National College of Ireland

Spam Detection in Short Message Service Using Natural Language Processing and Machine Learning Techniques

MSc Research Project
Data Analytics

Anchal Ora
Student ID: x18135846

School of Computing
National College of Ireland

Supervisor: Dr. Catherine Mulwa

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Anchal Ora
Student ID:	x18135846
Programme:	Data Analytics
Year:	2019
Module:	MSc Research Project
Supervisor:	Dr. Catherine Mulwa
Submission Due Date:	12/12/2019
Project Title:	Spam Detection in Short Message Service Using Natural Language Processing and Machine Learning Techniques
Word Count:	
Page Count:	27

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	28th January 2020

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Spam Detection in Short Message Service Using Natural Language Processing and Machine Learning Techniques

Anchal Ora
x18135846

Abstract

As the usage of mobile phones increased, the use of Short Message Service increased significantly. Due to the lower costs of text messages, people started using it for promotional purposes and unethical activities. This resulted in the ratio of spam messages increasing exponentially and thereby loss of personal and financial data. To prevent data loss, it is crucial to detect spam messages as quick as possible. Thus, the research aims to classify spam messages not only efficiently but also with low latency. Different machine learning models like XGBoost, LightGBM, Bernoulli Naïve Bayes that are proven to be very fast with low time complexity have been implemented in the research. The length of the messages was taken as an additional feature, and the features were extracted using Unigram, Bigram and TF-IDF matrix. Chi-Square feature selection was implemented to further reduce the space complexity. The results showcased that Bernoulli Naïve Bayes followed by LightGBM with the TF-IDF matrix generated the highest accuracy of 96.5% in 0.157 seconds and 95.4% in 1.708 seconds respectively.
Keywords: Spam SMS, Text Classification, Natural Language Processing, Machine Learning, Bernoulli Naïve Bayes, LightGBM, XGBoost

1 Introduction

Short Messaging Service (SMS) is mainly used for unofficial communication such as promoting new products and services but at times also used for official communication like information about any bank transaction or confirmation of the order on an online portal etc. Due to advancements in technology, the costs of sending an SMS have reduced drastically. This has proved to be a boon for some whereas a bane for many. People are misusing the SMS facility to promote products, services, offers and schemes and so on. How annoying this has become can be assessed by the fact that people have started ignoring SMS they receive because twenty to thirty percent of the total SMS received is spam (Kim et al. (2015)). This menace is growing at a rapid rate. As a result, people miss out on genuine informative messages such as bank transactions. At times the ignorance towards SMS can prove detrimental because some fraud transactions might have been performed, but the information was neglected. The motive behind this project is to apply machine learning algorithms to separate spam messages from genuine ones. Machine learning techniques along with Natural Language Processing techniques was used to make the process more agile and efficient.

1.1 Motivation and Background

For today's generation, usage of phones is not just confined to communication now but an array of different uses such as storing personal information like documents, notes, media, making financial transactions, shopping, etc. Owing to a wide range of information stored on devices some of which are personal and critical, hacking phones is of utmost interest to people having unethical intentions. SMS is an easy way to target people because it is used by people from all walks of life and all ages and they are not aware of the implications. Hackers access phone and all its information when a phone is hacked, and people have absolutely no idea about it. Consequently, there is a loss of critical data that can be exploited for illegal purposes. It can be traumatic for the victims causing psychological despair and financial loses.

Messages are not just written in English but in various languages and those in English can also have words and abbreviations which are from other languages. Hence identifying spam SMS is a challenging work as stated by (Yadav et al. (2012)) since it does not have any header like in case of emails. So, the techniques used to figure out spam emails cannot be used for messages. New solutions must be devised for new problems and thus many researchers have been working in this area to develop new algorithms and techniques. In a recently concluded research by (Gupta et al. (2018)), a deep learning algorithm has been implemented with distinct traditional algorithms. Through this, the researcher was able to achieve an accuracy of 99.1 percent which is the best result achieved so far. According to (Yue and Elfayoumy (2007)), implementation of deep learning models comes with challenges. It is an arduous task to apply deep learning models to actual datasets since it becomes computationally expensive and time-consuming. Although a model generates accurate results, it takes time which is not good enough in case of SMS because people tend to access the SMS within seconds of receiving it. This gap in research demands the creation of a model that performs efficiently with high accuracy consuming very less time. Hence, below research question is put forward in this research.

1.2 Research Question

Spam SMS becomes very irritating and disappointing to the mobile phone users. This can lead to crucial information loss by clicking on any spurious links or even the genuine information could be missed by ignoring the actual message as spam. Hence there is a need to design a model that is not only efficient but quick in the detection of spam SMS. So the Research Question is:

RQ: "To what extent can classification of spam SMS using a combination of Natural Language Processing techniques (Bag of Words and TF-IDF) and machine learning techniques (XGBoost, LightGBM, SVM, Bernoulli Naïve Bayes, and Random Forest) improve the efficiency in detection of spam SMS with low latency to help mobile phone users?"

This research helps in filtering the spam SMS quickly and efficiently which can prevent data loss. In this research, the problem area is tackled by implementing models like Bernoulli Naïve Bayes, SVM, Random Forest and boosting techniques like XGBoost and LightGBM. Gradient Boosting technique has proven to be very efficient, flexible and quick in generating results from actual datasets and has won accolades on several data mining platforms (Tianqi and Guestrin (2016)). Along with these techniques, feature selection technique Chi-Square is implemented which further reduces dimensionality and

in turn decreases time complexity. To make the model more reliable, Stratified 10-Fold Cross-Validation technique was used.

To solve the research question, the research objectives are implemented as depicted in section 1.3 (Table 1).

1.3 Research Objectives and Contributions

One of the important objectives was to do an investigation on the work already done in the field of text analytics which helped in gaining the knowledge and achieving the below research objectives.

Table 1: Research Objectives

Objectives	Description	Evaluation Metrics
Obj-1	Spam SMS file dataset pre-processing	
Obj-2	Implement Natural Language Processing techniques	
Obj-2a	Implement Bag-Of-Words to create document term matrix for Spam SMS Detection	
Obj-2b	Implement TF-IDF to create document term matrix for Spam SMS Detection	
Obj-3	Apply feature selection technique i.e. Chi-Square method	
Obj-4	Implement, Evaluate and generate the outcome for several machine learning algorithms	
Obj-4a	Implementation, Evaluation, and Results of XGBoost	Accuracy, Precision, Recall, F1-Score, Execution Time
Obj-4b	Implementation, Evaluation, and Results of LightGBM	Accuracy, Precision, Recall, F1-Score, Execution Time
Obj-4c	Implementation, Evaluation, and Results of Bernoulli Naive Bayes	Accuracy, Precision, Recall, F1-Score, Execution Time
Obj-4d	Implementation, Evaluation, and Results of SVM	Accuracy, Precision, Recall, F1-Score, Execution Time
Obj-4e	Implementation, Evaluation, and Results of Random Forest	Accuracy, Precision, Recall, F1-Score, Execution Time
Obj-5	Compare the developed models(Obj-5a to Obj-5e)	Accuracy, Execution Time
Obj-6	Compare the developed models with the existing state-of-the-art models	Accuracy

Table 2: Research Objectives

Objectives	Description	Evaluation Metrics
Obj-1	Spam SMS file dataset pre-processing	
Obj-2	Implement Natural Language Processing techniques	
Obj-2a	Implement Bag-Of-Words to create document term matrix for Spam SMS Detection	
Obj-2b	Implement TF-IDF to create document term matrix for Spam SMS Detection	
Obj-3	Apply feature selection technique i.e. Chi-Square method	
Obj-4	Implement, Evaluate and generate the outcome for several machine learning algorithms	
Obj-4a	Implementation, Evaluation, and Results of XGBoost	Accuracy, Precision, Recall, F1-Score, Execution Time
Obj-4b	Implementation, Evaluation, and Results of LightGBM	Accuracy, Precision, Recall, F1-Score, Execution Time
Obj-4c	Implementation, Evaluation, and Results of Bernoulli Naive Bayes	Accuracy, Precision, Recall, F1-Score, Execution Time
Obj-4d	Implementation, Evaluation, and Results of SVM	Accuracy, Precision, Recall, F1-Score, Execution Time
Obj-4e	Implementation, Evaluation, and Results of Random Forest	Accuracy, Precision, Recall, F1-Score, Execution Time
Obj-5	Compare the developed models(Obj-5a to Obj-5e)	Accuracy, Execution Time
Obj-6	Compare the developed models with the existing state-of-the-art models	Accuracy

Contributions: The major contribution resulting from this research was the implementation of machine learning models such as XGBoost, LightGBM, Random Forest, SVM, and Bernoulli Naïve Bayes that are capable to detect spam SMS with good efficiency and less time.
The minor contribution of the research was the comparison of results of fully developed models along with the existing models on evaluation metrics using the visualizations. These contributions will help mobile phone users as it will effectively resolve the problem of spam SMS by detecting it with the minimum time.

The rest of the technical report contains the following chapters: Chapter 2 represents Literature Review, Chapter 3 consists of Scientific Methodology and Design Specifications, Chapter 4 talks about Data Pre-processing, Implementation, Evaluation, and Results of Spam Short Message Service Classification Models, Chapter 5 contains Discussion and Comparison of Results and lastly, Chapter 6 consists of Conclusion and Future Work.

2 Literature Review

2.1 Introduction

SMS Spam is a serious problem and thus many researchers have keen motivation to solve the problem using different approaches and methods. This section investigates the problem of spam and various other text mining problem and how it is tackled by implementing different machine learning algorithms and techniques by reviewing the literature from the year 2007 to date. The review is divided into the following subsections: (i)A Critique of Spam Short Message Service Detection (ii)A Critical Review of Algorithms and Techniques Used in Spam Classification and Identified Gaps (iii) A Critique on Dataset and Features used in Short Message Service Spam Detection (iv)A Comparison of the Reviewed techniques.

2.2 A Critique of Spam Short Message Service Detection

Over the past few years, mobile phones have encountered an enormous amount of growth and therefore SMS has become the common platform for the exchange of information. It was reported that there are around 3.5 billion active mobile phone users in the year 2010 (Shirani-Mehr (2012)). The researchers also mention that SMS had gained popularity among the youth and as the SMS charges reduced over the years, around 30% of total messages were found to be spam in Asia in the year 2012. The number of spam SMS detection software available is very limited, which makes spam SMS detection an interesting problem that can be solved by using machine learning techniques.
As per the researcher (Fernandes et al. (2015)), the short message service was started in the year 1992 and has gained substantial popularity with revenue close to 153 billion dollars in the year 2016. A humongous amount of SMSes are exchanged all over the world on a daily basis. Several misuses of SMSes have been highlighted leading to loss of personal and financial information by unknowingly clicking on bogus links. The researcher (Sethi et al. (2018)) revealed a very appalling fact that the amount of spam SMS circulated the world over surpass the number of spam emails. The problem is so serious that some countries like Japan took legal action against it. One of the main reasons for spam SMS is because the cost of sending an SMS is so low that it becomes irrelevant compared to the benefit hackers can reap if they get hold of the sensitive information.
Researcher (Agarwal et al. (2016)) mentions that spam is the junk or unwanted messages which are broadcasted to a group of users with bad intention like stealing personal information, etc. The spam messages are growing at a rapid rate with almost 500% increase every subsequent year. The researcher also mentions that the quantum of spam SMS is not the same across the regions. For instance, in North America only one percent of total SMS were spam in the year 2010, however, more than thirty percent were spam in some parts of Asia. In the year 2008, the spam messages received by the people in China in a week were around 200 billion.
Looking at the above-stated facts mentioned in research papers regarding the growth of spam SMS, it is high time to solve the problem using the latest techniques which are efficient and can be implemented in real-time.

2.3 A Critical Review of Algorithms, Techniques Used in Spam Classification and Identified Gaps

2.3.1 A Review of Short Message Service Spam Classification

An analysis of spam SMS filtering was done on the UCI machine learning dataset in the year 2015 by (Kim et al. (2015)) who chose the frequency ratio feature selection technique while implementing the algorithms like Naïve Bayes, Logistic Regression and J-48 Decision Trees where the 10 fold cross-validation technique was applied. It was seen that Naïve Bayes generated results in a minimum time with the highest accuracy of 94 percent. In the year 2018, a similar analysis was conducted by (Gupta et al. (2018)) using 2 sets of data, one with UCI machine learning which is the same corpus as of Kaggle with a total of 5574 ham and spam messages and another dataset contains 2000 spam and ham messages. TF-IDF matrix was created and then machine learning algorithms like Naïve Bayes, Random Forest, SVM, Decision Tree, Convolutional Neural Network and Artificial Neural Network were applied on both the datasets. The results obtained by CNN were the state-of-art in this area with an accuracy of 99.10 followed by Naïve Bayes and SVM.
A research conducted by (Ma et al. (2016)), on spam SMS detection proposes a message topic model which is a form of probability topic model. It uses the KNN algorithm to remove the sparsity problem in the messages. The symbol terms, background terms were considered and it was found that the model generated better results than the standard LDA. The classifier GentleBoost was used for the first time in the research done by (Akbari and Sajedi (2015)) on SMS spam detection in the year 2015. For unbalanced data and binary classification, boosting algorithms work well. GentleBoost is a combination of two algorithms, namely AdaBoost and LogitBoost. GentleBoost is well known for its higher accuracy and less consumption of storage as it removes unwanted features. It obtained an accuracy of 98% on the dataset consisting of 5572 text messages.
The author (Agarwal et al. (2016)) states that the short length of the messages and the use of casual words in the text messages do not allow it to perform well with the already established solutions of email spam filtering. In this research, it can be seen that SVM followed by Multinomial Naïve Bayes (MNB) shows outstanding results in terms of accuracy with 98.23 and 97.87% respectively. MNB took the least execution time of 2.03 seconds. The researcher further suggests that features like the number of characters in the messages or definite threshold to the length of the message can increase the performance.
Looking at the above research, it can be stated that the performance of traditional algorithms like Naïve Bayes and SVM is superior to other algorithms. Also in this research, the length parameter can be taken as an additional feature to check if it enhances the performance of the model.

2.3.2 A Review of Email Spam Classification

In the text classification area, the Naïve Bayes algorithm is very popular in generating good results. The research conducted by (Almeida et al. (2011)), on email spam filtering used 7 different versions of Naïve Bayes such as Basic Naïve Bayes, Bernoulli Naïve Bayes, Gaussian Naïve Bayes to name a few. The feature selection techniques applied were Chi-Square and Mutual Information. The model implementing Bernoulli Naïve Bayes in combination with Chi-Square generated better results.

Significant power and memory are needed if the number of features in a classification problem is excessive. Higher the features, more is the dimensionality and greater is the need for power and memory. By removing the features that are not pertinent or redundant and pulling out the beneficial ones, performance can be amplified as stated by the paper (Ergin and Isik (2014)) which is on email spam filtering. Eliminating the stop words, stemming and normalizing the dataset is followed by the creation of Document Term Matrix using Bag of Words. Techniques like Gini Index, Chi-Square and Information Gain were used for feature selection on machine learning algorithms like Artificial Neural Networks (ANN) and Decision Tree. An outstanding result was generated by the coalition of Bag of Words and Chi-Square techniques on ANN with the accuracy as 91%.

In the research performed by (Islam et al. (2009)) on the spam filtering techniques such as Naïve Bayes and Artificial Neural Networks (ANN) were applied. The features from the header and the body of an email were taken into consideration. It was found that Naïve Bayes outperformed ANN with higher accuracy, recall, and precision. A paper by (Lee et al. (2017)) found that Weighted Naïve Bayes is not only computationally effective but also very efficient in case of spam detection even with new destructive campaigns. The evaluation was carried on eight datasets that were sourced from two sites. The accuracy attained is around 95% for both the sources. The researcher (Yue and Elfayoumy (2007)) argues that despite neural networks producing good results, it takes a huge amount of time in generating results. Also, once the model is built it stops learning from new emails, unlike Naïve Bayes which is adaptive and trains from new emails. The researcher suggested that applying techniques like boosting in the future which can produce more quick results.

The researcher (Yu and ben Xu (2008)) discusses the pitfalls of email spams as these messages not only waste time and energy of the end-users but lead to issues like utilizing high mailbox space and bandwidth. The researcher tried to solve this problem by implementing 4 machine learning models like Support Vector Machine, Naïve Bayes, Neural Network and Relevance Vector Machine. The performance of the models was computed using the training set of varying sizes and the results demonstrated that Neural Networks are not suitable for spam filtering as they are susceptible to the size of training data. SVM and RVM performed better where SVM took less training time than RVM.

Looking at this section, it can be interpreted that Naïve Bayes along with feature selection like the Chi-Square technique shows great results in text classification. The results produced are not only better in terms of accuracy but also in terms of time complexity. Also, the deep learning techniques like Neural Networks were not found to be productive in spam detection as it is sensitive to the size of the training data. It takes a large amount of time to train and test the dataset which makes them computationally ineffective.

2.3.3 An Investigation of Other Text Classification Areas

Phishing is a type of cybercrime where sensitive information such as user credentials, card details, etc. are stolen through counterfeit websites or emails. The paper by the researcher (Li et al. (2019)) provided the solution by implementing Word2Vec as a feature extraction technique that gathers the feature from HTML code. Along with the feature extraction, a joint model using algorithms like LightGBM, XGBoost, and Gradient Boosting Trees are built which raised the performance to 97% accuracy. The researcher (Liew et al. (2019)), detected phishing tweets on a real-time basis utilizing the feature from the phishing

URL. Random Forest technique was implemented and found to produce an accuracy of 94.5%. The researcher (Koray et al. (2019)) detects the phishing websites by analyzing and extracting features from the URL. The Random Word Detection module is used which decomposes URL into small features which are then used to classify if the websites are legitimate or not. Seven different machine learning algorithms like Naïve Bayes, KNN, SVM, Random Forest, Decision Tree, Adaboost, K-star were implemented on a humongous amount of data. It was seen that Random Forest produced the highest accuracy of around 97.98 percent among all the techniques applied.

In the year 2018, the researcher (Yuan et al. (2018)) proposed a blend of features pertaining to URL and web page for the detection of phishing websites. Along with the basic features like the length of the URL, unusual characters, etc, statistical features like mean, median and variance and lexical features like title and the content of the web page were also considered. Several algorithms like KNN, Logistic Regression, Random Forest, Deep Forest, XGBoost were applied. It was found that Deep Forest followed by XGBoost manifested high accuracy and less training time.

Nowadays, the success of any business heavily depends on authentic reviews. However, not all reviews are authentic. The ratio of genuine reviews to sham reviews varies significantly from case to case and higher the bogus reviews, more is the image maligned of a business. Implementing sentiment analysis using natural language processing techniques can effectively detect the opinion spams. Earlier, researchers like (Jindal and Liu (2007)), (Ren and Ji (2017)) used supervised and semi-supervised algorithms for the detection of spam opinions. These models have a few restrictions such as low flexibility, high computational time and poor accuracy. These limitations were overcome by the researcher ((Hazim et al. (2018)) using the Gradient Boosting models like XGBoost, GBM Bernoulli, GBM Adaboost and GBM Gaussian. The opinion spam detection was performed on multilingual datasets. It was found that XGBoost outperformed the other models for the English language dataset generating high recall percentage whereas GBM Gaussian produced good results for the Malay language dataset. The researcher (Prieto et al. (2016)) detects opinion spam using neural networks and it was observed that model complexity increases due to the large set of details provided to the neural network and thus increases the overall computational cost.

In the world of online advertising, fraud clicks are one of the most momentous issues. The research done by (Minastireanu and Mesnita (2019)) tackles the problem of fraud clicks by using the latest machine learning technique viz. LightGBM on the dataset which contains millions of clicks. The K-Fold Cross-Validation technique is used as a feature engineering which helps in improving the performance. The accuracy achieved by the model was 98% and was found to be the fastest with respect to computational speed and low on memory consumption. Looking at the above research papers, it was found that LightGBM and XGBoost are suitable as it performs faster with a less computational speed, unlike the deep learning techniques. Also, Random Forest performed well giving high accuracy and hence in this research, these algorithms are chosen.

2.4 A Review of Datasets and Features used in Short Message Service Spam Detection

Spam SMS is a serious problem in Vietnam due to the cheap message pre-paid bundle available. The research done by (Pham (2016)) focuses on the detection of spam SMS for the Vietnamese language. The dataset contains 6599 messages marked as ham or spam.

Out of these 1042 were spam and rest are legitimate. The length feature was added as the spam messages tend to be longer than ham. The data was pre-processed using NLP techniques and then Bag of Words (BoW) and Term Frequency - Inverse Document Frequency (TF-IDF) were chosen as feature selection techniques. A content-based spam message filtering method proposed by (Balli and Karasoy (2018)) having a dataset of 5574 messages of which 747 are spam uses the semantic relationship between the SMS words. The feature selection is done using the Word2Vec algorithm which calculates the distance between the vector of the words and thus features are extracted for each message. The work presented by the researcher (Aich et al. (2019)) for spam detection on imbalanced datasets of SMS, implemented the SMOTE approach which generated great results in combination with the SVM algorithm and the accuracy increased by seven points.

The researcher (Najadat et al. (2014))investigates spam SMS detection by implementing 12 different types of classifiers on the dataset which contained 5574 messages. The research used the technique of down-sampling where the ham messages were reduced to the count of spam messages. The main contribution of this research is to examine the impact of class imbalance on performance and the researcher found that a balanced dataset produces more accurate results. The results are compared to the previous research done with the imbalanced dataset and found that performance degrades due to the under-fitting issue.

It can be seen that the datasets which have the issue of class imbalance can lead to bias results and poor performance whereas the balanced dataset can improve the performance. Also, feature selection techniques like Bag Of Words and TF-IDF works well in the case of text classification.

2.5 A Comparison of the Reviewed techniques

A high-level comparison of the related work in text analytics classification problem based on feature used and classifiers applied. It can be seen that classifiers like Naïve Bayes, Random Forest, LightGBM, XGBoost, and SVM outperform other classifiers and hence these are chosen to be implemented in the research.

Table 3: Comparison of Reviewed Features and Classification Techniques

Area of Classification	**Applied Classifiers**	**Features Extracted**	**Results**	**Author**
Spam SMS Classification	Naive Bayes, J-48, Logistic Regression	Message-Keyword Matrix (Bag of Words)	Naive Bayes with 94.7% accuracy	(Kim et al. (2015))
Spam SMS Classification	Naive Bayes, SVM, Random Forests, Adaboost, CNN, ANN,Logistic Regression, Decision Tree	TFIDFVectorizer	CNN with 99.1% accuracy	(Gupta et al. (2018))
Email Spam Classification	ANN, Na¨ıve Bayes	Multiple features from header and body like URL, images, links, etc	Naive Bayes with 92% with low execution time than Neural Networks	(Islam et al. (2009))
Phishing Classification	Logistic Regression, Decision Tree, GBDT, Deep Forest, Random Forest, XGBoost, KNN	TF-IDF for feature extracted from URL and links	Deep Forest with 97.7% and XGBoost with 97.1%	(Yuan et al. (2018))
Phishing Classification	Naïve Bayes, KNN, SVM, Random Forest, Decision Tree, Adaboost, K-star	NLP features, word vectors	Random Forest with 97.98%	(Koray et al. (2019))
Fraud Advertisement Classification	LightGBM, XGBoost. Stochastic Gradient Boosting	IP, OS, channel, device, click time	LightGBM with 98% accuracy	(Minastireanu and Mesnita (2019))

2.6 Conclusion

Looking at the related work done in the area of text classification, it is clearly seen that gradient boosting techniques produce quick and accurate results. As suggested, the length feature can be a useful parameter in deciding the type of SMS. Hence the same was implemented in this research. There is an urgent need for developing a spam SMS detection model by combining the best-reviewed machine learning techniques with the NLP techniques which can generate good results and can answer the research question (section 1.2) and the research objectives (section 1.3). The next chapter discusses the scientific methodology and design specifications chosen to develop the spam detection model which helps mobile phone users.

3 Scientific Methodology and Design Specifications

In data mining projects, different methodologies are used. The most common ones are CRISP-DM, KDD, and SEMMA. Knowledge Discovery and Data Mining (KDD) suits well for the research as the deployment step is not required. KDD is a very precise and complete approach that focuses not only on the business process but also on implementation(Shafique and Qaiser (2014)). The design architecture is a two-tier which contains a Presentation Layer and Application Layer.

3.1 Methodology for Short Message Service Spam Detection

The methodology for Short Message Service Spam Detection (Figure 1) is an iterative process that has the following phases :
(a) Data Selection phase where the UCI machine learning repository provided data to Kaggle, and the data was downloaded from Kaggle which is in .csv format.
(b) Data cleaning and preprocessing was done in python using the NLTK library.
(c) The data transformation was done by normalization and creation of Document Term Matrix followed by feature selection using a statistical technique like Chi-Square.
(d) In the data mining step, machine learning models such as XGBoost, LightGBM, Bernoulli Naïve Bayes, SVM, and Random Forest were implemented.
(e) All the models were evaluated and interpreted on the basis of different parameters like Accuracy, Precision, Recall, F1-Score and Execution Time.

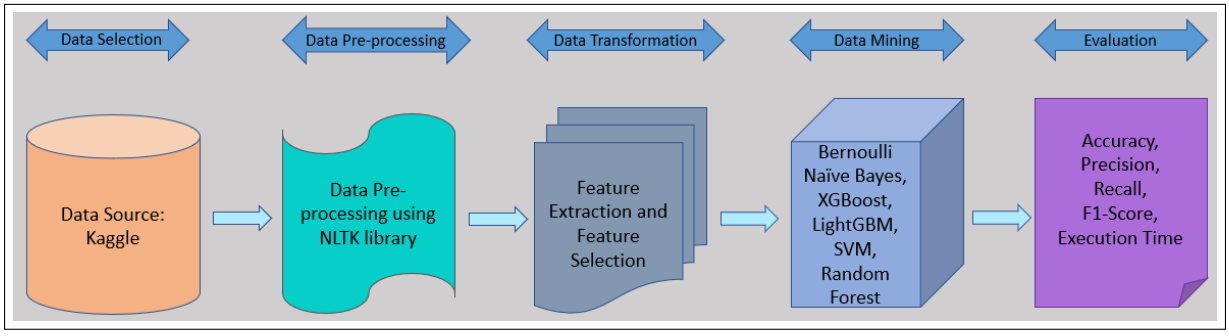

Figure 1: Methodology for Short Message Service Spam Detection

3.2 Project Design Specification

As depicted in Figure 2, the design process for the Short Message Service Spam Detection model consists of two-tier that is the Presentation Layer and the Business Logic Layer. The reason for choosing 2-tier architecture over 3-tier architecture was the format of the dataset. The data was already in a structured format and did not require any additional conversion apart from the pre-processing. The presentation layer consists of interpretation of results in the form of visualizations which was done in the Microsoft Power BI tool. In the Business Logic Layer, data was fetched from Kaggle which was then pre-processed, transformed, trained and evaluated using different classification models.

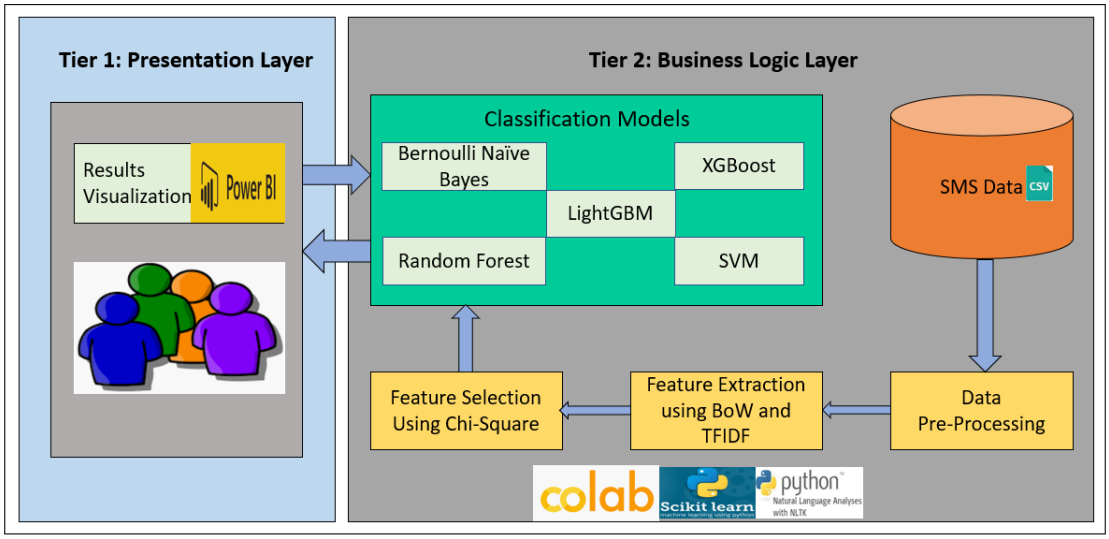

Figure 2: Design Process for Short Message Service Spam Detection

Google cloud platform Colab was used for the implementation of the project. Colab is a free to use platform as a service provided by Google. It is the preferred platform of a large number of developers for machine learning since it has almost all the packages pre-installed and does not require any special installations.

The implementation, evaluation, and results are discussed in detail in the next chapter.

4 Data Pre-processing, Implementation, Evaluation and Results of Spam Short Message Service Classification Models

4.1 Introduction

The section discusses all the steps involved in the implementation along with evaluation and results. Figure 3 illustrates the Workflow Diagram for the Short Message Service Spam Detection.The Workflow of the project is explained in detail step by step in this section.

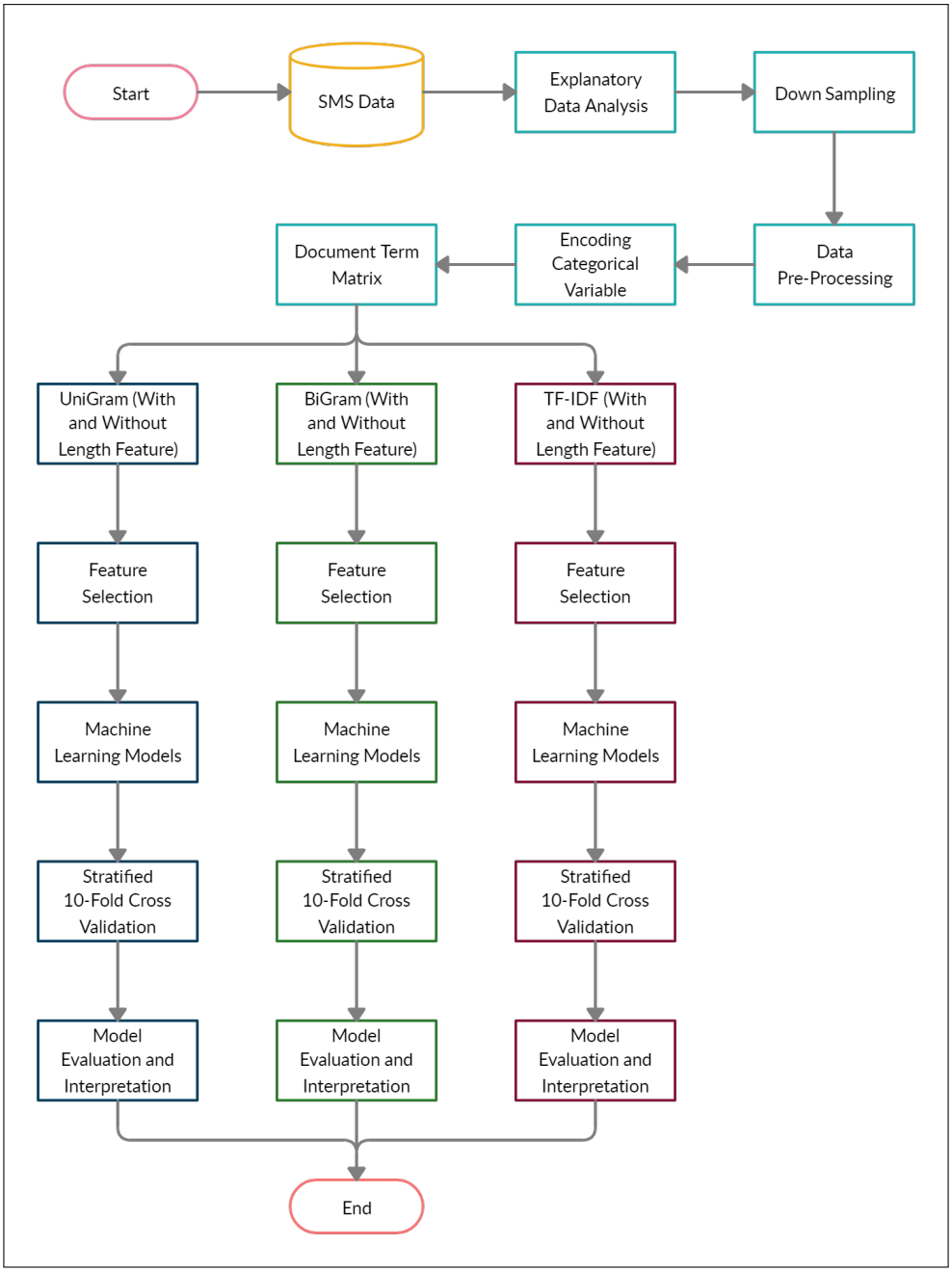

Figure 3: Work Flow Diagram for Short Message Service Spam Detection

4.2 Exploratory Data Analysis

Exploratory Data Analysis is a crucial process where the data is analyzed to uncover underlying patterns, spot abnormality and test the hypothesis . It is the best practice to understand the data and then carry out the data mining process. Missing value analysis was carried out using libraries of python like Pandas. An additional feature of the "length of text messages" was considered as there was a substantial correlation of over 0.6 between the length and the type of SMS as seen in Figure 4. For visualization of most frequent words appearing in spam messages and ham messages, the Matplotlib library with WordCloud technique was used (Refer Figure 5 and 6).

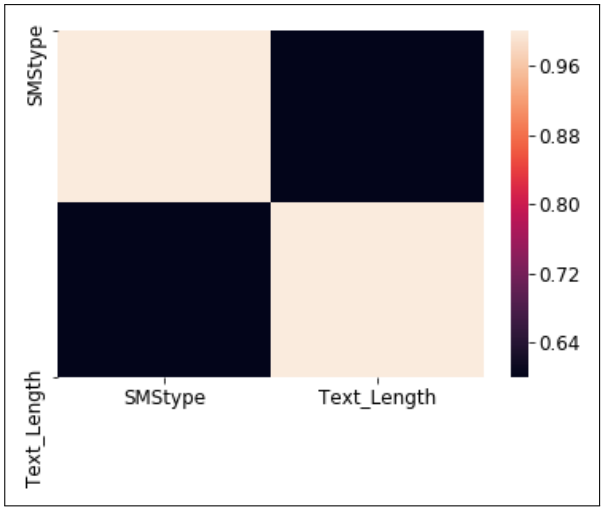

Figure 4: Correlation Matrix For SMS Type and Length Feature

Figure 5: Word Cloud for Spam

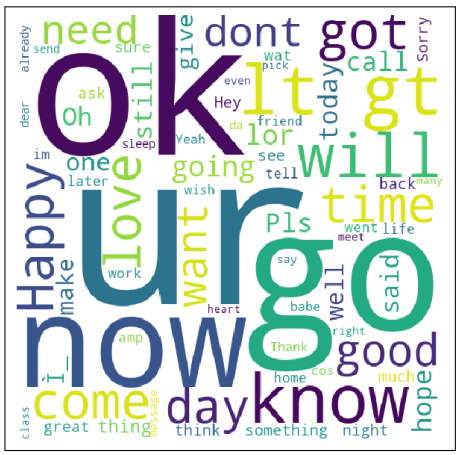

Figure 6: Word Cloud for Ham

4.3 Data Sampling

Class imbalance is most commonly observed in machine learning classification problems. It occurs where the 2 classes have disproportionate data. Under such circumstances, most of the algorithms generate biased results more towards the majority class.(Longadge and Dongre (2013)) The problem can be solved by implementing 2 techniques that are either up-sampling or down-sampling. Up-sampling is a technique where the minority class data is artificially created or duplicated and reached to the level of the majority class. Up-sampling is majorly done when the minority class is highly low in number.The other technique that is down-sampling is where the majority class samples are reduced to the same level as minority class sample size. In this research down-sampling is done using random sampling as minority class samples were in decent number. To implement this technique no external libraries were imported. Figure 7 shows the data before and after down sampling.

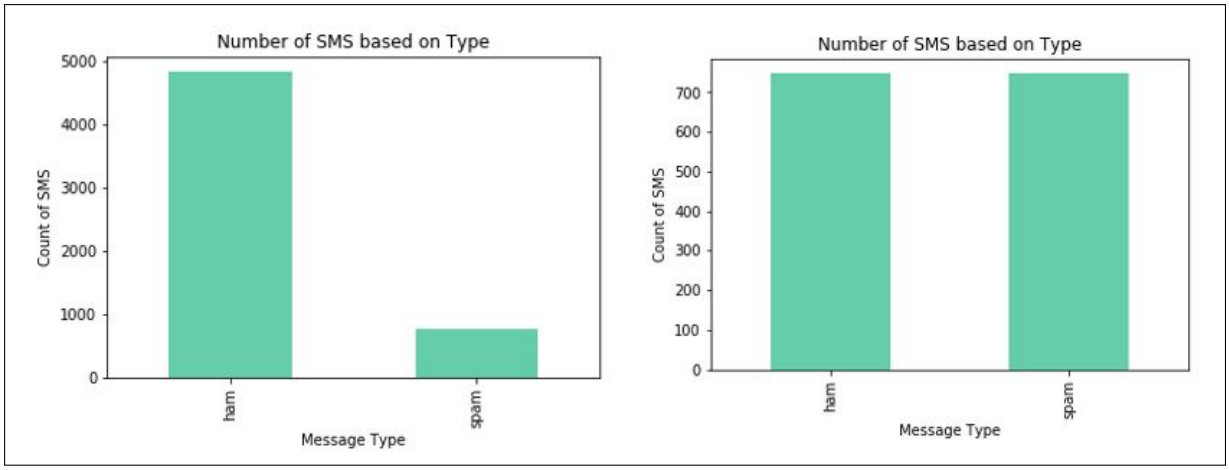

Figure 7: Message Count before and after Down Sampling

4.4 Data Pre-processing

Data pre-processing is mainly cleaning of data by removing unwanted rows, columns, missing values, outliers, etc. For the research, the following pre-processing steps have been taken:

Removal of Unwanted Columns It was found that there were 3 extra columns without data in it, which adds extra noise to the model hence removed.

Cleaning of Text Messages For cleaning the text messages, the following steps are involved:

•**Tokenize the words**: In this step, the words are split into tokens based on white spaces or punctuation. To achieve it, word_tokenize function from NLTK library was used.

•**Removal of stop words from text messages**: Stop words are basically the most commonly used words (such as "a", "an", "the") which increases the dimensionality and impacts the efficiency of the model (Vani and Gupta (2014)). Stopwords package from NLTK library was used where the stop words were removed if they appear in the pre-defined stop words list, thereby filtering the text messages.

•**Lemmatizing words**: Lemmatization in simple terms refers to the removal of duplicate data. For example, words like "study", "studying", and "studies" are considered 3 different words after the creation of a Document Term Matrix and hence increases the dimensionality. WordNetLemmatizer package from the NLTK library was implemented which helped in converting the inflected word to its base format.

•**Normalize the words**: The word-stock can be decreased if all the words are in either lowercase or uppercase. For example, "OFFER" and "offer" are considered 2 separate words before normalization. The lower() function in python helped in achieving the normalization of data.

Encoding the categorical column LabelEncoder() function from the skelearn.preprocessing library is used to encode the categorical dependent column.The label encoding is preferred over one-hot encoding because the SMS type is a 2-class column and not multi-class, which is either spam or ham.

4.5 Document Term Matrix using NLP Techniques

As the machine learning models work only on mathematical data, a matrix was created which contains the word and its frequency of occurrence. The following two techniques were used for the creation of the document term matrix.

Bag of Words (BoW): Bag of Words is a way of extracting the features from the set of text messages. In this, a matrix called a bag of words is created (shown in Figure 8) which describes the text based on the frequency of the word appearing in the document.This was implemented using the CountVectorizer package in python. The CountVectorizer can be implemented in the form of n-grams. In the research, the feature was extracted using Unigram and Bigram matrix. The Unigram matrix comprises a single word whereas the Bigram matrix consists of two consecutive words from a document.

	Text_Length	1	2	3	4	5	6	...	3615	3616	3617	3618	3619	3620	3621
0	29	0	0	0	0	0	0	...	0	0	0	0	0	0	0
1	101	0	0	0	0	0	0	...	0	1	0	0	0	0	0
2	141	0	0	0	0	0	0	...	0	0	0	0	0	0	0
3	59	0	0	0	0	0	0	...	0	0	0	0	0	0	0
4	31	0	0	0	0	0	0	...	0	0	0	0	0	0	0

Figure 8: Bag Of Words Matrix

Term Frequency-Inverse Document Frequency (TF-IDF) : Term Frequency-Inverse Document Frequency commonly known as TF-IDF is one of the common techniques used to represent the text data into vector form. This matrix (shown in Figure 9) helps in understanding the importance of the word in the corpus of documents. Frequently occurring term in a given document that seldom occurs in other documents in the corpus has high TF-IDF value. To implement the TF-IDFVectorizer() function was used with the n-gram as a unigram. The machine learning models were implemented after the creation of the matrix.

	Text_Length	1	2	3	4	5	...	3616	3617	3618	3619	3620	3621
0	29.0	0.0	0.0	0.0	0.0	0.0	...	0.000000	0.0	0.0	0.0	0.0	0.0
1	101.0	0.0	0.0	0.0	0.0	0.0	...	0.286945	0.0	0.0	0.0	0.0	0.0
2	141.0	0.0	0.0	0.0	0.0	0.0	...	0.000000	0.0	0.0	0.0	0.0	0.0
3	59.0	0.0	0.0	0.0	0.0	0.0	...	0.000000	0.0	0.0	0.0	0.0	0.0
4	31.0	0.0	0.0	0.0	0.0	0.0	...	0.000000	0.0	0.0	0.0	0.0	0.0

Figure 9: TF-IDF Matrix

4.6 Feature Selection

It is one of the most important steps in the area of text mining. It is the process of selecting the important attributes which contribute the most in making the classification. Feature Selection can enhance the training time of a model. It reduces over-fitting and thus increases the accuracy and performance of the model. The more the features, the more complex a model, hence it reduces the complexity and is easy to understand. There are many techniques used for feature selection. For the research, the Chi-Square Technique was implemented.

Chi-Square Feature Selection Technique: Chi-Square is a statistical test which eliminates the less important features by checking the relationship between dependent and

independent feature. It checks the deviation of the observed count from the expected count. Chi-Square technique performs well in the case of categorical data and the Bag-Of-Words and TF-IDF matrix consist of categorical data only and therefore the technique is preferred in the research. Also, from the literature (Ergin and Isik (2014)) and (Almeida et al. (2011)) it can be seen that Chi-Square generated excellent outputs in the field of text analytics. Chi2 package from the sklearn library was used for selecting the best features.
This accomplished the research objective from Obj-1 to Obj-3 mentioned in Chapter 1, Section 1.3.

4.7 Implementation, Evaluation, and Results of Spam Short Message Service Classification Models

After the selection of features, machine learning models like Naïve Bayes, Random Forest, XGBoost, LightGBM, and SVM. The evaluation of the models was done using a **Stratified 10-Fold cross-validation** approach on the basis of Accuracy, Precision, Recall, F1-Score and Execution Time. The accuracy and execution time being the most significant and were used to solve the research question. As mentioned by (Gupta et al. (2018)) choosing the correct metric for evaluation is important in interpreting the results from the observations. The following parameters which help in better understanding of the evaluation metrics which is considered in the research:
a. **True Positive(TP)**: The positive class is correctly predicted by the model
b. **True Negative (TN)**: The negative class is correctly predicted by the model
c. **False Positive (FP)**: The model predicts the class as positive but it should be negative
d. **False Negative (FN)**: The model predicts the class as negative but it should be positive

Accuracy: It measures how close the observed value is from the actual value. As per (Aich et al. (2019)) the classification accuracy metric is more transparent when the classes are balanced. It is formulated as:

$$\textbf{Accuracy} = \textbf{(TP+TN)/(TP+TN+FP+FN)}$$

Precision: It is the ratio of the correctly predicted positive results to the total number of positive results that a model predicts. It can also be interpreted as how much the model is relevant when it predicts. (Gupta et al. (2018)) It is formulated as :

$$\textbf{Precision} = \textbf{(TP)/(TP+FP)}$$

Recall: It is the true positive rate or sensitivity. It is the proportion of correctly predicted positive instances to the total number of instances in the actual class. Recall shows the potential of a model to find all the positive instances. It can be formulated as:

$$\textbf{Recall} = \textbf{(TP)/(TP+FN)}$$

F1-Score: F1-Score is the weighted average of recall and precision. It measures the incorrectly classified instances. It is formulated as:

F1-Score = 2*((Precision*Recall)/(Precision+Recall))

Stratified 10-Fold Cross-validation Technique: It is a statistical method where the data was divided into 10 folds in such a way that each fold represents whole data.The data was reshuffled after every fold and was repeated 10 times for this research. This technique helped in reducing the variance as different training and testing sets were used. After the cross-validation, the average of all the evaluation parameters were taken.

4.8 Experiment 1: Implementation, Evaluation and Results of XGBoost Model

4.8.1 Implementation

Extreme Gradient Boosting is an ensemble boosting machine learning technique based on the decision-tree algorithm and works in sequential manner. As per the researcher,(Tianqi and Guestrin (2016)) the model generates output very quickly compared to traditional machine learning algorithms. The other advantages of XGBoost are:
It has the capability to handle sparse data like BoW matrix.It uses parallelization which helps it to generate the results faster and suitable for low latency applications.

XGBoost Classifier was implemented using Stratified 10-Fold Cross-Validation to divide the dataset into training and test sets. The function used to implement the model was the XGBClassifier() of the sklearn library. The model was implemented on Unigram, Bigram, and TF-IDF with and without length feature.

4.8.2 Evaluation and Results

Looking at the Figure 10, it was evident that for the XGBoost model, there was a performance upgrade on taking the length feature into consideration. An increase in Accuracy, Recall, and F1-Score was depicted by the length feature. Length feature had a high impact on the Bigram matrix. The model performed well with the Unigram and TF-IDF matrix giving an Accuracy of 0.944, F1-Score of 0.943, and Recall of 0.929. Precision for Bigram without length feature was the highest at 0.997.

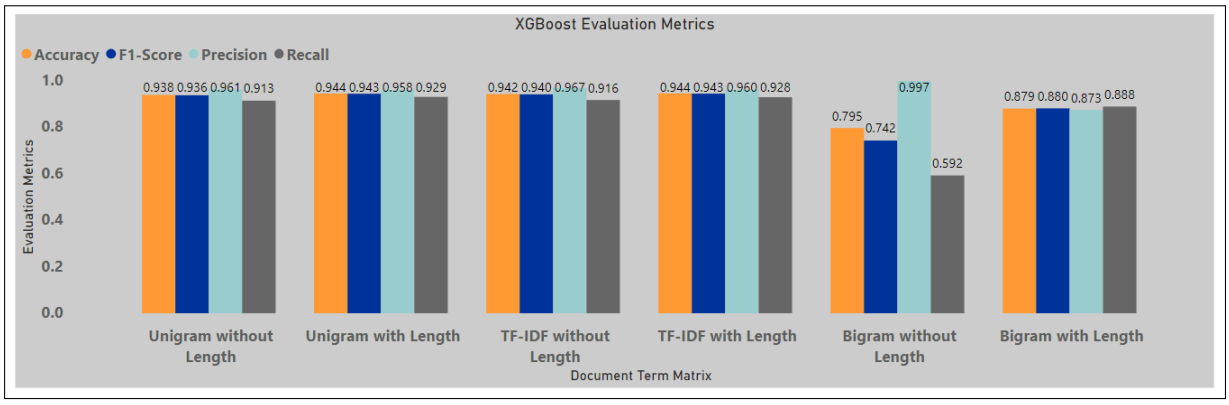

Figure 10: Evaluation Metrics for XGBoost Model

4.9 Experiment 2: Implementation, Evaluation and Results of LightGBM Model

4.9.1 Implementation

Light Gradient Boosting machine is recently developed machine learning model in 2017. It is a boosting technique and similar to XGBoost with advantages like high efficiency, low memory usage, fast training speed, better accuracy, capable to handle large datasets and support parallelization [1]. As it is new, comparatively less research has been done on it but looking at the literature, it can be seen that it performed well in the field of text analytics. (Minastireanu and Mesnita (2019) (Li et al. (2019))

LightGBM classifier was implemented using Stratified 10-Fold Cross-Validation to divide the dataset into training and test sets. The function used to implement the model was the LGBMClassifier() of the sklearn library. The model was implemented on Unigram, Bigram, and TF-IDF with and without length feature.

4.9.2 Evaluation and Results

As observed in the Figure 11,the LightGBM model generated nearly the same accuracy for the TF-IDF and Unigram matrix. The length feature contributed the most to the Bigram matrix where the performance was enhanced and there was a significant improvement in the Accuracy, Recall, and F1-Score. It can be seen that the model performed the best with the TF-IDF matrix generating an Accuracy of 0.957, Precision of 0.958, Recall of 0.956 and F1-Score of 0.957.

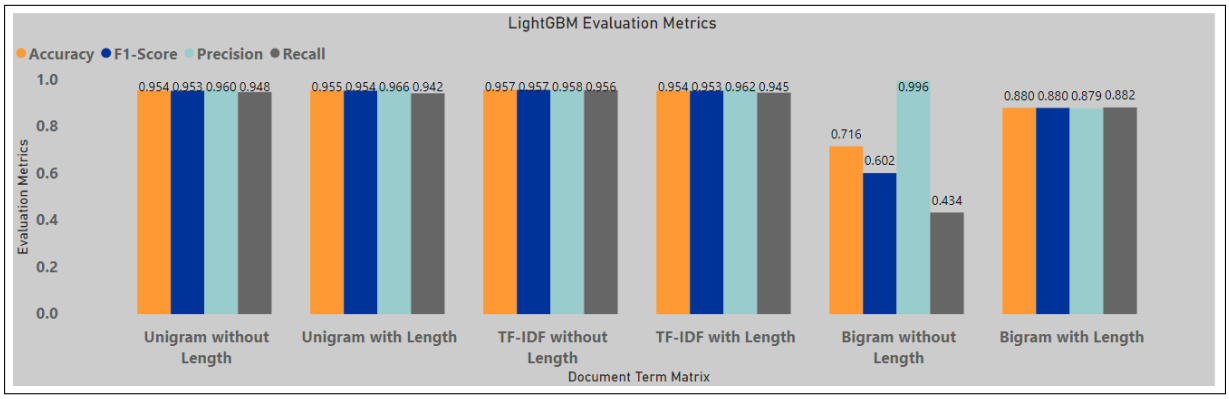

Figure 11: Evaluation Metrics for LightGBM Model

4.10 Experiment 3: Implementation, Evaluation and Results of Bernoulli Naïve Bayes Model

4.10.1 Implementation

Naïve Bayes is a simple algorithm which works on conditional probability and is mostly used in classification problems. The results generated by the researcher (Almeida et al. (2011)), (Lee et al. (2017)) showed that Naïve Bayes not only performed well in terms of accuracy and time.Bernoulli Naïve Bayes is a different form of classical Naïve Bayes

[1] https://lightgbm.readthedocs.io/en/latest/

and it works well with categorical data(Xu (2018)). Therefore it makes the classifier a suitable choice to implement in the research.

Bernoulli Naive Bayes classifier was implemented using Stratified 10-Fold Cross-Validation to divide the dataset into training and test sets. The function used to implement the model was BernoulliNB() of the sklearn library [2]. The model was implemented on Unigram, Bigram, and TF-IDF with and without length feature.

4.10.2 Evaluation and Results

From the Figure 12, it can be seen that the Bernoulli Naïve Bayes model generated the highest accuracy of 0.965 with the Unigram and TF-IDF matrix. The length feature does not add any value to this model. The positive predicted value i.e. Precision is the highest for the Bigram matrix having the value 1. The Recall and the F1-Score of the Bigram model are 0.652 and 0.789 respectively which are comparatively low compared to the Unigram and TF-IDF matrix having the values of 0.93 and 0.99 respectively.

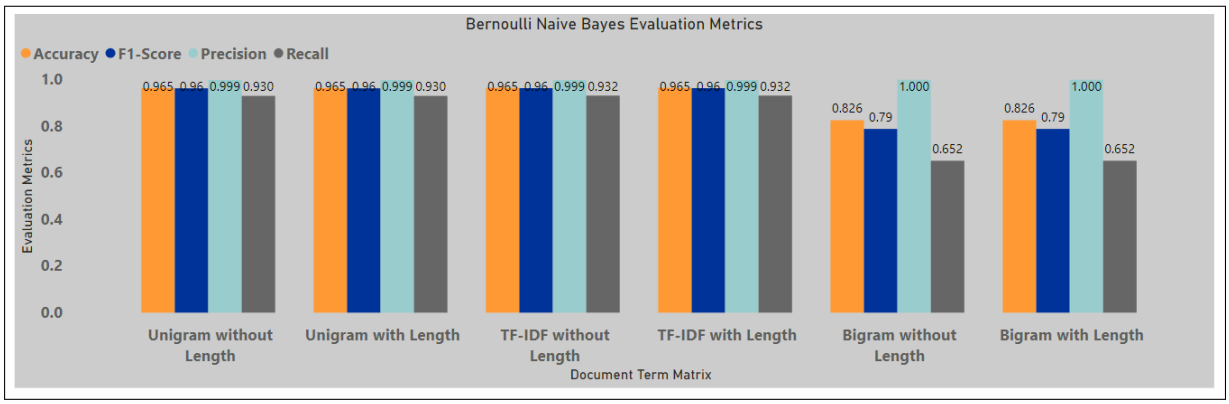

Figure 12: Evaluation Metrics for Bernoulli Naive Bayes Model

4.11 Experiment 4: Implementation, Evaluation and Results of Support Vector Machine Model

4.11.1 Implementation

Support Vector Machine is a supervised machine learning technique that uses an optimal hyperplane that differentiates between 2 or more classes. (Basu et al. (2002)) SVM can deal a large number of features with minimum error rate. The default parameters work very well and hence explicit tuning is not required. SVM with linear kernel has shown great results in the field of text analytics (Agarwal et al. (2016)), (Shirani-Mehr (2012)), and hence implemented in the research as a baseline model.

Support Vector Machine classifier was implemented using Stratified 10-Fold Cross-Validation to divide the dataset into training and test sets. The function used to implement the model was the SVC() of the sklearn library. The kernel was set as linear. The model was implemented on Unigram, Bigram, and TF-IDF with and without length feature.

[2]https://scikit-learn.org/stable/modules/classes.html

4.11.2 Evaluation and Results

SVM model evaluation metric is shown in the Figure 13. A comparison of all the matrices shows that the Unigram matrix has surpassed others with an Accuracy of 0.963, Recall of 0.945 and F1-Score of 0.962. The length feature did not contribute much to the model. Precision for Bigram without length feature was the highest of 0.993.

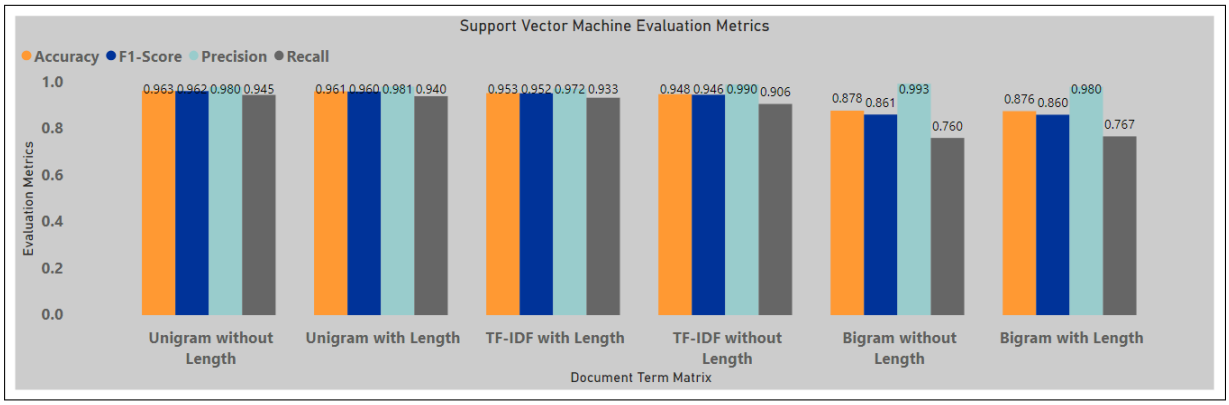

Figure 13: Evaluation Metrics for Support Vector Machine Model

4.12 Experiment 5: Implementation, Evaluation and Results of Random Forest Model

4.12.1 Implementation

Random Forest is an ensemble tree-based machine learning classifier. It is known for generating high accuracy as mentioned in the research (Koray et al. (2019)). Random Forest is capable of handling large datasets with high dimensionality and hence is a good fit for the research as it handles a large number of features created by a bag of words.

Random Forest classifier was implemented using Stratified 10-Fold Cross-Validation to divide the dataset into training and test sets. The function used to implement the model was the RandomForestClassifier() of the sklearn library. The number of estimators were taken as 1000. The model was implemented on Unigram, Bigram, and TF-IDF with and without length feature.

4.12.2 Evaluation and Results

Analyzing the Figure 14, it can be observed that the length contributed to the increase in the performance of the model for all the 3 matrices. TF-IDF with length generated the best results in terms of Accuracy, Recall, and F1-Score. The Accuracy stood at 0.964, Recall at 0.955 and F1-Score at 0.96. Precision for Bigram without length feature was the highest amongst all at 0.993.

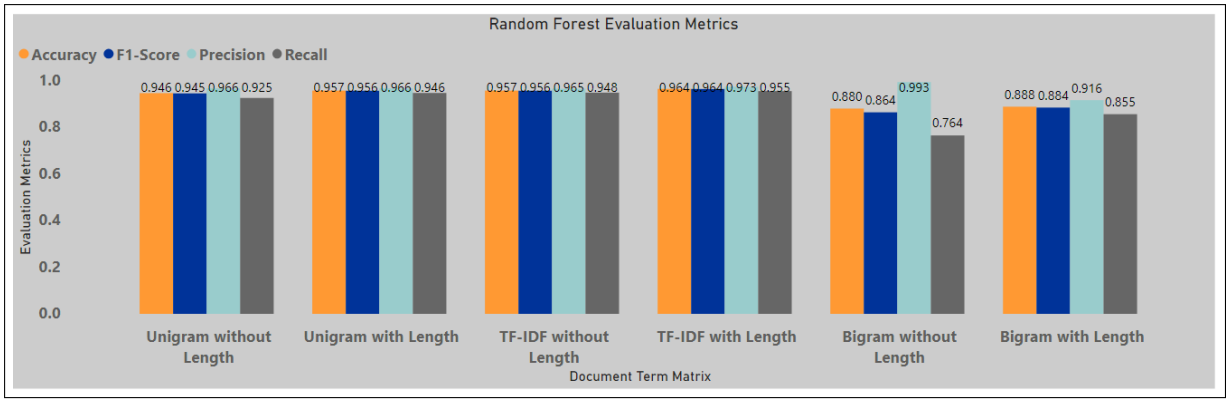

Figure 14: Evaluation Metrics for Random Forest Model

4.13 Conclusion

On the basis of implementation and generated results, the research objectives till Obj-4 (section 1.3) along with the research question (section 1.2) have been accomplished. The models developed will contribute remarkably to mobile phone users in detection of spam SMS.

5 Discussion and Comparison of Results

5.1 Comparison of Developed Models

As can be clearly seen from the above section, the TF-IDF matrix with length produced the best results amongst all the matrices used. Hence, the TF-IDF matrix was considered for comparing the various models. Fig 15 and 16 illustrate the comparison of the different models applied in this research in terms of accuracy and execution time. Evidently, the Bernoulli Naïve Bayes model generated the highest accuracy of 0.965 followed by Random Forest and LightGBM having the accuracy of 0.964 and 0.954 respectively. SVM and XGBoost resulted in an accuracy of 0.953 and 0.944 respectively. Random Forest and SVM gave an accuracy that was marginally lower than Bernoulli Naïve Bayes. However, the time taken by Random Forest (34.957 seconds) and SVM (30.285 seconds) was way higher than Bernoulli Naïve Bayes which took just 0.157 seconds. The gradient boosting models LightGBM and XGBoost took 1.708 and 7.919 seconds respectively. The time taken by LightGBM was slightly higher than the Bernoulli Naïve Bayes model but much lower than the traditional base models like Random Forest and SVM. This made LightGBM a good fit for the detection of spam SMS in a real-world scenario. This attained the research objective 5 mentioned in Chapter 1(Section 1.3).

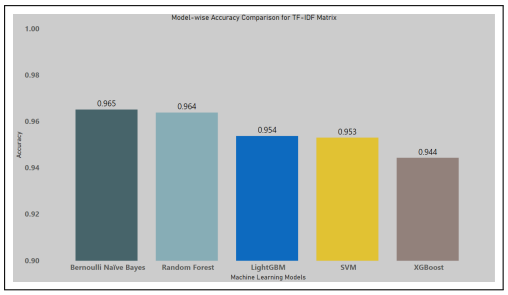

Figure 15: Model-wise Accuracy Comparison

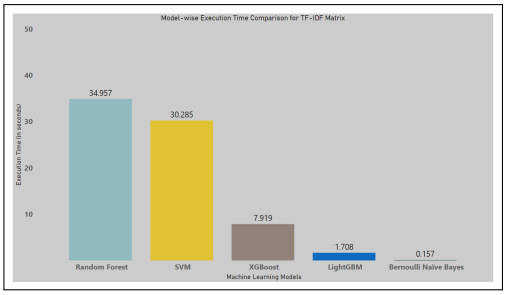

Figure 16: Model-wise Execution Time Comparison

5.2 Comparison of Developed Models with Existing Models

In this section, the developed models are compared with the existing ones that are discussed in the literature review (Table 2). The researcher (Gupta et al. (2018)), achieved an accuracy of 99.1% with the CNN model and (Kim et al. (2015)), achieved the highest accuracy of 94.70% with Naïve Bayes. As shown in the comment section of Table 3, the reason for high and low accuracy is mentioned. The CNN model produced good results but was computationally very expensive in terms of complexity and time (Yue and Elfayoumy (2007)).

Table 4: Comparison of Developed Models with Existing Models

References	Algorithm	Accuracy	Comments
Gupta et al. (2018)	Convolutional Neural Network	99.10%	Researcher did not handle class imbalance and hold out technique was used to split train and test dataset.TF-IDF matrix was used for feature.
Kim et al. (2015)	Naive Bayes	94.70%	The class imbalance was not handled. The frequency ratio technique used for feature selection.Researcher used 10-fold Cross Validation Technique to split test and train dataset
Current Research	Bernoulli Naive Bayes	96.50%	The class imbalance issue was handled, Feature selection implemneted and Stratified 10-Fold Cross Validation technique used to split the dataset.

5.3 Critique of the Research Conducted

During the course of this research, some issues and challenges were encountered such as: 1. The SMS spam dataset used for the research was highly imbalanced. The dataset contained 5574 records in all out of which 747 records were spam and the rest were legitimate records. This resulted in a disparity between the spam and the ham classes

and the ratio was extremely uneven. This was handled by applying the down-sampling technique to the ham class to ascertain the records were equal in ham as well as spam class. This reduction in the record size of the ham class could have impacted the training of the machine learning models used for this research.
2. Due to the down-sampling technique, the effective record size used for training the machine learning models was very limited. The performance achieved using this approach may vary when the data size grows exponentially. In the future, this can be tackled by aggregating different datasets and training the models again for better results.
3. The research in its current form is confined to the English language only as the dataset used contained English words. Also, the models are trained to identify proper English language words and not the slang and short forms. The result may vary when non-English words, slang language or short forms are used.

6 Conclusion and Future Work

Spam SMS is a grave threat and it is getting more and more serious with each day. It can cause significant harm and the consequences can be drastic. Countering this menace with high accuracy and low latency was the main motivation behind this research. A sample dataset was used to find an effective solution to the above problem. In the initial stage, Exploratory Analysis was conducted on the dataset wherein it was established that the length feature was a contributing factor in identifying the ham and spam. This also revealed that there was a high imbalance in the ham and spam class of the dataset. This was taken care of by the down-sampling technique to match the ham and spam class counts. Data pre-processing and cleaning was done to reduce the noise from the data. Furthermore, the features were extracted using the Bag of Words and TF-IDF models. To achieve low latency, the extracted features were selected using the Chi-Square feature selection technique. Then the machine learning models Bernoulli Naïve Bayes, LightGBM, and XGBoost were applied along with the traditional base models SVM and Random Forest.

The research objectives were accomplished and the spam SMS were filtered with high accuracy within a short time. Hence the research can be termed successful. The results section demonstrated that the suggested models like Bernoulli Naïve Bayes and LightGBM combined with TF-IDF were apt for solving the research question since they produced an accuracy of 96.5% and 95.4% respectively. Also, the time taken by these models was 0.157 and 1.708 seconds which was significantly better than the other traditional models.

Future Work: In an endeavor to improve on the results, the machine learning models should be trained using datasets from different sources and also using datasets having a large number of records. This will improve the reliability of the models. The millennial in today's time use slang and short forms in texting which cannot be detected by the models at present. This can be improved upon to better classify the genuine ham messages from the spurious ones. In-depth research can be conducted on this. Also, non-English languages can be included for spam SMS detection in the future. Apart from Chi-Square, other techniques like Information Gain, Gini Index, etc. can be used to evaluate the impact on performance.

7 Acknowledgment

I would like to extend my heartfelt gratitude towards my mentor Dr. Catherine Mulwa. This research would not have been possible without her guidance, help, and support. She always went the extra mile to help me whenever I was in need. I would also like to thank my family for bestowing their trust in me.

References

Agarwal, S., Kaur, S. and Garhwal, S. (2016). SMS spam detection for Indian messages, *Proceedings on 2015 1st International Conference on Next Generation Computing Technologies, NGCT 2015* (September): 634–638.

Aich, P., Venugopalan, M. and Gupta, D. (2019). Content based spam detection in short text messages with emphasis on dealing with imbalanced datasets, *Proceedings - 2018 4th International Conference on Computing, Communication Control and Automation, ICCUBEA 2018* .

Akbari, F. and Sajedi, H. (2015). SMS spam detection using selected text features and Boosting Classifiers, *2015 7th Conference on Information and Knowledge Technology, IKT 2015* pp. 1–5.

Almeida, T. A., Almeida, J. and Yamakami, A. (2011). Spam filtering : how the dimensionality reduction affects the accuracy of Naive Bayes classifiers, *Journal of Internet Services and Applications* **1**: 183–200.

Balli, S. and Karasoy, O. (2018). Development of content based sms classification application by using word2vec based feature extraction, *IET Software* .

Basu, A., Watters, C. and Shepherd, M. (2002). Support Vector Machines for Text Categorization, *Proceedings of the 36th Hawaii International Conference on System Sciences* pp. 1–7.

Ergin, S. and Isik, S. (2014). The assessment of feature selection methods on agglutinative language for spam email detection: A special case for Turkish, *INISTA 2014 - IEEE International Symposium on Innovations in Intelligent Systems and Applications, Proceedings* pp. 122–125.

Fernandes, D., d. Costa, K. A. P., Almeida, T. A. and Papa, J. P. (2015). Sms spam filtering through optimum-path forest-based classifiers, pp. 133–137.

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.

Hazim, M., Anuar, N. B., Ab Razak, M. F. and Abdullah, N. A. (2018). Detecting opinion spams through supervised boosting approach, *PLoS ONE* **13**(6): 1–24.

Islam, M. S., Khaled, S. M., Farhan, K., Rahman, M. A. and Rahman, J. (2009). Modeling spammer behavior: Naïve Bayes vs. artificial neural networks, *2009 International Conference on Information and Multimedia Technology, ICIMT 2009* pp. 52–55.

Jindal, N. and Liu, B. (2007). Analyzing and detecting review spam, pp. 547–552.

Kim, S. E., Jo, J. T. and Choi, S. H. (2015). SMS Spam filterinig using keyword frequency ratio, *International Journal of Security and its Applications* **9**(1): 329–336.

Koray, O., Buber, E., Demir, O. and Diri, B. (2019). Machine learning based phishing detection from URLs, *Expert Systems With Applications* **117**: 345–357.
URL: *https://doi.org/10.1016/j.eswa.2018.09.029*

Lee, C. N., Chen, Y. R. and Tzeng, W. G. (2017). An online subject-based spam filter using natural language features, *2017 IEEE Conference on Dependable and Secure Computing* pp. 479–484.

Li, Y., Yang, Z., Chen, X., Yuan, H. and Liu, W. (2019). A stacking model using URL and HTML features for phishing webpage detection, *Future Generation Computer Systems* **94**: 27–39.
URL: *https://doi.org/10.1016/j.future.2018.11.004*

Liew, S. W., Sani, N. F. M., Abdullah, M. T., Yaakob, R. and Sharum, M. Y. (2019). An effective security alert mechanism for real-time phishing tweet detection on Twitter, *Computers and Security* **83**: 201–207.
URL: *https://doi.org/10.1016/j.cose.2019.02.004*

Longadge, R. and Dongre, S. (2013). Class imbalance problem in data mining review, *Int. J. Comput. Sci. Netw.* **2**.

Ma, J., Zhang, Y., Liu, J., Yu, K. and Wang, X. (2016). Intelligent sms spam filtering using topic model, *2016 International Conference on Intelligent Networking and Collaborative Systems (INCoS)* pp. 380–383.

Minastireanu, E.-A. and Mesnita, G. (2019). Light GBM Machine Learning Algorithm to Online Click Fraud Detection Light GBM Machine Learning Algorithm to Online Click Fraud Detection, *Journal of Information Assurance & Cybersecurity* **3**(April): 1–15.

Najadat, H., Abdulla, N., Abooraig, R. and Nawasrah, S. (2014). Mobile SMS Spam Filtering based on Mixing Classifiers, *International Journal of Advanced Computing Research* **1**.

Pham, T.-h. (2016). Content-based Approach for Vietnamese Spam SMS Filtering, *2016 International Conference on Asian Language Processing (IALP)* pp. 41–44.

Prieto, A., Prieto, B., Ortigosa, E., Ros, E., Pelayo, F., Ortega, J. and Rojas, I. (2016). Neural networks: An overview of early research, current frameworks and new challenges, *Neurocomputing* **214**.

Ren, Y. and Ji, D. (2017). Neural networks for deceptive opinion spam detection: An empirical study, *Information Sciences* **385-386**: 213–224.

Sethi, P., Bhandari, V. and Kohli, B. (2018). SMS spam detection and comparison of various machine learning algorithms, *2017 International Conference on Computing and Communication Technologies for Smart Nation, IC3TSN 2017* **2017-October**: 28–31.

Shafique, U. and Qaiser, H. (2014). A comparative study of data mining process models (kdd, crisp-dm and semma), *International Journal of Innovation and Scientific Research* **12**: 2351–8014.

Shirani-Mehr, H. (2012). SMS Spam Detection using Machine Learning Approach, *tech. rep., Stanford University* pp. 1–4.

Tianqi, C. and Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System Tianqi, *Il Friuli medico* **19**(6).

Vani, K. and Gupta, D. (2014). Using K-means cluster based techniques in external plagiarism detection, *Proceedings of 2014 International Conference on Contemporary Computing and Informatics, IC3I 2014* pp. 1268–1273.

Xu, S. (2018). Bayesian naïve bayes classifiers to text classification, *Journal of Information Science* **44**(1): 48–59.
URL: *https://doi.org/10.1177/0165551516677946*

Yadav, K., Saha, S. K., Kumaraguru, P. and Kumra, R. (2012). Take control of your SMSes: Designing an usable spam SMS filtering system, *Proceedings - 2012 IEEE 13th International Conference on Mobile Data Management, MDM 2012* pp. 352–355.

Yu, B. and ben Xu, Z. (2008). A comparative study for content-based dynamic spam classification using four machine learning algorithms, *Knowledge-Based Systems* **21**(4): 355–362.

Yuan, H., Chen, X., Li, Y., Yang, Z. and Gv Liu, W. (2018). Detecting Phishing Websites and Targets Based on URLs and Webpage Links, *Proceedings - International Conference on Pattern Recognition* **2018-August**: 3669–3674.

Yue, Y. and Elfayoumy, S. (2007). Anti-spam filtering using neural networks and Baysian classifiers, *Proceedings of the 2007 IEEE International Symposium on Computational Intelligence in Robotics and Automation, CIRA 2007* pp. 272–278.