National College of Ireland

Detecting Spam Campaigns on Twitter Using Machine Learning Approach

MSc Internship
Cyber Security

Adedoyin Alaba
Student ID: 19221436

School of Computing
National College of Ireland

Supervisor: Michael Pantridge

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: ……. Adedoyin Alaba……………………………………………………………………

Student ID: ……19221436……………………………………………………………….

Programme: …………Cyber Security……………………… **Year:** ……2021…………

Module: ………………………MSc Internship…………………………………………………

Supervisor: ………Michael Pantridge……………………………………………………………

Submission Due Date: ……………16/08/2021……………………………………………………………………

Project Title: Detecting Spam Campaigns on Twitter Using Machine Learning Approach

Word Count: ……………………6,577…………… **Page Count**……………..18…………………

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.
I agree to an electronic copy of my thesis being made publicly available on NORMA the National College of Ireland's Institutional Repository for consultation.

Signature: ………………………………………………………………………………………………………

Date: …16/08/2021………………………

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):	

Detecting Spam Campaigns on Twitter Using Machine Learning Approach

Adedoyin Alaba
19221436

Abstract

A secure voting system entirely depends on modern technology. Cybersecurity guards against election violence and manipulation. Spammers have taken advantage of the huge popularity of social media to distribute spam messages because it takes advantage of relationships between users. Social spamming is more effective than conventional techniques like email spamming. One of the most significant reasons is that social media assist in the development of intrinsic trust relationships between online buddies, even if they do not know each other in person. Detecting spam is the first and most important step in combating spam. This study is focused on Twitter, and proposes a novel, effective approach to detect and filter unwanted tweets, complementing earlier approaches in this direction. Previous studies rely on historical features of tweets that are often unavailable on Twitter after a short period of time, hence not suitable for real-time use. This study approached an optimized set of readily available features, independent of the historical textual features on Twitter. This paper focuses on identifying SPAM tweet patterns used during elections by assembling two machine learning algorithms and applying them on the combination of unigram and bigram words to produce a better accuracy of 99.2%.

Keywords: Spam, Twitter, Elections, Machine Learning, Tweet

1 Introduction

Technology has continued to advance at an exceptional pace and the only way to secure the modern voting system is using technology. This means that cybersecurity has become a part of the remedy to forestall manipulation and violence throughout the entire election process [1]. In the 2020 elections in Ukraine and Indonesia, phishing attacks threatened the moderately normal and long running democratic systems [2]. Government groups and other private actor now use phishing attacks to target democratic and other financial institutions [3].

Spammers have taken advantage of the huge popularity of social media to distribute spam messages because it takes advantage of relationships between users. Social spamming is more effective than conventional techniques like email spamming. One of the most significant reasons is that social media assist in the development of intrinsic trust relationships between online buddies, even if they do not know each other in person. As a result, people are more likely to consume the messages received or casually click on links from their online acquaintances. Because of this, spammers have taken advantage of social media and uploaded harmful or spam material to reach more people. Detecting spam is the first and most important step in combating spam. The social media for consideration will be Twitter. With over 200 million users, Twitter is one of the most popular microblogging sites.

Social media is no doubt one of the standout phenomena in an era that is highly technologically driven. Social media channels like including Instagram, Twitter and Facebook have become quite involved in improving the connectivity of everyone on earth. An estimated 2.46 billion users are said to be connected in 2020. This is equivalent to one-third of the population of the world[1]. Social media users can generate and consume information as they freely choose, and this results in enormous amounts of data. The importance of social analytics in enhancing productivity and improving the competitive advantage is obvious in different forms. Social media provides information which can be used in healthcare support for delivery of service, to engage fans in sports, for enhancing consumer experience in entertainment, complement business experience and intuition when taking decisions, and during the entire election process such as enhancing a wider supporter engagement and following of results and foretelling the outcome of pools [4, 5, 6, 7]. With all these benefits, spam generated using social media has cast doubts on the credibility of studies performed on this data analysis [8]. The Nexgate[2] report estimates an average of one spam post going live in every 200 posts made on a social media platform. More so, another new study reported that almost fifteen percent of Twitter user accounts that are currently active are merely automated bots masquerading as humans. Autonomous account otherwise called social bots and spam posts keeps growing and this poses a lot of concerns around the veracity, trustworthiness, and representation of data available for research [9].

In this research, Twitter is the focus and a novel and effective approach is proposed to not only detect but to filter out spam tweets. This will complement similar earlier progress made in doing this. However, past research relied on history feature of tweets, but this are seldomly available on Twitter after a while and this makes it unsuitable for real-time utilization. In this research, a different approach is taken which is an optimizes set of readily available features, that does not depend on Twitter's historical textual features. Features that have been selected were grouped as related to the Twitter account and the user (pairwise interaction between users). Two machine learning languages were trained, and a recursive feature used to eliminate to determine the strength and discriminatory power of each feature. Hence, when compared to an earlier study, the current study proposes features that demonstrate a more potent discriminative power with more stable performance in the different models.

This paper focuses on identifying SPAM tweet patterns used during elections using Naive Bayesian Classifier and Xgboost algorithms then combining these algorithms to give a better accuracy which in turns answers the research question: **How accurate are Naive Bayesian Classifier and Xgboost algorithms in detecting SPAM tweet patterns that may be employed during election campaigns? This fills the gap in literatures by conducting a supervised reinforcement approach of machine learning algorithms to detect SPAM campaigns that use botnets on Twitter.**

Following this introduction, this paper is organized into Section 2 which is an overview of related work in this area, then Section three which gives a brief on the research methods. Then Section four shows the design specification while Section five is all about the implementation

[1] Social media statistics and facts, Online: http://www.statista.com/topics/1164/ social-networks , Accessed: 04-08-2021.

[2] NexGate, State of Social Media Spam Research Report, NexGate. 2013.

of the model. Section 6 presents the evaluation and Section 7 concludes the work and elaborates on future work.

2 Related Work

2.1 Spam Campaigns

In addition to malware distribution, posting commercial URLs and false or abusive content, automating the creation of enormous of media creatives [10] and follow or tagging users in random [11] are all examples of online spamming activities. Social bots and machine learning models are other forms of online spamming [12]. With an estimated growth rate of 355 percent in 2013[2], spam is on the rise worldwide. Every 21 tweets on Twitter are spam and about 15% of users that are active are independent vehicles, i.e., social bots [14]. Due to the lack of physical contact between the communicating parties, spam volume has increased. Users' real identities and the legitimacy of their postings are therefore difficult to verify because social media data is often unrepresentative, it is obvious that relying on it without effective filtering could lead to inaccurate analysis and incorrect conclusions. Spammers, on the other hand, are constantly evolving to elude detection systems. Therefore, some strategies may become obsolete and ineffective as spammers develop new tricks.

2.2 Machine Learning

As a form of Artificial Intelligence, this is a technique that allows a computer system to acquire information without explicit programming. Most of the work involves developing and implementing software that enables computer programs to access data and self-learn. First, we examine our data to identify trends that will help us in improved decision making. The main objective is to activate systems to learn automatically, without human intervention. Although consciousness is not a prerequisite for information gathering of information, those patterns which the data presents are quite familiar. Thus, a human is not required in the learning process when machine learning algorithms such as supervised learning techniques, unsupervised learning techniques, semi-supervised learning techniques, and reinforcement learning techniques are applied [15]. Predictions are made with the help of supervised learning techniques, which use previous and current data and labels to make inferences. In unsupervised learning, the training data is neither classified nor numbered. Using both labelled and unlabelled data in the 4-training method, semi-supervised learning techniques fall between supervised and unsupervised machine learning techniques. Learning strategies that involve action and recognition of faults are known as reinforcement learning strategies.

2.3 Review of Related Works

SPAM can be described as actions that hurts or hinders other users online any form of activity that harms or disrupts other online users, regardless of how it is delivered. Social bot accounts are a prime example of unreliable sources of information that humans are inclined to spread. [16] have recently discovered that both true and false news spread at the same speed. In a short period of time, false information spread on Twitter. Social bots are used to speed up the process, and human users help to spread the word further through their social networks. Diverse techniques are discussed in this section to detect spam tweets. As part of his pioneering work on spam detection, [17] made use of graph models which were directed for the analysis of friend-follower relationships via Twitter and set out sets of features for detection of SPAM. The following categories of approaches to spam detection are commonly used: social graph analysis [18 - 20], activity patterns and text analysis [21], profile analysis of user meta-data analysis [21, 22, 23], interaction analysis [24, 10, 25], and the effect of URL blacklisting.

Each tweet can contain only one hundred and forty characters (it is now two hundred and eighty characters), this has led to the proliferation of URL shortening services [26]. Researchers [21, 22] and [23] examined the use of obfuscated URLs by spammers to take users to sites with malicious intents. They did this by analysing the URL streams. Consequently, the examination of URL streams was examined by [24] and even though approaching the problem using this method is effective, it is slow and lacks the ability to effectively identify malicious intending URLs. Additionally [22] studied URL usage to identify Facebook spamming activities and determined that spamming of this nature was related to commercial accounts but not those which are spam focused. Another study by [23], researched the statistical properties in accounts of users and how spam detection is affected by shortening services. Though Twitter services allow URLs and use URL shortening, it is cumbersome to pinpoint links with the potential to be malicious in bulk. Even so, many URLs detection are dependent on historical information, and this hinders in attempt at detection and analysis in real-time. Social network modelling was used by [20] to suggest legitimate user accounts which are managed by malicious users. Examining the posting of SPAM, [24] developed a social honeypot account group that imitated naïve group of Twitter users. Engagement with this honeypot group means that users could be violating the usage policy of Twitter. The link payloads and other features used to capture the follower-follower network dynamics were the basis for the identification and categorization of different types of users. A system for detection of social bot accounts using various features related to content, the network and user was produced by [10]. When [25] analysed tweets for SPAM, words that were probably going to appear in SPAM and non-SPAM tweets were sort for. A study carried out to perform an in-depth analysis of words that are intended to be deceptive and used by spammers on Twitter was carried out by [27]. The concept known as Twitter SPAM Drift is produced by SPAM makers who adopt and remove various tricks that are evasive in nature in perpetuity. Subsequently, machine learning classifiers were trained using similar phenomenon. [29] showed that detection can be avoided when datasets that are unbalanced are used. As it relates to the behaviour of SPAM, basic machine learning methods may be considered inadequate in most cases. The deep learning technique of Word2Vec [31] was used by [30] to capture the differences of challenges related to SPAM. However, those methods that solely rely on information that are text-based are not sufficient to differentiate between a notorious SPAM sending account and a non-SPAM sending account. Many techniques can be used to identify bots which are determined to influence the way discussions on Twitter [13]. These classes of autonomous entities seek to obtain a place of authority in on-going or fresh discussions and then generate false or unreliable data.

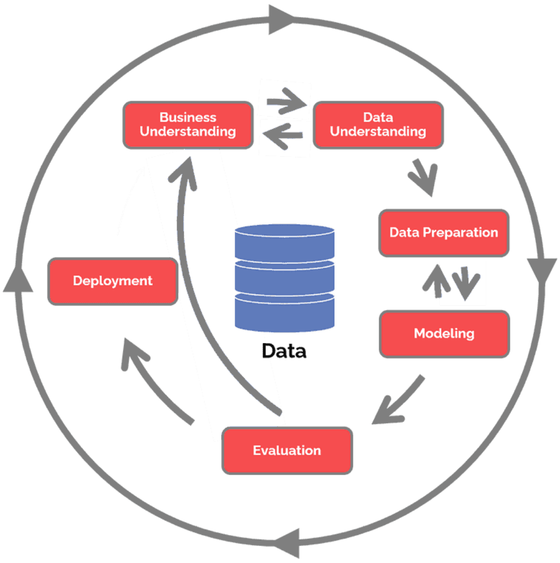
For the most part, those who have studied spam detection have relied upon extracting features from an individual's tweet history [33] or learning a limited number of features through unsupervised techniques. For enhanced execution and broader pertinence, the proposed approach of this study depends on available real-time features.

3 Research Methodology

SPAM tweets are created and deployed with the sole aim of gaining trust and spreading misinformation. During an election campaign, these tweets have the potential to sway the decision of voters and it is best to have a system that can detect them in real time.

Cross-industry method CRISP-DM is referred to as CRISP-DM and this was applied in this study because it is a solid, tried-and-true data mining method. The CRISP-DM methodology offers a systematic method that is used to plan any data mining endeavour. A project's phases, tasks, and roles, as well as the relationships between these roles, are described in detail in this document.

Figure 1: CRISP-DM Methodology



Although the CRISP-DM methodology involves 6 stages, these stages are flexible and allows for going back and forth between the different levels[3].

The first state involves the determination of the research objectives, and it needs adequate knowledge about what the task to be accomplished is trying to achieve. It is in this stage that the cost of the effort is determined and the proper tools to undertake the task are selected. The next phase is the data understanding phase. Here the data collected initially is described, explored the quality is verified. The data munging or preparation phase comes next and, in this phase, the final data set(s) for modelling are handled in five separate tasks – select, clean, construct, and format. Immediately following the data preparation phase is the fourth stage called modelling. Data modelling has four tasks – select the best modelling technique, generating the test data, build the model, and access the model. The penultimate stage is the stage where there is an evaluation of results after which the entire process is reviewed before the next step to take is determined. In the sixth and final stage called the deployment, this stage

[3] https://www.datascience-pm.com/crisp-dm-2/

is planned while plans are also made for maintenance and monitoring. In this stage, the final report is produced, and the entire process is then reviewed.

Based on the above, SPAM tweets can be identified with the help of supervised learning algorithms such as Naïve Bayesian classifier, Xgboost, decision tree classifiers, support vector machines and logistic regression classifiers, which use standardized header information as well as tweet body information. SPAM tweets are typically detected using classification, which involves inserting or removing features extracted from the text.

Naïve Bayesian Classifier

This classifier technique is based on Bayesian theorem, and it performs better when the dimensionality of data is high[4]. The Bayesian classifier can calculate the most possible output based on the input. It allows for the addition of new data to at run time and gives a better probabilistic classifier. This is a probabilistic machine learning algorithm[5].

This algorithm allows for the presence of a particular feature in a class even when it is unrelated to the presence of any other feature.

Bayesian theorem provides an equation for calculating posterior probability p(c|x) from p(c), p(x) and PP(x|c):

$$p(c|\ x) = \frac{p(c\ |\ x)p(c)}{p(c)}$$

- **p(c|x): the posterior probability of class (c, target) given predictor (x, attributes).**
- **p(c): the prior probability of class.**
- **p(x|c): the likelihood, which is the probability of predictor given class.**
- **p(x): the prior probability of predictor.**

Above equation demonstrates the number of times A occurs when B happens, this is shown as P(A|B) also called posterior probability. We now know how often B happens if A happens, presented as P(B|A) and how likely A is on its own, written P(A) and how likely B is on its own, written P(B).

What it does to classifier a data record in our case spam or no spam, the posterior probability is computed for each class.

XGBoost

It has recently dominated competitions on Kaggle and other sites that deal with structured or tabular data using XGBoost's applied machine learning algorithm[6].

It is a high-performance execution of gradient-boosted decision trees. Structured or tabular datasets are no match for XGBoost when it comes to classification and regression predictive

[4] https://towardsdatascience.com/naive-bayes-explained-9d2b96f4a9c0

[5] https://www.kdnuggets.com/2020/06/naive-bayes-algorithm-everything.html

[6] https://machinelearningmastery.com/gentle-introduction-xgboost-applied-machine-learning/

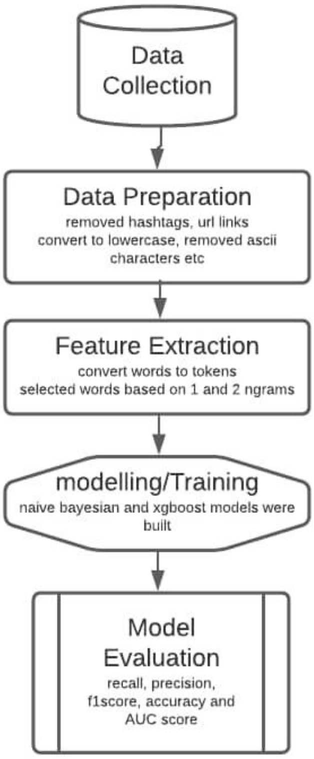
modelling. A decision tree algorithm based on gradient boosting is implemented in the XGBoost library.

New models are added to existing models as part of an ensemble technique known as boosting. Sequentially, new models are added until no further improvements can be made to the system. Unpopular algorithm AdaBoost weights data points that are difficult to predict, as an example. New models are created that predict the residuals or errors of previous models, and then they are added together to make the final prediction in a process known as gradient boosting. For example, when adding new models, the algorithm utilizes a gradient descent algorithm to reduce losses.

4 Design Specification

As shown in the activity diagram in figure 2, the process flow that works best to answer the research question. From the data collection to the implementation of the results, this is the execution flow. Below is a flowchart of the steps taken to implement the model.

Figure 2: The Proposed Model Architectural Design


Data Collection
Data Preparation
removed hashtags, url links
convert to lowercase, removed ascii characters etc
Feature Extraction
convert words to tokens
selected words based on 1 and 2 ngrams
modelling/Training
naive bayesian and xgboost models were built
Model Evaluation
recall, precision, f1score, accuracy and AUC score

4.1 Data Collection

In the first step, we created a dataset by randomly gathering the data from twenty (20) election bloggers, political parties, political office holders twitter accounts after which a hundred (100) followers per account were collected. This became the base point of collecting tweets from the identified twitter users.

The bloggers were selected based on the number of followers and twitter ranking. Furthermore, the reputation of the bloggers was checked based on the formular:
number of followers / (number of followers + number of following).

Thereafter, the tweets were manually labelled as ‘spam’ or ‘non-spam’. The process of manual labelling of tweets was based on identified keywords that have been reported by both Twitter and researchers[7] to be most common and used among spammers. Words like $$$, 100%, Act now, Action, Additional income, Affordable, cheap, Amazed, Apply now, Be amazed be your own boss etc has been flagged as making up spam tweets by researchers. When a tweet in our dataset contains at least four or more of these key words it is labelled as a spam tweet. Furthermore, to enrich the model, the user-based features such as followers count, friends count, user reputation etc, were considered to label each tweet.

Other methods used in manually classifying tweets as spam or non-spam include the number of URLs contained the 50 most recent tweets of the identified accounts, a baseline of 50 keywords identified in 50 most recent tweets and the level of retweets found within the 50 most recent tweets (a high number of retweets indicates that the account is a spammer).

In some N-gram analyses, words that frequently appear together are examined. A contiguous sequence of n items from a particular sample of text or speech is referred to as an n-gram.

A one-word sequence (or unigram) is called a 1-gram (or unigram). For the sentence, "I love reading blogs about data about science on Analytics", the unigrams would be: "I", "love", "reading", "blogs", "about", "data", "science", "on", "Analytics", and "Vidhya".[8]

In terms of grammar, a 2-gram (or bigram) is a two-word sequence of words, like “I love”, “love reading”, or “Analytics Today. A trigram (or three-gram) is a sequence of three words, such as “I love reading”, “about data science”, or “on Analytics today.

Some stop words, the most common words, are no longer important, as in a sentence they serve only as a connecting link to other words rather than conveying information about the sentence or situation (e.g., “a”, “the”, “and”, “but”, and so on). Using the NLTK Stopword Dictionary, we are provided with a list of the most common stopwords.

4.2 Data Pre-processing

To prepare data for training, it must first be cleaned and prepared. In this stage, the information is arranged and formatted in such a way that data is ready for training. Training information must be correct, complete, and appropriate after pre-processing data. Complete or raw data sent to a model can cause a variety of errors. These errors can lead to a much lower total accuracy in the long run. Data Pre-processing is the stage in every machine learning process where data is transformed or encoded so that the machine can easily parse it.

When using a text-based data, the first step was to remove all unwanted strings present in the data. For this Regex was used to remove "https tags", alpha numerical etc. before eliminating stop words because these will limit the model from distinguishing between real words. Then stemming was applied to simplify terms to their source words. These keywords are matched

[7] https://www.activecampaign.com/blog/spam-words

[8] https://towardsdatascience.com/text-analysis-basics-in-python-443282942ec5

against a pre-defined set of SPAM words. Where the words match, then the tweet is classified as SPAM.

4.3 Feature Extraction

In machine learning, feature selection is a way to prune down features by selecting subsets of the necessary features. Origin features and content features are the two categories of tweet profiling features that were used. Tweet headers contain information about the tweet sender, including the Twitter handle, display name, IP address, and location. All these information can be used to determine who sent the tweet. In this research this was done using two (2) methods CountVectorizer and TF-IDF Transformer.

Scikit-CountVectorizer learn's is a great tool. Based on the frequency of each word in the text, it is used to transform a given text into a vector. We can use this feature when we have many such texts, and we want to convert each word into a vector (for using in further text analysis). While IDF (Inverse Document Frequency) is an information retrieval and information extraction subtask that is used to show the relative significance of a word within a corpus. Search engines use it to get better results that are more relevant to a query.

With this approach text was converted to digits and then ready for modelling.

4.4 Modelling

To access the output of the model when the model was being designed, the data was divided into training and testing data after they were collected and prepared for evaluation and analysis in a 70 and 30 percent portions, respectively. Using Python libraries like Scikit-Learn and Natural Language Toolkit, Pandas, Matplotlib and Seaborn, classifiers such as Naive Bayesian and Xgboost as proposed for this research.

4.5 Evaluation

In this case, machine learning was used to identify spam tweets. An objective analysis of the work, its attributes, and its performance was also included in the evaluation, which demonstrates the influence and effectiveness of the study.

5 Implementation

In this section, the implementation tools and the methods are discussed. The technologies and tools employed in developing the model are also elucidated.

5.1 Python

Python is an object-oriented, high-level programming language that is interpreted and has dynamic semantics, according to the Python website. Fast application development is made possible by its high-level built-in data structures combined with Dynamic Typing and Dynamic Binding. In addition, Python's readability is enhanced by its simple, easy-to-learn syntax[9]. Modules and packages are supported in Python, which encourages program modularity and

[9] https://www.python.org/doc/essays/blurb/

code reusability. Neither the Python interpreter nor the vast Python standard library are charged for and can be freely distributed.

5.2 Libraries

Since Python is so flexible, many developers have turned to it to create their own machine learning libraries. Machine learning experts are increasingly turning to Python because of its large number of libraries such as TensorFlow, Scikit-Learn, NumPy, Keras, PyTorch, etc. The following are some of the libraries employed for this project:

Pandas: Data structures that make working with "relational" or "labelled" data simple and intuitive are provided by pandas, a Python package. With this module, Python users can perform practical data analysis in a high-level environment. Pandas was used for data manipulation in this study.

Scikit-Learn: This was used for the data pre-processing and features extraction stages. It was previously called scikits.learn and is now known as sklearn. NumPy and SciPy numerical and scientific libraries are designed to interoperate with it.

Tweepy: It's an open-source Python package that makes it easy to use Twitter's API with Python. Among Tweepy's classes and methods are those that represent Twitter's models and API endpoints. Tweepy was used during the data collection stage to extract tweets from Twitter.

Matplotlib and seaborn: This were used during the data visualization stage.

The Matplotlib package is a Python graphics package that allows data to be visualized in graphical form. Pandas and NumPy can be easily integrated into the program. The MATLAB plotting commands are closely mirrored by the pyplot module. In this way, MATLAB users can easily switch to plotting with Python after learning the language.

Working with Pandas data frames is easier using Seaborn. As an extension to the Matplotlib graphics library, it provides a more straightforward set of methods for creating beautiful graphics in Python.

Xgboost – The data modelling stage was accomplished using Xgboost.

Jupyter Notebook

This open-source web application gives the user the option to create code and share equations, visualizations, and informative text documents in real time. Live code is used. Machine learning, data cleaning, numerical simulation and mathematical modelling are among its applications.

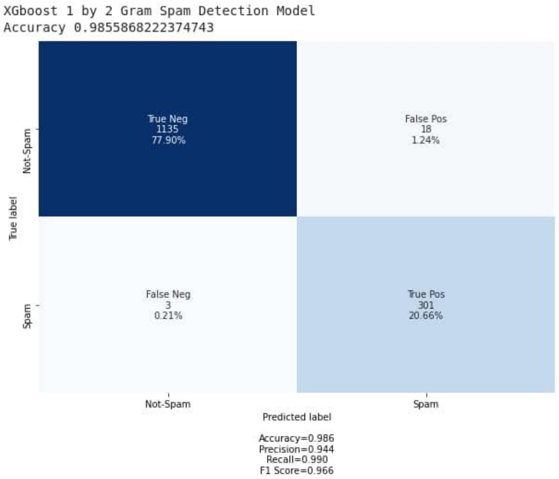
6 Evaluation

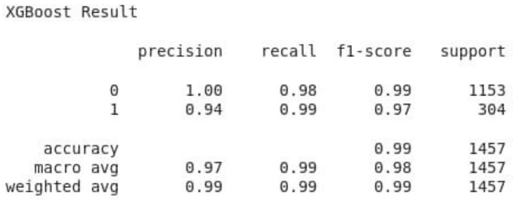
This chapter will present the results of the model's effectiveness. The experiments on the datasets were carried out. To determine how well the proposed model works in detecting SPAM tweets, in real time scenarios. Every classifier was tested against every other classifier, and the two classifiers that yielded the best results were merged to produce the model.

6.1 Experiment 1: Xgboost Detection Model

The result of the Xgboost model is shown in the diagrams below, the algorithm was able to get accuracy, precision, recall, F1-score of 98.6%, 94.4%, 99% and 96.6% respectively.

Figure 3: Xgboost Classifier Accuracy, Precision, Recall, F1-score, and ROC Curve



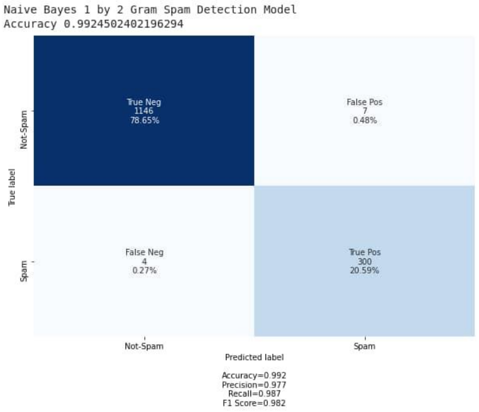


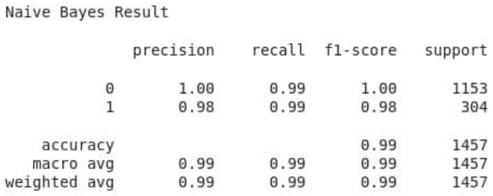
```
XGBoost Result

              precision    recall  f1-score   support

           0       1.00      0.98      0.99      1153
           1       0.94      0.99      0.97       304

    accuracy                           0.99      1457
   macro avg       0.97      0.99      0.98      1457
weighted avg       0.99      0.99      0.99      1457
```

6.2 Experiment 2: Naïve Bayes

The result of the Naïve Bayes model is shown in the diagrams below, the algorithm was able to get accuracy, precision, recall, F1-score of 99.2%, 97.7%, 98.7% and 98.2% respectively.

Figure 4: Naïve Bayes Classifier Accuracy, Precision, Recall, F1-score, and ROC Curve





```
Naive Bayes Result

              precision    recall  f1-score   support

           0       1.00      0.99      1.00      1153
           1       0.98      0.99      0.98       304

    accuracy                           0.99      1457
   macro avg       0.99      0.99      0.99      1457
weighted avg       0.99      0.99      0.99      1457
```

The results in table 1 show that the chosen algorithms Xgboost and Naïve Bayes have accuracies of 98.6% and 99.2% respectively. These were now used to ensemble the model.

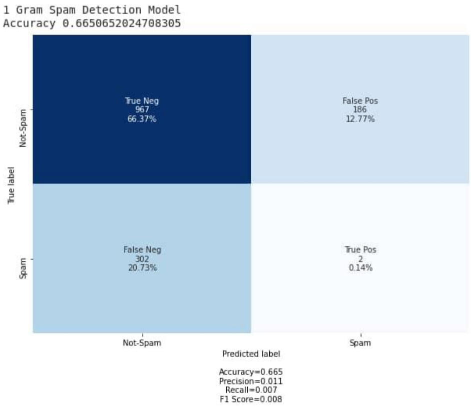
Table 1: Comparison of Results from All Models

Algorithm	Accuracy	Precision	Recall	F1 Score
Xgboost	98.6	94.4	99	96.6
Naïve Bayes	99.2	97.7	98.7	98.2

6.3 Experiment 3: Model Application on Unigram Words

The result of the ensemble of the classifiers when ensembled and applied on unigram words is shown in the diagram below, the algorithm was able to get accuracy of 66.5%, with a false negative of 12.77% and false positive of 20.73%. The precision, recall, F1-score of 0.1%, 0.7%, 0.8% respectively.

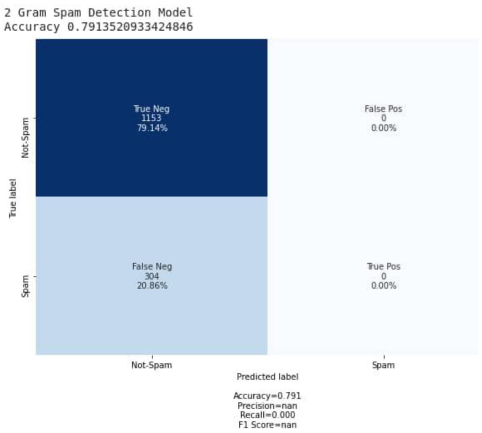
Figure 5: Classifier Ensemble on Unigram Words Accuracy, Precision, Recall, F1-score, and ROC Curve



6.4 Experiment 4: Model Application on Bigram Words

The result of the ensemble of the classifiers when ensembled and applied on bigram words is shown in the diagram below, the algorithm was able to get accuracy of 79.1%, with a false negative of 20.86% and false positive of 0%. The precision, recall, F1-score of 0%.

Figure 5: Classifier Ensemble on Bigram Words Accuracy, Precision, Recall, F1-score, and ROC Curve

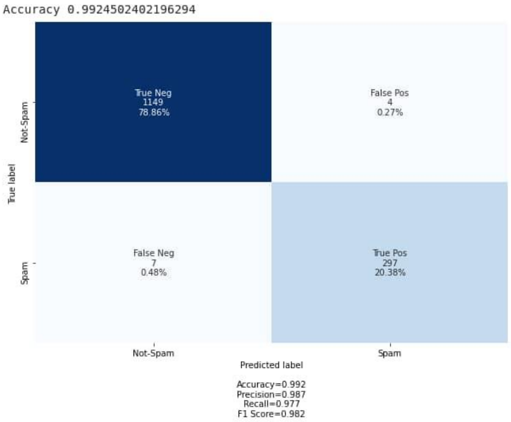


6.5 Experiment 5: Model Application on Unigram & Bigram Words

The result of the ensemble of the classifiers when ensembled and applied on both unigram and bigram words is shown in the diagram below, the algorithm was able to get accuracy of

99.2%, with a false negative of 48% and false positive of 27%. The precision, recall, F1-score of 98.7%, 97.7% and 98.2% respectively.

Figure 6: Classifier Ensemble on both Unigram and Bigram Words Accuracy, Precision, Recall, F1-score, and ROC Curve



6.6 Discussion

To answer the research question: **How accurate are Xgboost and Naïve Bayes algorithms in detecting SPAM tweets using in election campaigns?** It was found that two machine learning algorithms used in this research, Xgboost and Naïve Bayes produced the best accuracy, therefore the combination of the two algorithms into a single model. The ensemble was then applied on unigram and bigram words which resulted in an accuracy of 65.1% and 79% respectively.

The result in Table 2 shows that the ensembled algorithms when applied on the combination of unigram and bigram words gives a better accuracy of 99.2%, the false-positive rate of 0.27%, false-negative rate 48%, true positive of 20.38% and true negative of 78.86%.

Table 2: Comparison of the Unigram, bigram, and Unigram + bigram algorithms.

Ensembled Algorithm	Accuracy	True Negative	True Positive	F1 Score	False Negative	False Positive
Unigram	66.5	66.37	0.14	0.8	20.73	12.77
Bigram	79.1	79.14	0	0	20.86	0
Unigram + Bigram	99.2	78.86	20.38	98.2	48	0.27

7 Conclusion and Future Work

This study made use of supervised machine learning algorithms - Xgboost and Naïve Bayes to identify SPAM tweets. Both classifiers were combined to give better accuracy. The result demonstrated that when both classifiers were applied on a combination of unigram and bigram words it gave the accuracy of 99.2%. The suggested model considered identified SPAM keywords used in tweets and gave a good result. Hence, to answer the research question; How accurate are Xgboost and Naïve Bayes algorithms in detecting SPAM tweets used in election campaigns?

It is anticipated that future work will be carried out considering a bigger scope of dataset to further ascertain the validity of the system. The system is still in its infancy, and there is still room for further growth. One new and immediate danger is that in tweets, attackers are using malicious attachments to send the intended text. This method must be implemented in such a way that the text extracted from the tweet will be from both the text section and the attachment. File attachments may be an image file, not just a text file, with discernible text within it.

8 References

[1] S. N. a. A. N. Madheswari, "Prevention of phishing attacks in voting system using visual cryptography," in *2016 International Conference on Emerging Trends in Engineering, Technology and Science (ICETETS)*, 2016.

[2] M. V. P. M. H. P. a. D. N. M. M. M. V. Peter, "Finger Print Based Smart Voting System," *Asian J. Appl. Sci. Technol*, vol. 2, no. 2, pp. 357 - 361, 2018.

[3] B. Y. e. al., "Platform independent secure blockchain based voting system," in *International Conference on Information Security*, 2018.

[4] J. M.-G. M. S. D. C. E. Rojas, "Process mining in health- care: a literature review," *J. Biomed. Inf.*, vol. 61, p. 224–236, 2016.

[5] E. M. ,. C. A. K.C. Yee, "Perfect match? Generation Y as change agents for information communication technology implementation in healthcare," *Stud. Health Technol. Inf*, vol. 136, p. 496–501, 2008.

[6] T. Davenport, "Analytics in Sports: The New Science of Winning," *International Institute for Analytics, 2014 Technical report . White paper*, 2014.

[7] Deloitte, "Social Analytics in Media Entertainment the Three-minute Guide," *De- loitte Development LLC, 2014 Technical report*, 2014.

[8] B. C. ,. S. M. ,. L. S. ,. T. F. D. Contractor, "Tracking political elections on social media: applications and experience," in *Twenty-Fourth International Conference on Artificial Intelligence*, 2015.

[9] E. F. ,. C. .. D. ,. F. M. ,. A. .. F. O. Varol, "Online human–bot in- teractions: detection, estimation, and characterization," in *International AAAI Conference on Web and Social Media*, 2017.

[10] B. E. J. C. K. Lee, "Seven months with the devils: a long-term study of content polluters on twitter," in *Fifth International Conference on Weblogs and Social Media*, Catalonia, Spain, 2011.

[11] B. V. ,. J. C. ,. H. Z. ,. B. Z. Y. Yao, "Automated crowdturfing attacks and defenses in online review systems," in *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, Dallas, TX, USA, 2017.

[12] F. Braz, "A PROPOSAL FOR THE USE OF BLOCKCHAIN IN THE PORTUGUESE VOTING SYSTEM," *Universidade Nova de Lisboa,* 2021.

[13] A. G. M. S. M. P. a. s. K. B. Sujatha, "E VOTING APPLICATION USING BIOMETRICS & SMS OTP VERIFICATION," *Inf. Technol. Ind,* vol. 9, no. 3, pp. 256-260, 2021.

[14] N. U. P. S. T. W. a. M. S. A. A. Sabale, "UNIQUE FINGER IMPRESSION BASED SECURED VOTING FRAMEWORK USING IOT," 2021.

[15] N. L. E. K. a. A. P. M. Arapinis, "Definitions and Security of Quantum Electronic Voting," *ACM Trans. Quantum Comput,* vol. 2, no. 1, pp. 1-33, 2021.

[16] A. B. Ayed, "A Conceptual Secure Blockchain Based Electronic Voting System," *Int. J. Netw. Secur. & Its Appl,* pp. 01 - 09, 2021.

[17] M. A. S. a. J. A. Halderman, "Security Analysis of the Democracy Live Online Voting System," 2021.

[18] Z. D. a. A. Lee, "Technology and protest: The political effects of electronic voting in India," *Polit. Sci. Res. Methods,* vol. 9, no. 2, pp. 398 - 413, 2021.

[19] B. V. a. W. P. B. Kajal, "A Review of Online Voting System Security Based on Cryptography," *SSRN Electron. J.,* 2021.

[20] K. L. a. J. Punjwani, "Enhancing Electronic Voting With A Dual Blockchain Architecture," *Ledger,* 2021.

[21] Z. C. a. D. Subramanian, "An Unsupervised Approach to Detect Spam Campaigns that Use Botnets on Twitter," 2018.

[22] M. Bossetta, "A Simulated Cyberattack on Twitter: Assessing Partisan Vulnerability to Spear Phishing and Disinformation ahead of the 2018 U.S. Midterm Elections," *First Monday,* vol. 23, no. 12, 2018.

[23] M. A. S. a. J. A. Halderman, "Security Analysis of the Democracy Live Online Voting System," 2021.

[24] G. S. D. A. R. M. U. L. S. H. E. D. L. L. C. B. Himelein-Wachowiak M, "Bots and Misinformation Spread on Social Media: Implications for COVID-19," *J Med Internet Res,* vol. 23, no. 5, 2021.

[25] A. Wang, "Don’t follow me: spam detection in Twitter," 2010.

[26] R. H. ,. J. Z. ,. S. S. ,. G. G. C. Yang, "Analyzing spammers’ social net- works for fun and profit: a case study of cyber criminal ecosystem on twitter," in *Twenty-First International Conference on World Wide Web, WWW ’12,,* New York, NY, USA, 2012.

[27] M. K. ,. P. G. ,. A. F. H. Yu, "Sybilguard: defending against Sybil attacks via social networks," *IEEE/ACM Transa. Netw,* vol. 16, no. 3, p. 576–58, 2008.

[28] P. M. G. Danezis, "Sybilinfer: detecting Sybil nodes using social networks," in *Network and Distributed System Security Symposium (NDSS),* San Diego, CA, USA, 2009.

[29] J. H. ,. C. W. ,. Z. L. ,. Y. C. ,. B. Z. H. Gao, "Detecting and characterizing social spam campaigns," in *ACM SIGCOMM Internet Measurement Conference (IMC),* Melbourne, Australia, 2010.

[30] C. G. ,. J. M. ,. V. P. ,. D. S. K. Thomas, "Design and evaluation of a real-time URL spam filtering service," in *Thirty-Second IEEE Symposium on Security and Privacy (S&P),* Berkeley, CA, USA, 2011.

[31] J. K. S. Lee, "Warningbird: Detecting suspicious URLs in twitter stream," in *Nineteenth Annual Network and Distributed System Security Symposium (NDSS),* San Diego, CA, USA, 2012.

[32] G. M. ,. T. R. ,. V. A. F. Benevenuto, "Detecting spammers on twitter," in *Electronic Messaging, Anti-Abuse and Spam Conference, CEAS*, 2010.

[33] E. F. ,. C. .. D. ,. F. M. ,. A. .. F. O. Varol, "Online human–bot interactions: detection, estimation, and characterization," in *International AAAI Conference on Web and Social Media*, 2017.

[34] B. K. P.N. Howard, "Bots, #Strongerin, and #Brexit: Computational Propaganda During the UK-EU Referendum," 2016.

[35] T. Blog, "Giving you more characters to express yourself," 2018.

[36] K. T. ,. V. P. ,. C. Z. C. Grier, "@spam: the underground on 140 characters or less," in *Seventeenth ACM Conference on Computer and Communications Security (CCS),* Chicago, IL, USA, 2017.

[37] A. S. S. Sedhai, "Semi-supervised spam detection in Twitter stream," *IEEE Trans. Comput. Soc. Syst.,* vol. 5, no. 1, pp. 169 - 175, 2018.

[38] S. W. ,. J. Z. ,. Y. X. ,. J. O. ,. A. A. ,. M. H. C. Chen, "Investigating the deceptive information in Twitter spam," *Fut. Gen. Comput. Syst.,* vol. 72, p. 319–326, 2017a.

[39] Y. W. ,. J. Z. ,. Y. X. ,. W. Z. ,. G. M. C. Chen, "Statistical features-based real-time detection of drifted Twitter spam," *IEEE Trans. Inf. Foren. Secur,* vol. 12, no. 4, p. 914–925, 2017b.

[40] S. L. C. Li, "A comparative study of the class imbalance problem in Twitter spam detection," *Concurr. Comput. Pract. Exp,* vol. 30, no. 5, p. e4281, 2018.

[41] S. L. ,. J. Z. ,. Y. X. T. Wu, "Twitter spam detection based on deep learning," *Twitter spam detection based on deep learning,* p. 3, 2017.

[42] K. C. ,. G. C. ,. J. D. T. Mikolov, "Efficient Estimation of Word Repre- sentations in Vector Space," in *International Conference on Learning Representations (ICLR 2013)*, 2013.

[43] H. H. ,. A. M. N. Chavoshi, "Temporal patterns in bot activities," in *Twenty-Sixth International Conference on World Wide Web Companion, International World Wide Web Conferences Steering Committee,* 2017.

[44] A. A. ,. S. D. ,. V. K. ,. A. G. ,. K. L. ,. L. Z. ,. E. F. ,. A. F. ,. F. M. V.S. Subrahmanian, "The DARPA twitter bot challenge," *IEEE Comput.,* vol. 49, no. 6, pp. 38-46, 2016.

[45] A. Z. ,. M. L. ,. R. P. B. Wang, "Making the most of tweet-inherent features for social spam detection on Twitter," in *Fifth Workshop on Making Sense of Microposts, co-located with the Twen- ty-Fourth International World Wide Web Conference (WWW),* Florence, Italy, 2015.