A Proposed Data Science Approach for Email Spam Classification using Machine Learning Techniques

Aakash Atul Alurkar[*1]
[*]*Department of Computer Engineering, Smt. Kashibai Navale College of Engineering, Savitribai Phule Pune University, Pune, India*
[1]aakash.alurkar95@gmail.com

Sourabh Bharat Ranade[*2]
[*]*Department of Computer Engineering, Smt. Kashibai Navale College of Engineering, Savitribai Phule Pune University, Pune, India*
[2]sourabhranade96@gmail.com

Shreeya Vijay Joshi[*3]
[*]*Department of Computer Engineering, Smt. Kashibai Navale College of Engineering, Savitribai Phule Pune University, Pune, India*
[3]shreeya.joshi96@gmail.com

Siddhesh Sanjay Ranade[*4]
[*]*Department of Computer Engineering, Smt. Kashibai Navale College of Engineering, Savitribai Phule Pune University, Pune, India*
[4]ranade.siddhesh@gmail.com

Piyush A. Sonewar[*5]
Assistant Professor, Department of Computer Engineering, Smt. Kashibai Navale College of Engineering, Savitribai Phule Pune University, Pune, India
[5]pasonewar@sinhgad.edu

Parikshit N. Mahalle[*6]
Head of Department, Department of Computer Engineering, Smt. Kashibai Navale College of Engineering, Savitribai Phule Pune University, Pune, India
[6]aalborg.pnm@gmail.com

Arvind V. Deshpande[*7]
Principal, Smt. Kashibai Navale College of Engineering, Savitribai Phule Pune University, Pune, India
[7]principal.skncoe@sinhgad.edu

***Abstract* — With the facility of email being accessible to any individual with an internet connection, the proliferation of spam emails is one of the biggest problems which plagues our globally integrated communication systems. The various solutions to filter and hide spam previously included the manual detection of specific keywords and the blacklisting of certain domains created to send spam. However, these methods have certain shortcomings in classifying whether emails are spam or ham. This proposed system attempts to use machine learning techniques to detect a pattern of repetitive keywords which are classified as spam. The system also proposes the classification of emails based on other various parameters contained in their structure such as Cc/Bcc, domain and header. Each parameter would be considered as a feature when applying it to the machine learning algorithm. The machine learning model will be a pre-trained model with a feedback mechanism to distinguish between a proper output and an ambiguous output. This method provides an alternative architecture by which a spam filter can be implemented.**

***Keywords* — Email Classification, Spam, Ham, Spam Filter, Features, Spam Detection, Email Client, Ambiguous Output, Email Structure**

I. INTRODUCTION

Emailing is arguably the most convenient and ubiquitous form of communication for professional and personal use. It is also the fastest way to send complex information from one user to another, including not just text but also attachments such as images, videos, documents, URLs etc. This form of communication helps us save a lot of time and cost by eliminating overheads and inconveniences which are present in traditional methods such as letters and fax. The email system is inherent in the corporate and the academic world, so much so that all major communications and daily operations are managed by email. Since its inception, emailing has taken global operations to a new level of accelerated economic growth. It has become so ubiquitous in our daily lives that approximately 205 billion emails are sent every day globally. However, email protocols such as SMTP and POP being so easily accessible to everyone, plus their ease of use makes them vulnerable to be misused. Plenty of irrelevant and unsolicited emails are sent every day, a majority of which are auto-generated. Such kinds of spam emails are used for advertising, ransomware, phishing, fake purchase receipts, increasing traffic to

978-1-5386-3197-3/17/$31.00 ©2017 IEEE

malicious websites, loading scripts, ransomware, malicious websites, loading scripts, crimeware, rootkits and underlying executable files. According to a survey carried out by the Radicati Group, a research firm based in California, out of the 205 billion emails sent daily, about 18.5% is irrelevant to the recipient and 22.8% emails are sent unnecessarily. Spam emails cause a loss of around 20 million dollars annually, which is partly the result of employees wasting critical company time in reading and deleting them [1]. In another study conducted by the Kaspersky Lab Inc., handling spam makes the average user waste approximately 5 to 6 hours of time per month which is inimical to their productivity [2]. The arrival of spam also takes up memory space on servers which incur additional cost to either the provider, user or the company while being of no use at all, requiring them to purchase additional storage over a period of time. Moreover, with millions of users using the same email client, the size of this storage compounds exponentially. With normal emails bundled along with spam, it is easy for the user to overlook or accidentally delete emails which might be relevant. As critical communication on every level of an organisation is dependant upon email, the existence of spam affects an enterprise on all levels.

II. LITERATURE SURVEY

Spam filtering is a kind of email prioritization but it only focuses on filtering unwanted emails or two level prioritization systems. Sahami et al reported good results in Spam filtering using Naive Bayes classifiers [3]. After Sahami, lots of duplicated experimental results confirm Sahami's findings. Zhang et al reported similar outcomes on several different spam collections with various machine learning algorithms. They also declared that both header and body information was important in identifying spam [4]. However, spam filtering was identified with more difficult problems than what Sahami discovered because of the attacks against statistical classifiers [5]. One attack out of four identified attacks by Wittel is a tokenization attack, which works against the feature selection (tokenization) of a message by splitting or modifying key message features such as splitting up words with spaces and using HTML layout tricks. To overcome these attacks, Boykin and Roychowdhury utilized social networks to fight spam [6]. Gray and Haahr proposed collaborative spam filtering method [7]. Goodman et al summarized other advancements except machine learning in spam filtering and they reported that spam filtering was under control of the user, but the battle between the spam generator and the spam researcher was ongoing [8]. However, spam filtering alleviates the overload of the recipients to a certain degree but with these changes, it is possible to develop an email system which gives more efficient and accurate results. Along with this, a system which provides an output that is user-specific has been aimed for. This ensures a superior user experience for every individual who uses the system.

III. PROPOSED SYSTEM ARCHITECTURE

In the sequel, this paper proposes the classification of emails as spam and ham using a machine learning approach which facilitates the algorithm to recognize the necessary features more accurately, rather than specifying them manually. The main idea is to classify the incoming emails for a user using various parameters which are typically used by spammers. Its main objective is to group important emails and block the spam ones. Blocking senders who are likely to spam from a predefined list by a system administrator is an exercise in futility, due to the ease and open availability of different internet domains. This paper presents a model which uses machine learning over manual classification using parameters such as - To field, From field, Message-ID, Cc/Bcc field, etc. in the email header. The paper also takes into consideration the email body with commonly used keywords and punctuations.

A three-tier architecture which is also a client-server architecture is incorporated in the proposed model. The first module is mainly used for the processing of data. Here, the first step of acquiring emails from an email server is performed. Once the required data in the form of emails is obtained, the next step is data formatting. This helps us obtain more accurate results. In the second module, the fundamental logic behind the classification of spam/non-spam emails is implemented. The formatted emails are sent to the machine learning library. The next vital step which is Explore and Analyse Data (EDA) is executed. Herein the data is explored with the aim of analysing features. With the help of these patterns, required variables and keywords are obtained and thus the final output is achieved. According to Figure 1, the system architecture consists of four stages under which the entire functioning of the system takes place.

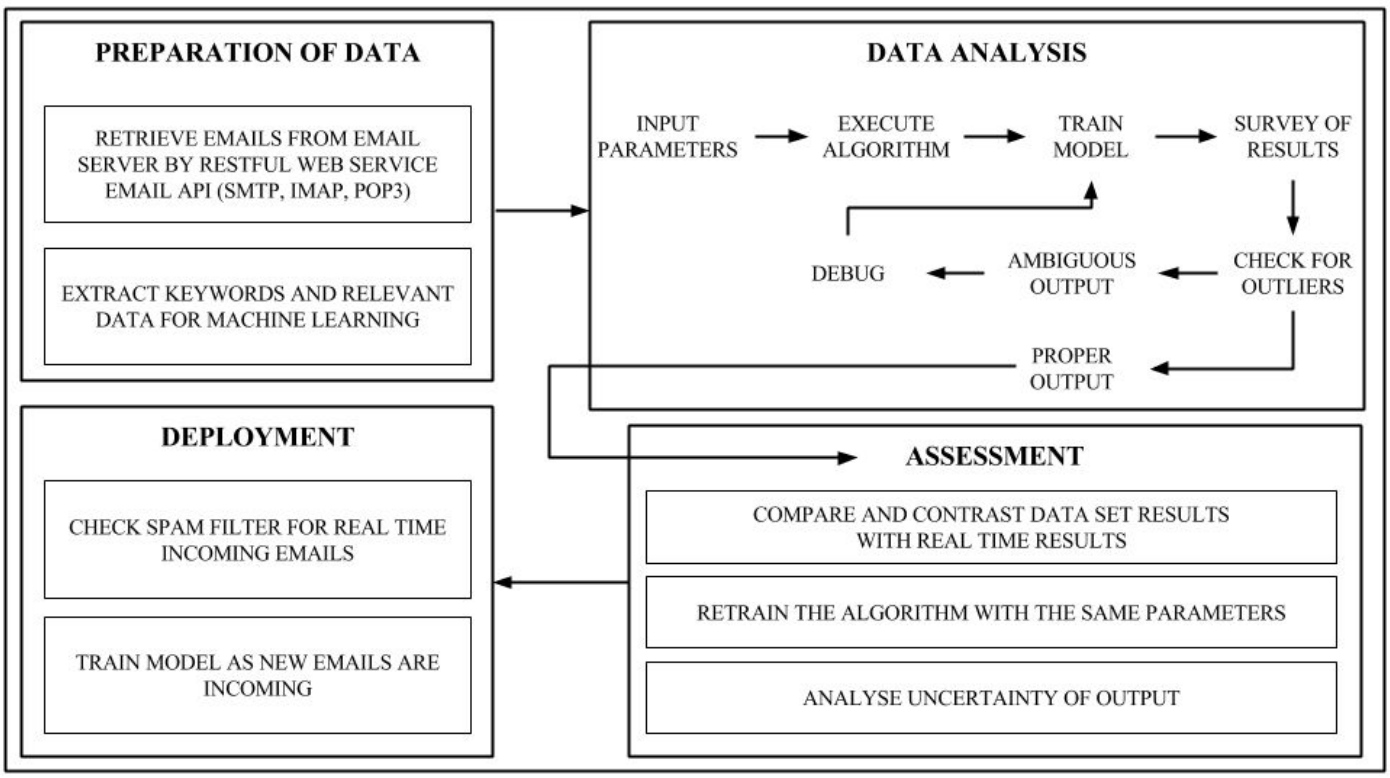

Fig 1: Proposed System Architecture

The entire system workflow, from the process of retrieving emails to their classification and spam filtering by the final concept, occurs under this umbrella of functionalities. As each stage is autonomous in functionality but not in the data it needs, it requires a different set of steps and utilises an independent part of the workflow architecture diagram. The system uses a pre-trained machine learning model to perform binary classification on emails. This model inherently creates some issues. Firstly, we need a large set of data based on which the algorithm 'learns' to classify emails. This dataset needs to have a diverse range of emails to optimise the accuracy of the algorithm. Secondly, the accuracy of the spam filtered emails will never be one hundred percent. As the algorithm performs testing on larger datasets and as more users sign up to use the progressive web application, the accuracy will increment gradually. Thirdly, the parameters to be used to identify whether an email is spam or ham are absolutely critical. Headers, Cc, Bcc, keywords etc. might vary depending on the user. The algorithm takes this slight variety into account. The four stages are delineated below.

i. Preparation of Data:

Since the classification algorithm needs a dataset upon which to perform its functionalities, it is of utmost importance that emails are retrieved with a hundred percent accuracy from their respective servers, irrespective of their domain. This retrieval is done using protocols such as IMAP (Internet Message Access Protocol), SMTP (Simple Mail Transfer Protocol), POP3 (Post Office Protocol 3) and so on. Once the emails are retrieved successfully onto a PWA or an application, only the issue of classification remains.

This is done by scanning the text in each view of the email, including the header and the body. Specific keywords are scanned for by parsing the text files to perform broad spam/non-spam classification. The algorithm then reads these parameters to determine the level of priority. The preparation of data thus consists of the retrieval of emails and their scanning.

Datasets such as the Enron email dataset and the UCI email dataset have been used previously to train algorithms to perform at optimum efficiency. The corpus contains a total of about 0.5M messages.

ii. Data Analysis:

After the data preparation phase, the input parameters are considered upon which the algorithm is to be executed. Using this, the classification model is 'trained' to recognize similar patterns in future emails and conserve the time that would've been required for further comparisons.

The accuracy of the final algorithm is directly proportional to the amount of training it receives. The results obtained after training need to be surveyed for further analysis. This is done primarily to find 'outliers': results that deviate from the norm in enough factors that they warrant being classified differently. This outlier analysis bifurcates the dataset into two types of outputs: ambiguous and proper. The proper output is the one on which further assessment is performed. If the output obtained in a certain case is ambiguous, it is not ideal. The result is debugged and the algorithm reiterates back to the training phase.

iii. Assessment:

The proper output now needs to be assessed one final time before being classified or filtered. We already have a data set on which the algorithm has been trained. Upon retrieving the user's emails, we also have a real-time result. The dataset results and the real-time results need to be compared and contrasted to optimize accuracy. The algorithm is further retrained with the same parameters. This enables us to analyze the uncertainty of the output obtained and further improvise the level and accuracy of filtering and prioritization.

iv. Deployment:

Once the emails have been accurately sorted into their respective folders, the only issue that remains is the one concerning the real-world application of the system. A detailed report needs to outline and recapitulate the overall working of the system in a way that experts, as well as laymen, can grasp its basic functionality. The successful deployment of the client online is paramount to the intended pervasiveness of the system for users. Finally, as the number of users increases, scalability without compromising the efficiency of the system becomes a necessity. The algorithm, as well as the PWA, should be scalable onto multiple devices without compromising the performance that is to be expected.

The paper proposes taking each dataset independently and that too on a personalised basis, as an email that is classified as spam for one user may not be treated as spam for others in certain rare cases. A weighted or statistical model has not been implemented on a very large scale. Further, the paper also proposes the development of a progressive web app (PWA) which is platform independent. There are a number of benefits to a progressive web app which solves the problem of having and showing two distinct user interfaces: one for the application and one for the web browser. Also, using a PWA reduces the size of the app dynamically without compromising on any of the functionalities, showing no discrimination on the browser and on the app. The data would be displayed in a well-structured method and in an extremely user-friendly format.

The workflow of the system takes place in four stages, but each stage depends on the flawless functioning of the previous one(s). It is thus a co-dependent but semi-autonomous series of utilities that result in emails being classified as spam and ham. The emails would be received by RESTful APIs. Subsequently, all the stages of machine learning would be implemented by using TensorFlow, which is an interface for expressing machine learning algorithms and an implementation for executing these algorithms. TensorFlow can be implemented on all kinds of devices ranging from phones and tablets to distributed systems [9]. This grants the proposed system greater flexibility.

IV. MATHEMATICAL MODEL

Let S be the system, such that

$$A = I, O, F, S, NSP, NSO$$

where
$I =$ Set of inputs
$O =$ Set of outputs
$F =$ Set of functions
$S =$ Set of spam emails
$NSP =$ Set of emails that are ham and higher priority
$NSO =$ Set of emails that are ham and lower priority

i. Input:

$I =$ Set of emails
Each input will have following structure:
$I1 = S1, CC1, B1$

Where

$S1 =$ Subject
$CC1$ = Cc and Bcc fields
$B1 =$ Body

ii. Process:

We follow the decision theory in view of binary classification of emails as spam/ham. The mathematical model presents an output wherein the definite probability of the output is unknown. Optimality of the solution depends on the amount of data referenced. Larger the dataset, higher the accuracy.

Assume that x is the set of attributes for Decision making. The attributes will be acquired from S, CC, B sets of all emails.

Let the set of decisions be denoted by D.
x will combines x and prior information, i.e. both previous as well as new attributes.
x is given as:

$$x = x.1, x.2, x.3, ., x.n$$

By using conditional probability,

$$P(D|x.1) = P(D \cap x.1)/P(x.1)$$
$$P(D|x.2) = P(D \cap x.2)/P(x.2)$$
$$P(D|x.n) = P(D \cap x.n)/P(x.n)$$

By combining the above equations ,we can write
$$P(D|x.j) = P(D \cap x.j)/P(x.j)$$

Where

$P(D|x.j)$ = what we know about D after considering attribute x.j (posterior); and
$P(x.j|D)$ = likely probability after observing certain value of x.j (likelihood)

Both the probabilities will be taken into consideration to obtain the output and verification of the obtained result.

iii. Output:

$$O1 = S$$

Where O1 is set of spam mails which need not be attended.

$$O2 = NSP, NSO$$

Where O2 is set of emails which are non-spam and can be considered for further prioritization.These set of emails will be required to be attended/replied urgently.

O2 also consists of emails which are not on priority.This set will consist of emails which are not spam.They might not require a reply or can be attended later.

V. CONCLUSION AND FUTURE SCOPE

This system mainly focusses on categorising emails in two categories, namely spam and non-spam. This has a myriad of implications for both organisations and individual users. At an organisational level, an effective and flexible classifier improves the soundness of its employees' email systems. For an individual user, a secure email client which automatically blocks spam emails is absolutely essential. A self-learning system which is customizable to each user and based on their dataset will only ensure greater accuracy as the dataset grows in size. Thus the system approaches an optimal solution as time passes.

Email spam is not merely an innocuous waste of time. It is a tool for malicious activities such as spear phishing, whaling, clone phishing, website forgery and much more. Classifying emails as spam or ham is thus of utmost importance from a security perspective for the user. The proposed system thus trains the algorithm and classifies emails by learning from a previously classified dataset, and then extends that functionality to classify incoming emails and display them in an organised manner. This not only results in an increase in productivity by reducing the distractions and clutter caused by spam but more importantly it protects the user from malevolent attacks. As the number of threats continue to grow, this simple yet effective measure is paramount for security and productivity.

REFERENCES

[1] Email Statistics Report, 2015-2019 Executive Summary http://www.radicati.com/wp/wp-content/uploads/2015/02/Email-Statistics-Report-2015-2019-Executive-Summary.pdf

[2] Kaspersky Lab Reports Significant Increase in Malicious Spam Emails in Q1 2016 https://www.kaspersky.co.in/about/press-releases/2016_kaspersky-lab-reports-significant-increase-in-malicious-spam-emails-in-q1-2016

[3] Sahami, M., Dumais, S., Heckerman, D., & Horvitz, E. (1998, July). A Bayesian approach to filtering junk e-mail. In *Learning for Text Categorization: Papers from the 1998 workshop* (Vol. 62, pp. 98-105)

[4] Zhang, L., Zhu, J., & Yao, T. (2004). An evaluation of statistical spam filtering techniques. *ACM Transactions on Asian Language Information Processing (TALIP)*, *3*(4), 243-269

[5] Yoo, S. (2010). Machine learning methods for personalized email prioritization. *PhD. Carnegie Mellon University*

[6] Boykin, P. O., & Roychowdhury, V. P. (2005). Leveraging social networks to fight spam. *Computer*, *38*(4), 61-68

[7] Wittel, G. L., & Wu, S. F. (2004, July). On Attacking Statistical Spam Filters. In *CEAS*.

[8] Tretyakov, K. (2004, May). Machine learning techniques in spam filtering. In *Data Mining Problem-oriented Seminar, MTAT* (Vol. 3, No. 177, pp. 60-79)

[9] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., ... & Ghemawat, S. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*.