 78.977945, in num_word equal 18.453279 and in num_sentences equal 1.947185. Std in num_characters equal 58.236293, in num_word equal 13.324793 and in num_sentences equal 1.372406. Max shown in num_characters equal 910.000000. Min shown in num_word equal and num_sentences same equal 1.000000. Table 7 shows the relationship between the new columns added with the target depending on mean, max and min.

Table 7 Shows the relationship between the new columns added with target



	num_characters			num_words			num_sentences		
	mean	amin	amax	mean	amin	amax	mean	amin	amax
target									
0	70.0	2	910	17.0	1	220	2.0	1	28
1	138.0	13	224	28.0	2	46	3.0	1	8

Step 2: EDA

Exploratory Data Analysis (EDA) is an important and essential part of the data science and machine learning workflow. It allows us to become familiar with our data by exploring it, from multiple angles, through statistics, data visualizations, and data summaries. This helps discover patterns in the data, spot outliers, and gain a solid understanding of the data we are working with.

In this Python EDA tutorial, we will cover the use of an excellent Python library called Pandas Profiling. This library helps us carry fast and automatic EDA on our dataset with minimal lines of code.

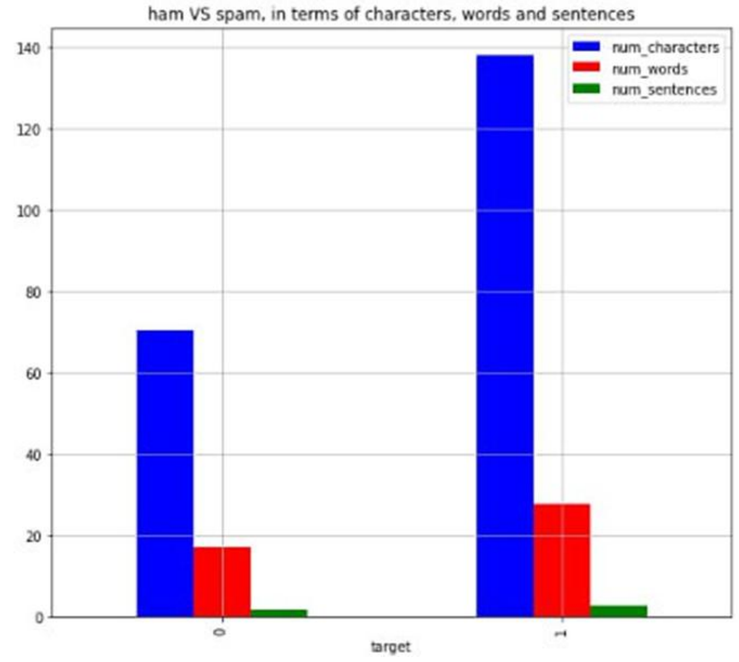


Figure 5 Ham VS spam in terms of characters, word and sentences

Figure 5 shows that num_characters in SMS its proportion is much greater compared with num_characters in ham as well num_words in SMS its proportion is greater compared with num_words in ham finally, num_ sentences in SMS its proportion is greater compared with the proportion_ sentences in ham.

Step 3: Text Preprocessing

It is necessary to evaluate data before using it for analysis or prediction. Text preprocessing is used to prepare text data for model generation. It is the very first stage of every NLP project. Preprocessing phases include:

a. Lower case

One of the most typical text preparation python processes is to convert the text to the same case, ideally lower case. Nevertheless, this step is not required every time you work on an NLP problem because lower casing might result in information loss in some cases. Such as example, if we are working with a person's emotions in a project, then phrases printed in upper case might indicate dissatisfaction or joy. So this step is to convert the letter letters into lowercase letters.

b. Tokenization

In this stage, the text is broken into smaller sections. We may not use sentence tagging or word tokenization based on our issue description.

Sent = "I'm a dog and it's great! You're cool and Sandy's book is big. Don't tell her, you'll regret it! 'Hey', she'll say"!

Word_tokenize(sent) ['I', "'m", 'a', 'dog', 'and', 'it', "'s", 'great', '!', 'You', "'re ," ' cool', 'and', 'Sandy', "'s", 'book', 'is', 'big', '.', 'Do', "n't", 'tell,' 'her', ',', 'you', "'ll", 'regret', 'it', '!', "'Hey", "'", ',', 'she', "'ll", 'say!']

c. Removing special characters

At this step, the text's punctuation has been removed. Python's string library has a built-in set of punctuation symbols such as: **'!"#$%&'()*+,-./:;?@[\]^_`{|}~'**

d. Removing stop words and punctuation

Stop words are often used terms that are removed from the text since they add nothing to the analysis. These words have little or no meaning. The NLTK library includes a list of stop words in the English language. Here are a few examples:

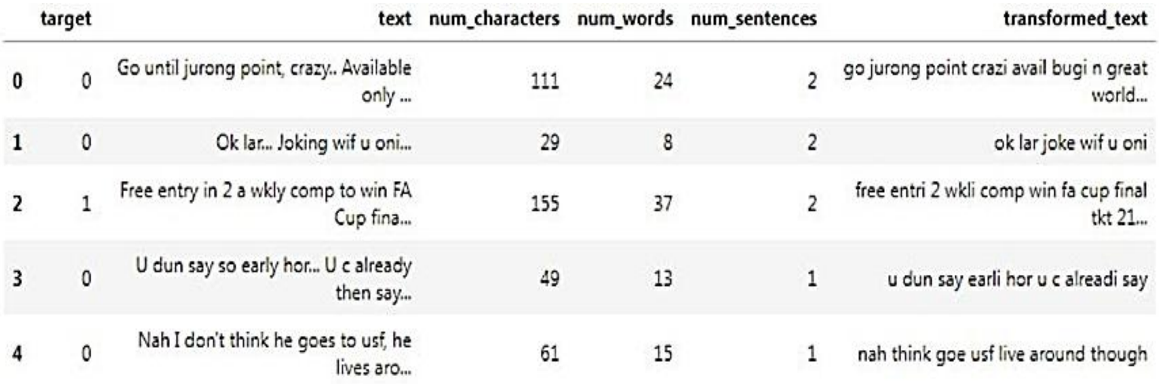
(i, me, my, myself, we, our, ours, ourselves, you, you're, you've, you'll, you'd, your, yours, yourself, yourselves, he, most, other, some, such, no, nor, not, only, own, same, so, then, too, very, s, t, can, will, just, don, don't, should, should've, now, d, ll, m, o, re, ve, y, ain, aren't, could, couldn't, didn't, didn't).

Nevertheless, it isn't required to utilize the given list of stop words since they should indeed be chosen judiciously project. While dealing with client inquiries, for instance, how could be a stop word for one model but crucial for another. For certain difficulties, we may create stop phrase properties.

e. Stemming

This is also referred as the language standardization stage, Words are stemmed or reduced to their root/base form. Terms like programmer, programming, and program will be shortened to program. After the above steps, we will return to save the message. Add column Transformed text as shown in Table 8

Table 8 Column Transformed-text.



	target	text	num_characters	num_words	num_sentences	transformed_text
0	0	Go until jurong point, crazy.. Available only ...	111	24	2	go jurong point crazi avail bugi n great world...
1	0	Ok lar... Joking wif u oni...	29	8	2	ok lar joke wif u oni
2	1	Free entry in 2 a wkly comp to win FA Cup fina...	155	37	2	free entri 2 wkli comp win fa cup final tkt 21...
3	0	U dun say so early hor... U c already then say...	49	13	1	u dun say earli hor u c alreadi say
4	0	Nah I don't think he goes to usf, he lives aro...	61	15	1	nah think goe usf live around though

Figure 6 shown chart of the most used words in SMS spam We note that the word ***cell*** is the most used word, while the word ***min*** is the least used word.

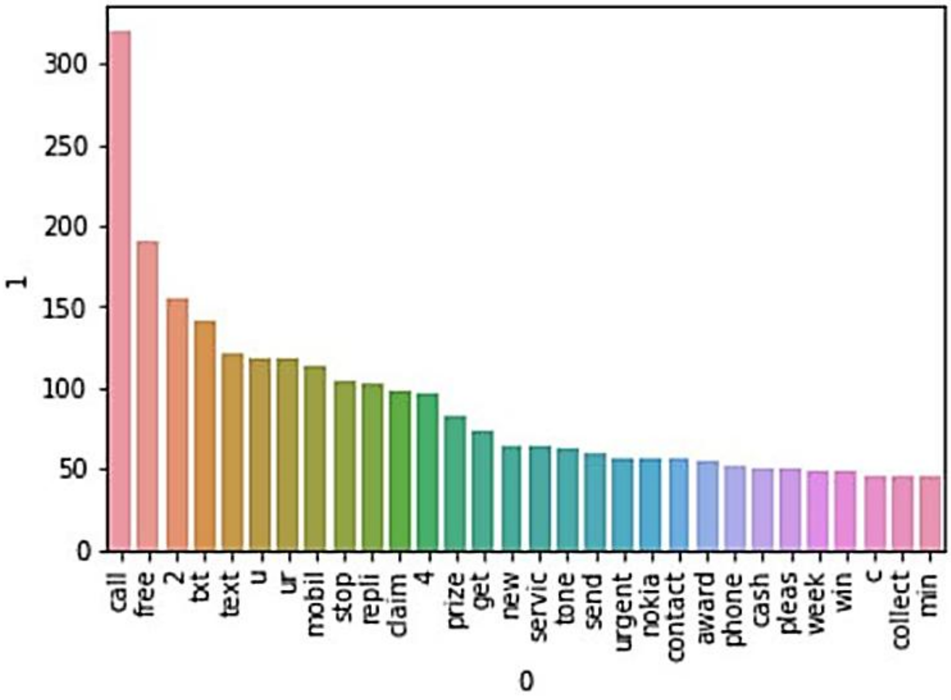


Figure 6 Chart of the most used words in SMS spam

Figure 7 shows a chart of the most used words in ham ham. We note that the character ***u*** is the most used word, while the word ***make*** is the least used word.

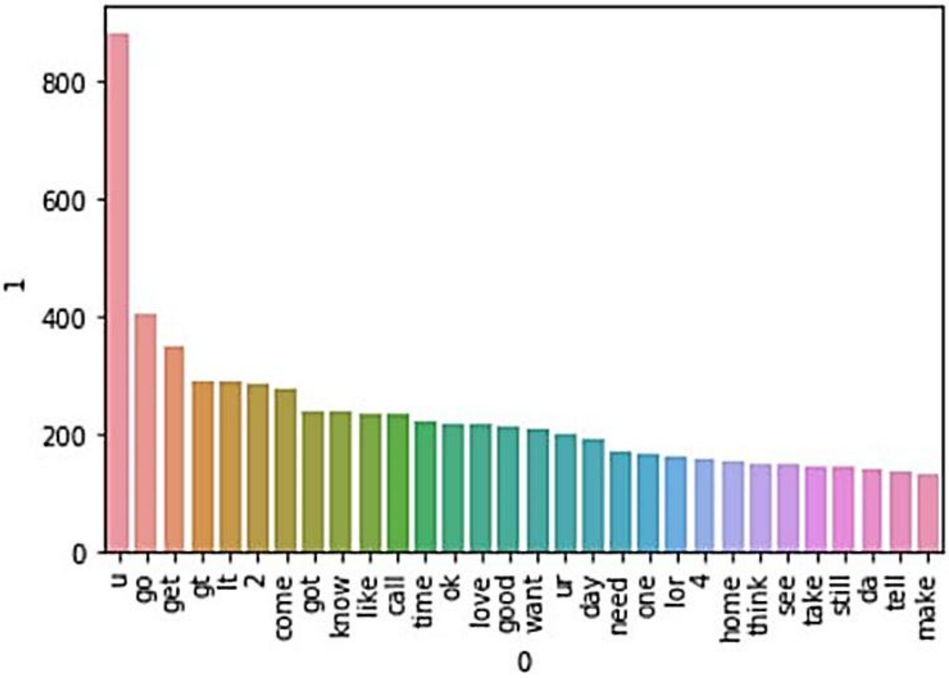


Figure 7 Chart of the most used words in HAM spam

Step 4: Model building

1-TF-IDF model

Used TF-IDF model to build the system. TF-IDF is a form of data retrieval that ranks the relevance of words in a text. It is based on the premise that words that appear more frequently in a document are more important to the material. The combination of Term Frequency and Inverse Document Frequency is TF-IDF. Below is the formula for TF-IDF computation, with equation 3.1 indicating the development of the term TF-IDF.

$$TF\text{-}IDF= [\text{Term Frequency } (TF) * \text{Inverse Document Frequency } (IDF)] \quad (1)$$

Term Frequency: it is an indicator of the quantity of phrases within a piece of writing. It is the ratio of the number of times a word appears in a document to the total number of words within this document.

Inverse Document Frequency: It's a measurement of how much detail the word gives on the topic of the paper. It is also the log of the percentage of the entire quantity of documents to the number of documents forming the term.

We use the log of this ratio because when the corpus develops in size, the IDF values may get too high, causing it to explode; hence, using the log will lessen this impact. Because we cannot divide by zero, we smooth the integer by adding 1 to the denominator. Equation 3.2 depicts the IDF, and equation 3.3 shows the TF-IDF.

$$IDF(t) = [\log(N/(DF + 1))] \qquad (2)$$

$$TF\text{-}IDF\ (t, D) = [TF(t, D) * \log(N/(DF + 1))] \qquad (3)$$

Where is t determine term (word), D determine document (set of words) and N determine count of corpus.

The TF-IDF approach has the advantage of eliminating the shortcomings of the Bag of Words approach. Because it does not give equal weight to all words, important terms that appear only a few times will be given a high weight. Bag of Words just provides a set of vectors reflecting the number of word occurrences in the text (reviews), but the TF-IDF model contains information on both the more significant and less significant occurrences.

2- BernoulliNB models (BNB)

BernoulliNB performs naive Bayes training and classification on data with multivariate Bernoulli distributions; that is, several features may exist, but each one is treated as a binary-valued (Bernoulli, Boolean) variable. As a result, samples must be saved as binary-valued feature vectors; if any other form of data is provided, a BernoulliNB object may binarize it (C.D. Manning, el al.,2008). The decision rule for Bernoulli naive Bayes is based on equation 4.

$$P\ (x_i \mid y)=P(x_i=1|y)x_i+(1-P(x_i=1|y)(1-x_i) \qquad (4)$$

This varies from the rule of multinomial NB in that it expressly penalizes the absence of a feature I that is an indication for class *y*.

Whereas the multinomial version would simply disregard a non-occurring feature. In the context of text identification, a word occurrences vector might be employed to train and test this classifier. BernoulliNB may perform better on some datasets, particularly those with shorter documents. If time allows, it is best to compare both models (A. McCallum and K. Nigam,1998).

3.6 Evaluation

a) <u>Accuracy:</u>

If a measurement is accurate, it means that it is close to the accepted norm for that quantity. For example, if we predict the size of a project to be x and the final project size is equal to or very close to x, it is accurate but not precise. The more accurate a system is regarded to be, the closer its measurements are to the recognized value.

Humans make mistakes all the time, but if you utilize project management software to help you scope your project, you will notice more exact project metrics and a refined process. The Accuracy Equation is depicted in Equation 5.

$$\text{Accuracy}=\frac{TP+TN}{TP+TN+FP+FN} \qquad \textbf{(5)}$$

where TP determines True Positive, FP determines False Positive, TN = True Negative and FN determines False Negative.

b) <u>precision:</u>

An accurate measurement has a value that agrees with other measures of the same thing. Consider an estimation of workload as an example of project scoping. If we estimate the size of several projects and discover that they are all close to or equal to what we predicted, we may begin to comprehend the precision of our forecasts. Above all, though, each project must be as exact as possible.

When it comes to project scoping, you want to get as near to the real workload as feasible. Establishing the scope entails you and your customer developing and documenting a list of precise project objectives. These might include project features, functionality, deliverables, timeframes, and, ultimately, project expenditures. The project scope aids in resource planning and time management. Equation 6 depicts the precision equation.

$$\text{Precision}=\frac{|\{\textbf{relevant docouments}\}\cap\{\textbf{elevant docouments}\}|}{|\{\textbf{elevant docouments }\}|} \qquad \textbf{(6)}$$

Accuracy and accuracy are employed in the context of measurement, such as project size, and are thus both useful when establishing the scope. Accuracy and precision are only similar in that they both pertain to measurement quality, but they are completely distinct markers of measurement.

c) F-Measure:

The F-Measure {Van Rijsbergen, 1979 #438} takes both the precision and the recall into account, for the calculation of the score. While the precision reading derives from the number of accurate results, divided by the number of every arrived result, the recall reading derives from the number of accurate results, divided by the number of results that ought to have arrived. Looked upon as a weighted average of precision and recall, the F-measure attains its greatest value at 1, and its least value at 0.

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

$$Recall = \sqrt{\frac{TP}{TP+TN}} \qquad \textbf{(7)}$$

$$F-Measure = \frac{2*Precision*Recall}{Precision+Recall}$$

4. Results and Discussions

The results of the proposed system showed an accuracy rate of 0.993559 using the BNB algorithm, while the RF algorithm was 0.974855 and the LR algorithm was 0.958414. The rest of the algorithms can be seen as shown in Table 9:

Table 9 Algorithms

Algorithms	Accuracy	precision	F1 Measure
KNN	0.905222	1.000000	0.803282
MNB	0.970986	1.000000	0.890785
BNB	0.993559	0.99187	0.975541
RF	0.974855	0.98276	0.878956

SVC	0.975822	0.97479	0.878862
ETC	0.974855	0.97458	0.872654
LR	0.958414	0.9703	0.908247
xgb	0.971954	0.94309	0.921761
AdaBoost	0.960348	0.9292	0.910547
GBDT	0.947776	0.92	0.897274
BgC	0.957447	0.86719	0.907846
DT	0.930368	0.81731	0.890567

Figure 8 the percentage accuracy, Precision and F1Measure of each algorithm is shown based on table 9, the figure shows that the BNB algorithm demonstrates the highest accuracy compared with other algorithms.

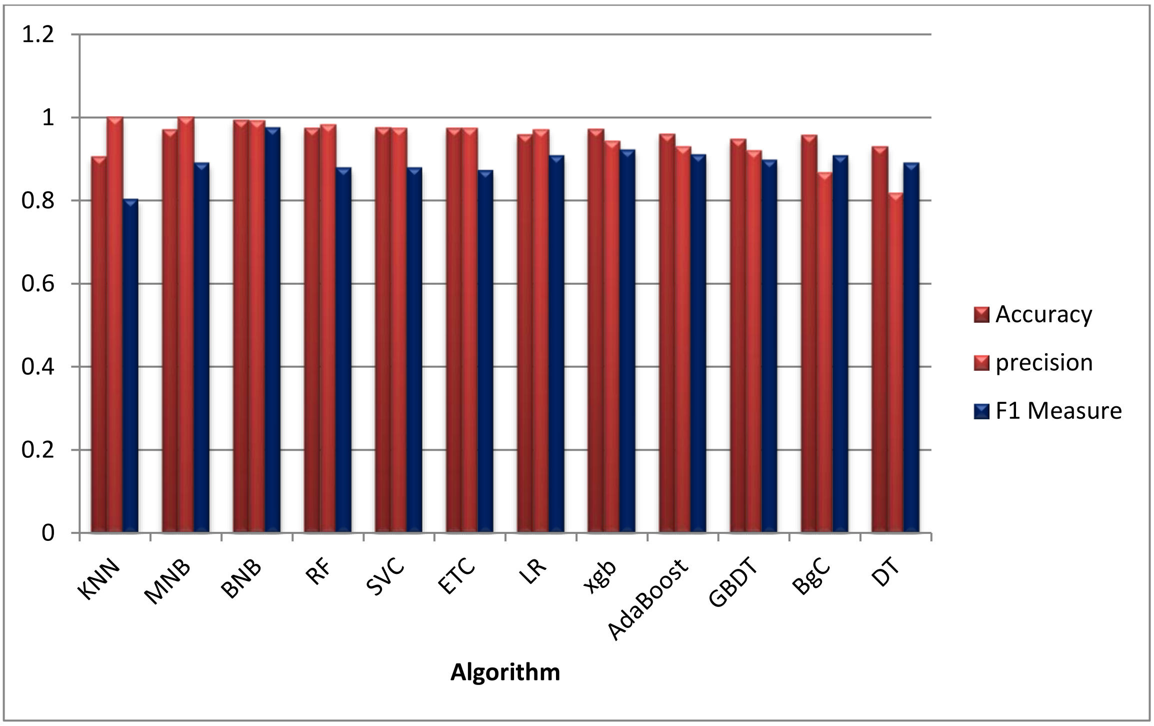


Figure 8Accuracy, Precision and F1 Measure of each algorithm in the proposed system

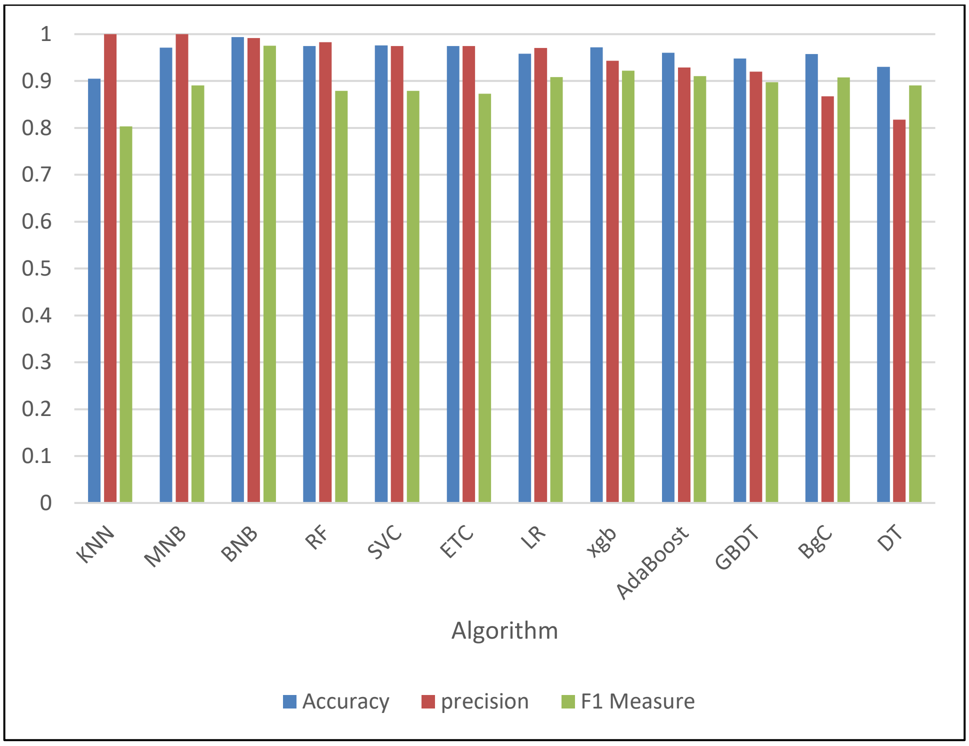


Figure 9 The percentage of Accuracy, Precision and F1 Measure of each algorithm in the comparative study system

In table 10, the top three algorithms used are BNB and LR. RF The ratio appears depending on the accuracy and precision of the algorithm of the proposed system TF-BNB and the algorithm of the comparative system IGM, as it shows that the algorithm of the proposed system is the highest percentage in terms of accuracy and is equal to 0.997559 compared to the algorithm in the comparative system, where the ratio was 0.89445. In figure 10, a chart shows the ratio of algorithms in the two systems.

Table 10 Top three algorithms in the proposed system

Method	**Accuracy TF-BNB**	**Precision TF-BNB**	**F1 Measure TF-BNB**	**Accuracy IGM**	**Precision IGM**	**F1 Measure IGM**
BNB	0.997559	0.99187	0.977523	0.89445	0.98265	0.87261
RF	0.974855	0.982759	0.965881	0.79009	0.97009	0.75206
LR	0.958414	0.970297	0.939473	0.94312	0.95454	0.92417

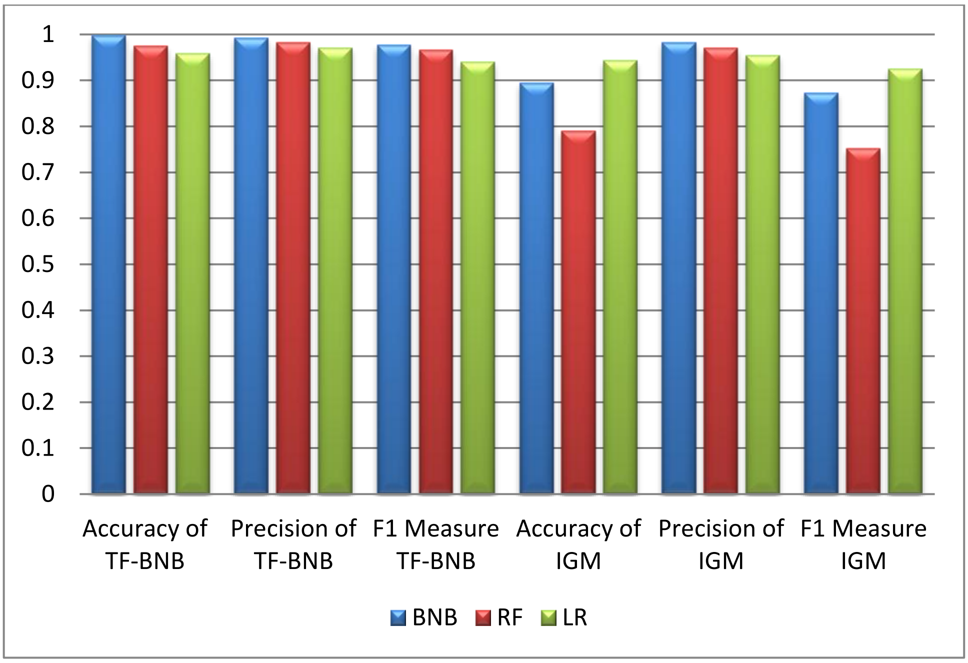


Figure 10 Accuracy, Precision and F1 comparison of algorithms

5. Conclusion

This chapter has given an introduction of a suggested system that uses machine learning to categorize and identify SMS spam, with a focus on the accuracy of the system using the Bernoulli Naive Bayes (BNB) algorithm. We started by bringing attention to the growing problem of SMS spam, which, despite being less prevalent than email spam, poses a serious risk to consumers' private and confidential information kept on their devices. We highlighted the negative effects of SMS spam, such as potential monetary losses and security breaches, emphasizing how it is a problem that impacts everyone. We introduced two important algorithms, BNB and TF-IDF, before moving on to the summary of findings. BNB, a form of the Naive Bayes algorithm, was selected for text classification due to its ease of use and effectiveness. BNB has some drawbacks, such as the assumption of binary characteristics and the inability to model feature correlations, it was stated. On the other hand, TF-IDF was used to assess the importance of terms in a text while considering word frequency. To assess the suggested approach, these algorithms were put to the test on two datasets: context and content. We acknowledged the suggested system's time-consuming development during the discussion of its limitations, which could lead to delays in finishing the necessary work. This emphasizes the requirement for more effective algorithms in further study.

REFERENCES

Alotaibi, Reem, Isra Al-Turaiki, and Fatimah Alakeel. "Mitigating email phishing attacks using convolutional neural networks." 2020 3rd International Conference on Computer Applications & Information Security (ICCAIS). IEEE, 2020.

Balubaid, M. A., Manzoor, U., Zafar, B., Qureshi, A., Ghani, N.: Ontology Based SMS Controller for Smart Phones. International Journal of Advanced Computer Science and Applications, 6(1), pp. 133–139 (2015)

Berns, Fabian, et al. "V3C1 dataset: an evaluation of content characteristics." Proceedings of the 2019 on International Conference on Multimedia Retrieval. 2019.

Bhushan, Bharat, Ganapati Sahoo, and Amit Kumar Rai. "Man-in-the-middle attack in wireless and computer networking—A review." 2017 3rd International Conference on Advances in Computing, Communication & Automation (ICACCA)(Fall). IEEE, 2017.

Bird, Steven; Klein, Ewan; Loper, Edward (2009). Natural Language Processing with Python. O'Reilly Media Inc. ISBN 978-0-596-51649-9.

Bosaeed, Sahar, Iyad Katib, and Rashid Mehmood. "A fog-augmented machine learning based SMS spam detection and classification system." 2020 Fifth International Conference on Fog and Mobile Edge Computing (FMEC). IEEE, 2020.

Bosaeed, Sahar, Iyad Katib, and Rashid Mehmood. "A fog-augmented machine learning based SMS spam detection and classification system." 2020 Fifth International Conference on Fog and Mobile Edge Computing (FMEC). IEEE, 2020.

Choudhary, Neelam, and Ankit Kumar Jain. "Towards filtering of SMS spam messages using machine learning based technique." International Conference on Advanced Informatics for Computing Research. Springer, Singapore, 2017

Delany, S. J., Buckley, M., Greene, D.: SMS Spam Filtering: Methods and Data. Expert Systems with Applications , Elsevier, 01(10) (2012)

Ethem Alpaydin (2020). Introduction to Machine Learning (Fourth ed.). MIT. pp. xix, 1–3, 13–18. ISBN 978-0262043793.

Gao, Min. "Account Takeover Detection on E-Commerce Platforms." 2022 IEEE International Conference on Smart Computing (SMARTCOMP). IEEE, 2022.

Gupta, M., Bakliwal, A., Agarwal, S. and Mehndiratta, P. (2018). A Comparative Study of Spam SMS Detection Using Machine Learning Classifiers, 2018 11th International Conference on Contemporary Computing, IC3 2018 pp. 1–7.

Gupta, Mehul, et al. "A comparative study of spam SMS detection using machine learning classifiers." 2018 Eleventh International Conference on Contemporary Computing (IC3). IEEE, 2018.

Harper, F. Maxwell, and Joseph A. Konstan. "The movielens datasets: History and context." Acm transactions on interactive intelligent systems (tiis) 5.4 (2015): 1-19.

Hu, J.; Niu, H.; Carrasco, J.; Lennox, B.; Arvin, F., "Voronoi-Based Multi-Robot Autonomous Exploration in Unknown Environments via Deep Reinforcement Learning" IEEE Transactions on Vehicular Technology, 2020.

Julis, M. Rubin, and S. Alagesan. "Spam detection in SMS using machine learning through textmining." Internatıonal Journal Of Scıentıfıc & Technology Research 9.02 (2020).

Kuhlman, Dave. "A Python Book: Beginning Python, Advanced Python, and Python Exercises". Section 1.1. Archived from the original (PDF) on 23 June 2012.

Mohammad, Rami M., Fadi Thabtah, and Lee McCluskey. "Tutorial and critical analysis of phishing websites methods." Computer Science Review 17 (2015): 1-24.

Mosquera, A., Aouad, L., Grzonkowski, S., Morss, D.: On Detecting Messaging Abuse in Short Text Messages using Linguistic and Behavioral patterns. Arxiv - Social Media Intelligence, http://arxiv.org/abs/1408.3934 (2014)

Narayan, A., Saxena, P.: The Curse of 140 Characters : Evaluating the Efficacy of SMS Spam Detection on Android. ACM (2013)

Nikiforakis, N., Maggi, F., Stringhini, G., Rafique, M. Z.: Stranger Danger: Exploring the Ecosystem of Ad- based URL Shortening Services. ACM, (April). doi:10.1145/2566486.2567983 (2014)

Nuruzzaman, M. T., Lee, C., Choi, D.: Independent and Personal SMS Spam Filtering. IEEE International Conference on Computer and Information Technology, pp. 429–435. doi:10.1109/CIT.2011.23 (2011)

Perkins, Jacob (2010). Python Text Processing with NLTK 2.0 Cookbook. Packt Publishing. ISBN 978-1849513609.

Ramzan, Zulfikar. "Phishing attacks and countermeasures." Handbook of information and communication security (2010): 433-448.

Rodrigues, Anisha P., et al. "Real-time twitter spam detection and sentiment analysis using machine learning and deep learning techniques." Computational Intelligence and Neuroscience 2022 (2022).

Sethi, G., Bhootna, V.: SMS Spam Filtering Application Using Android. International Journal of Computer Science and Information Technologies (IJCSIT), 5(3), pp. 4624–4626 (2014)

Sethi, Paras, Vaibhav Bhandari, and Bhavna Kohli. "SMS spam detection and comparison of various machine learning algorithms." 2017 international conference on computing and communication technologies for smart nation (IC3TSN). IEEE, 2017.

Sethi, Paras, Vaibhav Bhandari, and Bhavna Kohli. "SMS spam detection and comparison of various machine learning algorithms." 2017 international conference on computing and communication technologies for smart nation (IC3TSN). IEEE, 2017.

Shirani-Mehr, Houshmand. "SMS spam detection using machine learning approach." unpublished) http://cs229. stanford. edu/proj2013/Shir aniMeh r-SMSSpamDetectionUsingMachineLearningApproach. pdf (2013).

Shirani-Mehr, Houshmand. "SMS spam detection using machine learning approach." unpublished) http://cs229. stanford. edu/proj2013/Shir aniMehr SMSSpamDetectionUsingMachineLearningApproach. pdf (2013).

Shirani-Mehr, Houshmand. "SMS spam detection using machine learning approach." unpublished) http://cs229. stanford. edu/proj2013/Shir aniMehr SMSSpamDetectionUsingMachineLearningApproach. pdf (2013).

Siddique, Zeeshan Bin, et al. "Machine learning-based detection of spam emails." Scientific Programming 2021 (2021).

Simpson, Geoffrey, Tyler Moore, and Richard Clayton. "Ten years of attacks on companies using visual impersonation of domain names." 2020 APWG Symposium on Electronic Crime Research (eCrime). IEEE, 2020.

Song, G., Ye, Y., Du, X., Huang, X., Bie, S.: Short Text Classification: A Survey. Journal of Multimedia, 9(5), pp. 635–643. doi:10.4304/jmm.9.5. pp. 635-643 (2014)

Tu, Shanshan, et al. "Security in fog computing: A novel technique to tackle an impersonation attack." IEEE Access 6 (2018): 74993-75001.

Ullah, Abrar, Hannan Xiao, and Trevor Barker. "A dynamic profile questions approach to mitigate impersonation in online examinations." Journal of Grid Computing 17 (2019): 209-223.

Vural, I., Venter, H. S.: Combating Mobile Spam through Botnet Detection using Artificial Immune Systems. Journal of Universal Computer Science, 18(6), pp. 750–774 (2012)