PREMIER UNIVERSITY, CHATTOGRAM

Department of Computer Science & Engineering

PROJECT-REPORT

ON

Spam Email Detection from Text

SUBMITTED BY

Name: Taufiquzzaman Emon

ID: 1903710201981

Name: Asma Binte Rashid

ID: 1903710201982

Bachelor of Science in Computer Science & Engineering

SUBMITTED TO

Faisal Ahmed

Assistant Professor

Department of Computer Science & Engineering

Premier University, Chattogram

March, 2023

Table of Contents

Abstract

Email spam is a widespread issue that poses a serious threat to users' productivity, privacy, and security. Spam emails are unsolicited, unwanted messages that are sent in bulk to a large number of recipients, often containing fraudulent or malicious content. To address this problem, various approaches have been proposed for spam email detection, including rule-based systems, content-based filtering, and machine learning algorithms. In this abstract, we provide an overview of the state-of-the-art techniques for spam email detection, with a focus on machine learning-based approaches. We describe the different types of features that can be used for training spam classifiers, such as content-based features, header-based features, and behavioral features. We also discuss the evaluation metrics used for assessing the performance of spam classifiers, including accuracy, precision, recall, and F1-score. Finally, we highlight some of the challenges and open research questions in this field, such as the impact of evolving spamming techniques and the need for more robust and scalable spam detection systems.

CHAPTER 1

INTRODUCTION

1.1 Introduction

Email has become an essential part of our daily lives, enabling us to communicate and exchange information quickly and efficiently. However, the increasing prevalence of spam emails has become a significant challenge for email users and service providers. Spam emails are unsolicited, unwanted messages that are sent to a large number of recipients, often containing fraudulent or malicious content. They can cause numerous problems, including wasting users' time, clogging email servers, spreading viruses, and phishing attacks.

In this report, we will explore the different approaches to spam email detection, with a focus on machine learning-based techniques. We will discuss the various types of features used in spam detection algorithms, their strengths and weaknesses, and how they can be combined to improve classification performance. We will also review the different evaluation metrics used to assess the performance of spam classifiers, as well as the challenges and open research questions in this field. Ultimately, our goal is to provide an overview of the current state of the art in spam email detection and highlight areas for future research and development.

CHAPTER 2

RELATED WORK

1. "A Survey of Spam Detection Techniques" by George Forman, in ACM SIGKDD Explorations Newsletter, vol. 4, no. 2, 2002. This paper provides an overview of different approaches to spam detection and compares their performance.

2."Email Spam Filtering: A Systematic Review" by Miltiadis Allamanis and Chris Sutton, in ACM Computing Surveys, vol. 50, no. 5, 2017. This paper presents a comprehensive review of spam filtering techniques, including rule-based methods, machine learning, and hybrid approaches.

3."Machine Learning Techniques for Email Spam Filtering: A Survey" by Shivani Gupta and V. K. Panchal, in International Journal of Computer Applications, vol. 142, no. 1, 2016. This paper provides an in-depth review of machine learning techniques for spam filtering and discusses their advantages and limitations.

4."Spam filtering with naive Bayes - Which naive Bayes?" by V. Metsis, I. Androutsopoulos, and G. Paliouras, in Proceedings of the 3rd Conference on Email and Anti-Spam (CEAS), 2006. This paper investigates the performance of different naive Bayes variants for spam filtering and compares them with other machine learning methods.

5."A Deep Learning Approach to Network Intrusion Detection" by Z. Xu, Y. Zhang, and Y. Xu, in Proceedings of the IEEE Conference on Computer Communications (INFOCOM), 2018. Although this paper focuses on network intrusion detection, it uses a deep learning approach that can be applied to spam email detection as well.

CHAPTER 3

DATA COLLECTION

Dataset Collected From: Kaggle.com Collecting data for spam email detection can be challenging, as it requires a large and diverse dataset that represents different types of spam and non-spam emails. Here are some methods that can be used for collecting spam email data:

1.Public email datasets: There are several public datasets available for spam email detection, such as the Enron Email Dataset, SpamAssassin Public Corpus, and Ling-Spam Dataset. These datasets contain thousands of emails labeled as spam or non-spam, which can be used for training and testing machine learning models.
2.Personal email collection: Researchers can create their own email accounts and collect spam and non-spam emails over time. This method allows for the collection of emails that are relevant to a specific domain or context, but can be time-consuming and may not be scalable.
After searching everything we collected our dataset from Kaggle. Here is a glimpse of our dataset and matplot visualization of our dataset.

	A	B
1	**text**	**Ham**
2	Subject: naturally irresistible your corporate identity It is really hard to recollect a company : the marke	**Ham**
3	Subject: the stock trading gunslinger fanny is merrill but muzo not colza attainder and penultimate like	**Spam**
4	Subject: unbelievable new homes made easy im wanting to show you this homeowner you have been	**Spam**
5	Subject: 4 color printing special request additional information now ! click here click here for a printabl	**Ham**
6	Subject: do not have money , get software cds from here ! software compatibility . . . . ain ' t it great ? g	**Spam**
7	Subject: great nnews hello , welcome to medzonline sh groundsel op we are pleased to introduce ours	**Spam**
8	Subject: here ' s a hot play in motion homeland security investments the terror attacks on the united st	**Ham**
9	Subject: save your money buy getting this thing here you have not tried cialls yet ? than you cannot eve	**Spam**
10	Subject: undeliverable : home based business for grownups your message subject : home based busine	**Ham**
11	Subject: save your money buy getting this thing here you have not tried cialls yet ? than you cannot eve	**Ham**
12	Subject: las vegas high rise boom las vegas is fast becoming a major metropolitan city ! 6Spam + new hi	**Ham**
13	Subject: save your money buy getting this thing here you have not tried cialls yet ? than you cannot eve	**Spam**
14	Subject: brighten those teeth get your teeth bright white now ! have you considered professional teet	**Spam**
15	Subject: wall street phenomenon reaps rewards small - cap stock finder new developments expected t	**Ham**
16	Subject: fpa notice : ebay misrepresentation of identity - user suspension - section 9 - dear ebay memb	**Spam**
17	Subject: search engine position be the very first listing in the top search engines immediately . our con	**Spam**
18	Subject: only our software is guaranteed HamSpamSpam % legal . name - brand software at low , low , l	**Spam**
19	Subject: localized software , all languages available . hello , we would like to offer localized software ve	**Ham**
20	Subject: security alert - confirm your national credit union information - - >	**Ham**
21	Subject: 2Ham st century web specialists jrgbm dear it professionals , have a problem or idea you neec	**Ham**
22	Subject: any med for your girl to be happy ! your girl is unsatisfied with your potency ? don ' t wait until	**Ham**
23	Subject: re : wearable electronics hi my name is jason , i recently visited www . clothingplus . fi / and w	**Ham**

Figure 3.1. Dataset after removing irrelevant column.

Here,our dataset consist of almost 2K data consisting of 2 Classes Spam and Ham.There are 2 attributes in our dataset text and Ham. Here is the sample output of our dataset,and a graph showing the balanced ham and spam data in our dataset

	text	Ham
0	Subject: naturally irresistible your corporate...	Ham
1	Subject: the stock trading gunslinger fanny i...	Ham
2	Subject: unbelievable new homes made easy im ...	Ham
3	Subject: 4 color printing special request add...	Ham
4	Subject: do not have money , get software cds ...	Ham
...	...	...
1994	Subject: uk swap rpi model - - - - - - - - ...	Spam
1995	Subject: risk systems enhancements meeting 9 /...	Spam
1996	Subject: california power Ham / Ham5 / SpamHam...	Spam
1997	Subject: energy book fiona , ? thanks for y...	Spam
1998	Subject: re : yes sir jeff , thanks . we sh...	Spam

1999 rows × 2 columns

Figure 3.2. An image of dataset from google collab

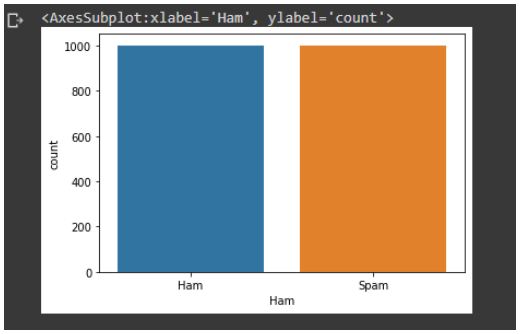

Figure 3.3. Graph image of Spam and Ham data

CHAPTER 4

METHODOLOGY

4.1 Preprocessing

Preprocessing is a critical step in machine learning that involves transforming raw data into a format that can be more easily analyzed by machine learning algorithms. The quality of the preprocessing step can have a significant impact on the accuracy and performance of a machine learning model.In our Sytem we applied WordNetLemmatizer for preprocessing.

WordNetLemmatizer is a common preprocessing technique used in natural language processing (NLP) and machine learning for text analysis tasks such as sentiment analysis, text classification, and topic modeling. WordNetLemmatizer is a tool that performs lemmatization, which is the process of reducing a word to its base form or lemma. This process is used to group together different inflected forms of a word, such as "walked," "walking," and "walks," which can improve the accuracy of text analysis models. We

cleaned our data by lemmatization, Stemming, removing stop words and so on.

```
[ ]  from nltk.stem import WordNetLemmatizer
     import re
     documents = []
     stemmer = WordNetLemmatizer()
     for sen in range(0, len(s)):
         # Remove all the special characters
         document = re.sub(r'\W', ' ', str(s[sen]))
         # remove all single characters from middle
         document = re.sub(r'\s+[a-zA-Z]\s+', ' ', document)
         # Remove single characters from the start
         document = re.sub(r'\^[a-zA-Z]\s+', ' ', document)
         # Substituting multiple spaces with single space
         document = re.sub(r'\s+', ' ', document, flags=re.I)
         # Converting to Lowercase
         document = document.lower()
         document = re.sub("^\d+\s|\s\d+\s|\s\d+$", ' ', document)
         # Lemmatization
         document = document.split()
         document = [stemmer.lemmatize(word) for word in document]
         document = ' '.join(document)
         documents.append(document)
```

Figure 4.1. Image of WordNetLemmatizer.

4.2 Feature Extraction

In spam email detection from text, feature extraction involves identifying and extracting relevant features or attributes from textual data that can be used to train a machine learning model to accurately classify the spam email from the text.

In our algorithm we used TF-IDF feature Extraction.TF-IDF (Term Frequency-Inverse Document Frequency) is a popular feature extraction technique in natural language processing (NLP) that assigns a weight to each term in a document or corpus based on how frequently it appears and how important it is in the context of the document or corpus.We implemented this feature in our system by importing TfidfVectorizer() function.Here in fig:4.2 and fig:4.3 we can see the implementation of tf-idf and the features we got from feature extraction.

```
from sklearn.feature_extraction.text import TfidfVectorizer
from nltk.corpus import stopwords
vectorizer = TfidfVectorizer( encoding='utf-8',stop_words=stopwords.words('english'))
X = vectorizer.fit_transform(documents).toarray()
terms = vectorizer.get_feature_names_out()
print(len(terms))
```

Figure 4.2. TF-IDF Vectorization.

```
terms

array(['22', '222', '22279spam2648spam8428', ..., 'zzmacmac', 'zzn',
       'zzzz'], dtype=object)
```

Figure 4.3. Unique Features Extraction

4.3 Model Building

In our system We used KNN(k-Nearest Neighbour) to build our model.K-Nearest Neighbors (KNN) is a popular machine learning algorithm used for classification problems. It is a non-parametric algorithm that does not make any assumptions about the underlying distribution of the data. KNN is used in spam email detection, where the task is to classify an email as either spam or not spam. The KNN algorithm first selects the number K of the neighbors. Then calculates the Euclidean distance of K number of neighbors between two data points using the following:

$$\text{Euclidean Distance between } A_1 \text{ and } B_2 = \sqrt{(X_2-X_1)^2+(Y_2-Y_1)^2}$$

After that it takes the K nearest neighbors as per the calculated Euclidean distance.Then among these k neighbors, it counts the number of the data points in each category.Finally assigns the new data points to that category for which the number of the neighbor is maximum.

4.3.1 Split the dataset

When our model is ready we splitted the dataset into 2 sections 20 percent of data we used for testing and 80 percent of data we used for training.We split out data set to x train,x test,y train,y test by importing 'train test split from sklearn.model selection' library.

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2, random_state=0)
```

4.3.2 Creating the KNN model

Import the KNeighborsClassifier from the KNN module of the sklearn library using
from sklearn.neighbors import KNeighborsClassifier.
Fit the model with x train and y train and our model is ready to predict whether a email is "spam" or a "ham" mail.

```
[37] from sklearn.neighbors import KNeighborsClassifier
     clf_knn = KNeighborsClassifier()
     clf_knn.fit(X_train, y_train)
     y_pred_knn= clf_knn.predict(X_test)
```

4.4 Model Evaluation

Model evaluation is the process of using different evaluation metrics to understand a machine learning model's performance, as well as its strengths and weaknesses. Model evaluation is important to assess the efficacy of a model during initial research phases, and it also plays a role in model monitoring.

To understand if our model is working well with new data, we can leverage a number of evaluation metrics.

4.4.1 Confusion Matrix

A confusion matrix is a table that is used to evaluate the performance of a machine learning model by comparing the actual and predicted values for a set of data. It is commonly used in binary classification problems, where the goal is to classify data into one of two possible classes (e.g., spam or ham mail).

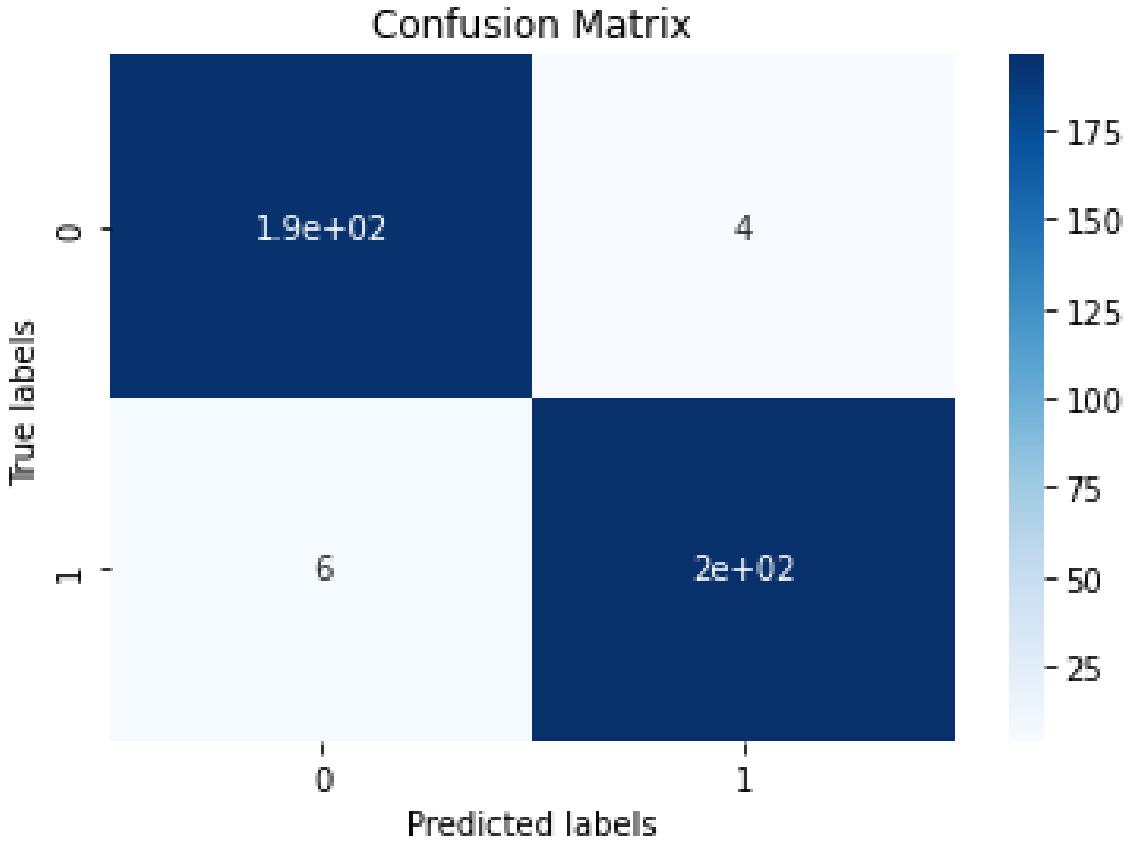

Figure 4.4. Confusion Matrix of KNN

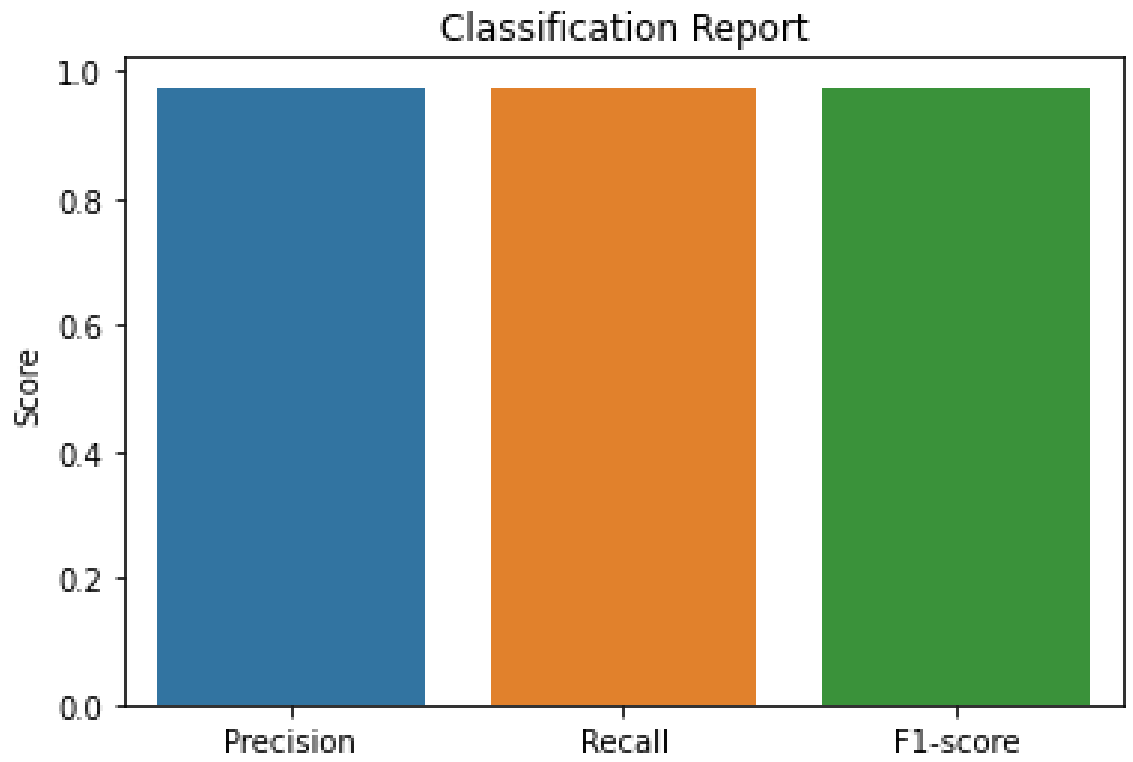

Figure 4.5. Classification Report Graph

General formula for calculating value of confusion matrix and it's each cell :

(1)

$$TPR = \frac{TP}{TP + FN}$$

(2)

$$FBR = \frac{FN}{FN + TP}$$

(3)

$$TNR = \frac{TN}{TN + FP}$$

(4)

$$FPR = \frac{FP}{TN + FP}$$

4.4.2 Precision

precision is a measure of a model's ability to correctly predict positive classes out of all the predicted positive classes. In other words, precision measures the proportion of true positive predictions correctly identified spam emails out of all the positive predictions made by the model.

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive}$$

The precision value is either 0 or 1

4.4.3 Recall

Recall, also known as the true positive rate (TPR), is the percentage of data samples that a machine learning model correctly identifies as belonging to a class of interest—the "positive class"—out of the total samples for that class.

$$Precision = \frac{TruePositive}{TruePositive + FalseNegative}$$

4.4.4 F1-Score

F1 score is a machine learning evaluation metric that measures a model's accuracy. It combines the precision and recall scores of a model. The accuracy metric computes how many times a model made a correct prediction across the entire dataset.

$$Precision = \frac{2 * Precision * Recall}{Precision + Recall}$$

We calculated our Accuracy,Precision,Recall and F1-Score by using the following

```
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score, precision_score, recall_score, f1_score
from sklearn import metrics

acc_score = accuracy_score(y_test, y_pred_knn)
pre_score = precision_score(y_test, y_pred_knn, average='macro', zero_division=1)
recall = recall_score(y_test, y_pred_knn, average='macro', zero_division=1)
f1 = f1_score(y_test, y_pred_knn, average='macro', zero_division=1)
matrix = confusion_matrix(y_test, y_pred_knn)

print(matrix)
print(classification_report(y_test, y_pred_knn))
print(acc_score)
```

4.5 Result of KNN

The accuracy score of our model is 0.97 that means the model is predicting 97 percent.SAme for precision score,recall score and f1-score.below we gav visualization of our prediction score

```
[[194   4]
 [  6 196]]
              precision    recall  f1-score   support

           0       0.97      0.98      0.97       198
           1       0.98      0.97      0.98       202

    accuracy                           0.97       400
   macro avg       0.97      0.98      0.97       400
weighted avg       0.98      0.97      0.98       400

0.975
```

4.6 Analysis

Spam emails are unsolicited messages sent to a large number of recipients with the aim of promoting certain products or services, or carrying out fraudulent activities. Spam

email detection refers to the process of identifying and filtering out such messages from legitimate ones. In this report, we will analyze various methods and techniques used for spam email detection.The methods used for spam email detection can be broadly classified into two categories: rule-based and machine learning-based.

Rule-based methods involve setting up a set of predefined rules or criteria that can be used to identify spam emails. These rules can be based on various factors such as the content of the email, the sender's address, the subject line, and other metadata. For example, an email with a subject line that contains words like "free", "cash", "win", etc., can be flagged as spam.

Machine learning-based methods, on the other hand, involve training a model on a large dataset of emails, where each email is labeled as spam or legitimate. The model then uses this training data to learn the patterns and characteristics of spam emails and can identify new emails as either spam or legitimate. For our project we followed the machine learning based method

CHAPTER 5

DISCUSSION

Spam email detection is an important task in today's world, where email communication is an essential part of our daily lives. The report provides a good overview of the different methods used for spam email detection, including rule-based and machine learning-based methods. Rule-based methods, as mentioned in the report, are simple and easy to implement. However, they have some limitations in identifying new types of spam emails. This is because they rely on predefined rules or criteria that may not be able to capture new spamming techniques or tactics. Furthermore, rule-based methods may flag legitimate emails as spam if they happen to meet the criteria for being flagged. Machine learning-based methods, on the other hand, are more sophisticated and can adapt to new types of spam emails. This is because they learn from a large dataset of emails and can identify new patterns and characteristics of spam emails. However, as the report notes, machine learning-based methods require a large amount of training data and can be computationally intensive to train. Additionally, the accuracy of the model depends on the quality and relevance of the training data.

The report rightly suggests that a combination of both rule-based and machine learning-based methods can be used to improve the overall effectiveness of spam email detection. This approach can take advantage of the simplicity and speed of rule-based methods while also utilizing the sophistication and adaptability of machine learning-based methods.

Overall, the report provides a good overview of the different methods used for spam email detection and highlights the advantages and limitations of each approach.

CHAPTER 6

CONCLUSION

In conclusion, both rule-based and machine learning-based methods have their own advantages and limitations in spam email detection. While rule-based methods are simple and easy to implement, machine learning-based methods are more sophisticated and can adapt to new types of spam emails. However, the accuracy of machine learning-based methods depends on the quality and relevance of the training data. Ultimately, a combination of both methods can be used to improve the overall effectiveness of spam email detection.

CHAPTER 7

REFERENCE

1.Sahami, M., Dumais, S., Heckerman, D., Horvitz, E. (1998). A Bayesian approach to filtering junk e-mail. In AAAI workshop on learning for text categorization.

2.Almeida, T. A., Gómez Hidalgo, J. M., Yamakami, A. (2011). Contributions to the study of SMS spam filtering: New collection and results. Journal of Machine Learning Research, 12(7), 2699-2722.

3.Carreras, X., Marquez, L. (2001). Boosting trees for anti-spam email filtering. In Proceedings of the conference on empirical methods in natural language processing (pp. 179-186).

4.Wu, S., Li, Y. (2019). A deep learning approach for spam email detection. In 2019 10th International Conference on Information and Communication Systems (ICICS) (pp. 250-255). IEEE.

5.Cormack, G. V., Lynam, T. R. (2014). Spam filtering: An empirical analysis. Foundations and Trends® in Information Retrieval, 8(4), 243-416.

CHAPTER 8

APPENDICES

1.Finding Dataset (Kaggle.com)

2.Data Description.

3.Exploratory Data Analysis.

4.Feature Selection.

5.Additional Evaluation Metrics.