

Adaptive reputation and linguistic classification of email users

Submitted in partial fulfillment of the requirements
of the degree of

Bachelor of Engineering

in

Computer Engineering

by

Anandteertha Rao 118A1067

Atharva Vaidya 118A1091

Yagnesh Narayanan 118A1095

Under the Guidance of:

Dr. Rizwana Shaikh



Department of Computer Engineering

SIES Graduate School of Technology

2021-22

CERTIFICATE

This is to certify that the project entitled “*Adaptive reputation and linguistic classification of email users*” is a bonafide work of the following students, submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of **Bachelor of Engineering in Computer Engineering**.

Anandteertha Rao

118A1067

Atharva Vaidya

118A1091

Yagnesh Narayanan

118A1095

Dr. Rizwana Shaikh

Internal Guide

Dr. Aparna Bannore

Head of Department

Dr. Atul Kemkar

Principal

PROJECT REPORT APPROVAL

This project report entitled (*Adaptive reputation and linguistic classification of email users*) by following students is approved for the degree of *Bachelor of Engineering* in *Computer Engineering*

Anandteertha Rao

118A1067

Atharva Vaidya

118A1091

Yagnesh Narayanan

118A1095

Name of External Examiner: -----

Signature:-----

Name of Internal Examiner: -----

Signature:-----

Date:

Place:

DECLARATION

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Anandteertha Rao

118A1067

Atharva Vaidya

118A1091

Yagnesh Narayanan

118A1095

Signature

Date:

ACKNOWLEDGEMENT

The acknowledgments by the candidate shall follow the citation of literature, signed by him/her, with date. *Sample acknowledgement is shown below. The student can change as per their requirements.*

We wish to express our deep sense of gratitude and thank to our Internal Guide, Dr. Rizwana Shaikh for her guidance, help and useful suggestions, which helped in completing our project work in time. We are also extremely grateful to our Project coordinator Prof Dr. Rizwana Shaikh for her guidance provided whenever required. We also thank to our Hod prof Dr. Aparna Bannore for her support in completing the project. We also thank to our Principal Dr. Atul Kemkar, for extending his support to carry out this project

Also, we would like to thank the entire faculty of Computer department for their valuable ideas and timely assistance in this project, last but not the least, we would like to thank our teaching and non-teaching staff members of our college for their support, in facilitating timely completion of this project.

Project Team

Anandteertha Rao

Atharva Vaidya

Yagnesh Narayanan

ABSTRACT

In today's modern day world E-mails have become a necessity and it has become one of the most widely used applications developed for smooth communication between 2 or more people. The only requirements for any person to use electronic mails are – a computer of any form, like desktops, mobile phones, tabs, laptops etc., and a network. Most of the times this network is the internet. The globe has seen an exponential rise in the usage of emails. But with the rise of this technology the hackers have found a new platform to trick common or naïve people into their traps. The hackers tend to use E-mails to extract information from the users to steal identity or money and they can also use the power of emails to flood the network of the user. All of this is done using unsolicited mails or SPAMS. Spam mails are unwanted mails and spammers make use of spams for phishing attacks and flooding networks.

This project aims at solving this problem in two steps. The first step is to classify emails based on optimized machine learning algorithm (like logistic regression, KNN, SVM, etc.) into 2 categories – spams and hams. The second step is to assign reputation values to all the spammers using an optimized algorithm which will consider 3 important factors: The frequency of spams sent, the frequency of hams sent and the behavior of the spammer across a network of multiple users. Most of the current implementations focus on the first part i.e., on the classification of mails but most of them do not consider the second part and even if they do there is some or the other gap or some test case that is left untouched and this can result into incorrect conclusions. This project aims at providing an optimal solution to this problem. Initial analysis and implementations in this project have shown that logistic regression has the best accuracy and hence has been used for the implementation of the classifier, logistic regression is also used by Google's Gmail for classifying emails which proved to be yet another strong reason for the use of logistic regression. A self-developed reputation algorithm has been developed as a part of research work and implementation.

Contents

		Page No.
Chapter 1	Introduction (as per UoM Guidelines)	1-5
	1.1 Introduction to topic and domain	1
	1.2 Need of the Project	2
	1.3 Scope	3
	1.4 Organization of report	5
Chapter 2	Literature Survey	6-9
	Survey of existing systems	6
	Limitations and gaps	9
	Objectives and Problem Statement	9
Chapter 3	Proposed System	10-12
	Proposed system	10
	Components and functionality	12
Chapter 4	Design and Methodology	13-16
Chapter 5	Results and Discussions	17-26
Chapter 6	Conclusion	27
	References	28-29

List of Figures

Figure No.	Figure Caption	Page No.
1.	Comparison of Accuracy of classification algorithms	10
2.	Dataset content with target values	11
3.	Plot of number of hams and number of spams in dataset	12
4.	Libraries used	12
5.	Flowchart of the project	14
6.	Flowchart for spam classification	15
7.	Algorithm for reputation calculation	17
8.	Accuracy, precision, recall and time of the 4 algorithms	18
9.	Modified comparison plot	19
10.	Analysis of KNN	19
11.	Confusion Matrix for KNN	20
12.	Analysis of Logistic Regression	20
13.	Confusion Matrix for Logistic Regression	21
14.	Analysis of Naïve Bayes	21
15.	Confusion Matrix for Naïve Bayes	22
16.	Analysis of Support Vector Machine	22
17.	Confusion Matrix for SVM	23
18.	Sample Logistic Regression Curve	23
19.	Login Page	24
20.	Login Page (filled)	24
21.	Output Page	25
22.	Testing(a)	26
23.	Testing(b)	26
24.	Testing(c)	27
25.	Testing(d)	27

List of Tables

Table No.	Table Caption	Page No.
1	Contents	viii
2	List of figures	ix

List of Abbreviations

In alphanumeric order

E-mails	Electronic-mails
KNN	K-nearest Neighbors
PAN	Permanent Account number
SVM	Support Vector Machine

INTRODUCTION

- **Introduction to your topic and domain**

E-mails or electronic mails have become one of the most frequently used methods for communication. The basic idea behind an email is to exchange messages through text bodies between two or more people using electronic devices. E-mails have noticed a significant rise in their usage parallel to the rise of the computing world. Emails primarily rely on two basic prerequisites- first, at least two electronic devices like mobile phones, tabs or computers of any form and second, a network. The most commonly used network for the purpose of exchanging messages through emails is the Internet. Since the prerequisites for using e-mail are two of the most readily available resources of the modern-day world (computer and internet), using emails has become one of the easiest ways to communicate irrespective of the location and time. Another major advantage provided by e-mails is that as long as you have an internet connection all the other functionalities are free of cost. Earlier restricted to text, e-mails have now evolved and are capable of transferring many multimedia files like images, audios, videos and all document formats like docs, pdfs, etc.

The growth in usage has been so significant that the governments from all over the globe have started considering email – ids as a basic necessity and individuals are required to have an email id to acquire almost all the official documents. For instance, citizens from India now have the privilege to link their Aadhar card and PAN cards to their email-ids for ease of use.

The mails stay between the sender and the receiver ensuring privacy of the users and this generates a sensation, among the people, that emails are completely safe to use. But this safety misconception has led people from having everything to going broke in a matter of seconds.

The wide usage email has been fruitful for the people but has also added a whole new domain for hackers, who are trying to exploit this platform as much as possible and as soon as possible. And the most commonly used technique by the imposters for tricking common people into their traps are **SPAM** e-mails.

- **Need of the Project**

Spam mails or simply spams are unwanted and unsolicited mails which are used to spread viruses and advertisements. Taking it further, hackers and imposters use spam emails for bank frauds. Since E-mails have become a necessity and since most of the legal documents are related to email-ids in one way or the other, the potential threat that comes along with email-ids has increased exponentially. Any person or organization who sends spam emails is said to be a Spammer. Spammers send spam emails for many reasons, some of them being - flooding the network, advertising their respective company/organization, spreading viruses like trojan horses and worms, and most importantly Phishing. Phishing is an attack where in the user is tricked into revealing sensitive and important information regarding bank records, account numbers, passwords by the spammer. The spammer thereafter has almost all the details required to extract/withdraw money from the user's bank account. These frauds are not just bank or money related but the spammers also tend to steal identity of the user to take advantage of the user's position.

Spammers also tend to flood the user's inbox with messages which is highly undesirable and needs to be addressed. This has led to all email service providers to develop spam classifiers which will help the organization/individual to automatically detect if the mail received is a legitimate email or a spam. Legitimate emails are also sometimes called as hams. Email service providers like Google's Gmail, Microsoft's Outlook and all similar organizations use machine learning techniques to classify emails into spams or hams and upon classification the mails are put in separate directories namely inbox (containing hams) and junk/spam mails (containing spams). Many machine learning algorithms with different optimization techniques are employed to serve this purpose. For instance, Support Vector Machine algorithm, Naïve bayes Classifier, Logistic regression, etc. are some of the commonly used techniques to classify the emails as spams or hams. Each one of these techniques has its own pros and cons and deciding which technique should be used is a matter of concern as one mistake can cause huge repercussions.

The point to be emphasized here is even though mails are classified into different folders but no further action is taken against a spammer who continues to take efforts for phishing or flooding the receiver's network. And this major problem needs to be addressed. Many studies have been carried out on which algorithm would be best for classification of emails but what action has to be taken further is yet to be considered. The classification of emails is based on text (content) present inside the mail. Once a sender is tagged as a spammer appropriate action regarding his future behavior has not been explored much yet. The frequency of spams sent, the behavior of the spammer across multiple users in a network and number of hams or legitimate sent by the spammer are all important factors and shouldn't be ignored while taking any further action/decision regarding the spammer.

This project primarily consists of two steps. First step is to classify the emails based on the most optimum machine learning to make sure highest accuracy is ensured while classifying the emails as spams or hams. The second step focuses on taking further action on the spammer, this will include assigning reputation to a spammer based on multiple factors like frequency of spams he has sent, the frequency of hams he has sent and also his behavior across a network of people (i.e., he might be tagged as spammer for one user but might be a hammer for many other users in the network). Hence first part of the project involves research and implementation for discovering the best classification algorithm, this also involves extensive study of previously published work from renowned publications and authors. Various machine learning algorithms have been studied and implemented to compare factors such as accuracy and speed, some of the algorithms compared are: Naïve Bayes Classifier, Support Vector Machine, K-nearest neighbors and Logistic regression. Speed and accuracy comparisons from the implementations proved that logistic regression proved to be the most optimistic algorithm to classify emails (Gmail also uses logistic regression along with optimization techniques for this purpose). The second part of the project has primarily focused on devising an algorithm for assigning reputation to tagged spammers and hence deciding further what action has to be taken against the one having a bad reputation value. This

action might be blocking him for a certain amount of time or permanently blocking him or reporting him.

- **Scope**

The project requirements include two of the most basic and readily available resources namely a computer and a network. Computer of any form: laptops, desktop computers, mobile phones, tablets etc., can use this product. This flexibility is attributed to the fact that the processing and classification is done on the server side. This eliminates the need for a high-performance computer with high end processing powers, rather normal systems capable enough to run any application which supports email handling will do the needful. Accessing the emails on web also work equally well. The second requirement is that of a network, and the most commonly available and accessible network is the internet and in present times the internet has reached almost each and every household. One important thing with emails is that the receiver need not be online while the transfer of emails and their processing for legitimacy takes place.

Since the requirements are readily available, the project can be scaled to a global level. In fact, almost everyone who uses Gmail or any similar service have already experienced classifiers and understand its core functionality. Due to the rise of spammers, hackers and imposters almost every email service providing organization has to take at most care of its user's safety and privacy. For this every organization implements a spam classification filter in their service so as to avoid any kind of malpractice. Emails are also extensively used by companies of all sizes, i.e., from startups to Multi-National Companies. Considering the high reliability of all the companies on emails, and the amount of sensitive and valuable information that companies deal with, the safety of emails has become one of the most important domains that need to be worked on. This project not only focuses on classifying the emails with the best algorithm but it also focuses on what steps have to taken further. Assigning a reputation to the spammers can greatly help in reducing the traffic and can avoid phishing attacks on naïve users. The reputation algorithm is focusing not only on how the spammer is behaving for a single user but spammer's behavior for multiple users is being considered. This has been done to ensure that mailers who have been wrongly tagged as spammers in one's system might actually be legitimate mailers. The module for reputation assignment alone can also be used by the other e-mail service providers. The

main deliverable of this project is a classifier with reputation assignment for spammers, and not the actual transfer of mails from sender to receiver.

- **Organization of the report**

1. Introduction

- a. Introduction to the topic and domain.
- b. Need of the project.
- c. Scope.
- d. Organization of the report.

2. Literature Survey

- a. Survey of existing system.
- b. Limitation of existing system.
- c. Problem Statement.
- d. Objectives.

3. Proposed System

- a. Overall proposed system.
- b. Components and Functionality.

4. Design and methodology

- a. Design (ER diagram).
- b. Methodology.
- c. Algorithm implemented.
- d. Details of hardware and software required.

5. Results and Discussions

a. Implementation

b. Analysis of output

6. Conclusion

LITERATURE SURVEY

- **Survey of existing systems**

As of now 49 papers have been published on which classifier has to be used for spam classification [8]. Following are the findings of 7 such research papers on the topic of email classification and reputation calculation.

- 1. Detecting Spammers with SNARE [1]: Spatio-temporal Network-level Automatic Reputation Engine.**

[Authors: Shuang Hao, Nadeem Ahmed Syed, Nick Feamster, Alexander G. Gray, Sven Krasser]

This paper investigates ways to infer the reputation of an email sender based solely on network-level features, without looking at the contents of a message. This paper talks particularly about SNARE. SNARE (SNARE is an automated response engine) is automated and lightweight enough to operate solely on network-level information. Even though SNARE maintains information about the IP address of a sender as an input to filtering, there may be cases where an IP address sends both spam as well as legitimate email. So, we will need more sophisticated classifiers involving time series-based features.

- 2. A Study of Machine Learning Classifiers for Spam Detection [2]**

[Author: Shrawan Kumar Trivedi]

In this paper, the aim was to distinguish between ham emails and spam emails by making an efficient and sensitive classification model that gives good accuracy with low false positive rate. Greedy Stepwise feature search method has been incorporated for searching informative feature of the Enron email dataset. The comparison has been done among different machine learning classifiers (such as Bayesian, Naïve Bayes, SVM (support vector machine), J48 (decision tree), Bayesian with Adaboost, Naïve Bayes with Adaboost). The concerned classifiers are tested and evaluated on metric (such as F-measure (accuracy), False Positive Rate, and training time). By analyzing all these aspects in their entirety, it has been

found that SVM is the best classifier to be used but it takes a little longer to build the model.

3. Email classification using Machine learning Algorithms [3].

[Authors: Anju Radhakrishnan, Vaidhehi V: Department of Computer Science, Christ University, Bengaluru, India]

In this paper email classification is done using machine learning algorithms. Two of the important algorithms namely, Naïve Bayes and J48 Decision Tree are tested for their efficiency in classifying emails as spam or ham. This paper showed that J48 Decision Tree Classifier is more efficient than the Naïve Bayes classifier for the dataset Enron Corpus. It gives an accuracy of 96.5971% in classifying the emails with a feature size of 400 attributes within a short span of time 0.06 seconds.

4. Social Network Based Reputation Computation and Document Classification [4].

[Authors: JooYoung Lee (Syracuse University, Syracuse, USA, jlee150@syr.edu), Yue Duan (Syracuse University, Syracuse, USA, yudian@syr.edu), Jae C. Oh (Syracuse University, Syracuse, USA, jcoh@syr.edu), Wenliang Du (Syracuse University, Syracuse, USA, wedu@syr.edu), Howard Blair (Syracuse University, Syracuse, USA, blair@syr.edu), Lusha Wang (Syracuse University, Syracuse, USA, lwang40@syr.edu), Xing Jin (Syracuse University, Syracuse, USA, xjin05@syr.edu)]

In this paper the authors develop two social network-based algorithms that automatically compute author reputation from a collection of textual documents. Other systems considered the reputation or rank of the documents/texts, this uses reputation of authors.

5. Machine Learning for E-mail Spam Filtering: Review, Techniques and Trends [5].

[Authors: Alexy Bhowmick (Tezpur University), Shyamanta M. Hazarika (IIT Guwahati)]

This paper focuses primarily on Machine Learning-based spam filters and their variants, and report on a broad review ranging from surveying the relevant ideas, efforts, effectiveness, and the current progress. Blacklisting, whitelisting or grey listing IPs will provide us with better and efficient solutions. Prioritizing emails based on feedback will give efficient solutions.

6. Machine learning for email spam filtering: review, approaches and open research problems [6].

[Authors: Emmanuel Gbenga Dada, Joseph Stephen Bassi, Haruna Chiroma, Shafi'i Muhammad Abdulhamid, Adebayo Olusola Adetunmbi, Opeyemi Emmanuel Ajibuwa]

This paper is also a comparison of various machine learning algorithms which can be used for classifying emails as spams or hams. SVM proved to get us great results for a sparse dataset.

7. Efficient email classification approach based on semantic methods [7]

[Authors: Eman M. Bahgat, Sherine Rady, Walaa Gad, Ibrahim F. Moawad]

This paper shows that TF-IDF Classification accuracy is found more when the TF-IDF value is used compared to simple word counts. Using SVM + Ant colony optimization yielded better results than kNN or NB or SVM.

8. Trends in email classification research [8]

[Authors: GHULAM MUJTABA, LIYANA SHUIB, RAM GOPAL RAJ, NAHDIA MAJEED, MOHAMMED ALI AL-GARADI, (department of Information Systems, Faculty of Computer Science and Information Technology, University of Malaya, Kuala Lumpur 50603, Malaysia)]

As of now 49 studies have been carried out on spam email classification, 1 on image-based spam classification, 1 on VIP email-based classification and 2 on multilingual email classification (i.e., most of the research has been based on spam email classification). This paper gives a deep insight as to which datasets are being used across various studies and researches that are being carried out, in the field of email classification, e.g., Out of all, 5 studies have been based on Enron corpus which is basically a dataset containing data from 150 users which are mostly senior management employees of Enron.

All the research papers mainly focused on comparison between various machine learning algorithms to classify emails into spams and hams.

- **Limitations in existing papers and research gaps**

All the papers emphasized on comparison between various algorithms with respective to speed and accuracy. Following are the drawbacks or gaps found in each of the papers referred during the preparation of this project. For the first paper which discusses about SNARE [1], special emphasis has been given to IP addresses sending spams but an increasing fraction of spam may be sent from an IP address that also send significant amounts of legitimate mail. Hence only considering amount spams sent might lead to erroneous results. The following paper [2] suggests SVM to be accurate but the major drawback that SVM has is that the time it takes to build the model is significantly higher than other algorithms. For [3] the drawback was that the dataset used was not diverse and contained data from emails of senior managers of a single company. For [5] the drawback was Isn't dynamic because we use knowledge-based engineering which creates a rule base, which is stagnant and if we used SMTP path analysis stand alone, then it is not of much use and doesn't provide added accuracy, also it proves to be a lot time consuming.

For paper [6] the major drawback was what serves as a advantage for SVM, proves to be a disadvantage for NB, and vice versa and Neural Network works efficiently providing good accuracy only when we have a large dataset, and requires a great computation power. Paper [7] was a paper about the studies that have been carried out on the topic of spam classification and it suggested that as of now 49 papers have been published on or around this topic and none of them talks about what has to be done after a mailer has been classified as a spammer. This served as the motivation for this project.

- **Problem Statement**

The problem statement for this project is divided into 2 parts: First part is to implement an algorithm to classify emails into 2 categories: spams (unsolicited emails) and hams (legitimate emails) which is optimum with respect to speed and accuracy. The second part is to assign reputation to the tagged spammers based on the following factors- the frequency

of spams sent, the frequency of hams sent and the behavior of the sender across a network of multiple users (This is done because a legitimate sender can sometimes be wrongly classified as a spammer for a particular receiver but his behavior might be good (hammer) with respect to some other receivers).

- **Objectives**

- To analyze various machine learning algorithms and compare their complexity with respect to time and accuracy.
- To implement the most suitable classifier to classify emails into 2 categories- spams and hams (legitimate emails).
- To devise and implement an appropriate algorithm to assign reputation values to spammers so further actions like blocking or marking as hammer can be taken depending on these values.
- To make sure behavior of the spammer is considered across multiple users and not just a single user.

PROPOSED SYSTEM

- **Proposed system to overcome drawbacks**

The proposed system consists of 2 modules for each serving different functionalities. The first module is used to classify emails into two classes namely spams and hams. This model makes use of Logistic Regression to classify emails, and this classification is particularly applied on the textual content present inside the mail. Logistic regression has been chosen following the analysis carried out during the initial phases of the project. Logistic regression provided the highest accuracy among the following machine learning algorithms: K-nearest neighbors algorithm, Naïve bayes algorithm, Support Vector Machine algorithm and Logistic regression.

	algo	acc
0	Naive Bayes	97777.777778
1	Support Vector Machines	97198.067633
2	K Nearest Neighbours	85700.483092
3	Logistic Regression	98357.487923

Fig. 1. Comparison of accuracy.

The second module consists of a self devised reputation algorithm which is used to assign reputation to the senders based on the following factors:

- Frequency of spams sent by the sender.
- Frequency of hams (legitimate mails) sent by the sender.
- Behavior of the sender across a network of multiple users.

Most of the studies that have been carried only focus on the number of spams that a sender or a spammer has sent. But this might lead to misconceptions if a legitimate mail has been wrongly classified as a spam as we can never a classification accuracy of 100%. Hence the frequency of hams that the sender has sent also plays an important role in assigning a reputation value to the sender. For instance, let's say a sender has sent 100 spam mails to the receiver. Most of the existing solutions directly tag this sender as a spammer. But there is a very high chance that this conclusion is wrong.

Consider the case where the sender has sent 100 spams but 2000 hams or legitimate mails, in such a case tagging the sender as a spammer is an incorrect decision taken by the system.

Let's say that there are mailers or senders say sender A and sender B. If sender A sends 2 spams and 2 hams, and sender B sends 100 spams and 100 hams then currently used systems treat both of them equally but we can clearly feel by intuition that this conclusion is wrong. Here the probability of sender A being a spammer is far lesser than the probability of sender B being a spammer. Hence such cases must also be considered. Another case possible is when the sender has sent 2 mails both which have been classified as spams for one particular user, but considering inboxes of more people across the network the sender has more hams and has a good reputation. Then assigning such a sender a bad reputation value doesn't make sense.

Our proposed system takes into consideration all the above-mentioned test cases and provides appropriate reputation values to all senders. These reputation values can prove to be of great use to email service providers as they'll already know about potential spammers and hence can take proper actions against them.

The dataset used for this project was Enron Corpus. It is the most used dataset for the Classification of emails [8] along with PU corpora which is another similar dataset. Enron corpus contains the emails from 150 senior employees from the company Enron which have been correctly tagged as spams or hams. The following figure (Fig. 3) shows the content of a spam and its target value as 1 and content of a ham and its target value as 0.

	text	target
0	b'Subject: nesa / hea \ ' s 24 th annual meetin...	0.0
1	b'Subject: meter 1431 - nov 1999\r\ndaren -\r\...	0.0
2	b"Subject: investor here .\r\nfrom : mr . rich...	1.0
3	b"Subject: hi paliourg all available meds . av...	1.0
4	b'Subject: january nominations at shell deer p...	0.0

Fig. 2. Contents with target as 1(spam) and 0(ham)

Since the classification is text based we can clearly see that text such as "mr. rich investor here" has been given the target (or label) as 1 which is nothing but a spam. And texts such as "Annual meet" have been given label/target value as 0 which is an indication for hams.

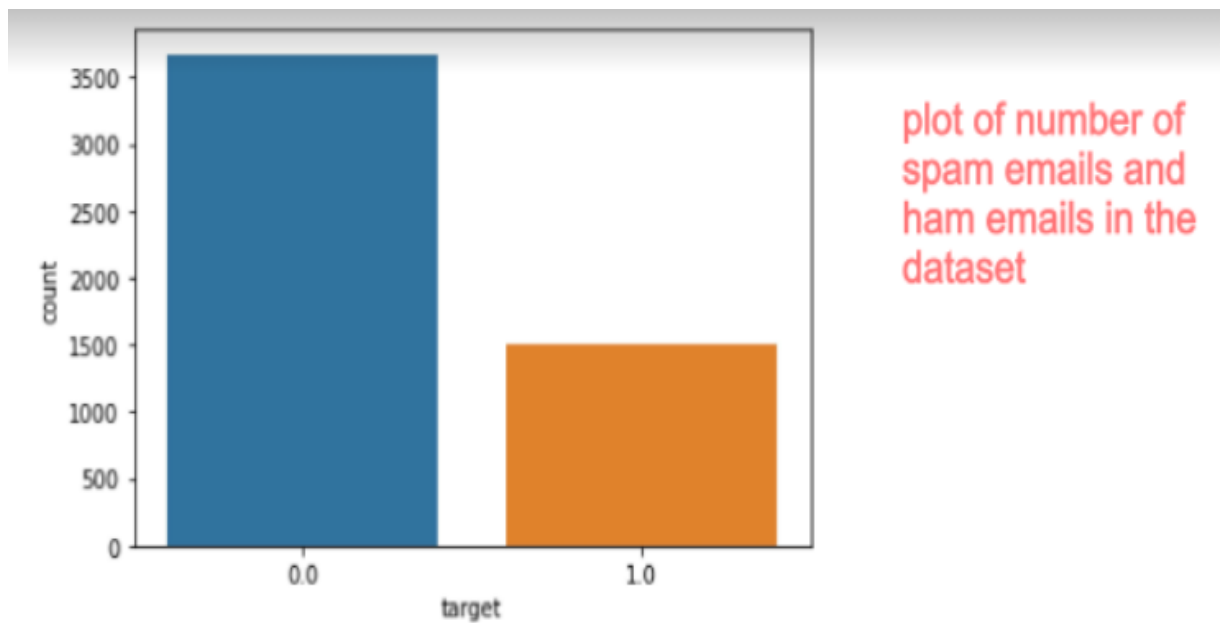


Fig. 3. Plot of number of hams and spams in the dataset

- **Components and functionality**

The proposed solution consists of 2 basic modules. The first module consists of the classifier. This module makes use of tools required for natural language processing, but the best part about this is that all the tools required for this purpose are free and open source. Following Natural language processing libraries have been used in this project:

- NLTK
- Pandas
- Sklearn
- NumPy

```
import pandas as pd
import numpy as np
from sklearn.datasets import load_files
```

```
import re
import nltk
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
from nltk.stem import WordNetLemmatizer
```

Fig. 4. Libraries used

DESIGN AND METHODOLOGY

- Design

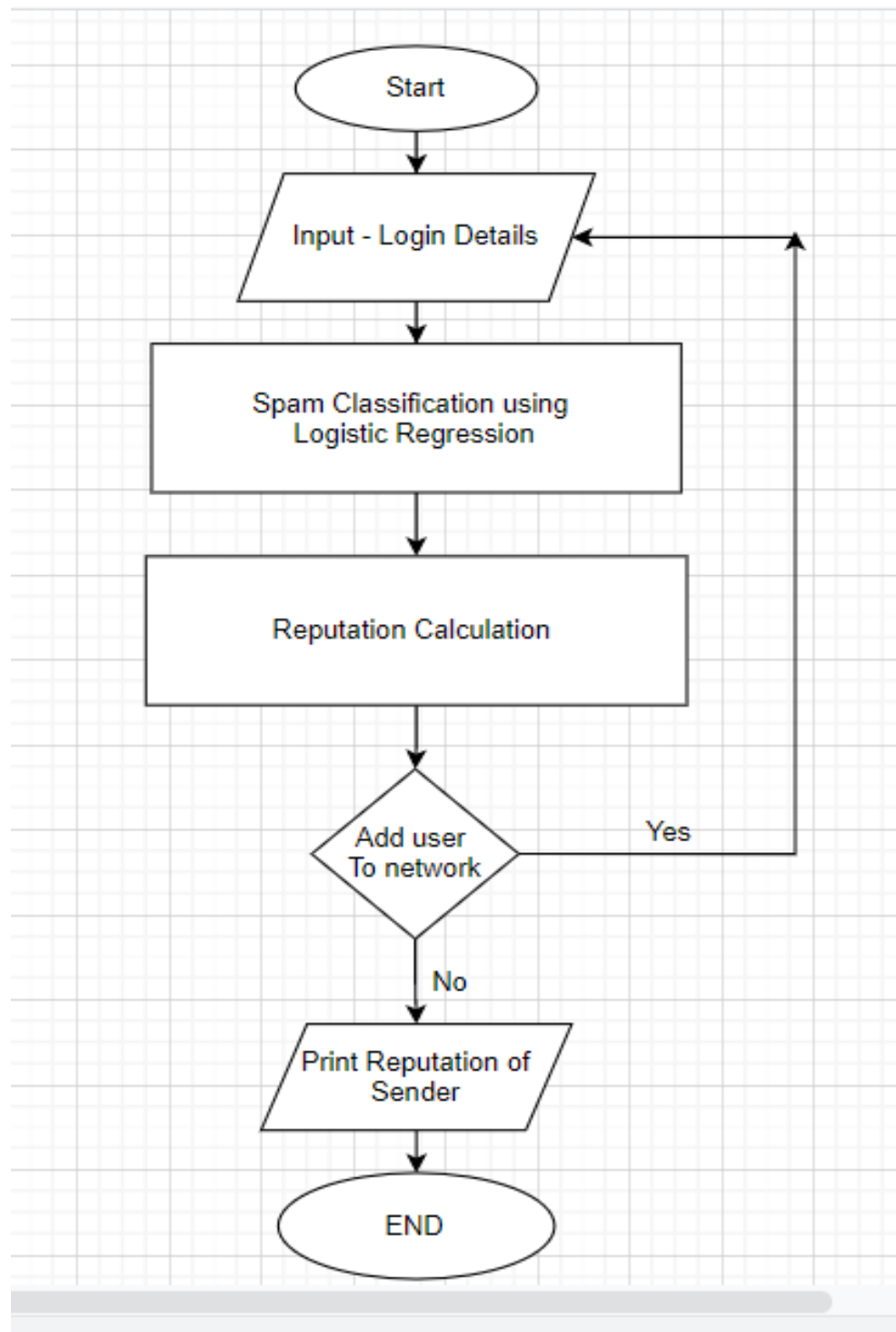


Fig. 5. Flowchart of the project

- **Methodology and Algorithms**

Our project is divided into two main categories, one includes the classification of emails as spam or hams, second includes the calculation of reputation of each email user.

Let us understand the first method.

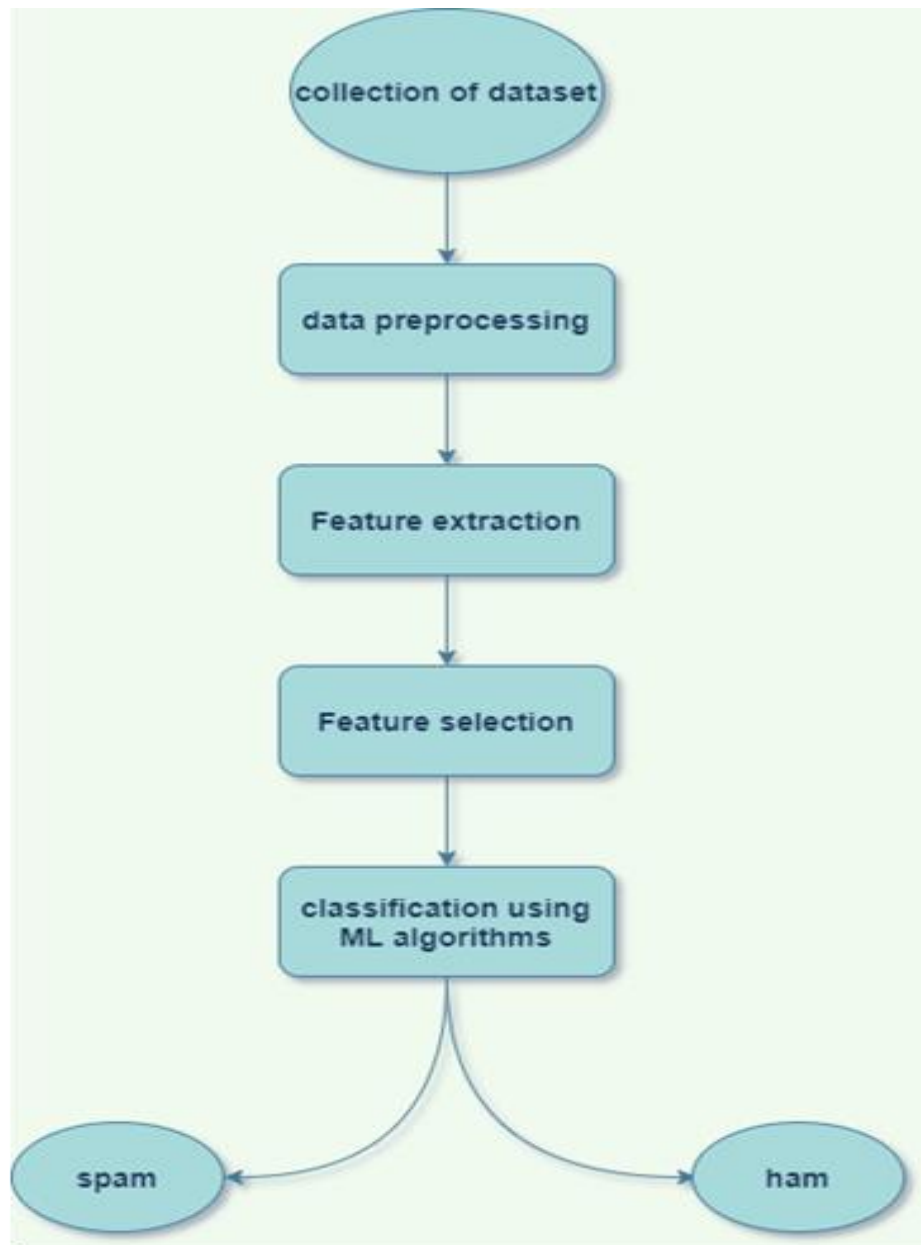


Fig. 6. Flowchart for Classification

It is lucid from the flowchart that we are using a machine learning algorithm to perform the classification using simple binary classification method. But before that we have to fabricate the input which is currently an email in the form of a linguistic data to a machine-readable data. To engineer this, we are using the techniques of Natural Language Processing.

A few of the steps of NLP are given below:

- 1) Dimensionality Reduction: In this phase we are executing the syntax analysis methods namely stemming and lemmatization in order to get the root words. While doing so, we came to a conclusion which is also a fact that lemmatization though takes more time to execute, provides better results. These stemming and lemmatization methods were used from the renowned package of nltk.
- 2) We removed special symbols, extra spaces, numbers, etc., and generated a corpus.
- 3) We selected the features by using stop words removal technique.
- 4) Finally, we extracted features using the BoW technique of TF-IDF. In this we converted our previously linguistic data to vector which is easily understandable by machines.

Once we got the vectors, we moved on the next part which is selecting the best algorithm of machine learning to classify if the email is spam or ham. We researched all the algorithms that would provide efficient results given the constraints. We found out that SVM (support vector machines), KNN (K-nearest neighbours), Naive Bayes & Logistic Regression are the few algorithms which might give us the required results.

Hence, we applied all the algorithms and calculated the time taken, the accuracy, precision, recall & the confusion matrix by each one.

There were a few notable observations that we made while we conducted this experiment to determine the best algorithm for our use-case, one was SVM took the least to execute and give us the results whereas KNN took the maximum time to execute. Logistic Regression had the highest accuracy consequently KNN had the least accuracy. We had to do a trade-off between time and accuracy. We came down in favour of accuracy and decided to opt for accuracy rather than time. Therefore, we got a high accuracy model to classify the emails as spam or ham using the logistic regression algorithm with the help of natural language processing.

The second part includes the calculation of reputation of each email user. This was an important part of our project as this helps us to classify each email user with a degree of how likely he/she is to send a spam. To realize this goal, we created a network of email users and studied the behavior of each one of them. While most of our attempts to draw meaningful insights were beneficial but not flawless. Finally, we got the formula to derive the reputation of each users which was pretty much unblemished.

Algorithm 1 Calculate the reputation of email users

```

1: procedure CALCULATE REPUTATION(email, email_user)
2:   classification = model.classify(email)
3:   if classification  $\neq$  SPAM then
4:     add_ham_count(email_user)
5:   else
6:     add_spam_count(email_user)
7:   end if
8:   rep = formula_for_rep_cal(email_user)
9:   return rep
10: end procedure

```

Fig. 7. Algorithm for reputation calculation

As shown in the algorithm we are creating a network which allows us to recognize the spamming pattern of each user which also taking into account his behavior while sending legitimate emails. At the end, we are scaling the reputation values and converting it into a percentage which depicts the chance of that user being a spammer or a hammer.

• Details of Hardware and Software

Two basic requirements are internet and any computing device like desktop computer, laptop or a mobile phone which is capable enough to handle email processing. Software used for building this project was a python editor (atom) and a flask server. Apart from this no other hardware or software while implementing this project.

RESULTS

- **Results and implementation**

Analysis –

The Following figure shows the parameter values and corresponding plot for comparison of classification algorithms:

```
dfNew = pd.DataFrame.from_dict(graphs)
dfNew['time'] = dfNew['time'].divide(dfNew['time'].max())
dfNew
```

	algo	acc	precision	recall	time
0	Naive Bayes	0.977778	0.960526	0.963696	0.432151
1	Support Vector Machines	0.971981	0.950658	0.953795	0.428121
2	K Nearest Neighbours	0.857005	0.687651	0.937294	1.000000
3	Logistic Regression	0.983575	0.967320	0.976898	0.461555

```
import matplotlib.pyplot as plt
dfdash = dfNew.drop(['algo'],axis=1)
dfdash.T.plot()
plt.show()
```

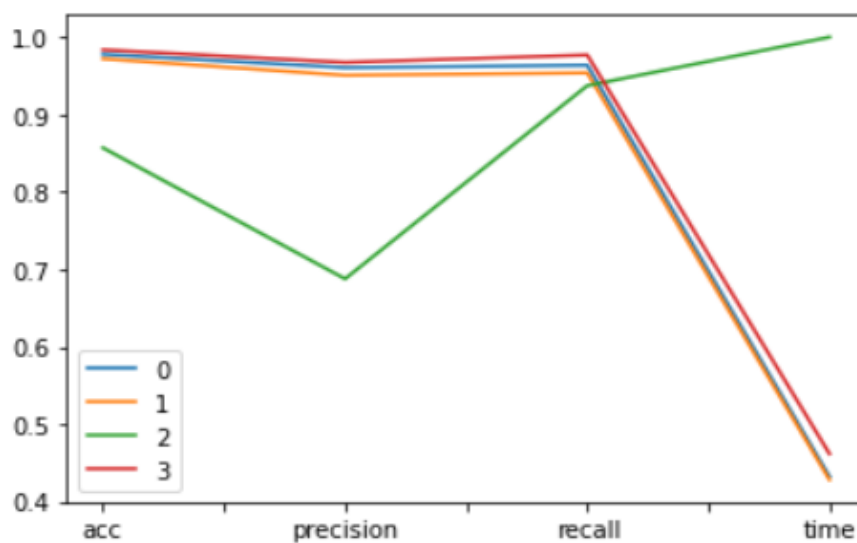


Fig. 8. Accuracy, precision, recall and time of the 4 algorithms

```

x = [0,1,2,3]
y = dfacc['acc']
plt.plot(x, y, label='accuracy')
y = dfpre['precision']
plt.plot(x, y, label='precision')
y = dfrecall['recall']
plt.plot(x, y, label='recall')
y = dftime['time']
plt.plot(x, y, label='time')
plt.legend()
plt.show()

```

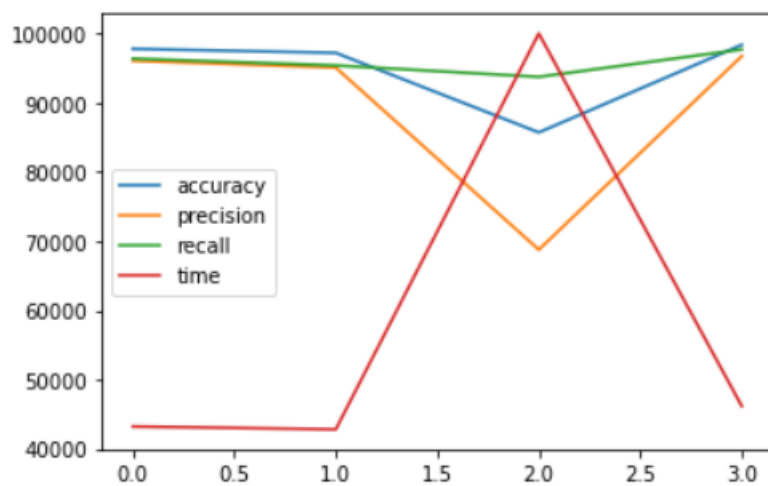


Fig. 9. Modified comparison plot

Following is an analysis of each machine learning based algorithm:

```

start_time = time.time()
from sklearn.neighbors import KNeighborsClassifier
neigh = KNeighborsClassifier(n_neighbors=3)
neigh.fit(X_train, y_train)
pred = neigh.predict(X_test)

accuracy = accuracy_score(y_test, pred)
precision = precision_score(y_test, pred)
recall = recall_score(y_test, pred)
conf_m = confusion_matrix(y_test, pred)

print(f"accuracy: %.3f" %accuracy)
print(f"precision: %.3f" %precision)
print(f"recall: %.3f" %recall)
print(f"confusion matrix: ")
print(conf_m)
timeforKNN = time.time()-start_time + timeforsplit + timeforfeaturesExtraction + timeforcorpus
print(timeforKNN,'seconds to execute')

accuracy: 0.857
precision: 0.688
recall: 0.937
confusion matrix:
[[603 129]
 [ 19 284]]
420.055198431015 seconds to execute

```

Fig. 10. Analysis of KNN

```

accuracy: 0.857
precision: 0.688
recall: 0.937
confusion matrix:
[[603 129]
 [ 19 284]]
411.2928352355957 seconds to execute

```

Fig. 11. Confusion Matrix for KNN

```

start_time = time.time()
from sklearn.linear_model import LogisticRegression
clf = LogisticRegression(random_state=0)
clf.fit(X_train,y_train)
pred = clf.predict(X_test)

accuracy = accuracy_score(y_test, pred)
precision = precision_score(y_test, pred)
recall = recall_score(y_test, pred)
conf_m = confusion_matrix(y_test, pred)

print(f"accuracy: %.3f" %accuracy)
print(f"precision: %.3f" %precision)
print(f"recall: %.3f" %recall)
print(f"confusion matrix: ")
print(conf_m)
timeforLR = time.time()-start_time + timeforsplit + timeforfeaturesExtraction + timeforcorpus
print(timeforLR,'seconds to execute')

accuracy: 0.984
precision: 0.967
recall: 0.977
confusion matrix:
[[722  10]
 [  7 296]]
190.32634902000427 seconds to execute

```

Fig. 12. Analysis of Logistic Regression

```

accuracy: 0.984
precision: 0.967
recall: 0.977
confusion matrix:
[[722  10]
 [  7 296]]
190.32634902000427 seconds to execute

```

Fig. 13. Confusion Matrix for Logistic Regression


```

start_time = time.time()
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score, precision_score, recall_score
from sklearn.metrics import confusion_matrix

model = MultinomialNB().fit(X_train, y_train)
pred = model.predict(X_test)

accuracy = accuracy_score(y_test, pred)
precision = precision_score(y_test, pred)
recall = recall_score(y_test, pred)
conf_m = confusion_matrix(y_test, pred)

print(f"accuracy: %.3f" %accuracy)
print(f"precision: %.3f" %precision)
print(f"recall: %.3f" %recall)
print(f"confusion matrix: ")
print(conf_m)
timeforNB = time.time()-start_time + timeforsplit + timeforfeaturesExtraction + timeforcorpus
print(timeforNB, 'seconds to execute')

accuracy: 0.978
precision: 0.961
recall: 0.964
confusion matrix:
[[720  12]
 [ 11 292]]
177.6290581226349 seconds to execute

```

Fig. 14. Analysis of Naïve Bayes

```

accuracy: 0.978
precision: 0.961
recall: 0.964
confusion matrix:
[[720  12]
 [ 11 292]]
177.74045181274414 seconds to execute

```

Fig. 15. Confusion Matrix for Naïve Bayes

```
start_time = time.time()
from sklearn.svm import LinearSVC
model = LinearSVC()
model.fit(X_train, y_train)
pred = model.predict(X_test)

accuracy = accuracy_score(y_test, pred)
precision = precision_score(y_test, pred)
recall = recall_score(y_test, pred)
conf_m = confusion_matrix(y_test, pred)

print(f"accuracy: %.3f" %accuracy)
print(f"precision: %.3f" %precision)
print(f"recall: %.3f" %recall)
print(f"confusion matrix: ")
print(conf_m)
timeforSVC = time.time()-start_time + timeforsplit + timeforfeaturesExtraction + timeforcorpus
print(timeforSVC, 'seconds to execute')

accuracy: 0.972
precision: 0.951
recall: 0.954
confusion matrix:
[[717  15]
 [ 14 289]]
176.49558329582214 seconds to execute
```

Fig. 16. Analysis of Support Vector Machine

```
accuracy: 0.972
precision: 0.951
recall: 0.954
confusion matrix:
[[717  15]
 [ 14 289]]
176.49558329582214 seconds to execute
```

Fig. 17. Confusion Matrix for SVM

From the analysis part it is clear that the most suitable algorithm for classification of mails is logistic regression.

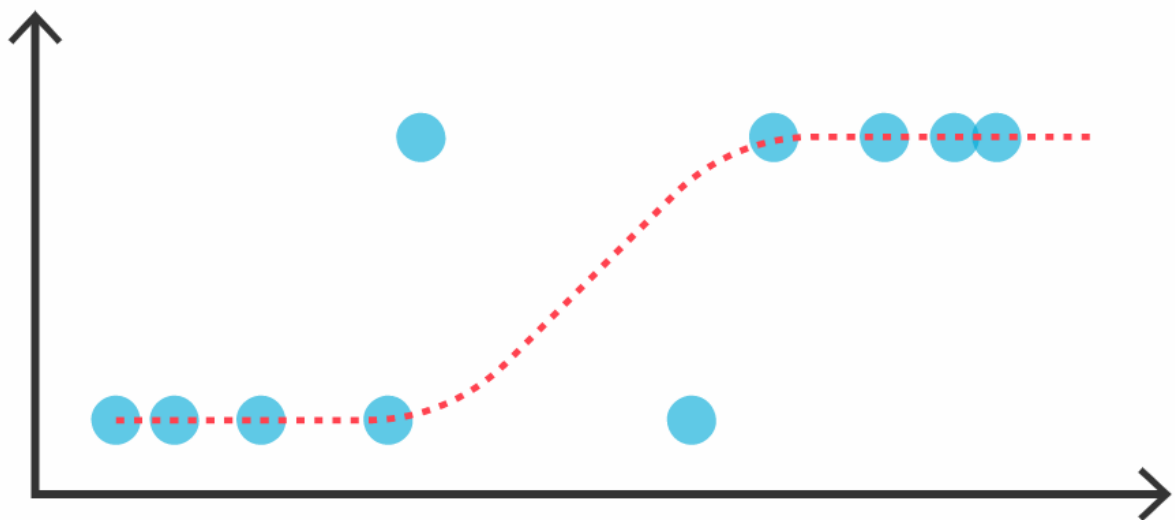
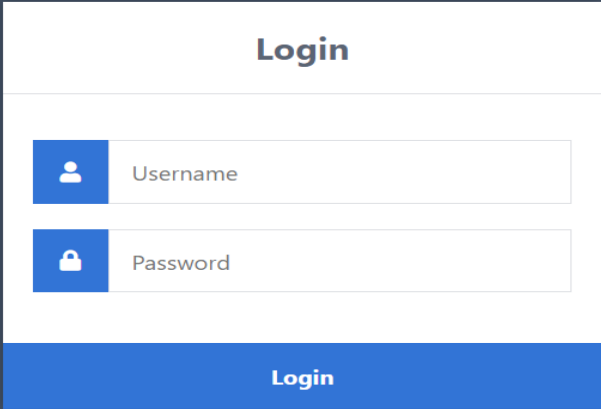


Fig. 18. Sample Logistic Regression Curve

Talking about the implementation, GUI was built with the help of flask. The GUI has been kept simple for ease of understanding and it starts with a login page. After the login details have been verified at the server side, backend processing starts which computes the reputation of senders by the algorithm that has been proposed. The user can open multiple browser tabs and

log into multiple accounts, hence creating a network of users. The calculations for the sender's reputation will now be done considering the sender's behavior in each one of the accounts that have logged in. This helps us predict and conclude more accurately.

Implementation



The image shows a login page with a dark blue background. In the center is a white login form. The form has a title "Login" at the top. Below the title are two input fields: "Username" with a person icon and "Password" with a lock icon. At the bottom of the form is a blue button labeled "Login".

Fig. 19. Login Page

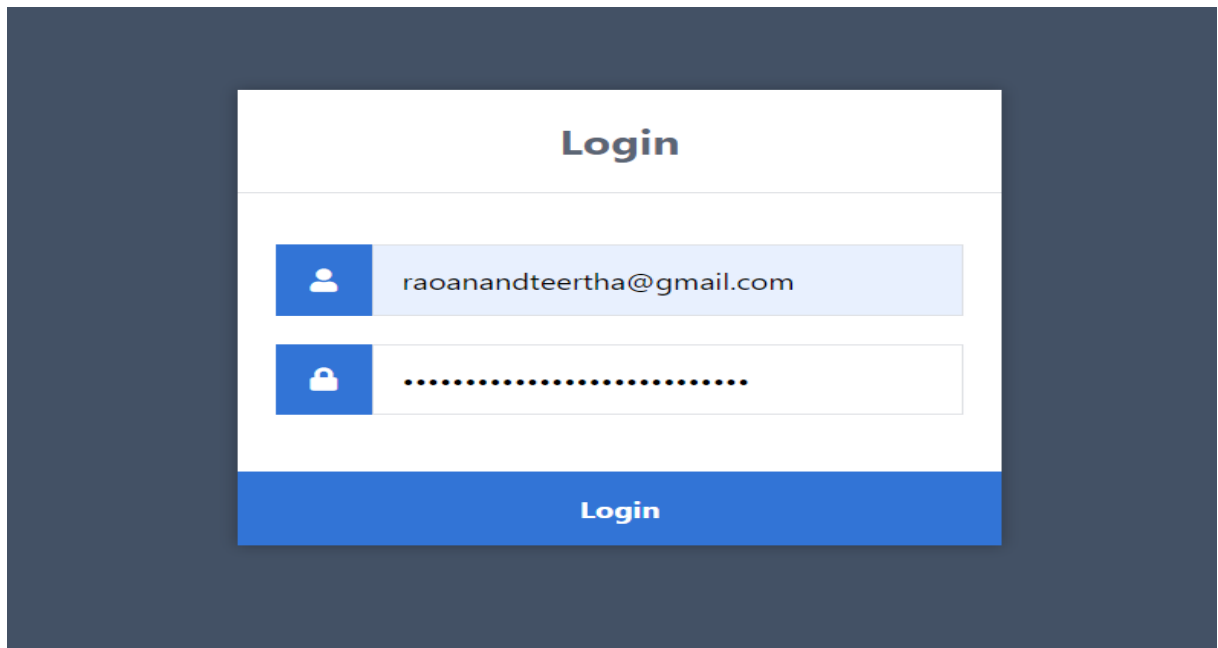
A screenshot of a login page. The page has a dark blue background. In the center, there is a white rectangular box. At the top of this box, the word "Login" is written in a bold, dark blue font. Below this, there are two input fields. The first input field has a blue icon of a person on the left and contains the email address "raoanandteertha@gmail.com". The second input field has a blue icon of a padlock on the left and contains a series of black dots, indicating a password. Below these two input fields, there is a solid blue rectangular button with the word "Login" written in white text.

Fig. 20. Login Page(filled)

Final Output Page

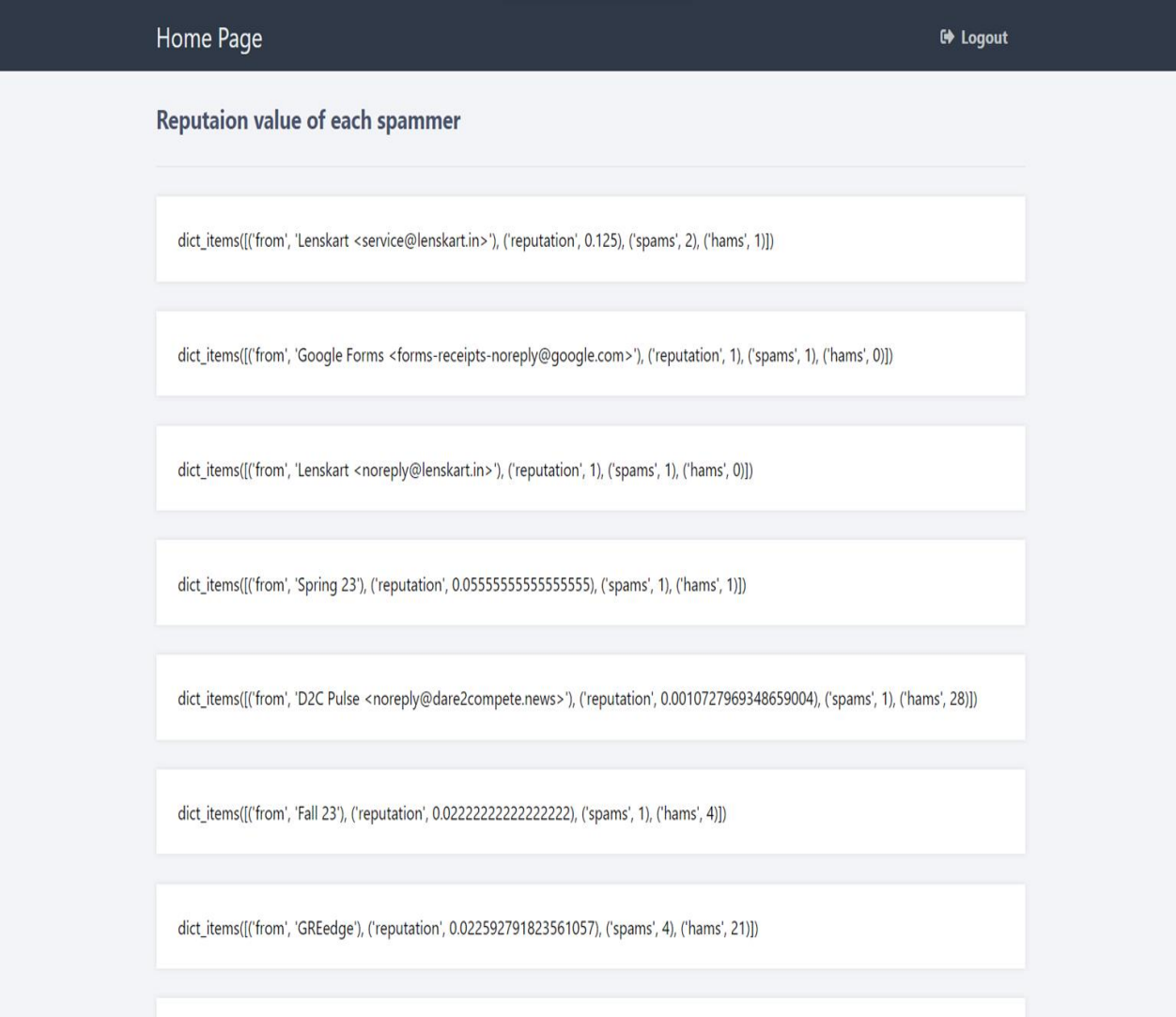


Fig. 21. Output Page

The output page displays the senders email address, along with the total number of spams and total number of hams considering multiple users. The reputation value is also displayed along with all of the other items.

Testing

Reputaion value of each spammer
<p>from = yagnesh narayanan <yagnesh.narayanan18@siesgst.ac.in> reputation = 0.0555555555555555 spams = 1 hams = 1</p>
<p>from = VAIDYA ATHARVA MAHENDRA <atharvavce118@gst.sies.edu.in> reputation = 1 spams = 1 hams = 0</p>
<p>from = Anandteertha Rao <raoanandteertha@gmail.com> reputation = 1 spams = 1 hams = 0</p>
<p>from = anandteertha rao <anandrao67@outlook.com> reputation = 0.0555555555555555 spams = 1</p>

Fig. 22. Testing(a)

Reputaion value of each spammer
<p>from = yagnesh narayanan <yagnesh.narayanan18@siesgst.ac.in> reputation = 0.1066666666666667 spams = 2 hams = 2</p>
<p>from = VAIDYA ATHARVA MAHENDRA <atharvavce118@gst.sies.edu.in> reputation = 1 spams = 2 hams = 0</p>
<p>from = Anandteertha Rao <raoanandteertha@gmail.com> reputation = 1 spams = 1 hams = 0</p>
<p>from = anandteertha rao <anandrao67@outlook.com> reputation = 0.1066666666666667 spams = 2 hams = 2</p>

Fig. 23. Testing(b)

<pre>from = D2C Pulse <noreply@dare2compete.news> reputation = 0.0555555555555555 spams = 1 hams = 1</pre>
<pre>from = Fall 23 reputation = 0.04166666666666664 spams = 1 hams = 2</pre>
<pre>from = GREedge reputation = 0.2222222222222222 spams = 4 hams = 1</pre>

Fig. 24. Testing(c)

<pre>from = D2C Pulse <noreply@dare2compete.news> reputation = 0.0555555555555555 spams = 1 hams = 1</pre>
<pre>from = Fall 23 reputation = 0.06530612244897958 spams = 2 hams = 4</pre>
<pre>from = GREedge reputation = 0.2962962962962963 spams = 6 hams = 2</pre>

Fig. 25. Testing(d)

CONCLUSION AND REFERENCES

Conclusion

- Emails have become one of the most widely used applications in today's world for easy and smooth communication. Since there is an exponential rise in usage of emails, there is also an increase in unsolicited emails or spam mails where the hacker tries to extract user information to steal their identity or money which is carried out through emails.
- Existing solutions based on content filtration or reputations have been considered for blacklisting certain senders based on their IP addresses, but this mechanism could be easily breached. In this, we have optimized an efficient solution which provides a cutting edge over the existing solutions.
- Based on the research papers and considering all the parameters collectively we could say that Logistic Regression is best used for classification of emails as it gives out most accurate results as well as best precision and recall rate. This could also be proved by the experimental implementation we have carried out [Fig.].
- After the model has classified the email as spam or ham, we are assigning a reputation value to each sender. This is necessary because there might be a situation where legitimate mail has been wrongly classified as spam as one can never have a classification with a 100% accuracy rate.
- Some of the current systems directly tag a sender as a spammer because the sender might have sent a few spam messages, but this might not always be the case. So we are also considering the amount of ham mails sent by the sender. This is because while considering both, the amount of spam mails and amount of ham mails we get a more accurate reputation value which finally helps us to classify whether a sender is a spammer or a legitimate user.
- In addition to this, we are also looking into the network where multiple users exist, in essence we are not just looking into a single user and classifying the sender but we are also considering multiple users in a network. When a sender sends spams to one user and ham mails to another user in a network we are considering the total amount of hams and spams sent by the sender across the network and based on this we are calculating the total reputation value to classify the sender as a spammer or a legitimate user.
- There are three feasible datasets that can be used based upon the number of spams and hams present. PU corpora, Enron corpus and SpamBase are some popularly used datasets in this field. However, we in this implementation have considered enron corpus as it was readily available and was of feasible size.

References:

- [1]: Detecting Spammers with SNARE: Spatio-temporal Network-level Automatic Reputation Engine [Authors: Shuang Hao, Nadeem Ahmed Syed, Nick Feamster, Alexander G. Gray, Sven Krasser]
- [2]: A Study of Machine Learning Classifiers for Spam Detection
[Authors: Shrawan Kumar Trivedi]
- [3]: Email classification using Machine learning Algorithms.
[Authors: Anju Radhakrishnan, Vaidhehi V: Department of Computer Science, Christ University, Bengaluru, India]
- [4]: Social Network Based Reputation Computation and Document Classification

[Authors: JooYoung Lee (Syracuse University, Syracuse, USA, jlee150@syr.edu), Yue Duan (Syracuse University, Syracuse, USA, yudian@syr.edu), Jae C. Oh (Syracuse University, Syracuse, USA, jcoh@syr.edu), Wenliang Du (Syracuse University, Syracuse, USA, wedu@syr.edu), Howard Blair (Syracuse University, Syracuse, USA, blair@syr.edu), Lusha Wang (Syracuse University, Syracuse, USA, lwang40@syr.edu), Xing Jin (Syracuse University, Syracuse, USA, xjin05@syr.edu)]
- [5]: Machine Learning for E-mail Spam Filtering: Review, Techniques and Trends
[Authors: Alexy Bhowmick (Tezpur University), Shyamanta M. Hazarika (IIT Guwahati)]
- [6]: Machine learning for email spam filtering: review, approaches and open research problems
[Authors: Emmanuel Gbenga Dada, Joseph Stephen Bassi, Haruna Chiroma, Shafi'i Muhammad Abdulhamid, Adebayo Olusola Adetunmbi, Opeyemi Emmanuel Ajibuwa]
- [7]: Efficient email classification approach based on semantic methods
[Authors: Eman M. Bahgat, Sherine Rady, Walaa Gad, Ibrahim F. Moawad]
- [8]: Trends in email classification research
[Authors: GHULAM MUJTABA, LIYANA SHUIB, RAM GOPAL RAJ, NAHDIA MAJEED, MOHAMMED ALI AL-GARADI, (department of Information Systems, Faculty of Computer Science and Information Technology, University of Malaya, Kuala Lumpur 50603, Malaysia)]






Websites Referred are:

- <https://en.wikipedia.org/wiki/Spamming>
- <https://en.wikipedia.org/wiki/Email>
- <https://www.malwarebytes.com/spam>
- <https://www.researchgate.net>
- <https://www.nltk.org>

Document Information

Analyzed document	final report (2)-converted.pdf (D134592650)
Submitted	2022-04-25T16:34:00.0000000
Submitted by	rizwana1
Submitter email	rizwana.shaikh@siesgst.ac.in
Similarity	9%
Analysis address	rizwana.shaikh.sies@analysis.arkund.com

Sources included in the report

W	URL: https://www.researchgate.net/publication/342113653_Email_based_Spam_Detection Fetched: 2021-01-27T06:37:41.5070000	 1
SA	SIES Graduate School of Technology / STAGE1 Project Report 118A1073,92,93.docx Document STAGE1 Project Report 118A1073,92,93.docx (D118436876) Submitted by: anindita.khade@siesgst.ac.in Receiver: anindita.khade.sies@analysis.arkund.com	 6
SA	SIES Graduate School of Technology / Final year project sem 7 Paper .pdf Document Final year project sem 7 Paper .pdf (D122019889) Submitted by: rizwana.shaikh@siesgst.ac.in Receiver: rizwana.shaikh.sies@analysis.arkund.com	 10
SA	NCITP42.docx Document NCITP42.docx (D75438092)	 2
W	URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6562150/ Fetched: 2019-10-22T11:16:47.4170000	 2