

School of Information Technology and Engineering

**Spam Mail Detection Using CNN**

**Project Report**

J Component of ITA6004 – Soft Computing

***in***

**M.C.A**

*by*

**Pritam Mondal - 22MCA0129**

**Arya Panja – 22MCA0355**

*Under the guidance of*

**Dr. Anitha A**

**SITE**

**JUNE 2023**

**Contents**

|  |  |  |
| --- | --- | --- |
| **No.** | **Content** | **Page No** |
| 01 | Abstract | 02 |
| 02 | Introduction | 03 |
| 03 | Related Work | 04 |
| 04 | Background Fundamentals | 04 |
| 05 | Proposed Methodology | 05 |
| 06 | Experimental Analysis | 06 |
| 07 | Results | 07 |
| 08 | Comparative analysis with benchmarking techniques | 10 |
| 09 | Conclusion | 10 |
| 10 | References | 11 |

**Abstract**

Emails are currently the most widely used means of communication, and as the number of users utilizing these platforms continues to grow, so does the prevalence of spam. Spam refers to any unwanted and unsolicited digital communication that is sent out in large quantities. Spam emails contribute to a significant waste of resources by needlessly overwhelming network links. While most spam is generated by advertisers attempting to promote their products, some are more malicious in nature, such as phishing emails that aim to deceive recipients into divulging sensitive information like website login credentials or credit card details. This form of cybercrime is commonly known as phishing. To combat the issue of spam, numerous research efforts have been undertaken to develop spam detection systems capable of identifying and filtering out spam messages and emails. In this research we build an email spam detector using BERT pre-trained model that outperforms traditional machine learning algorithms and achieves a high level of accuracy. The proposed approach involves pre-processing the email text data through tokenization and encoding before feeding it into the BERT model, the model effectively learns discriminative features specific to spam emails.

**Introduction**

Email is an indispensable communication tool used across various domains, including business, education, and personal use. To ensure effective communication, spam detection plays a crucial role in enhancing user experience and security. Spam emails, commonly referred to as junk mail, are sent to many recipients simultaneously and often contain suspicious content, scams, or malicious phishing attempts. A spam detector, illustrated in Figure 1, is a software program designed to identify and prevent unsolicited, unwanted, and virus-infected emails from reaching a user's inbox.

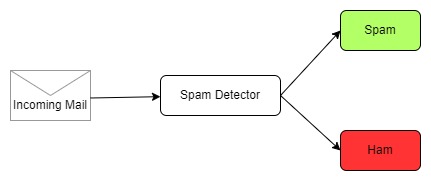


Fig 1: Spam Detector

Another approach for spam detection involves filtering emails based on their content. In this methodology, the spam detector examines the content of each email and categorizes it accordingly. This type of spam detection method exhibits high performance because spam emails often exhibit predictable patterns, such as offering deals, promoting explicit content, or targeting basic human emotions like desire and fear. Alternatively, a rule-based email filtering approach allows users to manually configure specific rules for incoming emails, classifying them as spam or not. These rules can be based on the sender's identity, email subject containing certain phrases, or specific words within the message. When an incoming email match one of these rules, it is automatically redirected to a designated spam folder.

Spam detection can be implemented using two approaches: machine learning-based and non-machine learning-based. Non-machine learning approaches involve manually constructing document classifiers with rules created by domain experts. However, the drawback of non-statistical approaches is the lack of a learning component, which may result in legitimate messages being misclassified as spam, leading to lower accuracy.

To overcome these limitations, researchers have turned to machine learning algorithms to develop their spam detectors. By training the spam detector model on a dataset, it can predict spam emails that exhibit similar patterns to those in the training data. The most common type of spam emails comprises advertising messages, accounting for 36% of all spam content worldwide.

**Related Work**

Janez-Martin et al [1] made the combined model of TF-IDF and SVM showed 95.39% F1score and the fastest spam classification achieved with the help of the TF-IDF and NB approach. Faris et al [2] proposed a Particular swarm optimization PSO-based Wrapper with a Random Forest algorithm that effectively detects spam messages. Marie-Saint et al [3] introduced Deep Learning-based approaches Deep learning mimics the human brain to solve the given the firefly algorithm with SVM and worked with Arabic text. This article concluded the proposed method outperformed SVM alone. [4] Built a spam detection model for Chinese, their model based on long-short attention mechanism that convert words to vectors based on the context of the sentence, they trained the model using Trec06C which is a Chinese spam dataset, their model achieved high performance with accuracy of 99.3, [5] Use modified Transformer model for SMS spam detection, they modified Transformer by adding memory and a linear layer with a final activation function that takes the output of Transformer model as an input to the final classification layer, they trained their model using two datasets, the first dataset is SMS Spam Collection v.1 [6] and the other dataset is UtkMl’s Twitter Spam Detection Competition [7] from Kaggle, the model f1-score was 98.92 for SMS dataset and 94.51 for twitter dataset [8] use fine tune BERT for spam detection task, they used various datasets to train and evaluate the model, model performance for each dataset was as follows: Spam Assassin dataset with f1-score of 97, Enron dataset achieved f1-score of 97, Ling Spam dataset f1-score was 94 and Spam Text dataset f1-score was 96.

**Background Fundamentals**  
This paper aims to develop an effective approach for identifying and classifying spam emails using advanced natural language processing techniques. The paper focuses on gathering and pre-processing a comprehensive dataset of emails, explaining the BERT model's architecture and its suitability for language understanding and representation. It discusses the extraction of features from email data using BERT embeddings and evaluates the model's performance through training and evaluation processes. The results and analysis of the experiments are presented, highlighting the strengths and limitations of using the BERT model for spam email classification. The paper concludes by discussing the implications of the research findings, potential future improvements, and the impact of the study in real-world applications.

The research aims to develop a spam mail detection system using the BERT model. The objectives include evaluating its performance compared to existing methods, analysing the impact of pre-training strategies, investigating robustness against attacks, and proposing improvements. The hypotheses suggest that the BERT model will outperform traditional techniques, benefit from fine-tuning and domain-specific pre-training, exhibit resilience against attacks, improve with ensemble methods, and excel in detecting complex spam patterns.

The theoretical framework for spam mail detection using the BERT model involves three key components: natural language processing (NLP), machine learning, and the BERT model itself. NLP techniques are employed to extract relevant features from spam emails, while machine learning algorithms classify emails as spam or non-spam. The BERT model, a powerful language representation model, captures contextualized word representations and improves classification accuracy. By integrating NLP, machine learning, and BERT, the framework enables effective identification and classification of spam emails based on their linguistic patterns and contextual understanding.

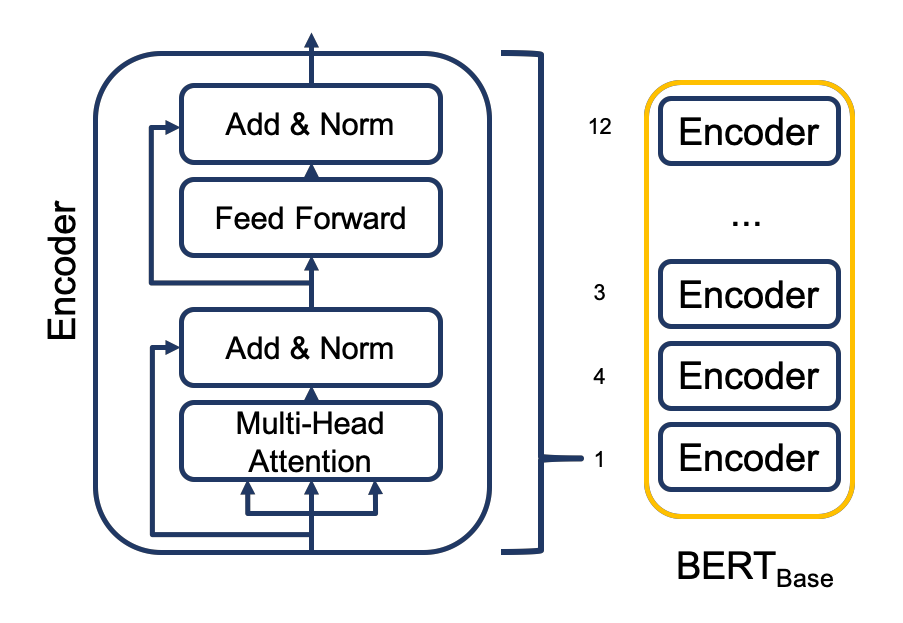


Fig 2. BERT Architecture

**Proposed Methodology**

We propose a novel model for spam mail detection that harnesses the capabilities of BERT (Bidirectional Encoder Representations from Transformers). Proposed model follows a sequential architecture comprising BERT preprocessing, a BERT base layer, a dropout-dense layer with ReLU activation, and a final dropout dense layer with sigmoid activation. Initially, text mails are tokenized and converted into a vector representation using BERT preprocessing. This tokenized vector is then fed into the BERT base layer, which captures contextual information and semantic meaning. A subsequent dropout-dense layer with ReLU activation facilitates feature extraction and pattern recognition. Finally, a dropout dense layer with sigmoid activation produces a binary output representing the probability of the mail being classified as spam. By leveraging BERT's contextual understanding and utilizing dropout-dense layers, the model aims to outperform traditional spam detection methods in accurately identifying spam mails.

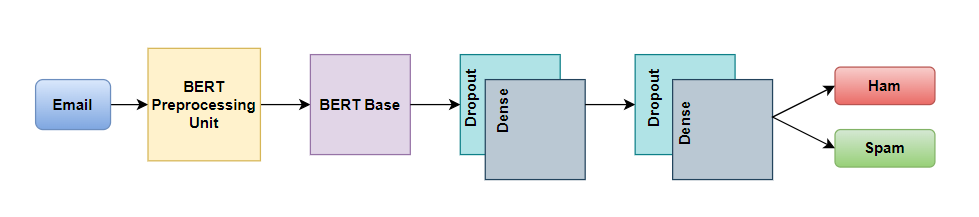


Fig 03. Model Flow Chart

BERT base consists of a stack of 12 encoders from Transformer model above each other’s and a hidden size of 768. Transformer encoders are all identical in structure and consists of a self-attention layer followed by a feed forward neural network layer, the self-attention layer is the main part of BERT and also Transformer is responsible for making the encoder while encoding a word to take in account the other words in the input sentence, so the word embeddings will be based on the sentence context, the output of the self-attention is then fed to a feed forward neural network layer to process the output from one attention layer in a way to better fit the input for the next attention layer.

To prepare text for BERT, it undergoes tokenization using Word Piece, where words are broken down into subworld units. Special tokens like [CLS] for classification and [SEP] for sentence separation are added to the tokenized input. Segment IDs are assigned to differentiate between different segments or sentences in the input. During pretraining, BERT is trained on a large corpus of unlabelled text using masked language modelling (MLM) and next sentence prediction (NSP) objectives. MLM involves randomly masking tokens and predicting them based on the context, while NSP predicts if two sentences are consecutive or not. After pretraining, BERT can be fine-tuned for spam mail detection, by adding two dropout-dense layers and training on labelled data. The activation function used in the output layer is sigmoid, which produces a binary output representing the probability of the input mail being classified as spam.

**Experimental Analysis**

**Dataset**

The dataset is downloaded from <https://www.kaggle.com/> containing 4500 ham emails and 1000 spam emails. The dataset has 2 columns named category and message.

**Python Code**

# !pip3 install tensorflow-text

import tensorflow as tf

import tensorflow\_hub as hub

import tensorflow\_text as text

from google.colab import drive

drive.mount('/content/drive')

import pandas as pd

df = pd.read\_csv("/content/drive/MyDrive/mail\_data.csv")

df.head(5)

#we change spam to 1 and ham to 0

df['spam'] = df['Category'].apply(lambda x: 1 if x == 'spam' else 0)

df.head()

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split( df['Message'], df['spam'], stratify=df['spam'])

#importing bert model

bert\_preprocess = hub.KerasLayer(

"https://tfhub.dev/tensorflow/bert\_en\_uncased\_preprocess/3")

bert\_encoder = hub.KerasLayer(

"https://tfhub.dev/tensorflow/bert\_en\_uncased\_L-12\_H-768\_A-12/4")

# model

text\_input = tf.keras.layers.Input(shape=(), dtype=tf.string, name='text')

preprocessed\_text = bert\_preprocess(text\_input)

outputs = bert\_encoder(preprocessed\_text)

# Neural network layers

l = tf.keras.layers.Dropout(0.2, name="dropout")(outputs['pooled\_output'])

l = tf.keras.layers.Dense(256, activation='relu',

name="dense")(l) # Added dense layer

l = tf.keras.layers.Dropout(0.2, name="dropout2")(l) # Added dropout layer

l = tf.keras.layers.Dense(1, activation='sigmoid', name="output")(l)

# Use inputs and outputs to construct a final model

model = tf.keras.Model(inputs=[text\_input], outputs=[l])

model.compile(optimizer='adam',

loss='binary\_crossentropy',

metrics=['accuracy'])

#accuracy on test data

model.evaluate(X\_test, y\_test)

print(X\_test)

#consufion matrix

y\_pred = model.predict(X\_test)

y\_pred = (y\_pred > 0.5)

from sklearn.metrics import confusion\_matrix

cm = confusion\_matrix(y\_test, y\_pred)

cm

#heatmap

import seaborn as sns

sns.heatmap(cm, annot=True, fmt='.0f')

def predict\_mail(mail\_string):

mail\_string = [mail\_string]

mail\_string = tf.convert\_to\_tensor(mail\_string)

mail\_string = tf.reshape(mail\_string, [-1])

pred = model.predict(mail\_string)[0][0]

if pred >0.5:

return 'SPAM'

else:

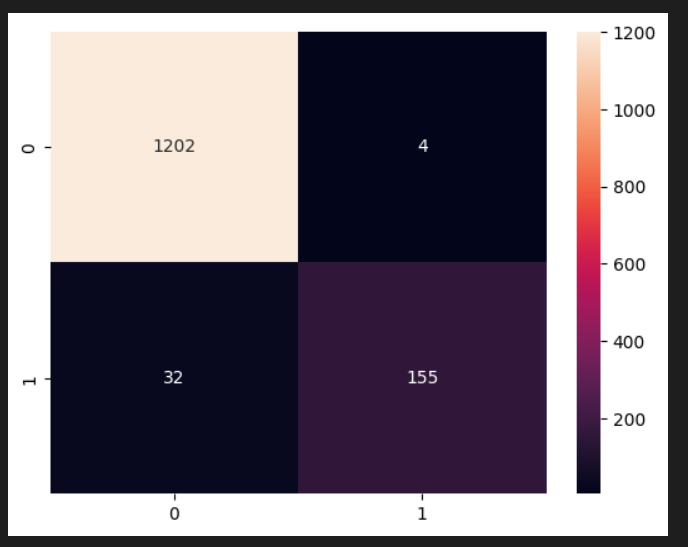
return 'HAM'

predict\_mail(" ")

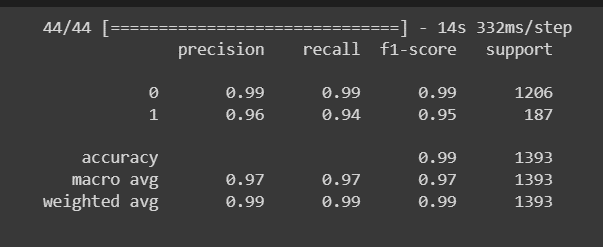
**Results**

After training our model with our dataset we got an accuracy of 98.64%. Our model achieved an overall accuracy of 99%, correctly classifying the majority class (0) with a precision, recall, and F1-score of 99%. The minority class (1) had a slightly lower performance with a precision of 96% and recall of 94%. The macro average of precision, recall, and F1-score across both classes is 97%. The weighted average takes into account the class distribution and is also 99%. Overall, the model performed very well, accurately predicting the majority class while still achieving a high level of performance for the minority class.

**Confusion Matrix**

  
  
Fig 04. Confusion Matrix

**Classification report**

  
Fig 05. Classification report

**Comparative analysis with benchmarking techniques**

Our model achieved the highest accuracy of 98.63% on a Kaggle dataset, while BERT-NB achieved 61.4% accuracy on the SPEMC-E dataset, and TF-IDF-LR achieved 94.6% accuracy on the same SPEMC-E dataset. It's important to consider the specific characteristics of each dataset, the pre-processing steps applied, the feature representation techniques used, and the mode. It could be seen that the pre-processing was poor in their model which was rectified in ours.

Our model, trained on a Kaggle dataset, achieved a higher accuracy of 98.63% compared to the BERT-based model's accuracy of 97.30% on the Spam base dataset. Our model's better pre-processing techniques contributed to its superior performance.

Our model achieved a higher accuracy of 98.63% on a Kaggle dataset, while the SVM model with "tok1" tokenization achieved an accuracy of 97.64% [tok1: tokens start with a printable character, followed by any number of alphanumeric characters, excluding dots, commas, and colons from the middle of the pattern. With this pattern, domain names and mail addresses will be split at dots, so the classifier can recognize a domain even if subdomains vary].

|  |  |  |
| --- | --- | --- |
| Model Used | Dataset | Accuracy |
| BERT-NB  TF-IDF-LR | SPEMC-E | 61.4 %  94.6 % |
| BERT based cased | Spam base data set (5000) | 97.30% |
| Spam Transformer | UtkMl’s dataset | 87.06% |
| SVM + tok1 | NUS SMS Corpus | 97.64% |
| Our model (BERT) | Dataset from Kaggle (5573) | 98.64% |

Table 01. Comparative Study

**Conclusion**

In conclusion, utilizing the BERT model for spam mail detection has shown promising results. Future research should focus on refining the BERT-based system through different fine-tuning and pre-training strategies, enhancing its resilience against adversarial attacks, and considering real-time implementation. Additionally, exploring multi-lingual spam detection and scalability of the system are important areas to explore. These efforts will contribute to the development of more effective and efficient spam mail detection systems in the future.

**References**

1. F. Jáñez-Martino, E. Fidalgo, S. González-Martı́nez and J. VelascoMata, "Classification of Spam Emails through Hierarchical Clustering and Supervised Learning," CoRR, vol. abs/2005.08773, 2020.
2. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser and I. Polosukhin, "Attention Is All You Need," 6 2017.
3. S. L. Marie-Sainte and N. Alalyani, "Firefly Algorithm based Feature Selection for Arabic Text Classification," J. King Saud Univ. Comput. Inf. Sci., vol. 32, pp. 320-328, 2020.
4. X. Tong, J. Wang, C. Zhang, R. Wang, Z. Ge, W. Liu and Z. Zhao, "A Content-Based Chinese Spam Detection Method Using a Capsule Network with Long-Short Attention," IEEE Sensors Journal, vol. 21, pp. 25409-25420, 2021.
5. X. Liu, H. Lu and A. Nayak, "A Spam Transformer Model for SMS Spam Detection," IEEE Access, vol. 9, pp. 80253-80263, 2021.
6. T. A. Almeida, J. M. G. Hidalgo and A. Yamakami, "Contributions to the Study of SMS Spam Filtering: New Collection and Results," in Proceedings of the 11th ACM Symposium on Document Engineering, New York, NY, USA, 2011.
7. Kaggle, UtkMl’s Twitter Spam Detection Competition.
8. Techniques Isra’a AbdulNabi∗, Qussai Yaseen, Spam Email Detection Using Deep Learning, March 23 - 26, 2021, Warsaw, Poland.