**Spam Message Filtering Based on Machine Learning**

**Algorithms and BERT**

Chandan J R1, Glennis Elwin Dsouza 2, Merin George3and Jayati Bhadra4

1St. Josephs College, Bengaluru

1chandanjr18@gmail.com

2 St. Josephs College, Bengaluru

2glennis25dsouza@gmail.com

3St. Josephs College, Bengaluru

[3georgencmerin@gmail.com](mailto:3georgencmerin@gmail.com)

St. Josephs College, Bengaluru

[4jayatibhadra@sjc.ac.in](mailto:4jayatibhadra@sjc.ac.in)

**Abstract.** The constant traffic of messages keeps increasing whether email or SMS. This leads to a direct increase of attacks from spammers. Mobile spams are particularly threatening as spam messages are often disguised as messages from banks, which cause immense harm. Due to this having a method to identify if the messages are spam or not becomes pivotal. Several techniques exist where deep learning tasks show higher accuracy compared to classic machine learning tasks. Hence, this paper compares the BERT with traditional machine learning techniques which are used in this paper are Logistic Regression, Multinomial NaiveiBayes, SVM and Random Forest. This paper uses an open-source dataset from Kaggle of labelled spam and not spam messages. It has a total of 5585 messages of which 4825 are labelled as spam and 760 as spam. We use various machine learning algorithms and an algorithm with BERT. In comparison to the other machine learning techniques, we discovered that the BERT algorithm delivered the highest testing accuracy. BERT has an encoding layer which generates encodings such that they are not biased and are then fine-tuned for spam detection. With this attribute, the algorithm obtains a higher accuracy.

**Keywords:** Short message service, machine learning algorithms, deep learning, NLP, BERT.

**1 Introduction**

Mobile phones have become companions to most of the population. The need to constantly check messages and ignore messages has become so innate that we never second guess as we swipe. This makes us easy prey to spammers. The number of people who get scammed are on a constant increase. Give a comprehensive study as to the statistics of people getting scammed in various ways and their new anatomy.

Knowing how to identify spam messages is vital to avoid potential risks of scams and space on devices. Earlier, one could identify them as messages sent from a Nigerian prince, or just sending money to an unknown person. But scammers have changed tactics, Alkhalil et al. [7] give a comprehensive study on them and provide a new anatomy as to how they work. Spam messages are described as unsolicited digital communication transmitted in large quantities. Spam wastes a lot of time and resources. From their paper on can note using SMS as a mode of phishing has gained popularity to aide scammers. Of the organizations faced targeted phishing attacks in 2019, 84% SMS/text phishing was discovered (SMishing)[7].

A basic outline of a spam detection model involves, collection of data, text processing and building a model for the classification of spam or not. The dataset we have used comprised of 5585 messages of which 4825 are not spam and 760 are spam. We referred to a dataset already existing from Kaggle. Our objective was to identify messages if they were spam or not, this led us to use classification techniques as a base. We compared several techniques that have been mentioned in previous studies as classifiers. And we also used BERT.

A crucial part of spam detection involves filtering the data(texts). There are many ways of text processing for this task. A common approach is using Natural language processing (NLP). This is seen in several papers. An example would be in A. Nabi et. Al [8], they had used the Keras tokenization programme which divides words into tokens depending on space. Using Keras' Tfidf Vectorizer, each token was encoded as a vector for classical classifiers. The SPAM 1 and HAM 0 binary formats are used to encode the label target variable.[8]. Another one was seen in Roy et. al [4] where supervised model has been used. Nouns, Adjectives, Verbs, Difficult Words, Fletch Reading Score, Dale Challe Score, Length, setilength, Stop Words, Total Words, Wrong Words, Entropy, One Letter Words, Two Letter Words, and Longer Letter Words were all extracted from the texts. In addition, a full description of the chosen features was defined. Using classifiers, the messages were categorised as spam or not-spam based on these characteristics. [4].

Our next task would involve building a classification model and comparing it with other existent models. Some of the basic models we had chosen were Logistic regression, Naïve Bayes, Support vector machines and random forest. Our approach involved using BERT [9].

BERT is employed in this paper because it is useful for a wide range of linguistic tasks, and it only adds a modest layer to the main model. BERT is an acronym for Bi-Directional Encoder Representation from Transformers. The reason for using BERT was to generate encodings such that they are not biased and then later use them for spam detection. BERT has multiple encode layers which would do the pre-processing and then encode the inputs in some form of embeddings which was then fine-tuned to as a classification model.

**2 Related Work**

Several Techniques are available for detecting spam messages, as well as spam filtering technologies, are currently accessible. Many Android-based programmes for classifying spam messages and filtering spam communications are available in the market or online. Classification algorithms have been used to explain a variety of machine-learning systems. The following is a list of previous work in this field-

Sjarifi .et. al[1], On SMS spam messages data, employ Term Frequency-Inverse Document Frequency(TF-IDF) and Random Forest Algorithm approaches. The stages of spam detection are as follows, pre-processing, extraction, selection, and classification of features. For Feature extraction and selection TFIDF was used. And various other algorithms were used for classification. Based on their experiments, the Random Forest algorithm outperformed the other algorithms have a 97.50 percent accuracy [1].

Bosaeed et. al [2], We generated numerous machine learning (ML) based classifiers utilising three classification methods – Nave Bayes (NB), Support Vector Machine (SVM), and Multinomial Nave Bayes – as well as five pre-processing and feature extraction methods. The technology is designed to operate in cloud, fog, or edge layers. The precision of PF5 is shown in the results. Overall, SVM appears to be the most effective.

Choudhary et. al [3], After doing extensive research into the characteristics of spam messages, 10 elements that can effectively filter SMS spam messages were discovered. After extracting features, the WEKA tool uses five machine learning methods to test classification accuracy: Nave Bayes, Logistic Regression, J48, Decision Table, and Random Forest. For the Random Forest classification technique, their proposed approach achieved a 96.5 percent true positive rate and a 1.02 percent false positive rate. [3].

Roy et. al [4], Text messages were classified as Spam or Not-Spam using deep learning. Convolutional Neural Network and Long Short-Term Memory models were used in particular. The models they proposed were only based on the feature set and textual data was self-extracted. A remarkable accuracy of 99.44 percent was achieved on a benchmark dataset consisting of 747 Spam and 4,827 Not-Spam text messages. [4]

Crawford et. al [5], carried out a survey of the most popular ML algorithms offered to handle the problem of review spam detection, as well as the performance of various approaches for categorization and detection of review spam. This research provided insight into the stages and methods utilised in spam detection, but not when it comes to SMS spam detection. They discovered that the majority of current research has concentrated on supervised learning approaches, which necessitate labelled data, which is scarce in the case of online review spam. They conducted a thorough and comprehensive evaluation a summary of latest studies on detecting fake reviews using multiple machine learning approaches, as well as developing methodologies for further exploration.[5]

Kaliyar et al. [6], Using various known machine learning classification methods, on condition that a general model which can recognise and filter spam transmission. Their approach makes a generalised SMS spam-detection model that can clean messages from a different number of sources (singapore, smerican, Indian English etc.). Initially results based on publicly available Singapore and Indian English datasets were used in their methodology. Their method shows potential in achieving high precision using big datasets of Indian English SMS and other background datasets.[6].

From the previously mentioned papers, a frequent strategy is to use machine learning algorithms. On further reading, a BERT based algorithm was used for spam detection [9]. Considering this, this paper compares various machine learning techniques with BERT for spam detection.

Diagram

Description automatically generated**3 Experıment Desıgn**

**Fig 1:** Diagram showing the text processing

**3.1 Pre-Processing the Data**

The data was gathered from the Kaggle Repository. and to the same dataset some more spam messages from our side has been added.

The dataset used contained 5585 rows of records, tagged accordingly whether the message is spam or ham messages and 5 columns in which first two columns were named as V1 and V2 and other 3 were having null values.

While pre-processing the data the 3 columns which were unnamed and were blank were dropped since they were of no use and the column v1 and v2 were renamed as “CLASS” and “TEXT” where the class column contained the label of the messages like Spam and Ham and text column contained the messages.

Label encoding was done to convert the label which was in text format to numerical values that is Spam as 1 and Ham as 0. There were no null values in the data.

The proportion of class variables were like 0 that is ham messages were 7302 and 1 that is spam messages were 1143. Then the length of each message was checked and displayed.

**3.2 NLP techniques**

**Stemming**: Words were turn downed to their attained form. For e.g., ‘runs’, ‘ran’, ‘running’ are replaced to ‘run’. The stemmer sometimes trimmed additional character from the end e.g., ‘include’, ‘included’, ‘includes’ becomes include.

**Removal of Stop words**: Words such as ‘the’, ‘an’, ‘in’ etc. which does not make significance sense in a sentence were removed. Even punctuations are removed. In addition to these even tabs, newlines, and spaces are assumed to be a single space character.

**Lower-casing**: All of the words are written in lower case, with no capitalization.

Due to imbalanced in the data, to balance the data set, class weights were given. The class weights were set for the loss function. 'Spam' variable is set to weight 8x more.

**3.3 Modelling**

**Multinomial Naïve Bayes:** From J. Eberhardt [10] it was understood that multinomial Bayes is an optimized Naïve Bayes model that is used to make the filter more accurate.

Once the data is given, a filter is developed in the Nave Bayes section that assigns a chance to each feature being in spam. In this scenario, probabilities are expressed as numbers between 0 and 1.

Now that the filter can detect a spam based on training instances, it will review whether a mail is spam deploy on the probability of every single word uniquely.

Equation 1: Given the word Y in the Text, the probability of P(X|Y) being a spam Text message is being shown using the Bayes theorem. The requested word is represented by Y, a spam Text is represented by X, and a non-spam Text is represented by Z.

(1)

In addition, the probabilities of all the items in an SMS should be merged. To express the probability, we use equation 2. P(X|Y) is the chance that an item is -spam if a set of terms Y appears in the SMS. N represents the total number of features sorted, i.e., the total number of words in the SMS.

(2)

Multinomial we adopt the same procedures as Nave Bayes, but we also keep a count of how many times each term arises. As a result, in the third equation: A multi-set of words in an SMS is denoted by Y. This means that Y has the correct number of appearances for each term. This implies that the SMS contains spam.

(3)

Once you have typed down the Naive Bayes equation, you can consider that the words are derived using a multinomial distribution and are independent. Because evaluating whether a SMS is spam is binary, We understand that our classes could only be 0 or 1. Equation 4 shows the Multinomial Bayes technique to spam detection based on these constraints. Where fY denotes the amount of times the term has been used Y appeared in the multiset Y, and Y denotes the product of each word in B.

(4)

Although Nave Bayes is frequently used to discover spam communications, it is rarely employed since it implies that the attributes it is classifying, in this example the specific words in the email, are independent of one another. The filter is more precise.[10]

**Support Vector Machine for Classification:** SVM is a supervised machine learning technique for solving classification and regression issues. It is, however, largely used to solve classification problems.

We take the same strategy as N. Krishnaveni et al. [11]. Where each data item is represented as a point in n-dimensional space (where n might be one or more features) with the value of each variable becoming the value of a certain point at certain points in the SVM classifier. The classification is then carried out by discovering the hyper-plane that best picks out the two classes.

**Random Forest Classifier:** A random forest is a machine learning method that can be used to address problems like regression and classification. It makes use of ensemble methods, which is an approach for solving complex situations that involves multiple classifiers. [12]

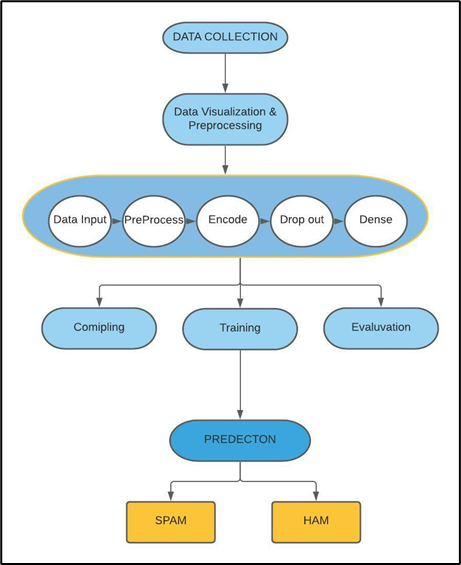
Akinyelu et al. [13] used the same technique. However, we used 3-fold cross-validation to train and test the classifier in this study. The dataset is separated into eight parts in 3-fold cross-validation; seven of the eight parts are utilized to train the classifier, while the information gained during the training process is used to authenticate the eighth part.

**Logistic regression:** From N. Englesson [14] it was understood that logisticiregressioniisia statistical techniqueiwhich is used to show is a binary response variable is dependent on one or more independent variable. It is often used for building a model in situations where there is a two-level categorical response variable.

(5)

The Logistic function, often known as the Sigmoid function, is depicted in equation 5 above. The Sigmoid curve is defined as a result of this. An S-shaped curve is applied to the Sigmoid function. The output of the Sigmoid function tends to 1 as z increases and tends to 0 as z decreases. As a result, the Sigmoid/logistic function always produces a value for the dependent variable that is between [0,1], i.e. the probability of belonging to a class.

**BERT:** A Transformer is essentially the basic structure of BERT (Bidirectional Encoder Representations from Transformer). The transformer is an attention mechanism which learns the contextual relation amongst the words in a text. At its core, the Transformer consists of two distinct mechanisms: An encoder, which reads the imputed text, and a decoder, which is used to produce a prediction for the specific task. The main aim of BERT is to generate a language model; hence, the encoder mechanism is sufficient.



**Fig 2:** Block Diagram of BERT model in spam detection

Unlike traditional directional models, wherein the text is read in a sequence; the Transformer encoder reads the whole sequence of words at once. This implies that it is bidirectional, but in its essence, it is non-directional. It is with this feature that BERT learns the context of the word based on the sequence imputed.

The main idea of these two steps is: With general pre-trained models, one can fine-tune them on datasets that are smaller and have specific task orientations. An example of this would be problems regarding, answering questions [16] and sentiment analysis [15]. The main benefit of this approach is that the improvement of accuracy when compared to regular methods of training a model on task specific datasets, is much greater. This is evident in this paper.

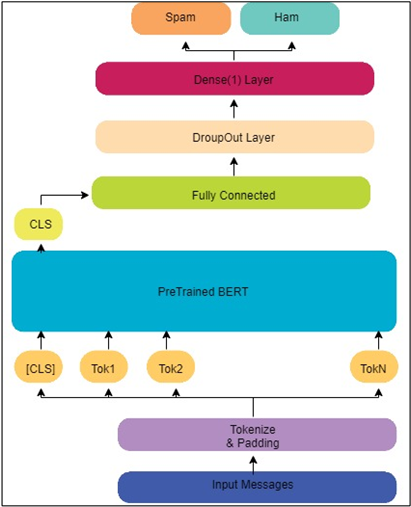
*Pre-training of Bert*

There are two main aspects as to which how BERT is trained: The first one is MLM(Masked Language Modelling) and NSP(Next Sentence Prediction).

In MLM, a percentage of the words are masked. There is no pre-determined way that this is done, hence it is random. They are masked (i.e., Replaced) with a token ([MASK]). It (the BERT model) is then trained as to identify these masked words with only the context of the leftover words. An issue with this is that this element is pretrained, from the paper [9], we know that 15% masked tokens would exist. But in the process of fine-tuning, MLM does not occur. This puts a strain as to how the task specific dataset would work. To rectify this issue, 80% (of the 15% mentioned earlier) are truly substituted with the token [MASK], the rest 10% are substituted with other random tokens, and the remaining are not substituted.

Using NSP training, BERT understands the relationship amongst two sentences. Having the input of pairs of sentences, the model then must predict if the next sentence is truly the next one or not. There is a 50% probability in this type of prediction. Random sentences must be disconnected from the initial sentence is the key assumption, in its context.

*Fine-tuning Bert*



**Fig 3:** Detailing Fine Tuning of BERT fine tuning

The main purpose of fine tuning is to avoid major task specific changes in the main architecture by using the learned weights form the pre-trained phase. In our case we aim to fine tune the BERT base model to do text classification in the circumstenses of spam recognition. This falls in the category of sequence level tasks.

From the figure 3, we note that the token [CLS] had been used for the previously mentioned task. The [CLS] token is always at the beginning of any text. There is another token [SEP], this one is used by BERT to tell the difference between two sentences. Both these tokens are crucial and required to be mentioned in the fine tuning of BERT. The main technique used in this paper for fine-tuning is that we 'froze' the entire architecture. This means that the layers of BERT base were frozen (left unaltered) while for our case 5 layers were added for the purpose of spam (text) classification.

**4 Results**

As seen in the table below, The BERT had outperformed the ML classifiers in terms of their accuracy and greatly in terms recall. But in terms of F1 scores and precision, BERT had comparable scores compared with the ML classifiers.

**Evaluation Metrics:** We will analyse the following possible outcomes to assess the effectiveness of our suggested approach: True Positive Rate, False positive rate, True negative rate, False negative rate, f1 score, accuracy, precision, and recall. Those are the benchmarks against which any spam detection system is measured. In a nutshell, these evaluation metrics are as follows: -

* TRUE POSITIVE RATE – It represents the propotion of spam texts which the machine learning algorithm correctly classified. If spam communications are denoted by the letter S, and spam messages that were correctly represented by the letter P, then:

TP = P/S

* TRUE NEGATIVE RATE: It is the percentage of ham messages correctly classified as such by the machine learning system. If we call a ham message H and a ham message that was correctly classified as ham by Q, we get:

TN = Q/S

* FALSE POSITIVE RATE - It represents the ratio of ham messages that the machine learning algorithm incorrectly classified as spam. If we refer to ham communications as H and ham messages that R incorrectly labelled as spam as R, we can say:

FP = R/H

* FALSE NEGATIVE RATE - It is the proportion of spam messages that the machine learning algorithm incorrectly identified as ham communications. If spam messages are denoted by S, and the number of SMS spam messages wrongly labelled as ham by T, then:

FN = T/S

* PRECISION - It represents the percentage of spam messages classified as spam by the classification algorithm. It demonstrates that everything is correct. It is written as follows:

Precision = TP/(TP + FP)

* RECALL - It represents the percentage of spam messages which were classed as spam. It exhibits thoroughness. It is written as follows:

Recall = TP/(TP + FN)

* F-measure - Precision and Recall are combined to form the harmonic mean. It's written as:

**Table 1.** Comparative study of our method and other state of art methods

|  |  |  |
| --- | --- | --- |
| Models | Accuracy | F1 score |
| Logistic Regression | 0.96 | 0.97 |
| Multinomial NB | 0.97 | 0.97 |
| SVM | 0.96 | 0.98 |
| Random Forest | 0.97 | 0.97 |

From the above work we can conclude that the BERT model had performed the best when compared to other machine learning models. It had shown the best accuracy compared to machine learning models mentioned. We will aim to add more features in the future, as the best spam features assist in the more accurate detection of spam messages. We'll also make an effort to collect more datasets from the real world.

**5 Conclusions**

With the increased use of text messaging, the problem of SMS spam is becoming more prevalent. Filtering SMS spam has been a major concern in recent years. In this research, we offer a method for filtering SMS spam. We find that the BERT algorithm provided the highest testing accuracy as compared to other machine learning techniques. BERT has an encoding layer which generates encodings such that they are not biased and are then fine-tuned for spam detection. With this attribute, the algorithm obtains a higher accuracy.

For future work, a method to further improve the results would be by taking an even larger dataset, due to limitations in processing speeds, we had limited our dataset to 5585 records. The spam identification task can also be used to languages of different texts, an example would be Kannada, Tamil, Arabic etc. We can also differentiate languages within the messages for example a message with English and French (similar scripts) and English and Kannada (different scripts).

**References**

1. Amir Sjarif, N. N., Mohd Azmi, N. F., Chuprat, S., Sarkan, H. M., Yahya, Y., & Sam, S. M. (2019). “SMS Spam Message Detection using Term Frequency-Inverse Document Frequency and Random Forest Algorithm”. *Procedia Computer Science*, 161, 509–515.
2. Bosaeed, S., Katib, I., & Mehmood, R. (2020). “A Fog-Augmented Machine Learning based SMS Spam Detection and Classification System”. *2020 Fifth International Conference on Fog and Mobile Edge Computing (FMEC).*
3. Choudhary, N., & Jain, A. K. (2017). “Towards Filtering of SMS Spam Messages Using Machine Learning Based Technique”. *Advanced Informatics for Computing Research*, 18–30.
4. Roy, P. K., Singh, J. P., & Banerjee, S. (2019). “Deep learning to filter SMS Spam. Future Generation Computer Systems”. DOI: 10.1016/j.future.2019.09.001
5. Crawford, M., Khoshgoftaar, T. M., Prusa, J. D., Richter, A. N., & Al Najada, H. (2015). “Survey of review spam detection using machine learning techniques”. *Journal of Big Data, 2(1).* DOI:10.1186/s40537-015-0029-9
6. Kaliyar, R. K., Narang, P., & Goswami, A. (2018). “SMS Spam Filtering on Multiple Background Datasets Using Machine Learning Techniques: A Novel Approach”. *2018 IEEE 8th International Advance Computing Conference* (IACC). DOI:10.1109/iadcc.2018.8692097
7. Zainab Alkhalil, Chaminda Hewage, Liqaa Nawaf and Imtiaz Khan. (2021). “Phishing Attacks: A Recent Comprehensive Study and a New Anatomy”. *Front. Comput. Sci*., https://doi.org/10.3389/fcomp.2021.563060
8. A. Nabi, I., & Yaseen, Q. (2021). “Spam Email Detection Using Deep Learning Techniques”. *Procedia Computer Science*, 184, 853–858. DOI: 10.1016/j.procs.2021.03.107
9. J. Devlin, M. W. Chang, K. Lee, K. Toutanova (2019). “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. arXiv:1810.04805v2 [cs.CL] 24 May 2019.
10. J. Eberhardt (2015). “Bayesian Spam Detection”. *Scholarly Horizons: University of Minnesota, Morris Undergraduate Journal,* vol. 2, no. 1, Article2
11. N. Krishnaveni and V. Radha (2021). “Comparison of Naive Bayes and SVM Classifiers for Detection of Spam SMS using Natural Language Processing”. *ICTACT Journal on Soft Computing, January 2021, Volume: 11, Issue: 02* DOI: 10.21917/ijsc.2021.0323
12. L. Breiman and A. Cutler, “Random forests-classification description,” Department of Statistics Homepage, 2007, http://www.stat.berkeley.edu/∼breiman/RandomForests/cc home.htm.
13. A. A. Akinyelu and A. O. Adewumi (2014). “Classification of Phishing Email Using Random Forest Machine Learning Technique”. *Hindawi Publishing Corporation Journal of Applied Mathematics* Volume 2014, Article ID 425731, 6 pages <http://dx.doi.org/10.1155/2014/425731>
14. N. Englesson (2016) “Logistic Regression for Spam Filtering”. *Kandidatuppsats 2016:9 Matematisk statistik* June 2016
15. N. C. Dang, M. N. Moreno-Garcia and F. De la Prieta (2020), “Sentiment Analysis Based on Deep Learning:A Comparative Study”, *Electronics 2020*, 9, 483; doi:10.3390/electronics9030483. <https://arxiv.org/ftp/arxiv/papers/2006/2006.03541.pdf>
16. S. Siriwardhana, R. Weerasekera, E. Wen, and S. Nanayakkara (2021). “Fıne-Tune The Entıre Rag Archıtecture (Includıng Dpr Retrıever) For Questıon-Answerıng”, <https://arxiv.org/pdf/2106.11517v1.pdf>