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

Mustafa Tareq[1], Saad Adnan Abed[2]

[1]College of Computer Science, University of Technology, Baghdad, Iraq

[2]Department of Computer Science, College of Computer Science and Information Technology, University of Anbar, Ramadi, Anbar 31001, 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. Since individuals of all ages and socioeconomic statuses are heavily engaged in SMS, many of who are unaware of potential impacts of such, SMS is an easy way to attack individuals.

Once a phone is hacked, the attackers will access the phone and all the information stored in it without their awareness. 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 SMS industry is a multi-billion dollar sector that has grown as a result of the high rate of mobile phone use. Simultaneously, the decreasing price of messaging services has helped to increase spam aimed at the mobile devices significantly. 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 retrieved information on a learning repository and used an SVM algorithm, having a total accuracy of 97.64. The second-best model in the set was Naive Bayes with an accuracy 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). A Naive Bayes algorithm can effectively classify text as spam or not and so it obtained a high accuracy of 98.445 percent according to the information gain matrix. Figure 1 gives a comparative analysis of different machine learning algorithms.

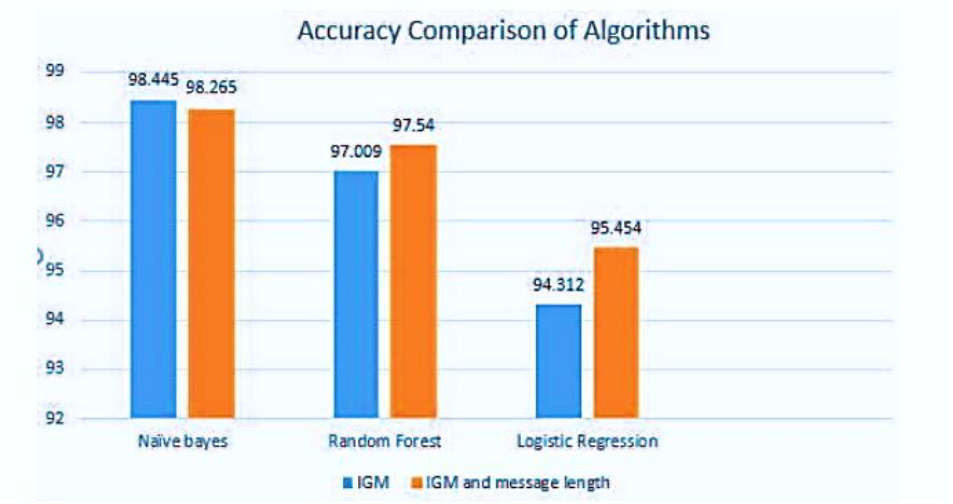

Figure 1. Comparative study of different ML algorithms

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 datasets 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. Based on the results of the classifier evaluation, the convolutional neural network recorded the highest accuracy of 99.19% and 98.25% with the AR of 0.9926 and 0.9994 on the two datasets.

Despite the fact that CNNs are highly applicable in image classification, they perform better than conventional classifiers and are also the most accurate in text classification. This has made them applicable to text classification tasks like sentiment analysis and review classification. In the meantime, more traditional classifiers such as SVM and Naive Bayes are also effective, with their results being similar to CNN on both datasets. The results indicate that there is a high probability of application in the real world in SMS spam detection because of the high levels of performance.

Even though this method can be used on both incoming and outgoing SMS, Bosaeed et al. created a detection and classification system, which specifically offers a tool to identify the spam in 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. The system detects spam messages and suggests appropriate spam filters and classifiers in response to user settings, including rating accuracy, true negatives (TN) and computing resource requirements. The analysis indicates that the PF5 filter and the SVM is the best in overall performance in SMS classification.

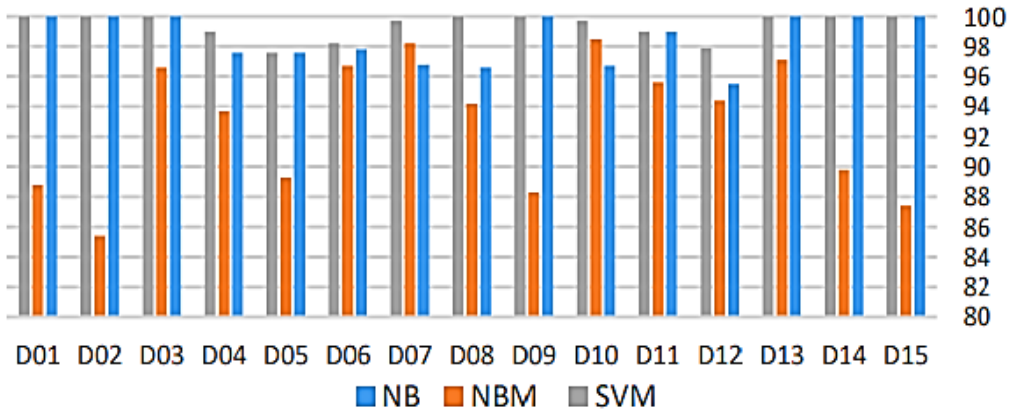

Figure 2 TNR using PF5

They created this strategy, as a reaction to the spam issue, which is common in the most popular social networks like Twitter, Facebook and Quora. The objectives of spam accounts on these platforms are to trick actual users into clicking the malicious links or to create spam messages by use of bots, which can influence user interaction. Although there is a lot of literature on the means of identifying certain forms of spam, sentiment analysis on such posts is a viable solution to the problem.

The main aim of this research is to come up with a system that can evaluate the sentiment of a tweet and at the same time identify whether the tweet is spam or junk. In the case of spam detection, different classifiers, including Decision Trees, Logistic regression, Polymorphic naive bayes, Support Vector Machines, random forests and Bernoulli naive bayes are used on features obtained after preprocessing of tweets. Deep learning is used in sentiment analysis, which includes Stochastic Gradient Descent, Support Vector Machines, Logistic Regression, Random Forests, Naive Bayes, and neural network models, such as Simple Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM), Bidirectional LSTM (BiLSTM), and 1D Convolutional Neural Networks (1D CNN).

Each classifier's performance is evaluated. The obtained classification results show that one can reliably identify whether a tweet is spam based on its extracted features and create a learning model, which correlates the tweets with a particular sentiment.

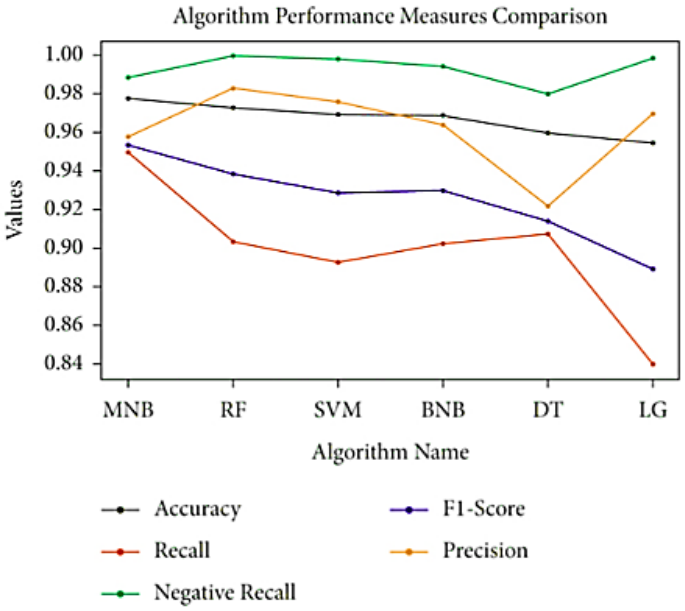

Figure 3 Algorithm performance measures comparison

The validation accuracy of the Polynomial Naive Bayes classifier in recognizing Twitter spam was 97.78% and deep learning LSTM model had a better validation accuracy of 98.74%. This type of classification displays that the features obtained through tweets can be successfully applied to tell whether 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 part introduces the use of data sets in the research and an overview of the proposed research 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 SMS spam dataset that is collected to research on SMS spam. It has 5,572 messages written in English language, with 5,572 messages classified as ham (legitimate) and 5,572 messages as spam (Harper, F. el al.,2015).

B. Dataset Content

One message is represented in each line and there are two columns in each line: v1 which contains the label (ham or spam) and v2 which contains the actual text of the message. This corpus was compiled from free or low-cost research sources on the Internet: (Berns, Fabian, et al.,2019)

- Grumble Text Website: A total of 425 SMS spam messages were selectively sampled on the Grumble text site, a UK based forum where mobile users report spam SMS. The spam message itself is not presented in most reports and hence is a difficult and time-consuming task that involves the systematic review of hundreds of web pages.
- NUS SMS Corpus (NSC): The sample size was 3,375 SMS messages which were sampled out of the NSC, a corpus of about 10,000 legitimate messages collected by the Department of Computer Science at the National University of Singapore. The messages are primarily Singaporean, mostly university students and were gathered among the participants after they were informed that their submissions would be published.
- Doctoral Study of Caroline Tag: It supplied 450 valid (ham) SMS messages.
- SMS Spam Corpus v.0.1 Big: It contains 1,002 ham messages and 322 spam messages.

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.

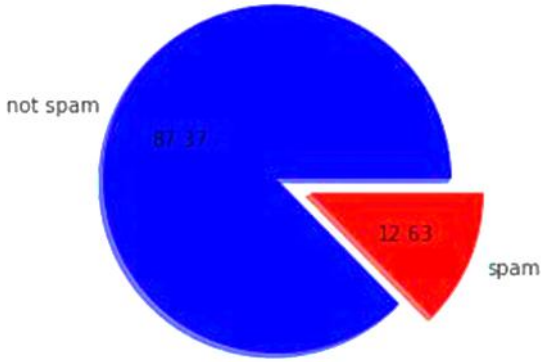

Figure 4 The percentage of spam and ham

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 correlation between new columns added and the target variable according to their means, maximum, and minimum values.

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

The Exploratory Data Analysis (EDA) is an important part of the data science and machine learning process. It allows us to comprehend our data because it is analyzed through various perspectives with the help of statistics, visualizations and summaries. EDA aids in revealing patterns, outliers and creating an overall perspective of the data.

In this Python EDA tutorial, we shall illustrate how to use Pandas Profiling, which is an effective library that enables quick and automated EDA to data with minimum code.

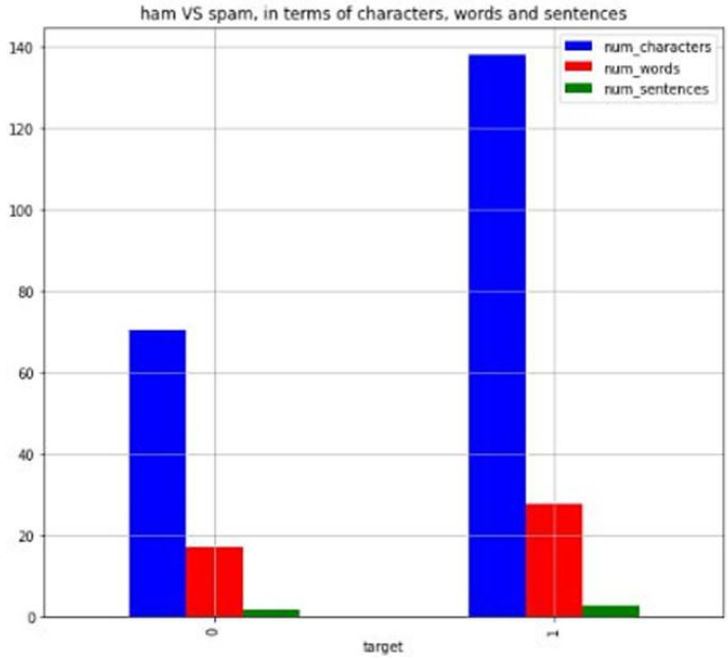

Figure 5 Ham VS spam in terms of characters, word and sentences

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 judge data prior to its analysis or prediction. Text pre processing is used in text based projects to format the data to be used in the development of models. It is the primary step of an NLP project. The preprocessing phases normally involve:

a. Lower case

Conversion of text to a consistent case, usually in lowercase, is one of the most frequently used text preprocessing operations that are performed in Python. This is not always required though sometimes the lowercasing can cause information to be lost. An example can be in sentiment analysis where the uppercase words can be used to express strong emotions like anger or excitement. On application, this step converts all letters into lowercase.

b. Tokenization

The text is subdivided into small units in this step. We can either do sentence tokenization or word tokenization depending on the kind of problem in question.

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 typical words that are normally removed in the text since they usually do not carry much meaningful information to analyze them. Such words that include the, is, and tend not to add to predictive strength of a model. The NLTK library offers by default a list of English stop words:

(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, a predefined list of stop words is not always required, and they need to be chosen carefully regarding a particular project. As an example, when examining customer inquiries, a word that may be regarded as a stop word in one model may be significant to another. We might even make problem specific stop word characteristics in some instances.

e. Stemming

This step is also called language standardization, and consists of simplifying the words to their root or base word with the help of stemming. e.g. fear of words such as programmer, programming, and programmed are all shortened to program. Once this has been done the processed text can then be saved to be used later. 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.

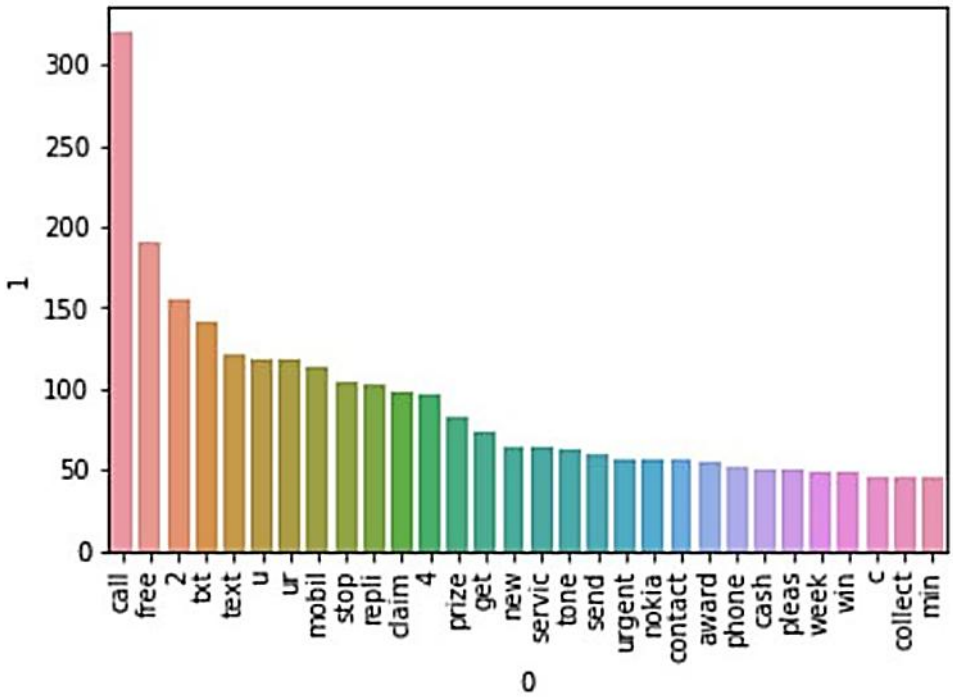

Figure 6 Chart of the most used words in SMS spam

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.

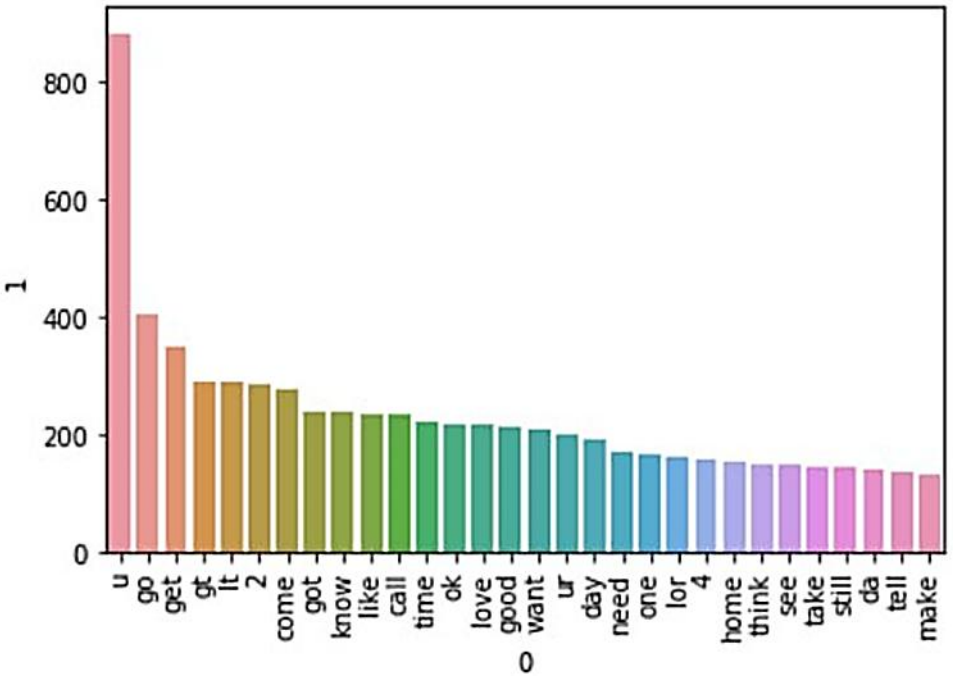

Figure 7 Chart of the most used words in HAM spam

Step 4: Model building

1-TF-IDF model

The model of building the system was TF-IDF. TF-IDF (Term Frequency-Inverse Document Frequency) represents a representation method used in text and determines the significance of words within a document. It is based on the fact that the more words and document appear to be more important whereas the more common words and documents are then less informative. TF-IDF is a combination of TF and IDF.

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

Term Frequency: The Term Frequency (TF) represents the frequency of a word that is used in a document. The ratio between the occurrence of a word in the document and the number of words in the same document is calculated.

Inverse Document Frequency: It's a measurement of how much detail the word gives on the topic of the paper. The Inverse Document Frequency (IDF) determines the significance of a term in all documents. It is estimated by the logarithm of the amount of documents to the number of documents that included the word.

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.

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

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

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

TF-IDF approach can address the weakness of the Bag of Words (BoW) approach. In contrast to BoW where all words are given the same weight and only word frequencies are counted, TF-IDF puts more emphasis on meaningful words that might be represented in a very small number. Although BoW is a simple representation of text by the frequency vectors, TF-IDF not only captures the importance of words but also their distribution across the documents, which is a more informative representation.

2- BernoulliNB models (BNB)

BernoulliNB BNB is the Naive Bayes training and classification which uses multi-variate Bernoulli distributions to model data. In this model, individual features are considered as binary (Bernoulli or Boolean) variables although there may be more than one feature. Thus, the samples of input are to be presented as feature vectors with binary values. In case of alternative data form, it is possible to have the data binarized automatically using the object BernoulliNB (C.D. Manning et al., 2008).

The Bernoulli Naive Bayes decision rule is determined by the following equation 4.

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

This is in contrast to the Multinomial Naive Bayes rule that the presence of an absence of a feature (i) correctly indicates the absence of a specific feature (y) in class (i) at the cost of the absence of non-occurring features being overridden by the presence of occurring features though not explicitly stated in the rule. A word occurrence vector can be trained and evaluated through this classifier in text classification. BernoulliNB can be more effective with some sets of data, especially those that have short texts. It is advisable to compare the two models when feasible in an attempt to establish which one works best (A. McCallum and K. Nigam, 1998).

3.6 Evaluation

a) Accuracy:

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.

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

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

b) precision:

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.

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

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:

F-Measure (Van Rijsbergen, 1979) takes into account both the precision as well as the recall to determine one performance score. The precision is the count of correct results out of the total number of results returned and recall is the count of correct results out of the total number of them that are supposed to be returned. The F-Measure is viewed as a weighted average of precision and recall: the maximum F-Measure is 1.

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

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

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

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.

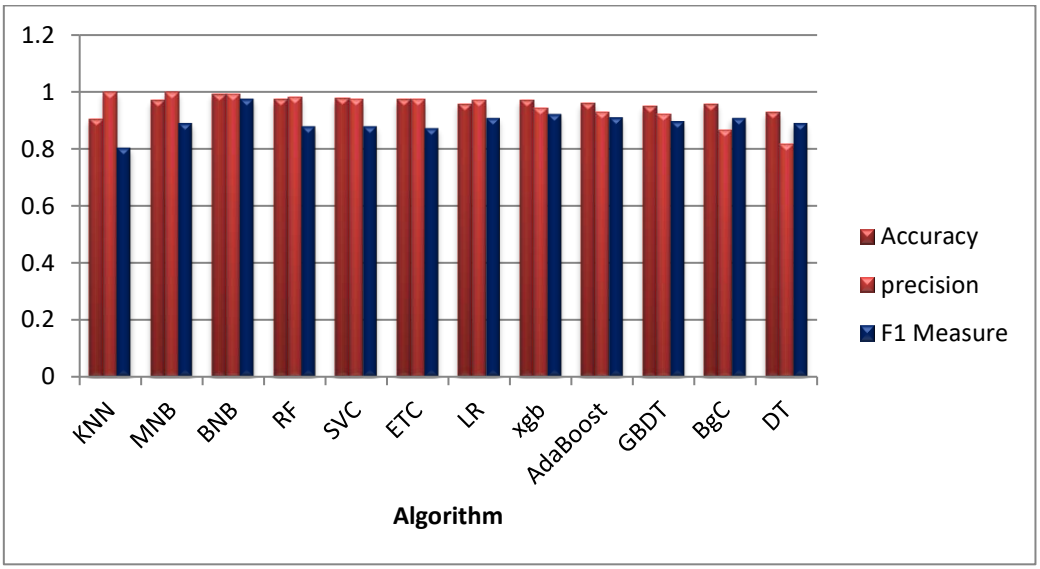

Figure 8Accuracy, Precision and F1 Measure of each algorithm in the proposed system

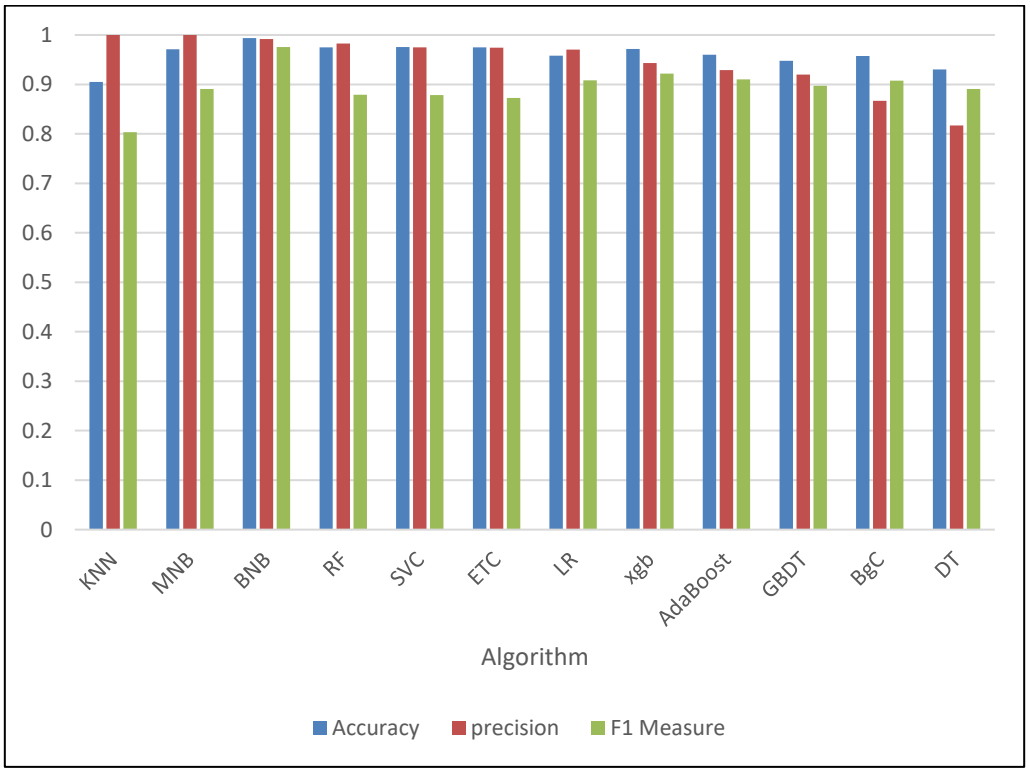

Figure 9 The percentage of Accuracy, Precision and F1 Measure of each algorithm in the comparative study system

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

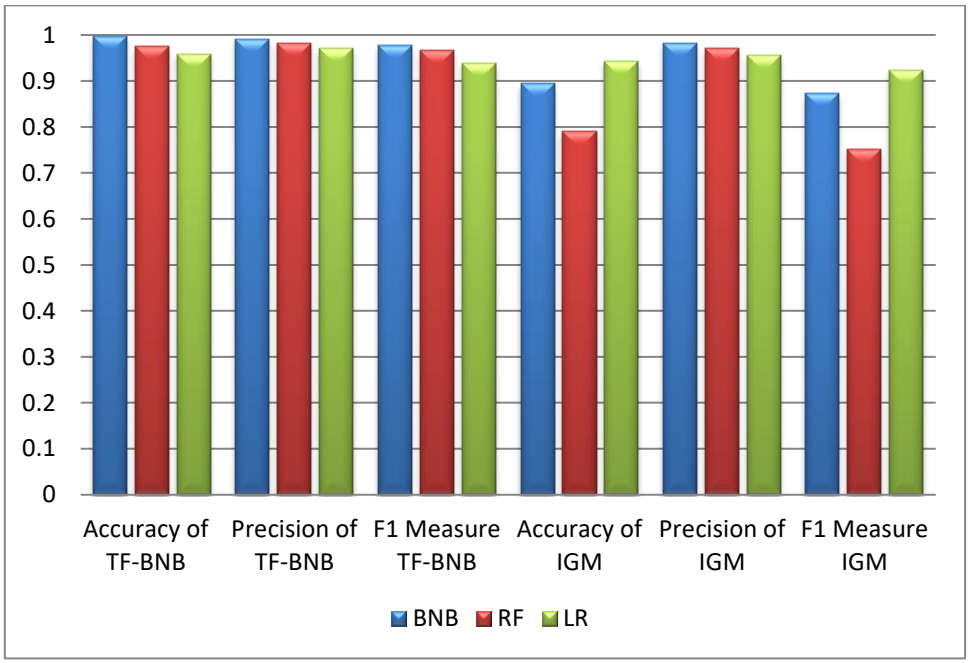

Figure 10 Accuracy, Precision and F1 comparison of algorithms

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. Conversely, the importance of words in a text using the

method of TF-IDF was used to consider the frequency of words within a document and spreads the words within a complete corpus. 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.

6. 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 Scıentıfic & 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)