Phishing email detection using learning vector quantization

Author

Aqsa Mumtaz

Reg No. 170802

Supervisor

Dr. Usman Qamar

MASTERS

In

SOFTWARE ENGINEERING

Department of Computer Engineering

COLLEGE OF ELECTRICAL & MECHANICAL ENGINEERING

NATIONAL UNIVERSITY OF SCIENCES & TECHNOLOGY

ISLAMABAD

December, 2019

Phishing email detection using learning vector quantization

Author

Aqsa Mumtaz

NUST201464507MCEME35214F

A thesis submitted in partial fulfilment of the requirements for the degree of MS Software Engineering.

Supervisor: Dr. Usman Qamar

Department of Software Engineering

COLLEGE OF ELECTRICAL & MECHANICAL ENGINEERING

NATIONAL UNIVERSITY OF SCIENCES & TECHNOLOGY

ISLAMABAD

December, 2019

Declaration

I hereby certify that I have developed this thesis titled as "Phishing email detection using learning vector quantization" on the basis of my personal findings under the guidance of my supervisor Dr. Usman Qamar. All of the sources used in this thesis have been cited and contents of this thesis have not been plagiarized. No portion of the work presented in this thesis has been submitted in support of any application for any other degree of qualification to this or any other university or institute of learning.

Student Signature
Aqsa Mumtaz
Reg.

Language Correctness Certificate

This thesis has been read by an English expert and is free of typing, syntax, semantic, grammatical and spelling mistakes. Thesis is also according to the format given by the university.

Student Signature
Aqsa Mumtaz
Reg

Supervisor Signature
Dr. Usman Qamar

Copyright Statement

Acknowledgement

All praise and glory to Almighty **Allah** (Subhanahu Wa Taalaa) who gave me courage and patience to carry out this work. Peace and blessing of Allah be upon last **Prophet Muhammad** (Peace Be upon Him).

First of all, I would like to express my sincere gratitude to my advisor **Dr. Usman Qamar** for the continual assistance throughout my research work, for his motivation, determination, and dedication. I am so blessed to have such a co-operative advisor and kind mentor for my Ms Thesis.

Along with my advisor, I would like to express my gratitude entire thesis committee: **--------** and **--------**, for their prudent suggestions and encouragement. I thank **-------** for your help in dealing with Particle Swarm Optimization throughout my work.

Abstract

Phishing attack is defined as the attempt to get valuable information such as password, credit card information by gaining the trust of the users in electronic communication. Phishing detection system could possibly protect many losses that are happening due to the phishing attack. A background research performed during the project showed that today internet world faces many security threats where one of the major security threat is phishing. Phishing attacks create a serious risk for end-users. A thorough survey of approaches of defense mechanisms for the detection of the phishing e-mail attack have been discussed.

Phishing attacks are growing in great number every day; hence it becomes vital for us to take a stance to defend the email users from phishing attack. The proposed phishing email detection system will detect phishing mails using a technique known as Learning Vector Quantization. This phishing detection system will retrieve the unread emails of the users, classify the emails, detect the attack and alert the user about the attack. In this project, the phishing detection system was designed, developed and tested. Datasets were developed to train and test the models.

Finally, the technique used in the application was evaluated against other similar techniques to determine the effectiveness of the system which proves that the proposed technique achieved higher accuracy and lower false positive and the future works have been suggested.

Table of Contents

Table of Figures

Table of Tables

List of Abbreviations

LVQ – Learning Vector Quantization

PC – Personal Computer

URL – Uniform Resource Locator

IP – Internet Protocol

SVM – Support Vector Machine

NBC – Naive Bayes classifier

DNS – Domain Name System

HTML – Hyper Text Markup Language

TP – True Positive

TN – True Negative

FP – False positive

FN – False Negative

1 Introduction

Internet is facilitating users by provding many types of services. The continual advancement in technologies attracts institutions globally, to offer their services online, including ecommerce provider, banks and stocks. Now a day, people of this century increasingly rely on internet services, which is the source of great threat to their privacy and safety. Users can easily communicate through internet in the whole world and share any type the information.

As we know that email has made the communication process easier. But the means of internet is not too much secure because it is facing different types of attacks such as denial of services Gupta et al, (2016). On contrary, it acts as facilitator, but at the same time, it also become a major source of threat. It can lead to financial loss. To develop a trust worthy relationship, attackers always used to send email, try to provide a comfort zone to the user, deceive them into providing their confidential, either of credit card or of social network.

Phishing e-mail is one of the attacks in which the attacker attempts to get secrets information such as passwords, credit card information by trust worthy entry in electronic communication. A serious threat to the information security and internet privacy is caused by phishing email. Criminals attempt by installing malicious software on user's computer to steal personal information. Phishes use email for attacks because that is the point of sharing and they easily malware the PCs of the user and misguide about the websites. Phishing e-mail is become one of the famous topic for researcher because phishes email have been increasing day by day.

It is research based conclusion that, approximately 20% people those who are victims of phishing threat, due to their loss, they refuse to open email or attachments even the email look-legitimate. According to APWG phishing attack trends reports (Abu-Nimeh, S., Nappa, D., Wang, X., & Nair, S. (2007, October))., the number of phishing email increase from 28,897 to 45,628 unique. According to research, number of phishing email attacks are increasing day by day. It's an alarming situation. There are lots of machine-learning techniques to detect and filter phishing attack. Attackers are more advanced than ant phishing techniques, in order to bypass the detection, they are always ready to challenge the anti-phishing techniques by hosting classy techniques. Those who are unfamiliar with security indicators, can easily be deceived by the attackers. By analyzing the email header information such as message-id, sender email and return path, as this information’s cannot be easily veiled by attackers. Variety of solutions have been proposed by researchers, some of them are as follow (Chandrasekaran, M., Narayanan, K., & Upadhyaya, S. (2006, June)):

One of them is securing the login process by adding multiple authentication factors. Warning the user, don’t take such actions that could result in compromising their confidential information. Attackers design professional emails, that are similar to legitimated emails and websites, through this they can sway users to disclose their credentials. A recent study from google, attackers force users to disclose their credentials, after getting information, they immediately change their password, in order to hijack their accounts. After hijacking accounts, attackers try to make victim friends fool, through communicating with them by using hijacked accounts. The knowledge discovery model and data mining techniques are used to build an intelligent model, that can learn from existing phishing attacks, how to detect them and how to reduce them. Mostly, users are asked to update their account by providing information such as, credit card number, password. These sites are actually fake or modified, by clicking on the given URL, it redirects to another site. Top detect the phishing emails, the system evaluates the keywords included in the database, and then determines the content of the mail. When attackers create phishing attack, usually they used them for unrealistic news. It can be created around major events, anniversaries. Victim receives malicious file injection message via email or link. Their goal is to direct the user to install the malicious software, through it they can track force them on disclosing personal and financial information.

Detection of phishes is one of the challenges for the researchers. Many types of techniques are used for detection of phishes Al-Shboul et al, (2016). Phishing email is also termed as fraud email. Because the users are traps by forcing them to read phish email and get involved in the attack Gupta et al, (2016). The main focus of the phishing is to create panic in reading material such as email contents. These attack plays trick with the users by attractively focus to ‘click the link’ and ‘see the detail in the link’ Sheng et al, (2010). Also they contain some type of warning that if does not click then the bank account may be stolen and all your credit information lost. Email containing such warning is result of phishing attack. Sometime phishing attack focus on individuals to be victims. Phishing emails are widely increases to disturb the user by attack again and again Kirda et al, (2005).

In this thesis the main focus is to propose an efficient technique that easily detects phishing attack in email. Phishes email disturbs the working of the people by flooding into the inbox of the peoples. To avoid this type of situation a new technique using learning vector quantization (LVQ) is proposed. We take all types of e-mail under consideration such as donation e-mail, advertisement e-mail and photographic e-mail. The performance of the system is checked by the two indicators such as detection rate and false alarm rate. Higher the detection rate and lower the false alarm rate will produce accurate result. The evaluation of the technique is done with help of machine learning algorithm. The performance of the system is check by using e-mail datasets to know how much is phishing or how much are live email. Result will show the LVQ technique is superior that other types of techniques in term of performance

1.1 Aims

The aim of the project is to create mail reader application with security features that read the email and detect phishing. This mail client automatically checks to detect phishing using our proposed algorithm which is Learning Vector Quantization. It will alert the user by sounding an alarm if phishing email is detected.

The phishing detection technique will be tested alongside with other classification algorithms to measure its performance. The aim is to achieve better accuracy and lower false rate. This test will help determine whether the phishing detection system will be considered as effective or not.

1.2 Objective

Phishing is known as one of the activity in which phishers can attack the important secrets of the email. This paper presents the methodology for detection of the phishing email. The main objective of this proposed technique is to find optimal solution using Learning vector algorithm. A classifier is used to separate the email into two categories such as phishing email and legitimate email. Following is the list of objectives of the report:

i. To study the literature that has been done on learning techniques for detecting phishing email.
ii. To critically analyze the various type of phishing detection systems that are and how they are implemented in real life.
iii. To design the system according to the research.
iv. To develop a model using LVQ for detection of phishing email
v. To test the models comparing different models using various evaluation measures.
vi. To implement the necessary changes.
vii. To finalize the report.
viii. To arrive at conclusion and check the validity with previous work.

1.3 Report organization

The report is organized by covering seven chapter explain the concept of effective framework. The proposed framework has used learning vector algorithm (LVQ). In this algorithm, a classifier is used to select features from the data set of email and then classify the data set into different category to detect the phishing email.

Chapter **one** covers the introduction of the project. The aims and objective of the project are also discussed in this project.

Chapter **two** also explains the related work done for the technique in phishing attacks.

Chapter **three** explain definition of phishing and how phishing is done and what type of people are involved in it. And also the procedure of how phishing attack happens and what different methodology is used to tackle the phishing attack.

Chapter **four** gives an overview about the design of the system and the requirements.

Chapter **five** explains the details of the proposed framework used for detection of the attack, how it is implemented

Chapter **six** explains the detail of the experiment and its result.

The last chapter **seven** explain what the conclusion of the project is and other features added to the methodology to increase the efficiency of that system.

2 Literature review

So many researchers have been done for detection of phishing attack. Many of the researchers proposed machine learning algorithm for phishing detection. Here is the work done by the previous researcher, explained in terms of literature.

2.1 Literature Review on Classification techniques

Islam et al., (2013) proposed a technique based on multi-tier classification model for detection of phishing attacks. For this purpose, feature selection of email is done. For feature selection email messages, contents and topic of the message are selected and then give them ranking. In multi-tier model, the feature construction is done by determining each phishing email separate header of mail from rest of the email. Model is constructed in which messages of email are classify in a sequential manner. The analyzer analyzes the classify messages and store them in the message box that contains email. This model reduces the features selection problem and hence over all functionality increases. The selection is made upon highest rank number. For better result, the algorithm is schedule again and again for filtering. Rearranging the algorithm produced different result every time for the change with highest accuracy. In proposed approach, the e-mail messages are classified as sequential flow by using two different types of algorithm. The output is analyzed by the analyzer. The experimental result shows the significant impact of classifier rescheduling upon the model. The proposed model is robust and the solution is non-optimal. The output analyzes by analyzer not produce the accurate result. He classifier involve in the rescheduling of the solution.

Form et al., (2015) proposed technique based on hybrid features selection for detection of phishing attack. The hybrid features selection based on three sections such as behavior based features, URL based and content of the email. The proposed method first extracts features from the incoming e-mail. The similarity is measured between senders of the domain and message ID. Black list words are collected by observing the common features. IP address URL is collected. DNS implemented a database for maintaining different Domain name. When user enters domain name it is translated by the DNS into IP address. Dot in URL also extracted because it is a significant feature in email. Links in email are counted and the number is not greater than 5. If number is greater than 5 then we normalize the number to reduce its value. Symbols in URL also collected. The sender of email is also observing that either it is a unique observer or not. Extracted feature the classifier is train which can classify email as phishing e-mail and non-phishing e-mail. The proposed method is tested by taking the dataset of public email and SVM is used as classifiers to classify emails. Promising results are performed by features selection with minimum false rate but also there is a limitation some of the keywords and not consider graphical elements. The result shows highest extraction rate in detection of a phishing email. The proposed method also has a higher computational rate as compared to other previously explained data.

Sami et al. (2015) presents an intelligent frame work based on extracting a set of features by taking under consideration different parts of email. The selected features are classifying using classification algorithm. The focus of this paper is upon the importance of preprocessing stage for extraction of features. Feature selection model is implemented using Java programming. Email header and email content are focused for selection of features. For training and testing data, cross validation is used. The preprocessing phase is used to obtain information from corresponding available data set. In step one feature extracted and then match with the data set and result store in file. In next phase, four features are extracted from email. Filters are also applied on the hyperlinks such as count the number of links and domain also counts the encrypted hyperlinks. Phishing features in email are collected. The header of email contains important information such as sender and receiver address, content of email and ID of message. Description of features is also shown by maintaining different files. The data mining based algorithm is used for the detection method. 23 features were selected from e-mail and using these features email classified. Dataset is used to test the proposed frame work and data mining algorithm is used for the classification of phishing e-mail. Random forest algorithm produced good result in this case. This method suggests features selection in pre-processing phase result as great influence on the classification model. Besides this wide range of metrics are also used.

2.2 Literature Review on Data mining

The authors Pandey et al., (2012) present text and data mining approaches for the detection of phishing e-mail. For evaluation of the technique, a dataset is used for analyzing phishing and non-phishing email. Rapid minor is used to select keywords from dataset of email. To select optimal structure among the dataset the relation is built in the data with help of machine learning algorithm. High frequency keywords are separated from other keywords. For purpose of sorting GMDH algorithm is used. This algorithm is also used for the consideration of the subset of the components. This algorithm is also known as neural network algorithm. The genetic algorithm is used. Genetic programming has been done with selected features. The reason for the selection of genetic programming is that it has solved simple problems. In genetic programming, the features are stored in form of tree structure. The operators used in genetic programming are cross over and mutation operator. Text mining involves understanding the working with documents. Text mining algorithm gathers the features from the dataset. The information is presented in form of characters and words. The text matrix based technique is used for conduction of the experiment. This matrix consists of the frequency of each term. Through frequency, it is easy to determine which term has the greatest ranking in the matrix. The experiment is conducted for the feature selection from email. Feature selection produces remarkable result. Genetic programming should be preferred because it contains if else rules which help in detection.

2.3 Literature Review on Machine learning

Nowadays, one of the lucrative types of fraud that is committed by many attackers. This fraud is done for the purpose of financial or personal gains. Different classification techniques can be used to detect phishing email detection. By communication well serene messages, attackers usually effect their evil.

In real application classification techniques involving large databases and better perform. Users are still very vulnerable to attackers, even though continuously increasing growing efforts to educate users and creating better detection tools. Data mining is the process searching used to detect phishing emails, through large amounts of data and picking out relevant information. Data mining is powerful new technique, help researchers with great potential to remain focused on the most important information. Data mining technique, predict future trends and behavior. As there are many factors that can be used to differentiate the original document from the phishing one. To classify phishing emails, datamining proposed methodology, based on two approaches such as content-based and non-content based approach. There are two important criteria to check either it is a phishing email or not. If one is Valid, it means that email is legitimate. And if the criteria are fraud, the email is considered as phishing email. The initial objective is to measure the risk of email, using these techniques.

Profiling phishing e-mail is presented by Hamid et al. (2013). Profiling helps to determine activities related to phishing. In presented technique, a model is created based upon machine learning algorithm that functions to predict unknown from known class of data set. The data is to be train and further training set of data make sure the classification algorithm to automatically generate the model for classification model. To find pattern in the data set clustering is best known technique. K means algorithm is used in which value is assigned to the centroid of the data sets and centroid updated repeatedly. Clustering algorithm can be done in two parts. In part one, the input data is scan and plans to determine it either with existing cluster or new cluster has been started. The distance is calculated if data variables are consisting of continuous attribute. In part two sub clusters are obtained from pre-cluster and then make group into new cluster. First, phishing emails are detected then profile of phishing email can be done. Profiles are generated on the base of clustering. The clustering algorithm is used to determine the clustering. Through phishing, email phishers are also detected. The features are selected based on ranking. An algorithm is proposed for phishing email detection and filtration, then cluster is created of phishing email which helps either clustering is generalized or not. General profile helps to determine attacks related to phishing. However, clustering accuracy is not so accurate. In this paper which produces satisfactory result with high accuracy.

Narayanan et al., (2006) proposed works in the way by separating phishing email from non-phishing email using different structural features. The obtaining features are combining with support vector machine. With help of SVM, the classification of email is done easily. To circumvent the forged email, we use URL as a tool for handling this issue. Today in email URL are displayed in the form of dot by increasing the factor of suspicions. To detect the attack of phishing the body parts of received message in HTML forms are parsed. For a new user, it is difficult to tackle with the attacks. IDN help to mitigate this type of attacks. We restrict to only concentrate upon attacks related to phishing email. Phishes can attacks and hijack the session of the user. So, installing the packet sniffer will avoid the phishing email reached by the user. The content of messages requires analysis which helps in categorization of email. So, the main goal of proposed technique is to classify phishing email using structural properties that remain same throughout the process. One of the challenges faced by the proposed methodology is under classification of the messages with great accuracy. That’s why weak features known as noise classifier. The experiment is carried upon two steps. In the first step, the email was preprocessed so that blank messages will be avoided. The result shows that with SVM minimum error rate is produces all results are remarkable by highly classify phishing e-mail. But through experiments no broader conclusion has been drawn.

To detect phishing e-mail Fernando et al. (2013) proposed a novel technique based on centric approach. The sender centric approach focuses on the information sent by sender in email rather than email titles and its content. This paper focuses on the email sent or received by the bank. Proposed technique consists of two steps. First step separate banking messages from non-banking messages and that separation of message help to identify phishing email and non-phishing email. In next step, the focus is on banking messages. In next step, suspicious sender is identified for detection of phishing email. In phishing attacks, the phishes plan situation in which recipients reply the email by clicking on the link. So, verification of sender plays very important role that makes sure the information is related to bank data. The two steps approach help in detection of phishing attacks. If the phishes send an email that is like the non-phishing email then SVM classifier can tag as banking message and in next steps, these messages are considered as phishing email by giving them high ranks. The approach is evaluated by using the dataset that contains both phishing and non-phishing bank e-mail. Sender centric approach is proved to be flexible as well as feasible for detection of phishing email. The experiment performed on real world data sets.

Phishing email contains different types of action to be performed one of them is clicking on the URL. Hajgude et al. (2012) proposed technique that takes benefit from blacklist for increasing the accuracy in detection attacks. Two different types of analysis are used in heuristic approach one of them is URL and other is textual. The content of email is mostly same and match with each other so proposed technique focusses on the analyzing the URL and text of email. Classification accuracy of email is increased with help of the using hybrid approach. Features extracted from email consist of text information and the corresponding email. The framework consists of three main parts, one is analyzer second part is classifier and third part is look up system. Through DNS analyzer check either the email is phishing one or not. Classifier task is to check email DNS in blacklist and white list. Black list and white list is maintained through look up system. Text analysis and lexical analysis both done in the approach. This proposed method seems to be very effective as compared to the previous one. The focus of this technique is to reduce the number of false rates. Both things help in reducing the false rate such as DNS of incoming email and its analysis of text.

To detect phishing attack is one of the emerging challenges. Volkamer et al. (2017) proposed a new technique based on the concept of TORPEDO. This method helps people by identifying the suspicious links which cause the phishing attack. It helps from the phishing attack by delay the suspicious link to be clicked. Moreover, this approach provides the flow chart for the detection of phishing attack. By observing the flow chart, the users alert him and avoid clicking any link. The working of proposed approach is divided into several steps. In the first step the authentic link is created prominent and when user is unaware of which link must be click then a message display to avoid it. In next step, the prediction power of the system is enhanced. The links in email are observed to determine either they are not concerned with attacks. It also highlights the important links in the email so that users play special attention. Firstly, user must pay attention towards URLs. In next step, focus is on the highlighted URLs. The user has special check the sender name in URL to safe from a phishing attack. After checking sender name the extension also checks to avoid by clicking and get involved in phishing attack.

Aburrous et al. (2010) presents idea for solving the problem of phishing attack in online banking. The author suggests that if fuzziness is controlled in online system than phishing attack will also be in control at a great level. An intelligent system is proposed which effectively reduced the attacks. The frame consists of machine learning algorithm combine with fuzzy logic. The technique based on fuzzy system so in first phase, the low, high and medium values are assigned to the set of variables. Expert of the system suggests the rules for implementation of the fuzzy logic. Probability is assigned to different website according to the great chance of being involved in phishing attack. The intelligent and efficient approach has been implemented by using Matlab. The final point scoring of e bank of phishing attack is high. A high number indicates maximum chance of the website to be part of phishing attack. The main goal of this approach is to determine attacks by using machine learning algorithm. However, the features set to be selected is not to be excellent. This approach also does not take into consideration of new modern ways to help in detection. The classification approaches produce satisfactory result.

One of the crucial security challenge faced by internet society is phishing attack. Abdelhamid et al. (2014) proposed technique that is based on the data mining technique. Associative classification is presented as a solution to deal the phishing attack by detecting them. This paper also surveys different technique used for detection of the phishing attack. A developed AC is tested and its results are compared with other previously proposed classification algorithms. The algorithm represents correlation between different website and comparison should be done. The whole procedure is done in two steps. Firstly, URLs defected with phishing attacks are separated and placed on black list. When new website must be launched then it is cross checked with black list to determine either it has phishing affected or not. The result shows that proposed solution produced remarkable result with great accuracy

This paper Shah et al., (2009) presents technique for removing phishing. The techniques based on the principle of detecting the phishing pages and remove them. If any affected page is appearing on the website it is detected by the system and system acts by immediately remove this page. When page is displayed the alarm tell the system that phishing attack has been happening and then system place a tracer to trace the location of the attacker. IP address is saved in the system. Then the system send message to administrator to remove the suspicious page. The result of this proactive approach is show result with high accuracy.

Kumar et al. (2015) presents anti phishing approach by Cryptographic methodology is used in this novel approach. The image of website is separated into two parts the first part is used by the user and second part is stored on the system. All the clients put data on the website. After uploading the data if some suspicious activity is generated then it is handled with the help of cryptography methodology. When user wants to share data taken from the website then website authenticate the user by asking to provide user name, password and secret question. If user fails to provide any of the information then system detect the unauthorized user and declare as a phishing attack. The problem with this approach is to centralized nature in centralized system security risk is high because attacker can easily attack and take control of the system.

Fang et al. (2012) the natural immune system inspired the people for developing artificial immune system. This paper presents approach based on immune system for phishing email detection. Immune system not only provides a defensive mechanism for the whole body and its parts but also can cater with new type disease attack. Immune system is proved to be very effective in terms of designing new arterial network. Phishing is considering to be most awful act and day by day increases. In proposed system two types of detectors are used in the system. Memory detector and mature detectors help in detection. In step one the email is analysis with help of email address and the subject of the email. The two detectors are based on actual phishing detail. In medical immune system not only provide inborn information from host parents. So, the memory detectors have ability to detect every type of email containing phishing attacks. To test the performance experiment is done by examine the true rate and false rate. Detectors work the set threshold value will remain static that is why working is not done properly by the system. Training data set is used for memory detector generation and system mutation process generate mature detector. Incoming email if match with the dataset is detected by memory detector and new incoming email is detected by mature detector. The overall system is flexible in nature. The result shows that present solution provided by immune system in phishing detection is remarkable.

2.4 Conclusion Table of literature review

Authors	Technique	Produced Result
Islam et al., 2013	Multi-tier classification model	Produced optimal solution at high accuracy
Pandey et al., 2012	Text mining	Feature selection is moderate
Narayanan et al., 2006	SVM classification	Reduce error rate
Form et al., 2015	Hybrid feature selection	Low false rate
Sami et al. 2015	Feature extraction by classification model	Low false rate
Fernando et al. 2013	Centric approach by different algorithm	Produce flexible result
Hamid et al.2013	Profiling approach based on ML	High accuracy
Fang et al. 2012	Artificial immune system	High result

Hajgude et al. 2012	heuristic approach	Control false rate
Volkamer et al. 2017	TORPEDO	Flexible result
Aburrous et al. 2010	Fuzzy logic with ML	High accuracy
Abdelhamid et al. 2014	Associative machine learning	Satisfied result
Shah et al., 2009	Removing analyser	Satisfied result
Kumar et al. 2015	Cryptographic methodology	High result

Table 1: conclusion table of literature review

3 Overview of phishing and detection approaches

3.1 How phishing attack?

Phishing is fraudulent attempt to obtain sensitive information, by disguising oneself as a trustworthy entity. It often directs users to enter personal information at a fake website, which looks like legitimate site. Phishing is attacking directly at specific individuals or companies. They gather information and then use that information against their target to increase their probability of success. Phishing attempts directed at senior executives and other high profile targets in which attackers get content and address of legitimate and previously delivered email and use it to create an almost identical or copied email. Images are used by phishers instead of using text, to make it difficult for anti-phishing filters to detect the text. Anti-phishing filters are able to recover hidden content using optical character recognition.

The question is arising that how phishing attack begins? An email phishing attack starts when an attacker sends an email that looks like to be sent from a genuine party to victims. In phishing, email receiver is invited to click the link because such link has some attractive material to trap the people. If the user clicks on these links then some of the malware has attacked and hold their system control Stajano et al, (2011).

Almost twenty years ago such attack begins. But these attacks increase rapidly with time. These emails contain URLs and ask the users to update their information by clicking the URL link within the e-mail and follow the instructions. To start a phishing attack two fields are combined such as software engineering with computer technology to carry on this type of attack Aburrous, Hossain, Dahal, & Thabtah, (2010) day by day increment shows in terms of phishing website. One of the reasons behind websites increment is their attractive design which captures the viewer. However, it also seems that phishing attacks become complex day by day because of the phishers can pass many filters and break the security made by the anti-phishing people Dhamija et al, (2006). Moreover, phishing attacks force user to click the link. Let’s take an example a user connected to a server as soon as a phisher become online capture the activity of victims and send link so that victim clicks it and get victimize *Litan et al, (2004*).

The phishing life activities are more explained through phishing life cycle in figure 1.in this life cycle the phishes initiate the attack by creating Phish in form of email message and after creating phish it sends on the internet to reach email address. The user receives the phishing email. The user is not able to detect such attack. He considers that it might contain some important information from authentic source either it is from bank, office or even from friend. Keeping such thoughts, the user's clicks on the email and read the message and then opens all the links to get information in this way the user email disclose in front of the Phish. The Phish easily stores all the info.

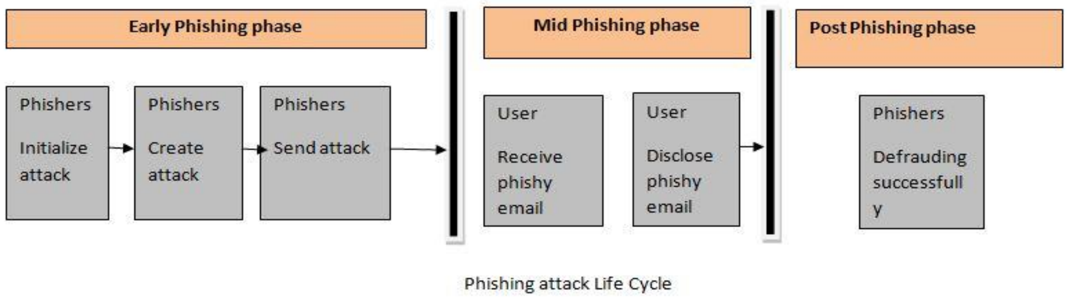

Early Phishing phase
Mid Phishing phase
Post Phishing phase
Phishers Initialize attack
Phishers Create attack
Phishers Send attack
User Receive phishy email
User Disclose phishy email
Phishers Defrauding successfull y
Phishing attack Life Cycle

Figure 1: phishing attack life cycle

3.2 Why people attempt phishing

To control phishing attack, it is important to know the factors that why people get involved in doing phishing attack. Phishers want to get some secret information such as passwords, data and account information. Some of the reasons for attempting phishers are as follows Jakobsson et al, (2008):

- People seem to have trust on email that is sent to their friends.
- People judge email by the design and logo used in email.
- Through research it is proved that phishing is done in emotion when a person is out of senses:

3.3 How to avoid phishing

If the user takes some counter measure, then he easily avoids phishing and saves his secret information being stolen. The user must be aware by using the email service. Below are the important points to keep in the mind Downs et al, (2006).

- First, focus on the layout of the email in the inbox and study the difference between layouts
- When you click on the download then whether it takes you from your URL to another
- Also, take under interest about the address mention in the email. If it from au authorized user then try not to click this
- Pay special attention to the address of the sender.
- Also, check the subject of the email
- Do not disclose the receipts. Make sure not to forward this type of email to the users
- Links related to the credit information never to be clicked

There are following basic guidelines to avoid phishing attacks that is most important for a email user to understand and apply.

Information about phishing techniques: Attackers are developing new phishing techniques all the time, you have to keep your eyes pared for news about phishing techniques. As early as possible, finding out about those attacks, you will be at lower risk.

Think before clicking: It is ok to click on link when you are on trusted sites. Attackers may claim that these emails are legitimate, it may exactly look like the same as original. They may ask you to fill the form, or may ask for your credentials.

Anti-phishing toolbars: Installing anti-phishing toolbars the best approach to avoid phishing attacks. These toolbars run quick checks on the sites and compare these links to the list on known phishing sites, and will alert you about any scam is found.

Keep updated browser: All the time, security patches are released for browser, so don’t ignore messages about updates, if you find any available updates, download and install it.

Firewalls are used: Providing high security to your system you have to use system firewalls and network firewalls as well. Using these two together, reduce the odds of hackers penetrating your system.

Anti-virus software: This will help you to protect your credentials from attackers. It helps to prevent damage to your system

3.4 Email

Over electronic communication, the process of sending and receiving messages is known as email. Email stands for electronic mail. Simple mail transfer protocol is used for transferring the email. Many other types of protocol are also used by the consumer over internet for sending and receiving email. In email, various email agent is involved. In this chapter, we go through the explanation of such agent.

- ✓ The email client is known as mail user agent. It is also known as a program to send an email and receive email. Example of mail user agent is outlook express for MS office and AIM mail from America and so on.
- ✓ The next agent is mail transfer agent. It is used to transfer email messages throughout the world with help of internet. Example of mail transfer is the send mail transfer.
- ✓ Delivery mail messages are the agent which is responsible for the delivery of messages from sender to receiver and vice versa.

3.4.1 Email flow

To understand phishing email, it is important to get the idea how email flow through internet. First, senders write a message and give it to the mail user agent which forward email to mail transfer agent. The sender also mentions the email address of the recipients. The sender also specifies the email domain while sending the email. Then from email domain, the location is determined. The mail transfer agent looks up into message exchange to confirm the address Siu et al, (2006).

3.4.2 Types of phishing attack

The objective and main purpose of phishing attack are to tackle the recipient activity according to the phishes attackers. A phishing messages or email looks like to come from the authenticated source such as from bank account that shows that user is satisfied *Kirda et al, (2005*). To trap the users there are certain types of phishing attacks such as spoofing email, phishing through social media and so on. This explains some different types of a phishing attack and the approaches used for their detection. The types of phishing attack make it difficult for recipient to identify the legitimate email from the phishing email. Spoofed email affects the great on the authentication of email user and website can harm by clicking on spoofed link. Some of the types of phishing attacks are listed as follows Hong et al, (2012). Spear Phishing Attack**,** Deceptive Phishing, Filter evasion, Whaling.

1. **Spear phishing**

Spear phishing is attacking directly at specific individuals or companies. They gather information and then use that information against their target to increase their probability of success. They attacked more than 1800 google accounts and implemented google .com domain to threaten the victim.

2. **Whaling**

Phishing attempts directed at senior executives and other high profile targets termed as whaling. Their content will be crafted to target an upper manager and the person's role in the company (Chandrasekaran, M., et. al. 2006).

3. **Clone Phishing**

It is a type of phishing attack, in which attackers get content and address of legitimate and previously delivered email and used to create an almost identical or copied email. The attachment is replaced by the malicious content; it may claim that it is the updated version of the previous one. This type of attack could be used to gain foothold on another machine.

4. **Link manipulation**

Commonly phishers use these tricks, misspelled URLs or using subdomains. To create web addresses visually identical, they use homograph attacks. Phishers have taken advantages of open URL redirectors of trusted organization to generate malicious URLs with trusted domain. It is quite possible for the phishers to purchase valid certificate, to takeoff a genuine website (Chandrasekaran, M., et al. 2006).

5. **Filter evasion**

Images are used by phishers instead of using text, to make it difficult for anti-phishing filters to detect the text. Anti-phishing filters are able to recover hidden content using optical character recognition.

Measure should be taken to control phishing. We should avoid opening an email from unauthorized users. To protect against phishing attack, the user must complete log in the account in two steps of verification to avoid trap from phishers. Some time for email the phishes set setting that when user enter to log in information on that spot all his credential information being stolen.

Phishing emails are increasing day by day. Many approaches are proposed for detection of phishing email. But no one proposed a solution to permanently get rid of from such attack. Much detection system has been proposed. We should take under consideration to trace phishes. And trace location and made some law to punish phishes.

3.5 Approaches for phishing detection attack

Phishing is technique that uses the combination of social and technology to gather sensitive information. Attackers use a trick, by employing different social engineering tactics such as threatens to suspend your account, if you don’t complete the process of account updates. According to the anti-phishing Working group, in march 2006, there were more than 18,450 phishing attacks and 9666 unique phishing sites. Millions of internet user are affected by phishing attacks. By incorporating, key structural features in phishing emails and employing machine learning techniques, we want to classify phishing emails. The main purpose is to learn labels of instances, either it is phishing or legitimate emails. Clustering is the type of unsupervised learning, assumed that there is no previous knowledge.

There are many approaches proposed for detection of phishing attack some of them are listed below such as:

1. Machine learning approaches for phishing attack
2. Neural network approach for phishing detection
3. Feature selection approach for phishing detection

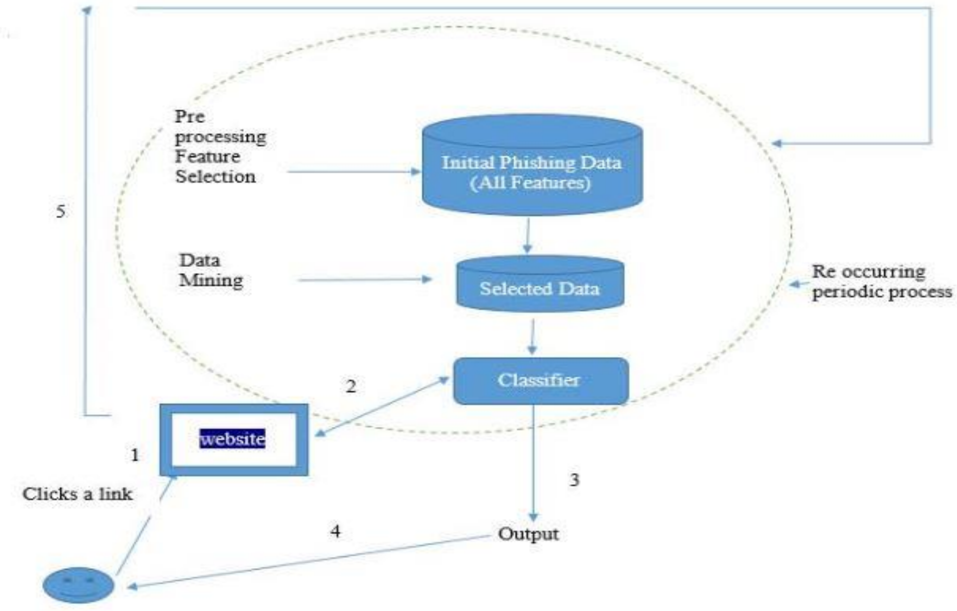
4. Filtering approaches
5. Data mining approaches

3.5.1 Machine learning approaches for phishing attack

Machine learning is most useful tool for the detection of phishing. Machine learning technique produced remarkable result for phishing detection as compared with traditional phishing approach. A detail has been shown here to determine the advantages and disadvantages to using ML in phishing attack detection. Different ML approaches are proposed to analyze their effect on detection attacks. To predict constructive model machine learning is one of the techniques used for this purpose explain by Abdelhamid et al, (2017). The phishing detection is converted into classification model that's why machine learning is best choice. The website often visited are saved in machine learning model and when user browses for the new site than the mode compare it with saved models and detect the phishing. Whenever user uses new browser website then it compares to determine the trend of the user. Automated phishing model is built with help of ML by capturing the features in database. Most of the machine learning deal with the phishing as binary problem and use ML algorithm to solve it Abdelhamid et al, (2017). To build the machine learning model must keep some of important points such as:

- ✓ To competing phishing by user what are important key role issues.
- ✓ For new user, how much knowledge is requiring building the model.

Basnet et al, (2008) explain that machine learning tools are as a data analyses tools. ML retrieves useful information from large data set to analyze the data. ML is embedded in the software for analysis purpose of features from large data set. ML solved problem related to large data set such as association between dataset, classification and pattern recognition system.



Pre processing Feature Selection
Initial Phishing Data (All Features)
5
Data Mining
Selected Data
Re occurring periodic process
2
Classifier
website
1
Clicks a link
3
4
Output

Figure 2: Machine learning Algorithm

ML is designed to give prediction about the target variable. The prediction is done with help of the classification model. The classifier is made from training data set. The main object achieved by classifier is to determine the target in large unseen data. Classification is done by classifier with higher accuracy by classifying the data.

Hence, as a result, the phishing involves an automatic system in which websites categorize automatically by set of features and classify the data. Some of the data have been taking as training data sets that contain specific data set. Automatically the besides are generated in full balanced. The ML based classifier easily classifies the websites into phishing and non-phishing attack. The accuracy of classifier is also checked to features is also extracted from the great check on the evaluation. Large data set is to be used for extraction. Cross validation methodology is used to use this service of classification; Php script plays important because it is embedded in the web browser to analyze different web sites. A special check is used upon the IP address of the computer system. The length of URL is also determined if it crosses the limit then the ML discover phishing email attack Xiang et al, (2011).

3.5.2 Neural network approach for phishing detection

To enhance the security of the system from phishing Nguyen et al, (2014) done feasibility study on phishing approach that uses the principle of neural network approaches. Through neural network, the phishing detection shows improvement in terms of accuracy. The reason behind the use of neural network is that the use of this approach in different fields such as vehicle system, process control would produce satisfactory results. Neural network supposed to be successful in phishing detection also. Neural network consists upon two methodologies that are supervised and unsupervised learning. Support vector machine is considered to under supervised learning. The drawback of SVM is that it cannot produce good phishing detection. So, to implement SVM considered some algorithm to increase the functionality of SVM.

Supervised learning algorithm consists of two main algorithms take under consideration. Global information is needed by the classifier to produce a result. For purpose of classification Adaline and back propagation is used in support vector machine. Adaline is the first stage for the classifier to look like. The network is containing weighted neuron. The similarity is observed in behavior of the weighted neuron as a result weights are adjusted by giving charge to input neuron cycle. The activation is activated by giving them the net value. Back propagation network is used for both purposes such as supervised and unsupervised learning. The activation function is also used in back propagation. To train artificial neural network this back propagation is used. The gradient function is used in this network approach. By considering all weights the gradient is calculated in this manner. Optimization procedure uses gradient function to update the weights in network Ma et al, (2009).

Support vector machine is just like classifier which analyzes the data pattern which helps in classification of data into different section. Classification of data is simple task done by vector machine. The phishing detection consists of different stages. In stage one, the data is collected which contain phishing and non-phishing email and websites. In stage two the knowledge is discovered from a set of data. If data is redundant then knowledge extraction is difficult. For knowledge, data filtration is involved which take time to produce result. The list is maintained for URLs, website etc. and block sites are discarded from the list. In last step, features are extracted from the data. To define characteristics of phishing features are extracted from the data. These features play important role in detection of phishing Sanglerdsinlapachai et al, (2010).

3.5.2.1 Genetic Algorithm approach for phishing detection

Phishing detection is also done by use of genetic algorithm stated by Kaui et al. (2015) which uses to assign weight to the features collected from data set. For assigning weight to the data first of all features has to be selected. After features extraction, the next step is used to assign weights. Almost ten different types of algorithm are used to apply on the features one of them included SVM, NBC etc. the purpose of genetic algorithm is to highlight the features and give maximum priority and hence accuracy of phishing detection increases. To protect user from financial loss the phishing detection is most important problem people face today. The genetic algorithm based approach for detection consists of four phases.

In phase 1 features are extracted from set of data. The features are extracted from each websites using the URL. The whole features are divided into two groups such as G1 and G2. G1 containing information related to DNS record and page information. G2 includes URL information and sub domain. In phase to data preprocessing has to be done in which each feature is separated into phishing class, non-phishing class and suspicious class. In second last phase 3, the best weight features are selected that describe the website in best manner. For this purpose features, weights are adjusted with help of genetic algorithm. In the last phase, the features with best phase are separated from the other features and best features are used to determine the phishing rate for detection. The threshold value is set and the each weighted feature is matched with threshold value to determine either dataset is phishing or legitimate one Ahmed et al (2012).

A java application is used for the purpose of feature selection. There are some rules defined for feature selection. After features selection weighted are assign. Information related to feature is adjusted as weight of the feature. Which feature provides the maximum information has maximum number of weight. Let's take example if A is attribute with values {a1,a2,a3……} then A is represented as different set such as A1, A2 ……..Am. By information gain, the accuracy of the system increased. In genetic algorithm, the chromosomes present weight for ten features. The genetic algorithm continuously applies cross over and mutation for obtaining better results. The weighted features efficiently calculated the fitness of the website.

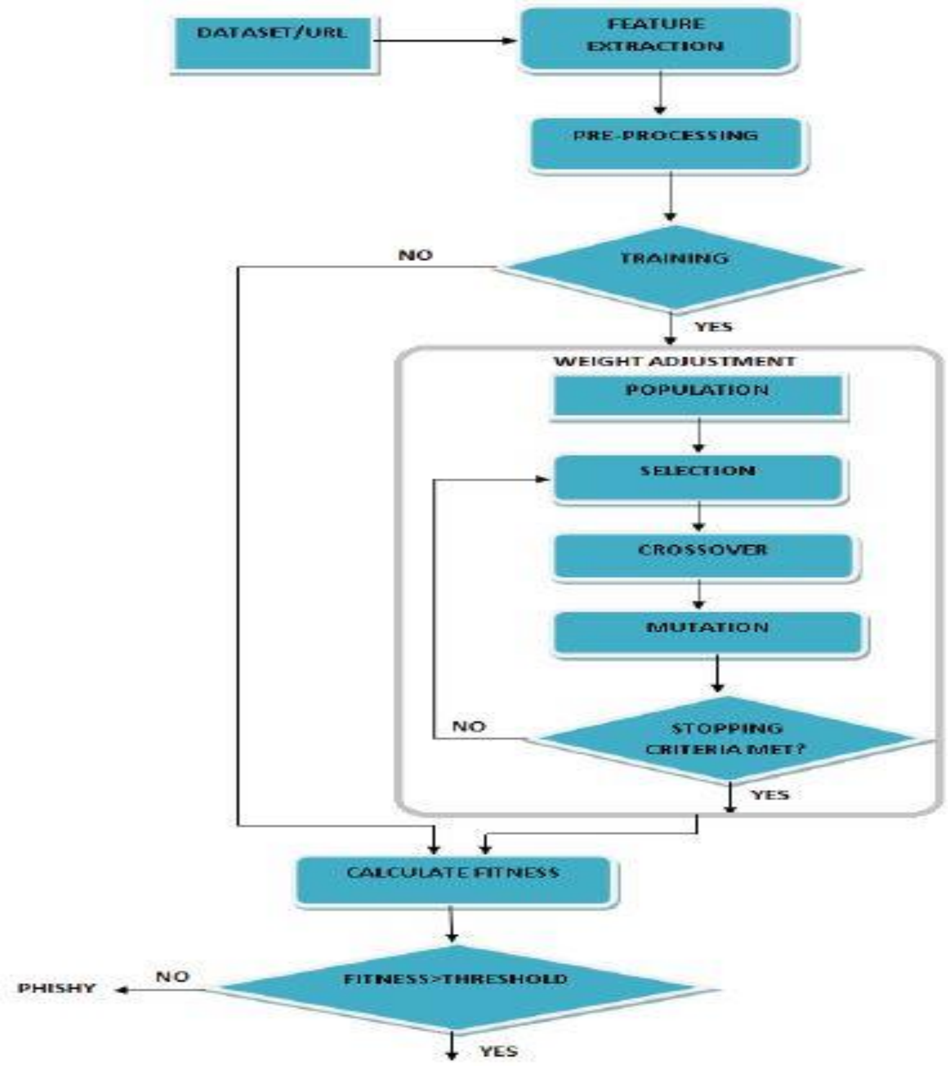


Figure 3: Genetic Algorithm approach

3.5.3 Filtering Approach

The filtering approach objective is to filter phishing email from other emails, so that phishing detection becomes easy. Phishing is a serious threat to global security and economy. Criminals are trying to convince email users, to reveal their personal data. According to the Anti-Phishing Working Group, there were 996 on average, unique phishing websites detected by APWD per day. On average, 141 unique brands were hijacked. There are large number of counter measures to scan phishing attacks. Keeping the filter up to date is extremely important as many new phishing scam are detected. We describe the active learning approach, which is able to help the emails provider in updating the filters to new phishing scams. It is also possible to use filters on client-side which would make processing of encrypted emails possible. There is a different between spammer and phisher, spammer only wants to contact the user and tried to offer them their products, on the other hand, phisher is very smart, they send an email with formal message, that email just look like the email come from some reputable institution. For the detection of phishing emails, it is natural to look at the external and internal structure of the email. Structural features contain the body parts of the email, containing four features, these are discrete and composite parts, alternative number of parts and different representation. It has also link features, that contains different properties of links contained an email. It also identifies, what kind of web techniques are used is reflected by element feature. By using training data, model for each class is generated, computed new message for different classes. To solve the problem of arbitrary sequence, a technique is developed. There are many other features. These are image features, including image distortion, in which, they introduce small distortions and stains, to defeat OCR tool, can be easily read by the humans, but reduce the accuracy of OCR. The other one is logo detection, in which, they completely design the same logo that is just look like the original one.

The filtering approach works on content based explain by Che et al, (2017). The filtering approach consists of two main parts one is separate email contents and second part preprocess these words for separation of phishing content. The last phase is classification phase in which legitimate email is separated from phishing email.

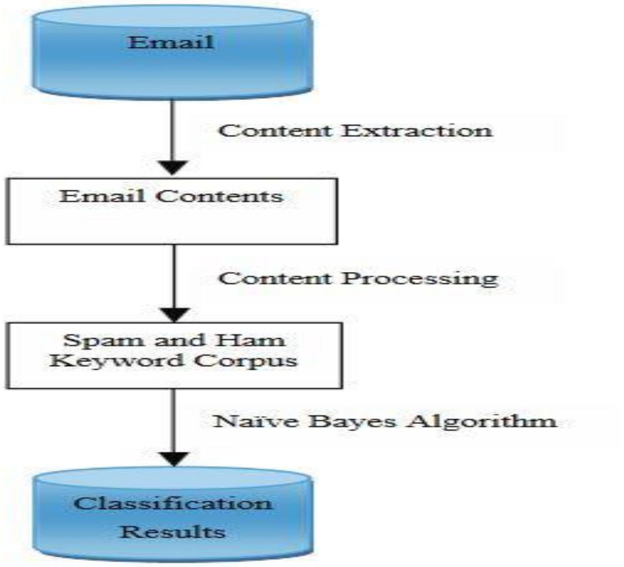


Figure 4: Filter Approach

The training step consists of many small phases. In phase one the data is extracted from training set and then is analyzed and separate phishing words to detect phishing. After extracted data, the data is converted in HTML form. Through classification algorithm, the words are separated consist on HTML and also separated special characters from the contents of email. After these steps, our extracted contents are saved for further process. In next process filter apply to separate the special alphabets. After applying filters the next step is to separate phishing words and determine frequency of each word present in the email contents. For experiment, the filter approach is implemented in either python or Java Ali et al, (2015).

4 System design and requirements

4.1 Classification algorithm

In learning technique, classification algorithm is the integral part of the system. In similar way, this phishing detection system need a classification algorithm to classify the emails to detect the phishing emails. Hence there is a requirement of classification algorithm.

As discussed in the literature review there are number of classification algorithm. This part of the report discusses the few selected classification algorithms which are also used for comparing the results with LVQ. The following are the algorithm that are discussed.

4.1.1 Decision Tree

A decision tree is a tree structure which classifies an input sample into one of its possible classes. Decision trees are used to extract knowledge by making decision rules from a large dataset. Decision tree classifier has a simple form which can be compactly stored and that efficiently classifies new data. The following example demonstrates working of decision tree algorithm Chourasia, Shikha, (2013).

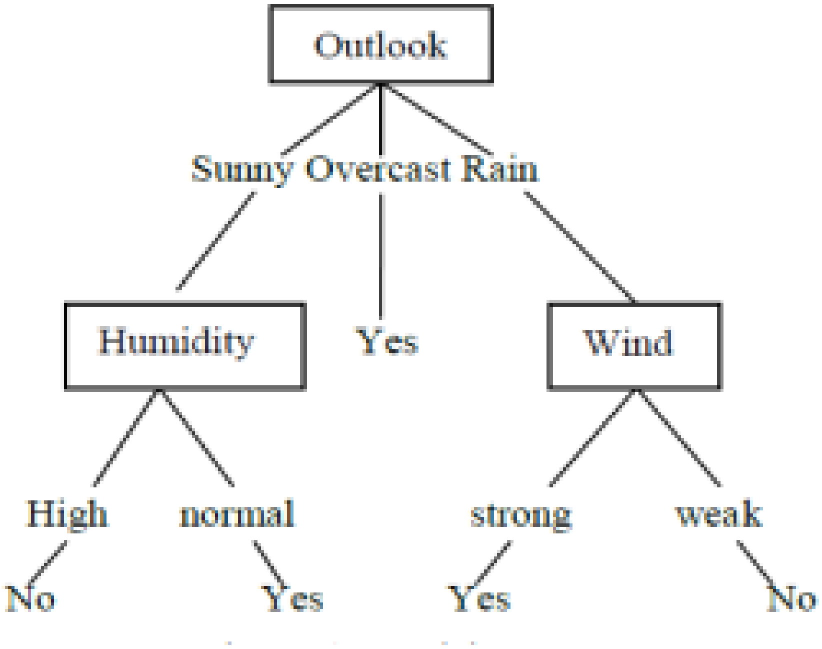


Figure 5: Decision tree Chourasia, Shikha, (2013).

4.1.2 Random Forest

Tree exists as soon as the training is used for sample collection and the replacement and leaves about one-third of the sample of all the circumstances. These data are to give a fair approximation of the classification of failure when the trees of the forest to join. Similarly, it consumes to make an approximation of varying degrees of importance Rodriguez-Galiano (2012).

Following made tree and adjacencies are calculated for each circumstance is used exactly the data in a tree. If a couple of circumstances control and conquer the like plant, is to improve the environment around them by 1. Last, dividing the number of trees that surround it are organized. Accustomed soon to replace the missing information, and identify outliers and create points of low-dimensional view of the disclosure of information. The following example illustrates working of Random Forest algorithm

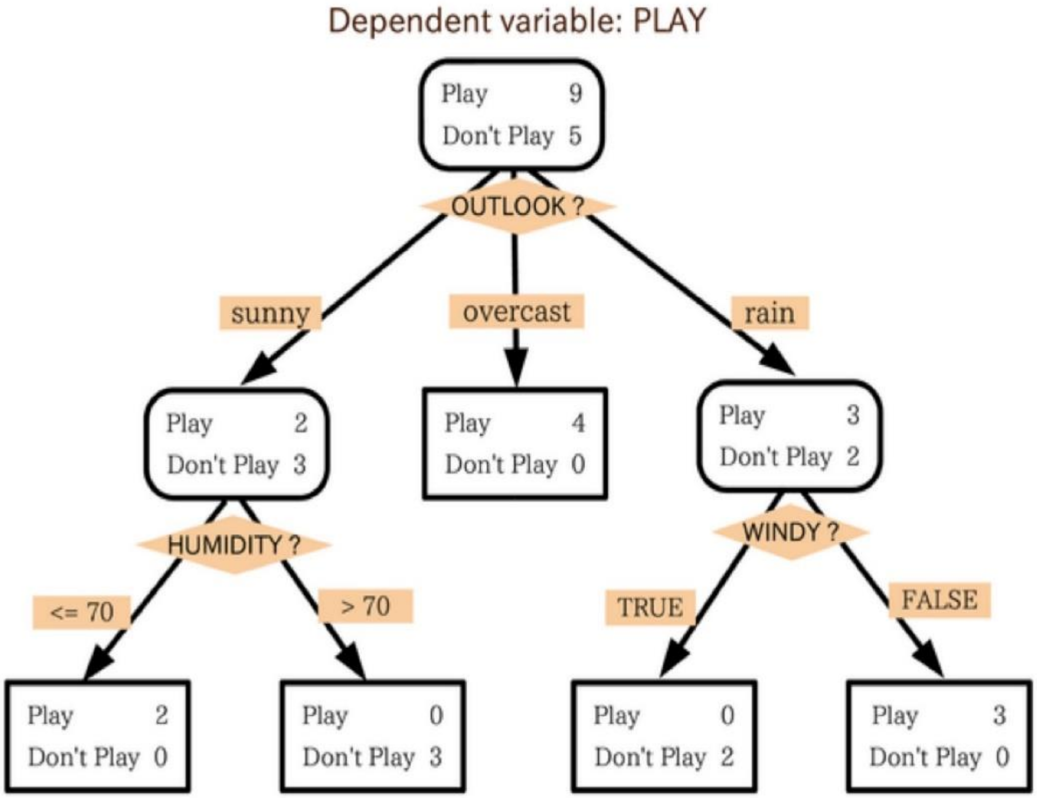


Figure 6: random forest Rodriguez-Galiano (2012).

4.1.3 K- Nearest Neighbor’s (k-NN)

In machine learning technique, (k-NN) is a method used for classification and regression. The input consists of the k closest training examples in the feature space Altman, N. S. (1992). The output depends on whether k-NN is used for classification or regression:

In k-NN classification, the output is a class membership. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors (k is a positive integer, typically small). If k = 1, then the object is basically assigned to the class of that single nearest neighbor. In k-NN regression, the output is the property value for the object. This value is the average of the values of its k nearest neighbors.

K-NN is a type of instance-based learning, or lazy learning, where the function is only approximated locally and all calculation is deferred until classification. The k-NN algorithm is among the simplest of all machine learning algorithms.

4.1.4 Naïve Bayes

Naive Bayes is the classification machine learning algorithm that relies on the Bayes Theorem. It can be used for both binary and multiclass classification problems. The main point of this algorithm is it relies on the idea of treating each feature independently. Naive Bayes method evaluates the probability of each feature independently, regardless of any correlations, and makes the prediction based on the Bayes Theorem.

The advantages of using this method include its easiness and simplicity of understanding. In addition to that, it performs well on the data sets with unrelated features, since the probabilities of them contributing to the output are low. Therefore they are not taken into account when making predictions. Moreover, this algorithm usually results in a good performance in terms of consumed resources, since it only needs to calculate the probabilities of the features and classes, there is no need to find any coefficients like in other algorithms. As already mentioned, its main drawback is that each feature is treated independently, although in most cases this cannot be true (Bishop 2006).

4.1.5 Learning Vector Quantization

Learning vector quantization is one of the best methods for classification of the data.This is the algorithm that is propossed for developing the system. The classification is done on the training data done by the content analysis. The architecture of LVQ is similar to neural network approach. In which input is obtained processed and given output and output it classifies into two different classes. In neural network approach, weight is assigned to the data and data with high weight is selected. This technique calculates weight in a heuristic manner. Neural network provides a better solution for classification of data. To train the neural network is considered being a difficult task. Neural network has the possibility for learning. Weight in network is adjusted by algorithm. The neural network model consists of two main parts. One is dealing with the input layer and the second part is the output layer. The input layer consists of six nodes where output layer consists of 1 layer. Neural network also has activation function that acts as a threshold. The neural network proposed two steps. In step one, it calculates the input for output and output for input. Training of network is also done. In this phase value below 1 are assigned to calculate weight of the data. In step two features are extracted from data for evaluation of the data. Two testing result is used for training the data.

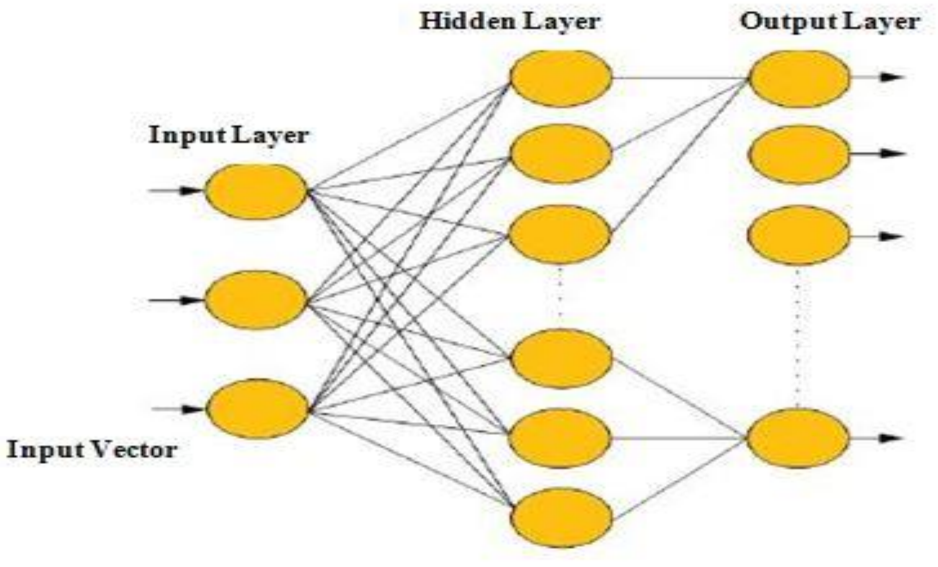


Figure 7: LVQ architecture

The contents of an email without header are given as an input. Weight is assigned to the contents in hidden layer and output is generated after assigning weight.

The LVQ algorithm consists of three steps:

Step 1: In step one, the training data is used to input in LVQ as input layer and consider G as input layer neurons. So, the formula for Input vector is as follows:

X= (y1, y2, y3 ……….yG)

Step 2: Weight assign in LVQ is represented by W. the W1 and W2 are the vectors for this weight. W1 is weight vector between input and hidden layer where W2 is weight vector between hidden and output layer.

Step 3: in step 3 each hidden layer is connected to the output layer. The weight vector is set to be one and zero. Hidden layer output and vector is calculated by formula:

Output= W1*X (W1 is weight vector between input and hidden layer)

Output Vector= W2 * O (W2 is weight vector between hidden and output layer)

The LVQ is used to classify input vector if the result of classification is correct then the connection of weights is updated. But if the result the result of classification is not correct then the value of weight is corrected the above process is continuously calculated until the highest classification rate is achieved. Learning data process has been done then the LVQ analyze new data for classification of email.

Proposed system using LVQ

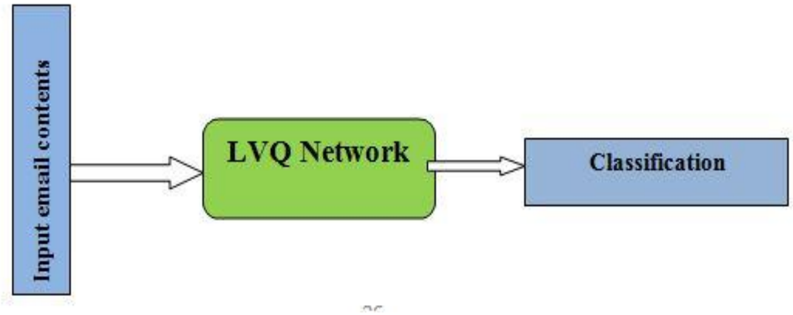


Figure 8: LVQ architecture for classification

After assigning weight to the selected content of email they are passed to LVQ network for classification of the contents. The classification process is done with help of feature vector using vector data. The feature qualifies the similarity between data with compare to the feature vectors. The feature vector captures repeating features used in the content of email.

The main objective in classification is to minimize the error in classification. Minimization of error is major objective in many classes. This is achieved through by making classes based on probability. But in practice it is not possible for us to do this. The methods based on neural network perform better result as compare to the other but prior knowledge of system cannot so much facilitate. To get rid of from such circumstances an ideal algorithm is one which is best and easy in learning perspective. This type of algorithm is found in the LVQ algorithm in which classification is done without any problem.

4.2 Design

4.2.1 Use Case Diagram:

Use case diagram shows the behavior of user, attacker and the system.

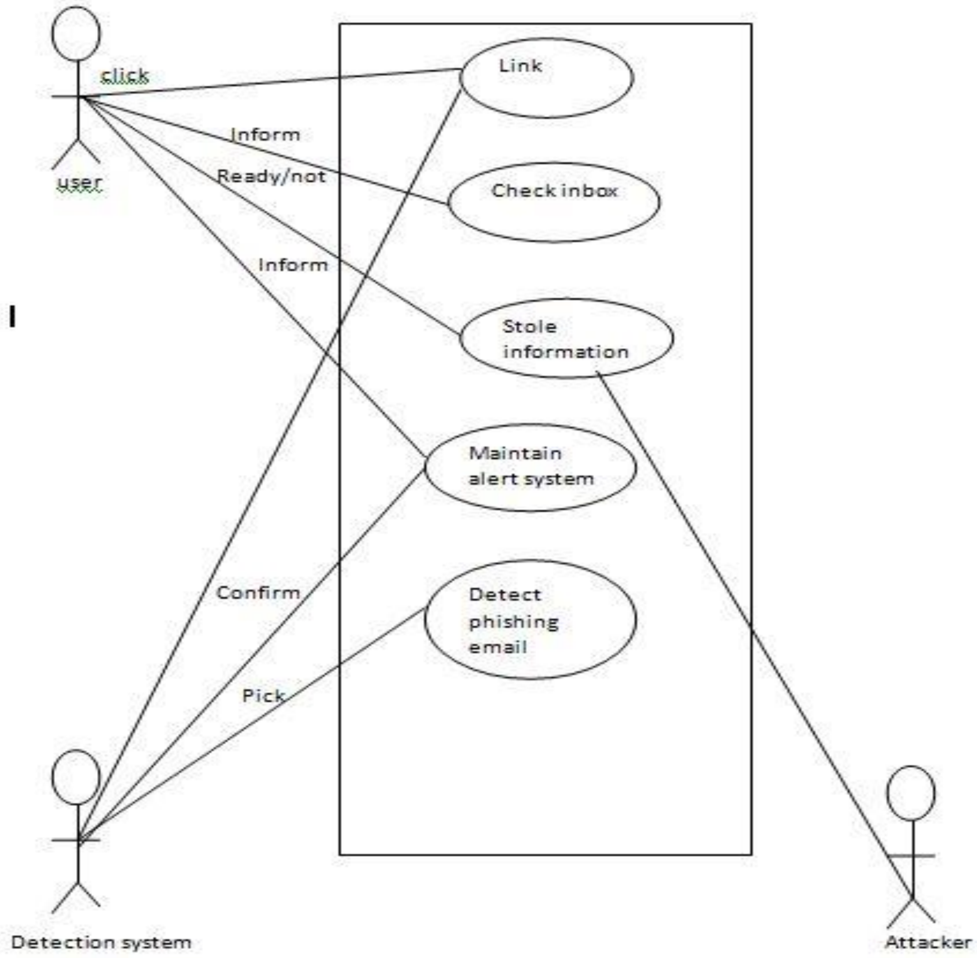


Figure 9: Use case diagram

4.3 Sequence Diagram

Sequence diagrams describe interactions among classes in terms of an exchange of messages over time.

- Sequence diagrams demonstrate the behavior of objects in a use case by describing the objects and the messages they pass.

- The horizontal dimension shows the objects participating in the interaction.

- The vertical arrangement of messages indicates their order.
- Fence Formats

Through sequence diagram the flow of data in the system is shown.

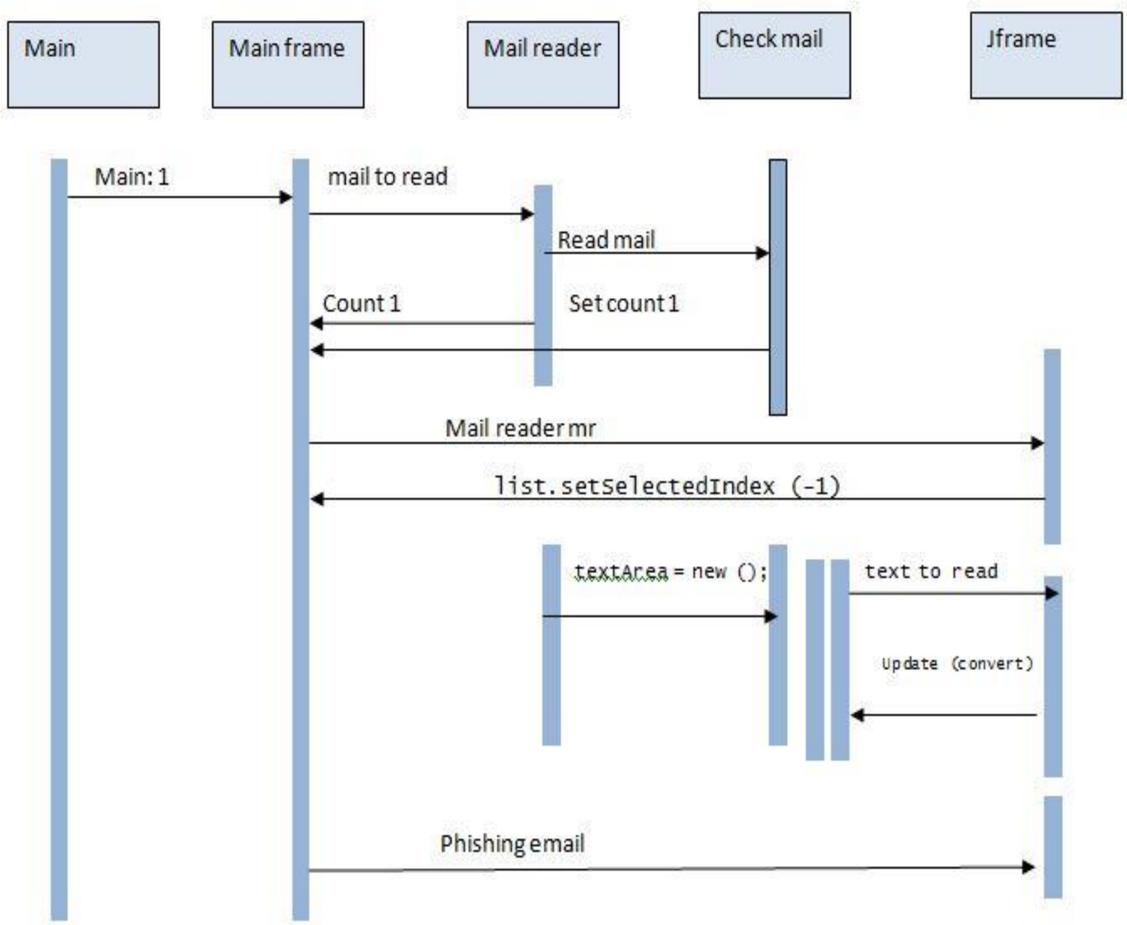


Figure 10: Sequence Diagram

4.4 Flow Chart

Flow chat represents the flow of system and actual working of the proposed code and it is represented by the following figure.

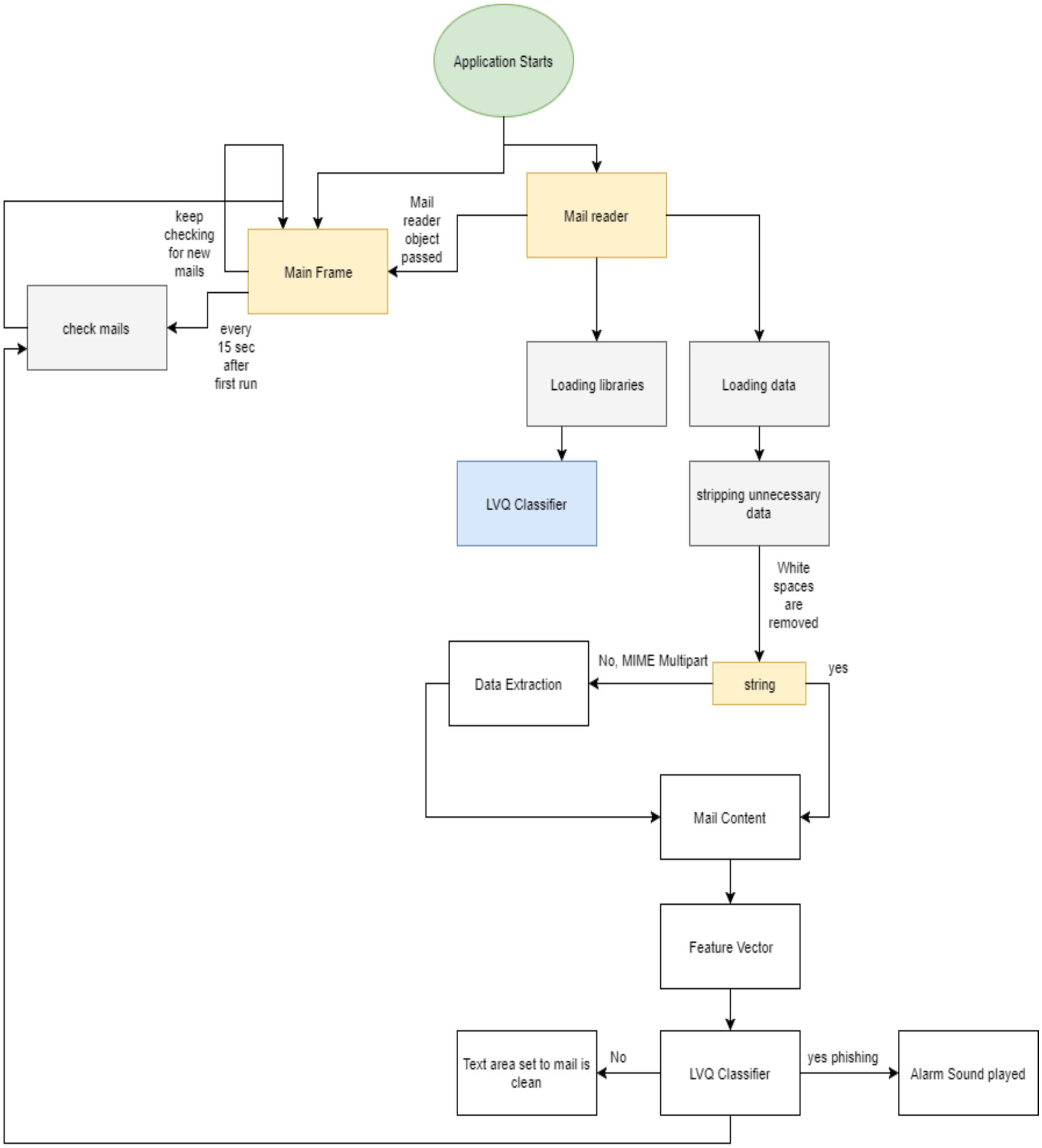
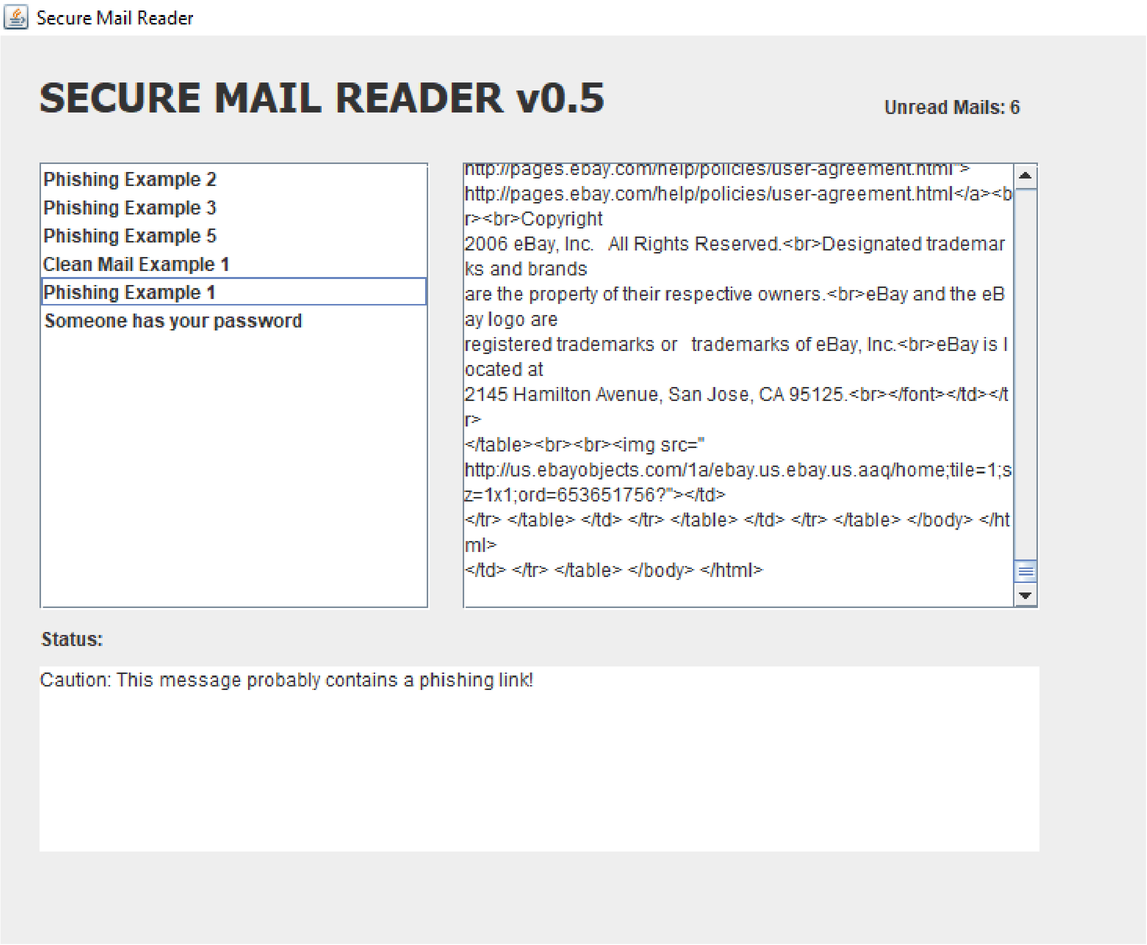


Figure 11: flow chart of the system

4.5 User Interface



```
http://pages.ebay.com/help/policies/user-agreement.html">
http://pages.ebay.com/help/policies/user-agreement.html</a><b
r><br>Copyright
2006 eBay, Inc.   All Rights Reserved.<br>Designated trademar
ks and brands
are the property of their respective owners.<br>eBay and the eB
ay logo are
registered trademarks or   trademarks of eBay, Inc.<br>eBay is l
ocated at
2145 Hamilton Avenue, San Jose, CA 95125.<br></font></td></t
r>
</table><br><br><img src="
http://us.ebayobjects.com/1a/ebay.us.ebay.us.aaq/home;tile=1;s
z=1x1;ord=653651756?"></td>
</tr> </table> </td> </tr> </table> </td> </tr> </table> </body> </ht
ml>
</td> </tr> </table> </body> </html>
```

Caution: This message probably contains a phishing link!

Figure 12: user interface

4.6 System Requirement

For the implementation of the phishing detection system resources required are given below

Software requirement

1. NetBeans: Java Development IDE
2. Language used: JAVA
3. Libraries- JSAT, JAVAMAIL API, Common-io-2.5
4. Datasets- Phishing corpus: Phishing dataset (https://monkey.org/~jose/) and HAM: Clean datasets. (http://csmining.org/index.php/spam-email-datasets-.html)

Overview of the resources

1. NetBeans – It is a software development platform which is written in Java and this platform allows applications to be developed in many different languages.

2. JAVA - It is the programming language that was used to develop the application

3. JSAT (Java Statistical Analysis Tool) library - These libraries are used in this project because it contains a lot of implementation that supports LVQ algorithm.

3.1 Common-io-2.5 – This is Input/ Output Library
3.2 JAVAMAIL API – This JavaMail API provides framework to build mail and messaging application

4. Datasets – Two datasets are used in which one is phishing dataset and other is the clean dataset. The dataset used for phishing is phishing corpus and the dataset used for clean mail is from the csmining spam email dataset. Only the clean mails are used. They are mainly used for developing the datasets. One is the training dataset and other is the testing dataset. This dataset is made using the phishing and the clean mail datasets.

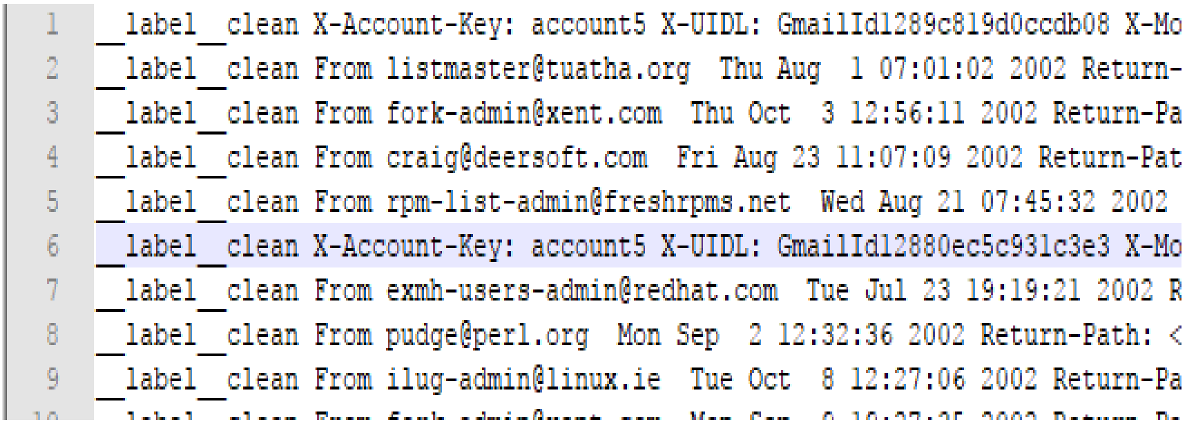
5 Methodology

In this chapter, the detailed explanation of how the proposed system for detection of phishing email is implemented is discussed.

5.1. Dataset selection

Dataset is required as the learning algorithm needs to be trained before it can predict. This is done using labeled datasets. Machine learning typically works with two data sets: training and test. The first set is the training set. Running a training set through a neural network teaches the network how to weigh different features, assigning them coefficients according to their likelihood of minimizing errors in your results.The data in numeric form will be contained in vectors, one for each layer of your network. They are the most important results will obtain from training a neural network.The second set is your test set. It functions as approving mechanism, and you don't use it until the end. After you've trained and optimized your data, you test your neural network against this final random sampling.

There seems to be only one dataset available publicly for Phishing e-mails. We also needed clean mails so we used clean mails from a SPAM detection dataset but we have nothing to do with spam, we use only HAM mails from it. So we used datasets that contains phishing mails and ham emails. The Phishing email dataset is taken from Phishing Corpus Monkey.org, (2019) and Ham Csmining.org, (2019)



```
__label__clean X-Account-Key: account5 X-UIDL: GmailId1289c819d0ccdb08 X-Mo
__label__clean From listmaster@tuatha.org  Thu Aug  1 07:01:02 2002 Return-
__label__clean From fork-admin@xent.com  Thu Oct  3 12:56:11 2002 Return-Pa
__label__clean From craig@deersoft.com  Fri Aug 23 11:07:09 2002 Return-Pat
__label__clean From rpm-list-admin@freshrpms.net  Wed Aug 21 07:45:32 2002
__label__clean X-Account-Key: account5 X-UIDL: GmailId12880ec5c931c3e3 X-Mo
__label__clean From exmh-users-admin@redhat.com  Tue Jul 23 19:19:21 2002 R
__label__clean From pudge@perl.org  Mon Sep  2 12:32:36 2002 Return-Path: <
__label__clean From ilug-admin@linux.ie  Tue Oct  8 12:27:06 2002 Return-Pa
```

Figure 13: CSDMC2019 mail dataset

Example of the clean labelled mails from the CSDMC2019 mail dataset.

Then we merged two datasets (actually 2000 entries from each) into one single phishing or not dataset which we call as model and we used this newly created dataset to train our algorithm. In many machine learning algorithms the training stage is required. The machine learning algorithms need to learn before they can decide on something they learn by training in this dataset there are 4000 mails.

Hence in this step of the process we have selected a dataset for training the…….

5.2 Dataset Preprocessing

After selecting original datasets, it is preprocessed. It is done so that the data is cleaned before converting to the vector. Things that are done in preprocessing are

1. Non utf8 characters removed,

2. New lines removed,

3. Headers removed,

4. Email addresses, proper nouns, company names, number removed,

5. Converted To Lowercase

6. Stop words ignored.

There are some words which are ignored known as stop words. They are words that are used very commonly in some language. They are ignored by this kind of software because they are considered ‘noise’ in most text processing applications. 99webtools.com, (2017).

5.3 Feature generation and selection

The feature that is used in the project are words, we selected our features using WordCounter.jar This program creates list of words in descending order along with the frequencies. We check whether the specific words exist or not. All the features are binary. Out of 4000 mails that were selected for the dataset are check to see if the words are present in them. A special check was done on emails by focusing on the most used words in all the 2000 phishing mails and then same process is repeated with the clean mails. After this we got the list of the most used words that are used in different email for 200 time. Then we check in how many different emails these words are included and we have removed the words that are more likely to exist in the clean mails. Hence there are no words for someone to exploit the system to show that phishing mail as clean. We didn’t check frequency of these words in mails, but only presence of them.

These are the feature words generated from the datasets

Feature words:

"please", "information", "thank", "click", "link", "security", "user", "protect", "help", "update", "produced", "member", "access", "agreement", "rights", "reserved", "customer", "notification", "service", "verify", "fraud", "notice", "attention", "recently", "ensure", "process", "department", "assistance", "included", "review", "activity", "need", "site", "visit", "personal", "preferences", "confirm", "secure", "matter", "apologize", "choose", "possible", "complete", "verification", "provide", "updated", "fraudulent", "password", "center", "protection", "inconvenience", "report", "request", "safety", "safe", "changes", "accordance", "problems", "suspended", "unusual", "change", "confidence", "suspension", "billing", "compromised".

While features have two distinct values and we said they are binary, they are not exactly 0 or 1. We use a custom metric to measure decisiveness (effect) of each word to the result. Then we weight every word of vector with that word’s importance.

Then weight is assign to every word of vector with according to word’s importance. This project applies decisiveness formula to calculate importance of the word in email. There is no zero occurrence of the word in each category. It is assumed that all words at least exist once in each of following category therefore as a result 0 occurrence has changed to Formula for Decisiveness:

The below words effect is calculated for the word please. The decisiveness for each selected word is calculated in the similar manner

By law of logarithm the affect is calculated.

log x x = loga a = 1

such as log6 6 = log10 10

LOG (phishing count/Clean count,2)

Log (1756/280,2)

phishing count= 1756

Clean count= 280

Log 2 ^ (1756/280)

X= Log 2 ^ (1756/280)

2^x= (1756/280)

2^x= 6.2714

Apply log on both sides:

Lox 2^x= log 6.2714

Xlog2= 0.797

X (0.30) = 0.797

X= 0.979/0.30

X= 2.648 Ans

For each word, the effect is calculated by using the same decisiveness formula using log.

Word	Phishing Count	Phishing %	Clean Count	Clean %	Decisiveness
Please	1756	87.80	280	14.00	2.6487941126
Account	1716	85.80	847	42.35	1.0186156782
Information	1413	70.65	239	11.95	2.5636789424
Thank	1321	66.05	53	2.65	4.6394942967
Click	1162	58.10	97	4.85	3.5824815112
Link	1105	55.25	124	6.20	3.1556343439
Security	1097	54.85	97	4.85	3.4994349682
User	1018	50.90	225	11.25	2.1777406549
Protect	922	46.10	22	1.10	5.3891913218
Help	913	45.65	225	11.25	2.0206898588
Update	896	44.80	114	5.70	2.9744649079
Produced	826	41.30	134	6.70	2.6239087810
Member	822	41.10	19	0.95	5.4350670702

Access	733	36.65	93	4.65	2.9785105770
Agreement	693	34.65	5	0.25	7.1147834472
Rights	665	33.25	106	5.30	2.6492900758
Reserved	645	32.25	97	4.85	2.7332425081
Customer	634	31.70	19	0.95	5.0604115167
Notification	634	31.70	12	0.60	5.7233765294
Service	623	31.15	137	6.85	2.1850562701
Verify	599	29.95	222	11.10	1.4319963264
Fraud	573	28.65	6	0.30	6.5774288280
Notice	525	26.25	37	1.85	3.8267202469
Attention	522	26.10	29	1.45	4.1699250014
Recently	521	26.05	77	3.85	2.7583530216
Ensure	514	25.70	21	1.05	4.6133071264
Process	513	25.65	114	5.70	2.1699250014
Department	509	25.45	13	0.65	5.2910821279
Assistance	506	25.30	8	0.40	5.9829935747
Included	503	25.15	56	2.80	3.1670596677
Review	478	23.90	68	3.40	2.8134039667
Activity	461	23.05	9	0.45	5.6786979390
Need	455	22.75	339	16.95	0.4245812719
Site	452	22.60	114	5.70	1.9872889483
Visit	449	22.45	50	2.50	3.1667154450
Personal	448	22.40	78	3.90	2.5219527032
Preferences	437	21.85	53	2.65	3.0435690149
Confirm	433	21.65	12	0.60	5.1732607140
Secure	423	21.15	134	6.70	1.6584246627
Matter	422	21.10	59	2.95	2.8384561393
Apologize	419	20.95	1	0.05	8.7108064337
Choose	417	20.85	53	2.65	2.9759831189
Possible	416	20.80	146	7.30	1.5106151593
Complete	416	20.80	57	2.85	2.8675497040
Verification	395	19.75	3	0.15	7.0407463423
Provide	394	19.70	52	2.60	2.9216121013
Updated	390	19.50	54	2.70	2.8524428116
Fraudulent	385	19.25	1	0.05	8.5887146356
Password	382	19.10	31	1.55	3.6232325176
Center	381	19.05	70	3.50	2.4443641705
Protection	381	19.05	30	1.50	3.6667565919
Inconvenience	364	18.20	1	0.05	8.5077946402
Report	364	18.20	138	6.90	1.3992701834
Request	356	17.80	42	2.10	3.0834160082
Safety	355	17.75	11	0.55	5.0122435958
Safe	354	17.70	26	1.30	3.7671658319
Changes	352	17.60	82	4.10	2.1018796140
Accordance	352	17.60	1	0.05	8.4594316186
Problems	344	17.20	180	9.00	0.9344116584
Suspended	323	16.15	2	0.10	7.3353903547
Unusual	323	16.15	13	0.65	4.6349506366

Change	321	16.05	199	9.95	0.6898048666
Confidence	284	14.20	11	0.55	4.6903155009
Suspension	269	13.45	1	0.05	8.0714623626
Billing	260	13.00	1	0.05	8.0223678130
Compromised	257	12.85	8	0.40	5.0056245492

Table 2: Decisiveness table

In above table record of 66 words with their decisiveness is shown in figure. Calculating the feature vector value this project load data in ARFF file. Then loaded ARFF file format is used in training the model. Feature extraction process creates a vector that is required to train the classifier each line represents one email in the vector form that was converted based on the selected feature vector. The last element of each line is clean itself so that our model knows when you see something like the number sequence the result is clean as you see our dataset is labelled by humans, this is marked as clean.

5.4 Feature extraction

This stage of the project is known as feature extraction which is one of the important stages in the process. The process of extracting data from the files is called feature extraction. The goal of feature extraction is to obtain a set of informative and non-redundant data. It is essential to understand that features should represent the important and relevant information about our dataset since without it we cannot make an accurate prediction. That is why feature extraction is often a non-obvious task, which requires a lot of testing and research. Moreover, it is very domain-specific, so general methods apply here poorly. Another important requirement for a decent feature set is non-redundancy. Having redundant features i.e. features that outline the same information, as well as redundant information attributes, that are closely dependent on each other, can make the algorithm biased and, therefore, provide an inaccurate result. In addition to that, if the input data is too big to be fed into the algorithm (has too many features), then it can be transformed to a reduced feature vector (vector, having a smaller number of features).

We extract information based on selected features from e-mails using FeatureExtractor.jar program.

Feature vector

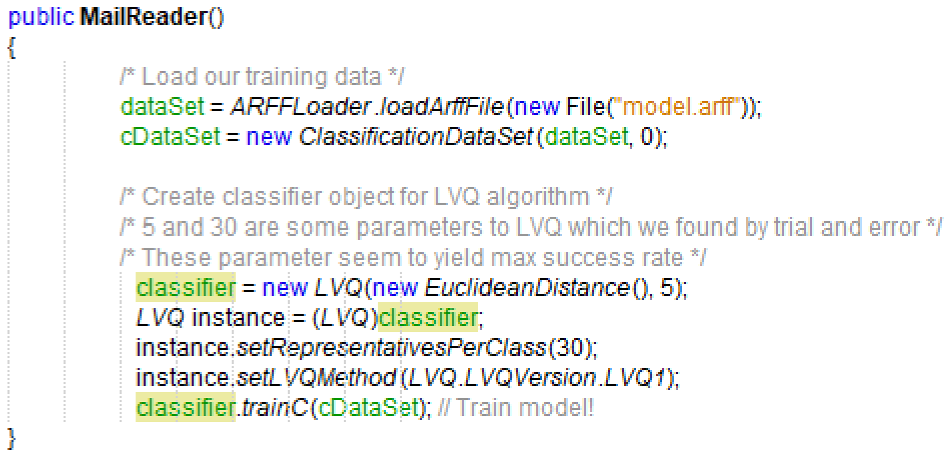
```
0,1.0186156782,0,0,0,3.1556343439,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1.4319963264,0,0,0,0,0,0,0,0,
0,0,0,0.4245812719,0,0,0,0,0,0,0,0,0,1.5106151593,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,clean
```

5.5 Training the classifier

Training the model is one of the basic of learning techniques. Training is done based on the input dataset, and the model that is built is subsequently used to make predictions. The output of such model depends on the initial task and the implementation.

The input dataset must contain the correct answer, which is known as a target or target attribute. The learning algorithm finds patterns in the training data that map the input data attributes to the features (the words that you want to predict), and it outputs a learning model that captures these patterns.

The next step after selecting the dataset and the feature extraction is to train the classifier with the dataset and the feature vector. The main purpose of training the data is to create a model from which the LVQ algorithm learn the behavior of system. We just convert dataset to list of feature vectors (ARFF) format and pass our learning implementation these vectors. It does itself automatically. It’s as how a neural network trains itself.



```
public MailReader()
{
        /* Load our training data */
        dataSet = ARFFLoader.loadArffFile(new File("model.arff"));
        cDataSet = new ClassificationDataSet(dataSet, 0);

        /* Create classifier object for LVQ algorithm */
        /* 5 and 30 are some parameters to LVQ which we found by trial and error */
        /* These parameter seem to yield max success rate */
          classifier = new LVQ(new EuclideanDistance(), 5);
          LVQ instance = (LVQ)classifier;
          instance.setRepresentativesPerClass(30);
          instance.setLVQMethod(LVQ.LVQVersion.LVQ1);
          classifier.trainC(cDataSet); // Train model!
}
```

Figure 14: Training Classifier

This function of mail reader is the constructor

Pseudocode

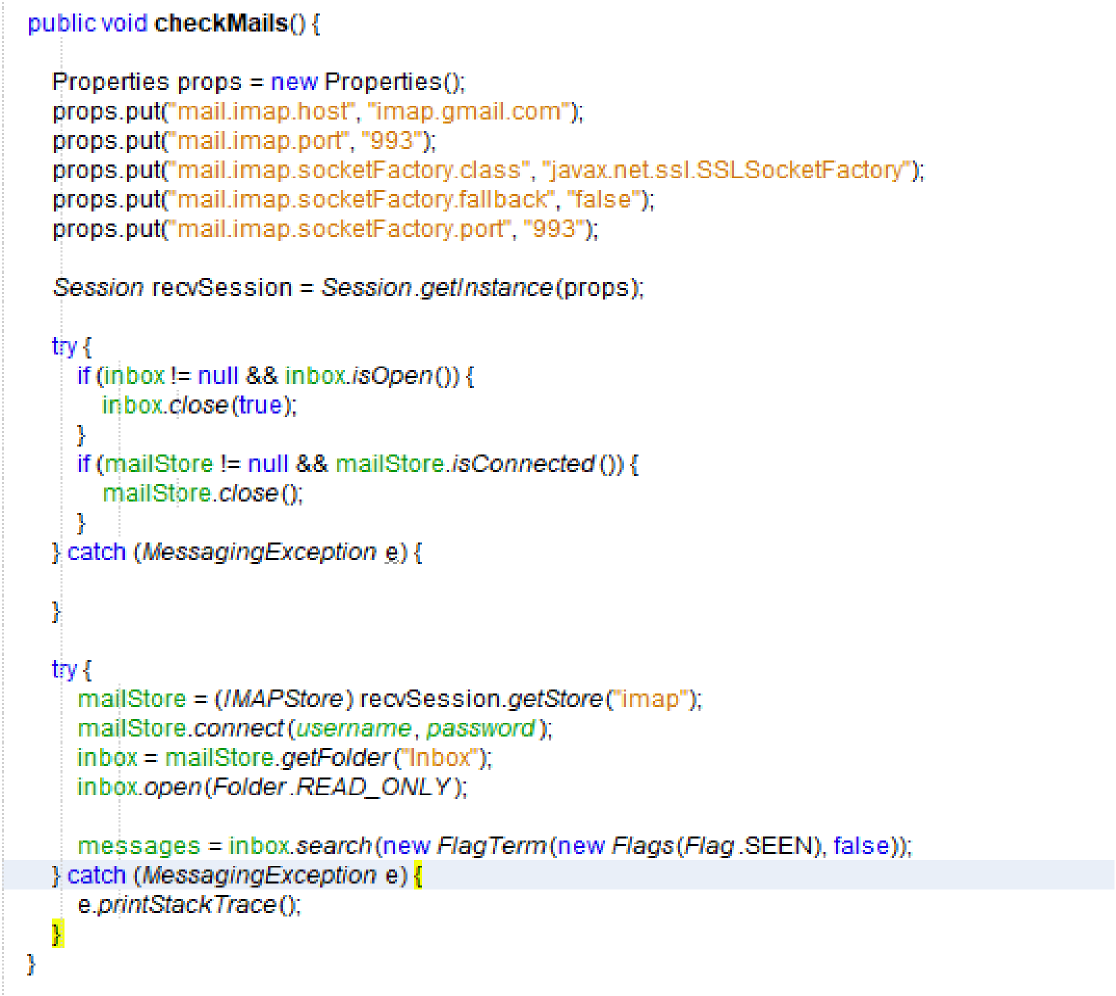
```
1. Load Training data
2. Create classifier object for LVQ algorithm
3. Parameters are provided to the object of LVQ.
4. Classifier is trained with the dataset.
```

5.6 Check mail

5.6.1 Log into email

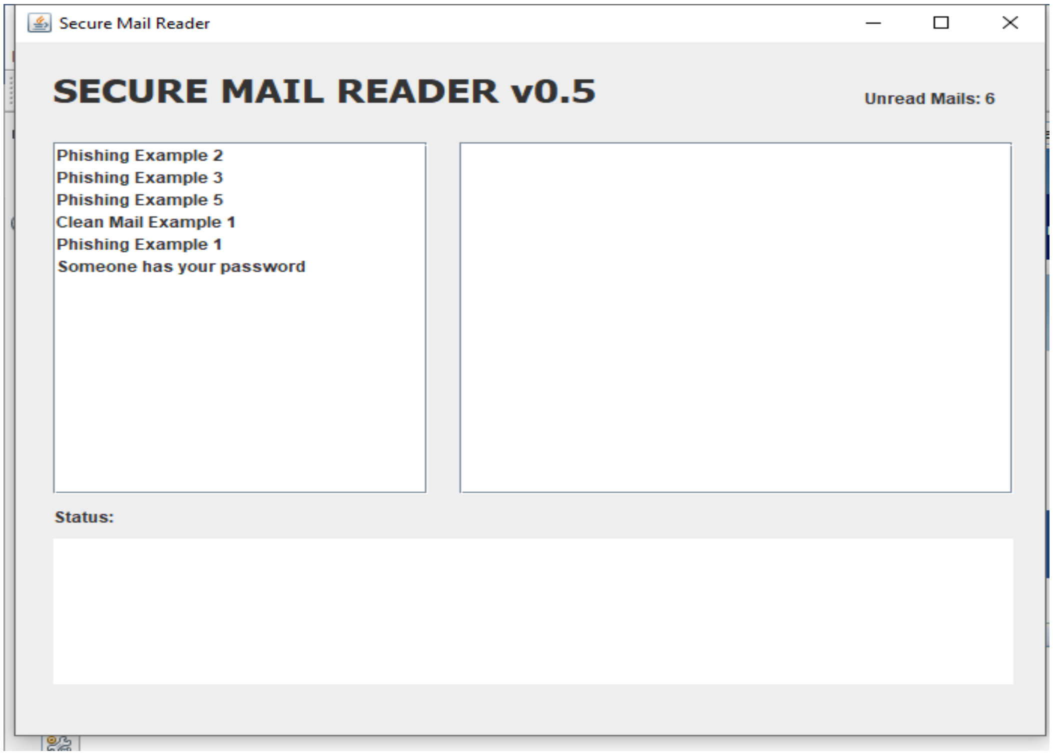
Text from mail body is extracted using API provided by Java. If it’s MIME Multipart only the first part taken into consideration currently, but we could check all. Gmail give other contents in other body parts so it doesn’t seem feasible in this project’s proof of concept implementation.

Using IMAP protocol and Java Mail API we just create a session, a store and connect using login credentials. IMAP is a protocol to retrieve mails from a mail server. User can login to the system by providing user name and password. As user provide a valid information he can enter his emails section. He can read unread email by clicking on them and we are using Gmail service for this system.



```
public void checkMails() {

    Properties props = new Properties();
    props.put("mail.imap.host", "imap.gmail.com");
    props.put("mail.imap.port", "993");
    props.put("mail.imap.socketFactory.class", "javax.net.ssl.SSLSocketFactory");
    props.put("mail.imap.socketFactory.fallback", "false");
    props.put("mail.imap.socketFactory.port", "993");

    Session recvSession = Session.getInstance(props);

    try {
        if (inbox != null && inbox.isOpen()) {
            inbox.close(true);
        }
        if (mailStore != null && mailStore.isConnected()) {
            mailStore.close();
        }
    } catch (MessagingException e) {

    }

    try {
        mailStore = (IMAPStore) recvSession.getStore("imap");
        mailStore.connect(username, password);
        inbox = mailStore.getFolder("Inbox");
        inbox.open(Folder.READ_ONLY);

        messages = inbox.search(new FlagTerm(new Flags(Flag.SEEN), false));
    } catch (MessagingException e) {
        e.printStackTrace();
    }
}
```

Figure 15: Log into mail



SECURE MAIL READER v0.5

Unread Mails: 6

```
Phishing Example 2
Phishing Example 3
Phishing Example 5
Clean Mail Example 1
Phishing Example 1
Someone has your password
```

Status:

Figure 16: Secure mail reader

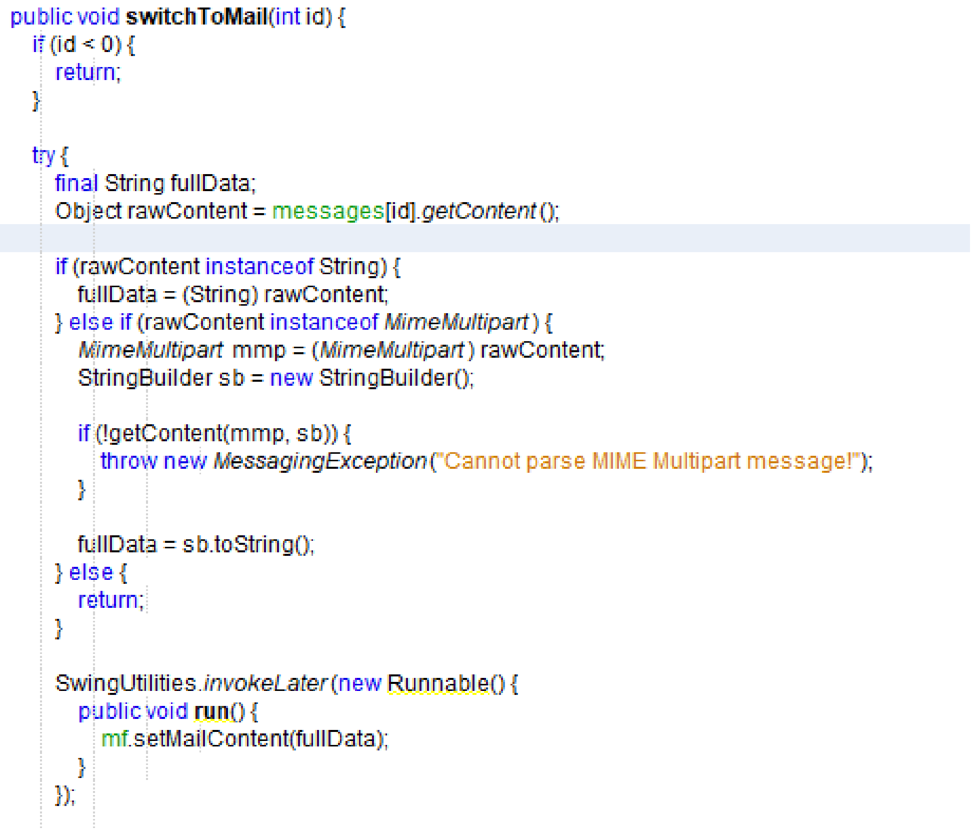
When the user logins with the correct credentials, the output shows the unread mails count and the mails.

Pseudocode:

```
1. Properties required for GMAIL IMAP connection using Java Mail API
2. Opening a mail transfer session
3. If there is already an object available then it is closed using imp to access mails
4. Request for connect using username and password
5. Retrieve contents of Inbox folder
6. Emails are open in read only mode as we are not changing anything
7. Emails are filtered so we can only see unread emails.
8. Extraction of text data from MIME multipart type mails.
```

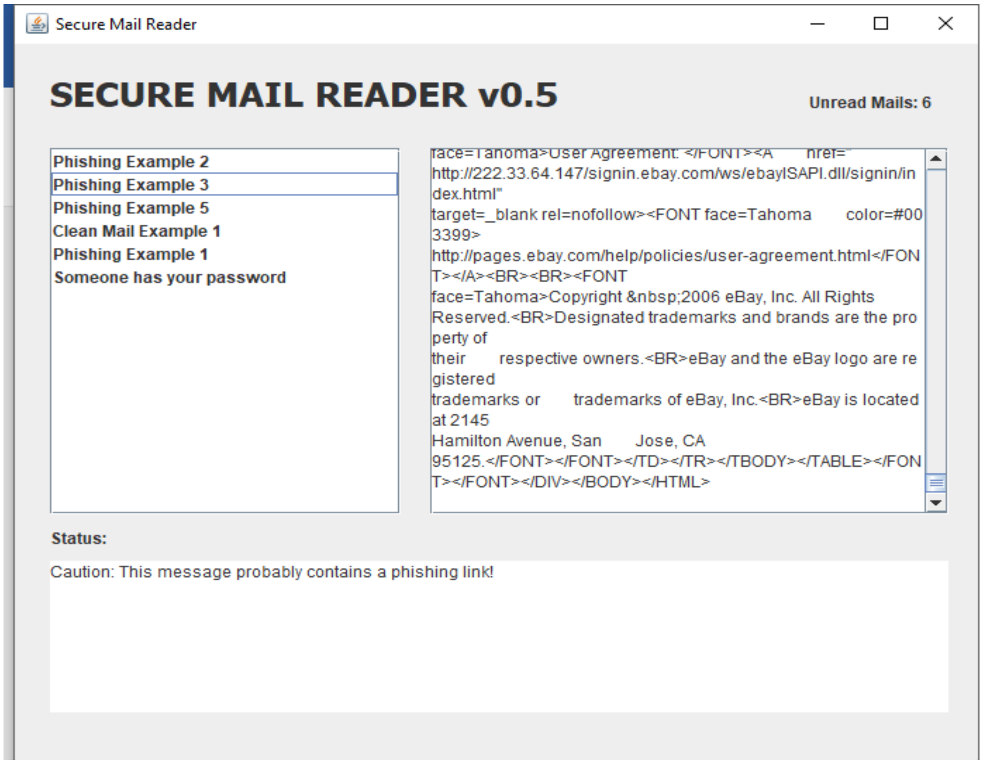
5.6.2 Extracting unread email

All the UN read emails are extracted from inbox. When user clicks on any emails its contents are retrieved and html and header tags are separated from the content of email with the help of filter approach.



```java
public void switchToMail(int id) {
    if (id < 0) {
        return;
    }

    try {
        final String fullData;
        Object rawContent = messages[id].getContent();

        if (rawContent instanceof String) {
            fullData = (String) rawContent;
        } else if (rawContent instanceof MimeMultipart) {
            MimeMultipart mmp = (MimeMultipart) rawContent;
            StringBuilder sb = new StringBuilder();

            if (!getContent(mmp, sb)) {
                throw new MessagingException("Cannot parse MIME Multipart message!");
            }

            fullData = sb.toString();
        } else {
            return;
        }

        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                mf.setMailContent(fullData);
            }
        });
```

Figure 17: extracting unread email



```
face=Tahoma>User Agreement: </FONT><A      href="
http://222.33.64.147/signin.ebay.com/ws/ebayISAPI.dll/signin/in
dex.html"
target=_blank rel=nofollow><FONT face=Tahoma      color=#00
3399>
http://pages.ebay.com/help/policies/user-agreement.html</FON
T></A><BR><BR><FONT
face=Tahoma>Copyright &nbsp;2006 eBay, Inc. All Rights
Reserved.<BR>Designated trademarks and brands are the pro
perty of
their      respective owners.<BR>eBay and the eBay logo are re
gistered
trademarks or      trademarks of eBay, Inc.<BR>eBay is located
at 2145
Hamilton Avenue, San      Jose, CA
95125.</FONT></FONT></TD></TR></TBODY></TABLE></FON
T></FONT></DIV></BODY></HTML>
```

```
Caution: This message probably contains a phishing link!
```

Figure 18: Email extraction

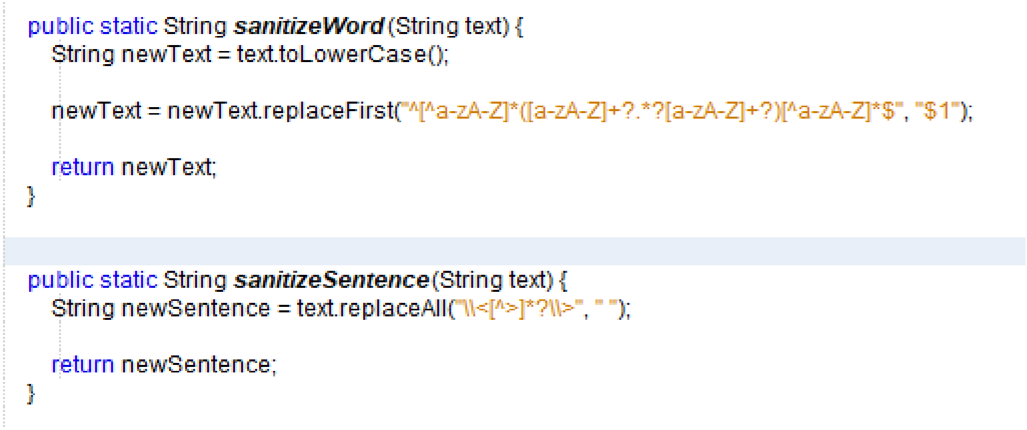
All the mail contents are shown in the text area as it can be seen in the output.

Pseudo Code

1- Id is provided to function
2- If id is less than 0 then return, so we can detect when an element of listbox is not selected.
3- Email content is retrieved against that id
4- If content contain text then all good
5- If content contain multipart
6- Text is extracted from multipart and returned.

5.7 Pre-processing

The data that is extracted from the email is pre-processed. The preprocessing is same as the pre-processing of the dataset which includes mail headers, non-utf8 characters, punctuation and numbers in both sides of words and all HTML tags. We call this approach sanitization.



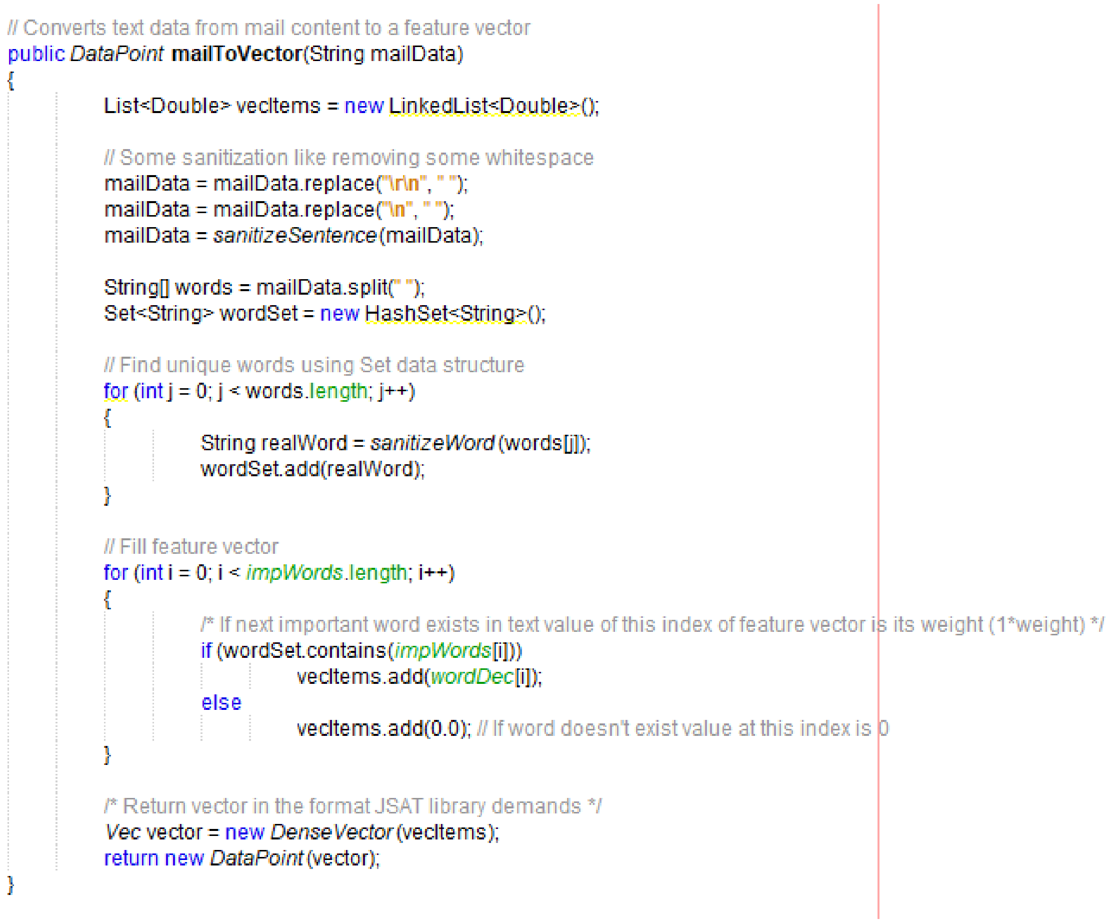
```
public static String sanitizeWord(String text) {
    String newText = text.toLowerCase();

    newText = newText.replaceFirst("^[^a-zA-Z]*([a-zA-Z]+?.*?[a-zA-Z]+?)[^a-zA-Z]*$", "$1");

    return newText;
}


public static String sanitizeSentence(String text) {
    String newSentence = text.replaceAll("\\<[^>]*?\\>", " ");

    return newSentence;
}
```

Figure 19: Pre-processing

5.8 Creation of feature vector for email

Feature vector is created using the mail data after replacing unnecessary data from the content and feature vector format is kept according to the standard of JSAT library.

JSAT holds feature vector in Data Point class.



```
// Converts text data from mail content to a feature vector
public DataPoint mailToVector(String mailData)
{
        List<Double> vecItems = new LinkedList<Double>();

        // Some sanitization like removing some whitespace
        mailData = mailData.replace("\r\n", " ");
        mailData = mailData.replace("\n", " ");
        mailData = sanitizeSentence(mailData);

        String[] words = mailData.split(" ");
        Set<String> wordSet = new HashSet<String>();

        // Find unique words using Set data structure
        for (int j = 0; j < words.length; j++)
        {
                String realWord = sanitizeWord(words[j]);
                wordSet.add(realWord);
        }

        // Fill feature vector
        for (int i = 0; i < impWords.length; i++)
        {
                /* If next important word exists in text value of this index of feature vector is its weight (1*weight) */
                if (wordSet.contains(impWords[i]))
                        vecItems.add(wordDec[i]);
                else
                        vecItems.add(0.0); // If word doesn't exist value at this index is 0
        }

        /* Return vector in the format JSAT library demands */
        Vec vector = new DenseVector(vecItems);
        return new DataPoint(vector);
}
```

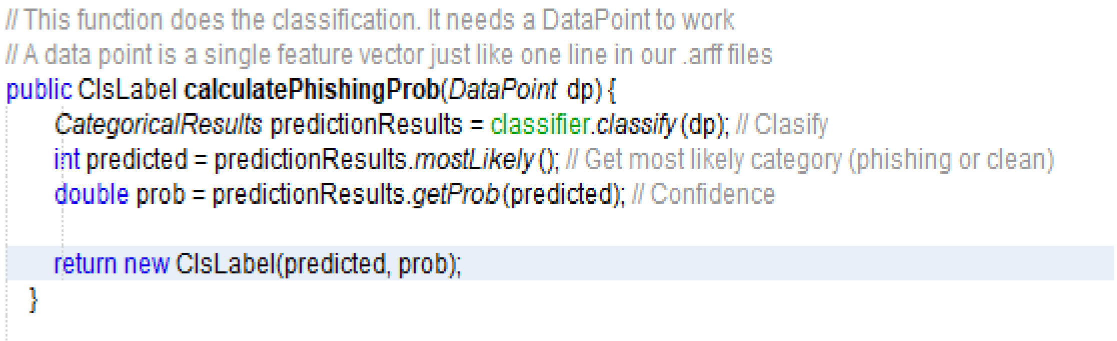
Figure 20: Feature vector creation

Pseudo code

1- Mail data is given to function
2- Feature vector is filled
3- If important word exists its value is 1*weight
4- If word doesn't exist its value is set to 0
5- As feature vector is formed according to JSAT library standard, then it is returned

5.9 Classify mail

Based on contents final decision made either email is phishing or legitimate. The contents of email after conversion into numerical data are passes through weighted structure of LVQ. Then distance is calculated through vector code book. Now the data is classifying through classification algorithm with help of LVQ.



```
// This function does the classification. It needs a DataPoint to work
// A data point is a single feature vector just like one line in our .arff files
public ClsLabel calculatePhishingProb(DataPoint dp) {
    CategoricalResults predictionResults = classifier.classify(dp); // Clasify
    int predicted = predictionResults.mostLikely(); // Get most likely category (phishing or clean)
    double prob = predictionResults.getProb(predicted); // Confidence

    return new ClsLabel(predicted, prob);
  }
```

Figure 21: email classification

Pseudo Code

1- Datapoint is provided as parameter to the function

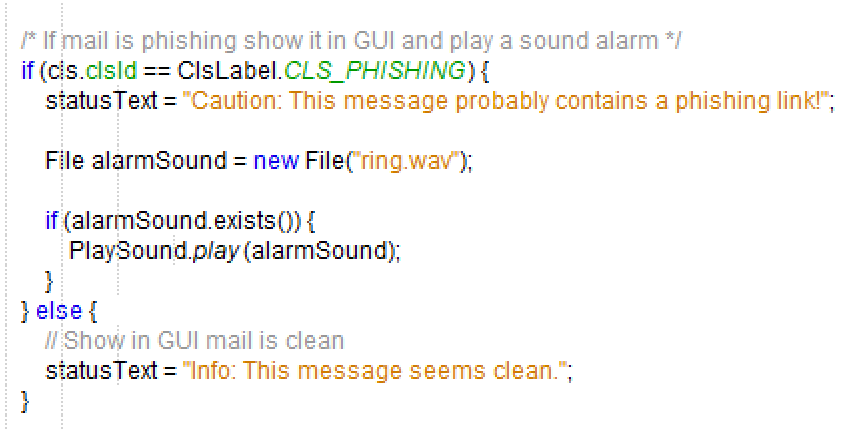
2- Single line feature vector from ARFF file is given to classifier for classification

3- Classifier returns prediction and confidence

4- New class label is returned with predicted and prob value.

5.10 Alert the user

If the classification algorithm in the phishing detection system detect that the mail is phishing email then the message is displayed in the GUI and the sound is played



```
/* If mail is phishing show it in GUI and play a sound alarm */
if (cls.clsId == ClsLabel.CLS_PHISHING) {
    statusText = "Caution: This message probably contains a phishing link!";

    File alarmSound = new File("ring.wav");

    if (alarmSound.exists()) {
        PlaySound.play(alarmSound);
    }
} else {
    // Show in GUI mail is clean
    statusText = "Info: This message seems clean.";
}
```

Figure 22: alert the user

Pseudo Code

```
1 – IF clasID – cls phishing then “Mail contains phishing link”
        Sound Alarm
2-  Else “email is clean”
```

5.2 Source code implementation

The source code of the Email Phishing application was implemented using as a guide the initial flowcharts, which was constructed throughout the design phase of the project, which the full source code listing is found in the appendices at the end of the report. Also, there are comments which were used for clarifying what it is achieved in each section of the source code.

This is the directory structure of the program

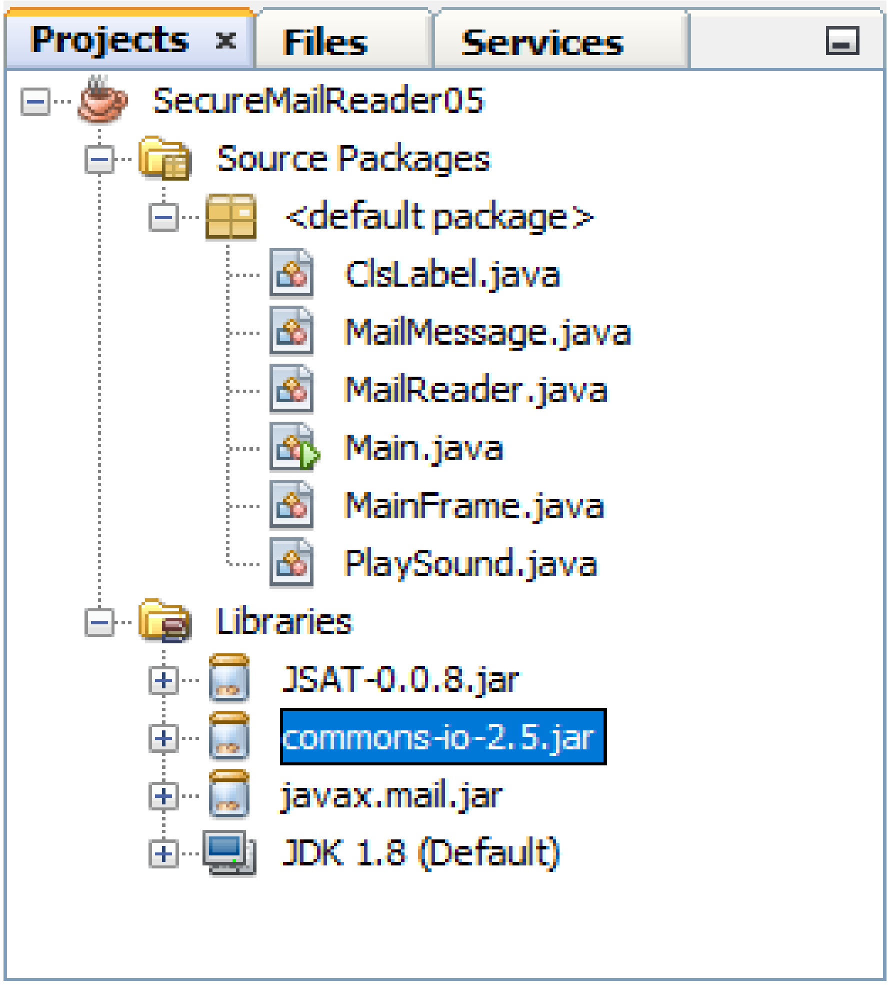


Figure 23: Source code implementation

5.2.1 Main

Within the Main.java file is the source code of the application's Main GUI. This is the main part of the code which is executed when the application is started. Creates objects of Mainframe and Mail Reader classes and pass them references to each other. Contains main method of the application. Code of this class runs in the main thread and it's used to check periodically for mails and any messages passed to Mail Reader class by Mainframe class (from GUI thread). Thread synchronization is done by employing a synchronized block and an object instance which works like a classic mutex.

5.2.2 Mail Readers

Within the MailReader.java file is the source code of the application's Mail reader. This class is main building block of whole application. Using functions provided by JSAT library, this class loads training data, creates LVQ based classifier and trains machine learning model using the data loaded. This class also has same feature extraction logic as our other tools. It has features and weight of each word hardcoded into it. It also has functions for stripping unnecessary data from mail content and word sanitization algorithm used by other tools and uses this function along with mail To Vector () function to build feature vector for selected e-mail message. When a mail is read, it uses this feature vector to classify mail. This class also has a messaging system to communicate with GUI thread to handle certain events. It uses GMAIL's IMAP server and Java Mail API's IMAP implementation to fetch e-mail messages from the mail account (in checkMails () function). When checking for unread mails it first closes if handles were previous used. Then gets a session, then an IMAPStore object from Java Mail API. After this it uses hardcoded account information to access mailbox. Then contents of the INBOX folder are fetched using filter to show only unread mails. When reading mail content, our program checks to see if it's MIME Multipart message or plain text. Text from MIME Multipart data are extracted using getContent () function. switchToMail () function is triggered by GUI thread when a mail subject is clicked. It shows mail content and status bar to tell user whether the mail is of phishing or HAM class. If phishing user is alarmed with a sound. This class also includes loop () function used by Main class (and main thread) to check mails periodically.

5.2.3 Main Frame

Within the MainFrame.java file is the source code of the application’s Mail reader. A sub-class of JFrame which includes all GUI logic the program has. It has functions for showing unread mail count, showing unread mails, showing mail content, showing whether the mail is phishing mail or not. It also handles events such as when a mail is selected. Consists of a listbox, two textareas, three labels and two scrollbar panes.

6 Testing and results

6.0 Testing the system

Testing is performed to check the working of the phishing detection system. All the functionalities of the system are checked

Test Case of system

ID	Description	Expected Output	Actual Output
1	The user can see the unread mails if the credentials are correct	Sign-up successful	Sign-up successful
2	User selects the mail, Contents are displayed	Emails seen in the textbox	Emails seen in the textbox
3	User doesn't submit same password in reconfirm password field	Mails are not seen	Mails are not seen
4	Phishing email detected	Alert sound	Alert Sound

Table 3: Test Case of system

Following the test of the system the testing is performed to check the efficiency of the LVQ in comparison to the other classification algorithm.

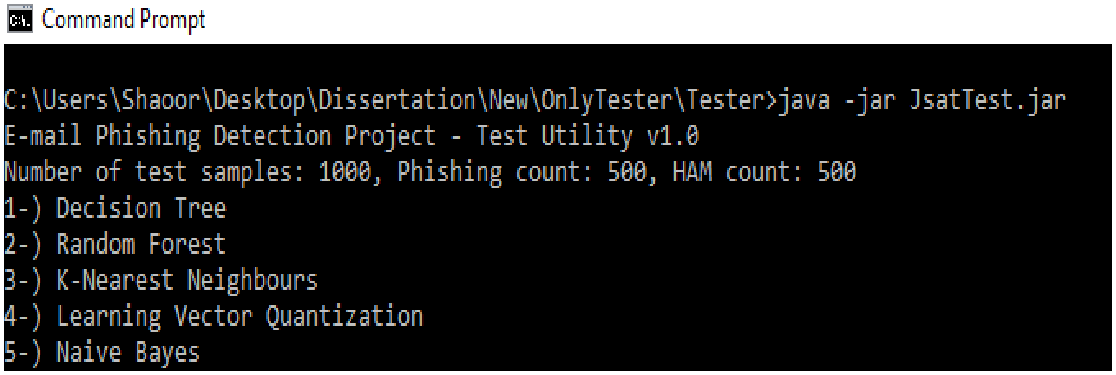
6.1 Dataset Explanation

The dataset for testing of the project was created in the same way the dataset was done for training the model. The original datasets used for creating the test dataset are Phishing Corpus and CSDMC2010 Ham mail datasets. From each dataset mentioned this project have selected 500 mails from each category for testing this dataset. This dataset is different from training dataset as different mails are selected from the original dataset.

In the testing process, this dataset is used to check the performance of our model. There are various evaluation measures that will be used to check the accuracy of the algorithm. In addition to that, getting the results of other classification algorithm and then comparing the results of it with our proposed framework of LVQ.

6.2 Experiment Methodology

In this project, utility program JsatTest (PhishingTestTool) was used to conduct tests using various machine learning classification algorithms based on custom training & test data.



```
C:\Users\Shaoor\Desktop\Dissertation\New\OnlyTester\Tester>java -jar JsatTest.jar
E-mail Phishing Detection Project - Test Utility v1.0
Number of test samples: 1000, Phishing count: 500, HAM count: 500
1-) Decision Tree
2-) Random Forest
3-) K-Nearest Neighbours
4-) Learning Vector Quantization
5-) Naive Bayes
```

Figure 24: testing process

As it can be seen in the figure above, the first line of the output shows the number of samples in dataset, which is 1000 as discussed above in dataset description, 500 samples are phishing and other half are clean samples. Then the program asks to input the number to select the algorithm, which then will give the output of the selected algorithm.

6.2.1 Working methodology

I. This program uses JSAT java library

II. Loads both the training data and test data and create objects for both datasets.

III. Outputs total number of test samples.

IV. Switch case to ask the user to select the algorithm.

V. Trains the system with training dataset.

VI. Start the testing and give results.

6.3 Results

The following results are calculated using the same methodology as discussed above. In this project, testing of 5 algorithms are performed on same testing dataset created which are:

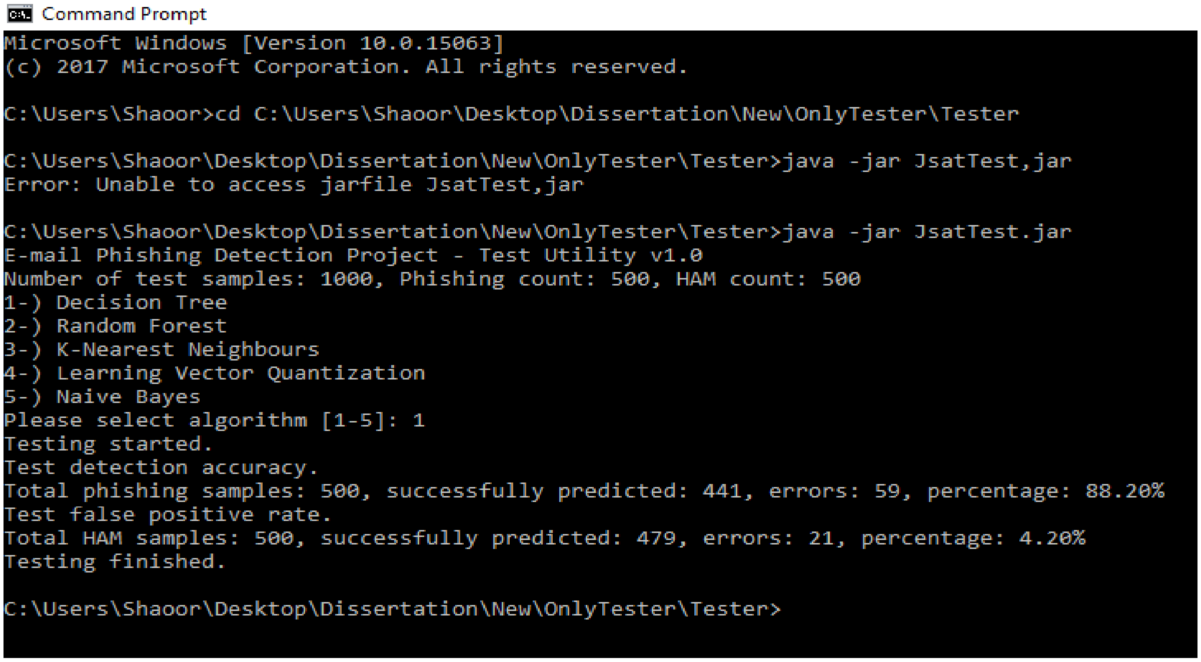
i. Decision tree

ii. Random forest

iii. K-nearest Neighbors

iv. Learning Vector Quantization

v. Naïve Bayes



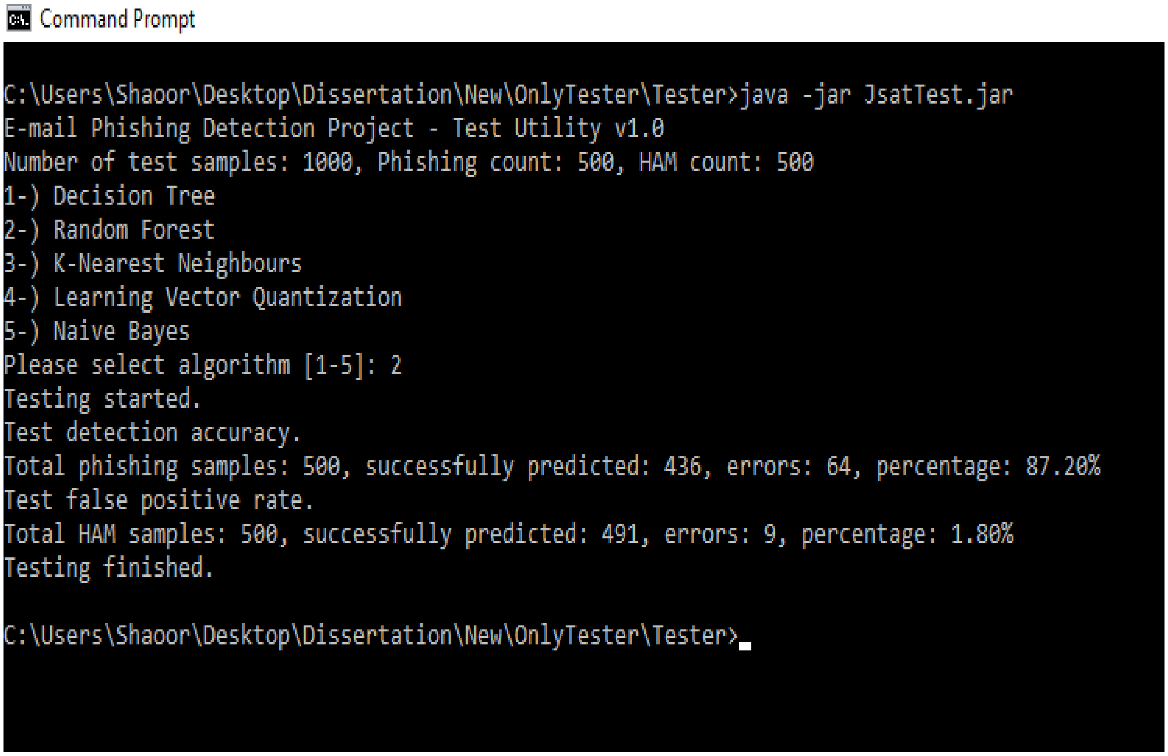
```
Microsoft Windows [Version 10.0.15063]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\Shaoor>cd C:\Users\Shaoor\Desktop\Dissertation\New\OnlyTester\Tester

C:\Users\Shaoor\Desktop\Dissertation\New\OnlyTester\Tester>java -jar JsatTest,jar
Error: Unable to access jarfile JsatTest,jar

C:\Users\Shaoor\Desktop\Dissertation\New\OnlyTester\Tester>java -jar JsatTest.jar
E-mail Phishing Detection Project - Test Utility v1.0
Number of test samples: 1000, Phishing count: 500, HAM count: 500
1-) Decision Tree
2-) Random Forest
3-) K-Nearest Neighbours
4-) Learning Vector Quantization
5-) Naive Bayes
Please select algorithm [1-5]: 1
Testing started.
Test detection accuracy.
Total phishing samples: 500, successfully predicted: 441, errors: 59, percentage: 88.20%
Test false positive rate.
Total HAM samples: 500, successfully predicted: 479, errors: 21, percentage: 4.20%
Testing finished.

C:\Users\Shaoor\Desktop\Dissertation\New\OnlyTester\Tester>
```

Figure 25: Decision tree

The results for Decision tree:

	Predicted Negative	Predicted Positive
Negative	479(TN)	21(FP)
Positive	59(FN)	441(TP)

Table 4: Decision tree results



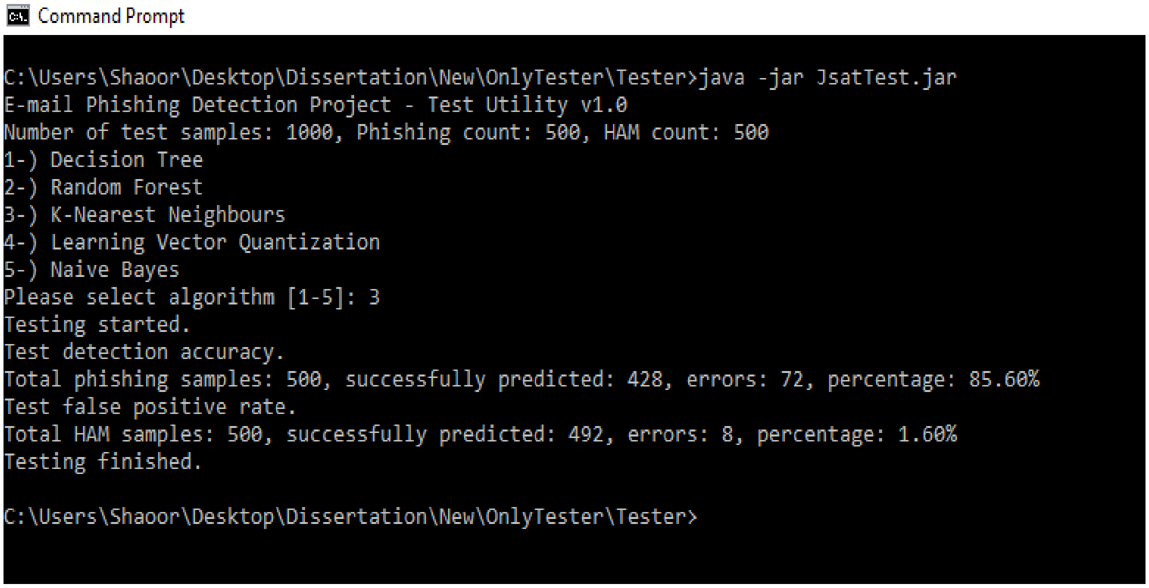
```
C:\Users\Shaoor\Desktop\Dissertation\New\OnlyTester\Tester>java -jar JsatTest.jar
E-mail Phishing Detection Project - Test Utility v1.0
Number of test samples: 1000, Phishing count: 500, HAM count: 500
1-) Decision Tree
2-) Random Forest
3-) K-Nearest Neighbours
4-) Learning Vector Quantization
5-) Naive Bayes
Please select algorithm [1-5]: 2
Testing started.
Test detection accuracy.
Total phishing samples: 500, successfully predicted: 436, errors: 64, percentage: 87.20%
Test false positive rate.
Total HAM samples: 500, successfully predicted: 491, errors: 9, percentage: 1.80%
Testing finished.

C:\Users\Shaoor\Desktop\Dissertation\New\OnlyTester\Tester>_
```

Table 5: random forest

The results of Random forest

	Predicted Negative	Predicted Positive
Negative	491(TN)	9(FP)
Positive	64(FN)	436(TP)

Table 6: random forest results



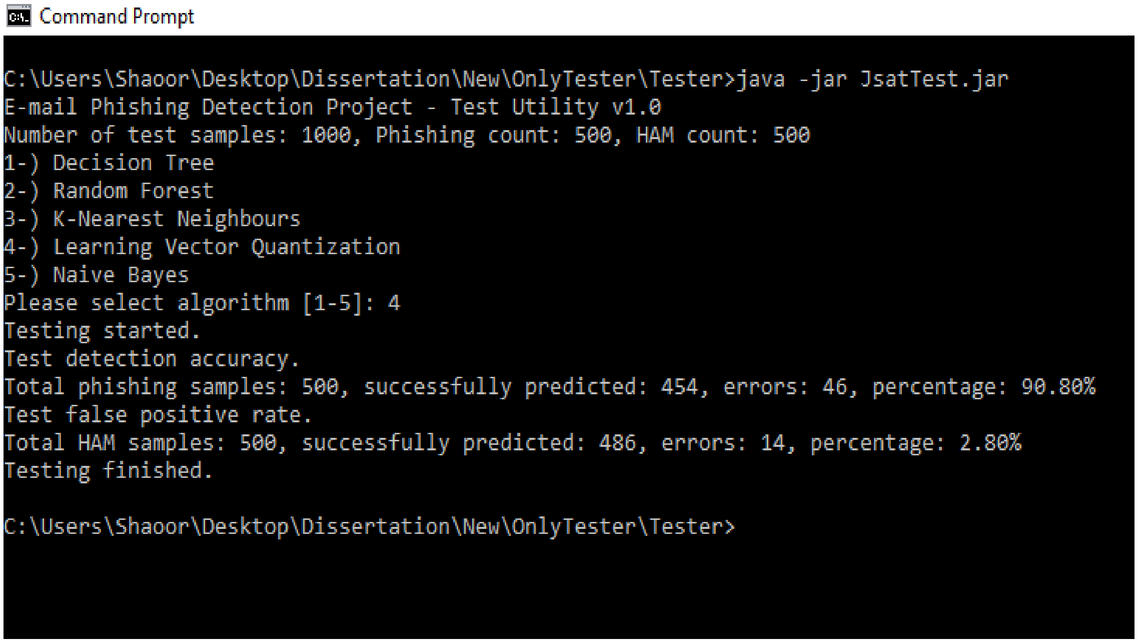
```
C:\Users\Shaoor\Desktop\Dissertation\New\OnlyTester\Tester>java -jar JsatTest.jar
E-mail Phishing Detection Project - Test Utility v1.0
Number of test samples: 1000, Phishing count: 500, HAM count: 500
1-) Decision Tree
2-) Random Forest
3-) K-Nearest Neighbours
4-) Learning Vector Quantization
5-) Naive Bayes
Please select algorithm [1-5]: 3
Testing started.
Test detection accuracy.
Total phishing samples: 500, successfully predicted: 428, errors: 72, percentage: 85.60%
Test false positive rate.
Total HAM samples: 500, successfully predicted: 492, errors: 8, percentage: 1.60%
Testing finished.

C:\Users\Shaoor\Desktop\Dissertation\New\OnlyTester\Tester>
```

Figure 26: K-nearest Neighbors

The result of K-nearest Neighbors

	Predicted Negative	Predicted Positive
Negative	492(TN)	8(FP)
Positive	72(FN)	428(TP)

Table 7: K-nearest Neighbors results



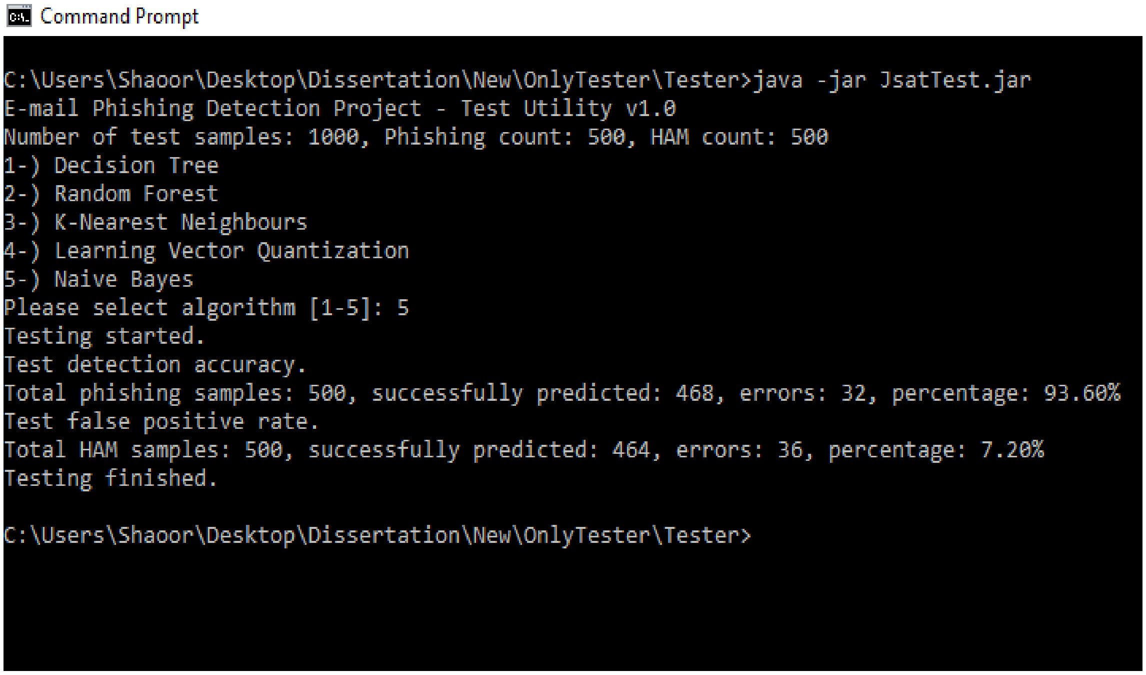
```
C:\Users\Shaoor\Desktop\Dissertation\New\OnlyTester\Tester>java -jar JsatTest.jar
E-mail Phishing Detection Project - Test Utility v1.0
Number of test samples: 1000, Phishing count: 500, HAM count: 500
1-) Decision Tree
2-) Random Forest
3-) K-Nearest Neighbours
4-) Learning Vector Quantization
5-) Naive Bayes
Please select algorithm [1-5]: 4
Testing started.
Test detection accuracy.
Total phishing samples: 500, successfully predicted: 454, errors: 46, percentage: 90.80%
Test false positive rate.
Total HAM samples: 500, successfully predicted: 486, errors: 14, percentage: 2.80%
Testing finished.

C:\Users\Shaoor\Desktop\Dissertation\New\OnlyTester\Tester>
```

Figure 27: result of LVQ

The result of LVQ

	Predicted Negative	Predicted Positive
Negative	486(TN)	14(FP)
Positive	46(FN)	454(TP)

Table 8: LVQ results



```
C:\Users\Shaoor\Desktop\Dissertation\New\OnlyTester\Tester>java -jar JsatTest.jar
E-mail Phishing Detection Project - Test Utility v1.0
Number of test samples: 1000, Phishing count: 500, HAM count: 500
1-) Decision Tree
2-) Random Forest
3-) K-Nearest Neighbours
4-) Learning Vector Quantization
5-) Naive Bayes
Please select algorithm [1-5]: 5
Testing started.
Test detection accuracy.
Total phishing samples: 500, successfully predicted: 468, errors: 32, percentage: 93.60%
Test false positive rate.
Total HAM samples: 500, successfully predicted: 464, errors: 36, percentage: 7.20%
Testing finished.

C:\Users\Shaoor\Desktop\Dissertation\New\OnlyTester\Tester>
```

Figure 28: Naïve Bayes

The result of Naïve Bayes:

	Predicted Negative	Predicted Positive
Negative	464(TN)	36(FP)
Positive	32(FN)	468(TP)

Table 9: Naïve Bayes results

6.4 Evaluation Measures

Evaluation measures is the way to evaluate the performance of the system. The measures are used to check how well the results satisfy the objective of the program. Some of the evaluation measure are mentioned below:

- True Positive (TP): The result is positive, and is predicted to be positive.
- False Negative (FN): The result is positive, but is predicted negative.
- True Negative (TN): The result is negative, and is predicted to be negative.
- False Positive (FP): The result is negative, but is predicted positive. Albon, (2017)

Example calculation of LVQ

In the similar way the calculation is done for all the other algorithms which can be seen in the figure table 2 of algorithm performance.

	Predicted Negative	Predicted Positive
Negative	486(TN)	14(FP)
Positive	46(FN)	454(TP)

Table 10: precision table

1. Precision is how many positive predictions are correct.

$$Tp/\ tp + fp$$

$$454/468\ =\ 0.97\ =\ 97\%$$

2. Recall is the percent of the positive prediction the algorithm catches.

$$Tp/tp + fn$$

$$454/500\ =\ 0.908\ =\ 90.8\%$$

3. Accuracy is proportion of all predictions that are correct. Accuracy checks how good a model is.

$$TP + TN/TP + FN + FP + TN =$$

$$correct\ predictions/all\ predictions$$

$$940/1000$$

$$94\%$$

4. False Positive Rate is the proportion of all negative observation that are predicted incorrectly. False positive rate checks how good a model is at predicting negative cases.

FPR=

$$FP/FP + TN =$$

$$FP/N = 14/500 = 2.8\%$$

5. F-measure.

The harmonic mean of precision and recall is known as f-measure. F1 score is an 'average' of both precision and recall. This project uses the harmonic mean because it is the appropriate way to average ratios.

$$F1 = 2 * Precision * Recall/Precision + Recall$$

$$2 * 97.0 * 90.8/97 + 90.8 = 0.938$$

Algorithm	Clean	Phishing	Detection Accuracy (%)	False Positives (%)	Precision	Recall	F-Measure
Decision Tree	479/500	441/500	92.0%	4.2%	95.4%	88.2%	0.917
Random Forest	491/500	436/500	92.7%	1.8%	97.9%	87.2%	0.922

K-NN	492/500	428/500	92.0%	1.6%	98.1%	85.6%	0.914
LVQ	486/500	454/500	94.0%	2.8%	97.0%	90.8%	0.938
Naive Bayes	464/500	468/500	93.2%	7.2%	92.8%	93.6%	0.932

Table 11: Algorithm performance

After testing the all the classification algorithm, algorithm performance table shows the results of all the model. The lowest accuracy was achieved by K- Nearest- Neighbors and Decision tree which is 92.0%, followed by Random Forest, and Naïve Bayes respectively (92.7% and 93.2%). The highest accuracy was achieved by LVQ which is 94.0%.

From the above table 10 LVQ gave the best result, Naive Bayes algorithm got 93.2% accuracy which is near to the LVQ which is 94%. But looking at the false positive rate the result of Naive Bayes is 7.2% which is not acceptable. Whereas the false positive rate of LVQ is less than Naive Bayes which is 2.8%.

According to f-measures, highest f-measure is of LVQ which is 0.938 hence LVQ performs the best among the algorithms this project has tried.

6.5 Discussion

After the complete training and testing process the proposed algorithm is evaluated and it produced satisfactory results. The produced results then mapped against already proposed papers in research. The two techniques including one is “Detecting Phishing Websites through Deep Reinforcement Learning” (Chatterjee et. al. 2019) and other is “Phishing URL Detection Through Top-level Domain Analysis: A Descriptive Approach” (Christou, O., et. al. 2020) were used for the comparison of results.

Comparison	Precision	Recall	F-Measure	Detection Accuracy (%)
Phishing Detection through Learning Vector Quantization	97.0%	90.8%	0.938	94.0%
Phishing URL Detection Through Top-level Domain Analysis: A Descriptive Approach (Random Forest)	85.0%	87.0%	0.859	89.2%
Phishing URL Detection Through Top-level Domain Analysis: A Descriptive Approach (SVM)	90%	88%	0.889	91.0%
Detecting Phishing through Deep Reinforcement Learning	86.7%	88.0%	0.873	90.1%

Table 12: comparison table

The proposed model is based on learning vector quantization for phishing detection and the compared models are based on SVM, Random Forest and deep reinforcement learning (Christou, O., et. al. 2020). In comparison to these approaches our proposed approach shows higher values in all four parameters of comparison including precision, recall, F-measure and detection accuracy which is shown in the figure below:

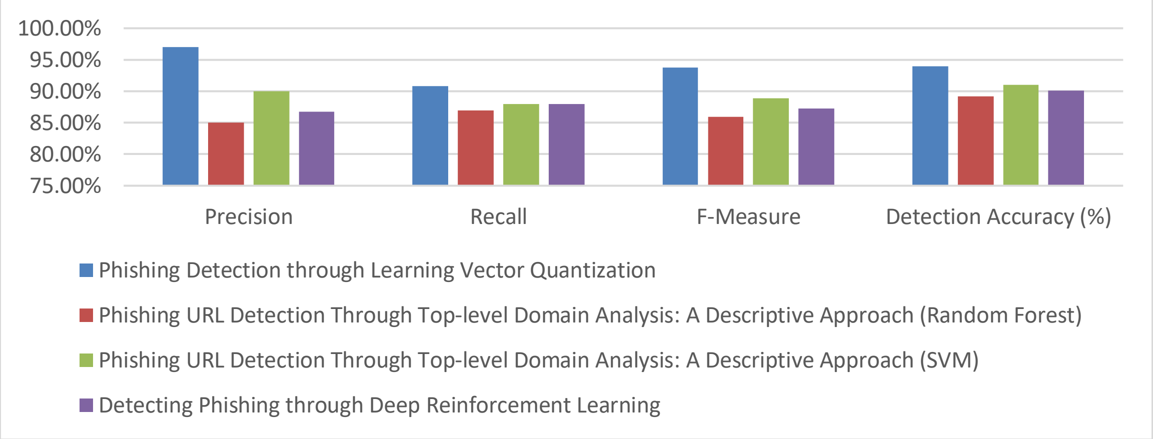


Figure 29: Graph of comparison

Through this comparison illustration it is evaluated that the proposed results shows a significant amount of progress in the detection of phishing emails and can be used to provide a comprehensive amount of defense against phishing email.

7 Project evaluation

7.1 Conclusion

As we have seen that the phishing attacks are becoming more popular, the project was an attempt to detect phishing emails by using new learning technique known as learning vector quantization. To achieve this, vast literature study was conducted to gain insights of how this type of techniques work. Analysis was done on various phishing detection system and how are they deployed in the real-life situation. The main purpose of the project was achieved which was to develop a mail reader application which will read all the mails before going to the inbox and when user select the emails, emails will be retrieved and the mail client checks for the phishing email. If phishing email found the user will be notified using the alarm sound.

The proposed phishing detection system was designed based on the working of the algorithm and the actual interface was developed using the design diagrams. Dataset used in the project for training the classifier was developed using different original datasets. Training model was developed using the dataset. In similar way dataset was also made for testing the LVQ with other algorithms which is known as testing dataset. The detection system checks the email service and when user selects the mail the LVQ classifier classifies the mail if it is phishing or not and then the user is also alerted as soon as phishing email is detected. The scope of the project is very well achieved. The phishing detection system was developed successfully, the classifier used in the project was tested with other classification algorithm using the same dataset for all. The result proved that the LVQ achieved highest accuracy and best f-measure.

We have a limitation in our system and it is with regards to very short e-mails because the text will be very less in small emails. Same with mails containing images instead of plain text and mails containing less text. Such emails will not be effective with our proposed detection system.

7.2 Future Work

Further improving the system, a more diverse dataset should be considered to make it more efficient. Adding more features to the algorithm can improve the accuracy. Also adding the feature responses from anti-virus services with their own calculated should be considered. This project didn't think word frequency did matter when detecting whether a mail is phishing. But this project could also go beyond the word presence and add features like finding <a> tags with attribute being different than its shown text, checking URLs for IP-based domains, checking age of domains, checking whether URL encoding is used or there is a link with double URLs. Finding the roots can also increase accuracy. Further improvements can also be done to trace the source of the phishing email sender. Overall the focus of the improvement should be to increase the rate of detection and reduce the false positive.

References

Islam, R. and Abawajy, J., 2013. Multi-tier phishing detection and filtering approach. Journal of Network and Computer Applications, 36(1), pp.324-335.

Pandey, M. and Ravi, V., 2012, December. Detecting phishing e-mails using text and data mining. In Computational Intelligence & Computing Research (ICCIC), 2012 IEEE International Conference on (pp. 1-6). IEEE.

Abu-Nimeh, S., Nappa, D., Wang, X., & Nair, S. (2007, October). A comparison of machine learning techniques for phishing detection. In Proceedings of the anti-phishing working groups 2nd annual eCrime researchers summit (pp. 60-69). ACM.

Chandrasekaran, M., Narayanan, K., & Upadhyaya, S. (2006, June). Phishing email detection based on structural properties. In NYS cyber security conference (Vol. 3).

Chatterjee, M., & Namin, A. S. (2019, July). Detecting phishing websites through deep reinforcement learning. In 2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC) (Vol. 2, pp. 227-232). IEEE.

Christou, O., Pitropakis, N., Papadopoulos, P., McKeown, S., & Buchanan, W. J. (2020). Phishing URL Detection Through Top-level Domain Analysis: A Descriptive Approach. arXiv preprint arXiv:2005.06599.

Form, L.M., Chiew, K.L. and Tiong, W.K., 2015, August. Phishing email detection technique by using hybrid features. In IT in Asia (CITA), 2015 9th International Conference on (pp. 1-5). IEEE.

Smadi, S., Aslam, N., Zhang, L., Alasem, R. and Hossain, M.A., 2015, December. Detection of phishing emails using data mining algorithms. In Software, Knowledge, Information Management and Applications (SKIMA), 2015 9th International Conference on (pp. 1-8). IEEE

Sanchez, F. and Duan, Z., 2012, December. A sender-centric approach to detecting phishing emails. In Cyber Security (CyberSecurity), 2012 International Conference on (pp. 32-39). IEEE.

Hamid, I.R.A. and Abawajy, J.H., 2013, July. Profiling phishing email based on clustering approach. In Trust, Security and Privacy in Computing and Communications (TrustCom), 2013 12th IEEE International Conference on (pp. 628-635). IEEE.

Fang, X., Koceja, N., Zhan, J., Dozier, G. and Dipankar, D., 2012, June. An artificial immune system for phishing detection. In Evolutionary Computation (CEC), 2012 IEEE Congress on (pp. 1-7). IEEE.

Hajgude, J. and Ragha, L., 2012, October. Phish mail guard: Phishing mail detection technique by using textual and URL analysis. In Information and Communication Technologies (WICT), 2012 World Congress on (pp. 297-302). IEEE.

Volkamer, M., Renaud, K., Reinheimer, B. and Kunz, A., 2017. User experiences of TORPEDO: TOoltip-poweRed Phishing Email DetectiOn. Computers & Security.

Aburrous, M., Hossain, M.A., Dahal, K. and Thabtah, F., 2010. Intelligent phishing detection system for e-banking using fuzzy data mining. Expert systems with applications, 37(12), pp.7913-7921.

Abdelhamid, N., Ayesh, A. and Thabtah, F., 2014. Phishing detection based associative classification data mining. Expert Systems with Applications, 41(13), pp.5948-5959.

Shah, R., Trevathan, J., Read, W. and Ghodosi, H., 2009, April. A proactive approach to preventing phishing attacks using Pshark. In Information Technology: New Generations, 2009. ITNG'09. Sixth International Conference on (pp. 915-921). IEEE.

Kumar, Vimal, and Rakesh Kumar. "Detection of phishing attack using visual cryptography in ad hoc network." Communications and Signal Processing (ICCSP), 2015 International Conference on. IEEE, 2015.

Du, Y., Xue, F. 2013. Research of the Anti-phishing Technology Based on E-mail Extraction and Analysis. 2013 International Conference on Information Science and Cloud Computing Companion.

Chandrasekaran, M., Narayanan, K., & Upadhyaya, S. (2006, June). Phishing email detection based on structural properties. In NYS cyber security conference (Vol. 3).

Tak, G.K. and Ojha, G., 2013. Multi-Level Parsing Based Approach Against Phishing Attacks With The Help Of Knowledge Bases. International Journal of Network Security & Its Applications, 5(6), p.15.

Inomata, A., Rahman, M., Okamoto, T. and Okamoto, E., 2005, August. A novel mail filtering method against phishing. In Communications, Computers and signal Processing, 2005. PACRIM. 2005 IEEE Pacific Rim Conference on (pp. 221-224). IEEE.

Kumar, B., Kumar, P., Mundra, A. and Kabra, S. 2015. DC scanner: Detecting phishing attack. 2015 Third International Conference on Image Information Processing (ICIIP).

Kjonji, M., Iraqi, Y., Jones, A. 2011. Lexical URL analysis for discriminating phishing and legitimate e-mail messages. 2011 International Conference for Internet Technology and Secured Transaction

Qabajeh, I. and Thabtah, F. 2014. An Experimental Study for Assessing Email Classification Attributes Using Feature Selection Methods. 2014 3rd International Conference on Advanced Computer Science Applications and Technologies.

Azeez, N. and Oluwatosin, A. 2016. CyberProtector: Identifying Compromised URLs in Electronic Mails with Bayesian Classification. 2016 International Conference on Computational Science and Computational Intelligence (CSCI).

Bozkir, A. and Sezer, E. 2016. Use of HOG descriptors in phishing detection. 2016 4th International Symposium on Digital Forensic and Security (ISDFS).

Gansterer, W. and Pölz, D. 2009. E-Mail Classification for Phishing Defense. Lecture Notes in Computer Science, pp.449-460.

Narayanan, Chandrasekaran, M., K. and Upadhyaya, S. 2006. Phishing email detection based on structural properties. In Proceedings of the NYS Cyber Security Conference

Abdelhamid, N., Thabtah, F. and Abdel-jaber, H., 2017. July. Phishing detection: A recent intelligent machine learning comparison based on models content and features. In Intelligence and Security Informatics (ISI), 2017 IEEE International Conference on (pp. 72-77). IEEE.

Basnet, Ram B., Srinivas Mukkamala, and Andrew H. Sung, 2008. "Detection of Phishing Attacks: A Machine Learning Approach." Soft Computing Applications in Industry 226 (2008): 373-383.

Nguyen, L.A.T., To, B.L., Nguyen, H.K. and Nguyen, M.H., 2014. October. An efficient approach for phishing detection using single-layer neural network. In Advanced Technologies for Communications (ATC), 2014 International Conference on (pp. 435-440). IEEE.

Kaui, S. and Kaur, A., 2015, October. Detection of phishing webpages using weights computed through genetic algorithm. In MOOCs, Innovation and Technology in Education (MITE), 2015 IEEE 3rd International Conference on (pp. 331-336). IEEE.

Che, H., Liu, Q., Zou, L., Yang, H., Zhou, D. and Yu, F., 2017, July. A Content-Based Phishing Email Detection Method. In Software Quality, Reliability and Security Companion (QRS-C), 2017 IEEE International Conference on (pp. 415-422). IEEE.

Ali, Mohd Mahmood, et al. 2015. "An approach for deceptive phishing detection and prevention in social networking sites using data mining and wordnet ontology." Electrical, Electronics, Signals, Communication and Optimization (EESCO), 2015 International Conference on. IEEE, 2015.

Ahmed, F. and Abulaish, M., 2012, June. An mcl-based approach for spam profile detection in online social networks. In Trust, Security and Privacy in Computing and Communications (TrustCom), 2012 IEEE 11th International Conference on (pp. 602-608). IEEE.

Monkey.org. (2017). Cite a Website - Cite This For Me. [online] Available at: https://monkey.org/~jose/ [Accessed 7 Sep. 2017].

Csmining.org. (2017). Spam email datasets * - Csmining Group. [online] Available at: http://csmining.org/index.php/spam-email-datasets-.html [Accessed 7 Sep. 2017].

99webtools.com. (2017). List of English Stop words | Web Tools. [online] Available at: http://99webtools.com/blog/list-of-english-stop-words/ [Accessed 7 Sep. 2017].

Gupta, S., Singhal, A. and Kapoor, A., 2016, April. A literature survey on social engineering attacks: Phishing attack. In Computing, Communication and Automation (ICCCA), 2016 International Conference on (pp. 537-540). IEEE.

Al-Shboul, B.A., Hakh, H., Faris, H., Aljarah, I. and Alsawalqah, H., 2016. Voting-based classification for e-mail spam detection. Journal of ICT Research and Applications, 10(1), pp.29-42.

Litan, A., 2004. Phishing attack victims likely targets for identity theft.

Kirda, Engin, and Christopher Kruegel. "Protecting users against phishing attacks with antiphish." Computer Software and Applications Conference, 2005. COMPSAC 2005. 29th Annual International. Vol. 1. IEEE, 2005.

Xiang, G., Hong, J., Rose, C.P. and Cranor, L., 2011. Cantina+: A feature-rich machine learning framework for detecting phishing web sites. ACM Transactions on Information and System Security (TISSEC), 14(2), p.21.

Zhuang, W., Jiang, Q. and Xiong, T., 2012, June. An intelligent anti-phishing strategy model for phishing website detection. In Distributed Computing Systems Workshops (ICDCSW), 2012 32nd International Conference on (pp. 51-56). IEEE.

Saberi, A., Vahidi, M. and Bidgoli, B.M., 2007, November. Learn to detect phishing scams using learning and ensemble? methods. In Proceedings of the 2007 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology-Workshops (pp. 311-314). IEEE Computer Society.

Hewamadduma, Shammi Ishara. "Detection and prevention of possible unauthorized login attempts through stolen credentials from a phishing attack in an online banking system." Research and Innovation in Information Systems (ICRIIS), 2017 International Conference on. IEEE, 2017.

Tian, Y., Yuan, J. and Yu, S., 2016, October. SBPA: Social behavior based cross Social Network phishing attacks. In Communications and Network Security (CNS), 2016 IEEE Conference on (pp. 366-367). IEEE.

Ren, Q., Mu, Y. and Susilo, W., 2007, July. SEFAP: An Email System for Anti-Phishing. In Computer and Information Science, 2007. ICIS 2007. 6th IEEE/ACIS International Conference on (pp. 782-787). IEEE.

Shaikh, A.N., Shabut, A.M. and Hossain, M.A., 2016, December. A literature review on phishing crime, prevention review and investigation of gaps. In Software, Knowledge, Information Management & Applications (SKIMA), 2016 10th International Conference on (pp. 9-15). IEEE.

Gupta, B.B., Tewari, A., Jain, A.K. and Agrawal, D.P., 2016. Fighting against phishing attacks: state of the art and future challenges. Neural Computing and Applications, pp.1-26.

Kirda, E. and Kruegel, C., 2005, July. Protecting users against phishing attacks with antiphish. In Computer Software and Applications Conference, 2005. COMPSAC 2005. 29th Annual International (Vol. 1, pp. 517-524). IEEE.

Banday, M.T. and Qadri, J.A., 2011. Phishing-A growing threat to e-commerce. arXiv preprint arXiv:1112.5732.

Stajano, F. and Wilson, P., 2011. Understanding scam victims: seven principles for systems security. Communications of the ACM, 54(3), pp.70-75.
Jakobsson, M., Johnson, N. and Finn, P., 2008. Why and how to perform fraud experiments. IEEE Security & Privacy, 6(2).

Downs, J.S., Holbrook, M.B. and Cranor, L.F., 2006, July. Decision strategies and susceptibility to phishing. In Proceedings of the second symposium on Usable privacy and security (pp. 79-90). ACM.

Siu, N., Iverson, L. and Tang, A., 2006, November. Going with the flow: email awareness and task management. In Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work (pp. 441-450). ACM.
Hong, J., 2012. The state of phishing attacks. Communications of the ACM, 55(1), pp.74-81.

Dhamija, R., Tygar, J.D. and Hearst, M., 2006, April. Why phishing works. In Proceedings of the SIGCHI conference on Human Factors in computing systems (pp. 581-590). ACM.

Ghosh-Dastidar, S. and Adeli, H., 2009. A new supervised learning algorithm for multiple spiking neural networks with application in epilepsy and seizure detection. Neural networks, 22(10), pp.1419-1431.

Ma, J., Saul, L.K., Savage, S. and Voelker, G.M., 2009, June. Beyond blacklists: learning to detect malicious web sites from suspicious URLs. In Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 1245-1254). ACM.

Sanglerdsinlapachai, N. and Rungsawang, A., 2010, January. Using domain top-page similarity feature in machine learning-based web phishing detection. In Knowledge Discovery and Data Mining, 2010. WKDD'10. Third International Conference on (pp. 187-190). IEEE.

Miyamoto, D., Hazeyama, H. and Kadobayashi, Y., 2008, November. An evaluation of machine learning-based methods for detection of phishing sites. In International Conference on Neural Information Processing (pp. 539-546). Springer, Berlin, Heidelberg.

Pan, Y. and Ding, X., 2006, December. Anomaly based web phishing page detection. In Computer Security Applications Conference, 2006. ACSAC'06. 22nd Annual (pp. 381-392). IEEE.

Salem, O., Hossain, A. and Kamala, M., 2010, June. Awareness program and ai based tool to reduce risk of phishing attacks. In Computer and Information Technology (CIT), 2010 IEEE 10th International Conference on (pp. 1418-1423). IEEE.

Sheng, S., Holbrook, M., Kumaraguru, P., Cranor, L.F. and Downs, J., 2010, April. Who falls for phish?: a demographic analysis of phishing susceptibility and effectiveness of interventions. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (pp. 373-382). ACM.

Lynch, J., 2005. Identity theft in cyberspace: Crime control methods and their effectiveness in combating phishing attacks. Berkeley Technology Law Journal, pp.259-300.

Rao, R.S. and Ali, S.T., 2015, April. A computer vision technique to detect phishing attacks. In Communication Systems and Network Technologies (CSNT), 2015 Fifth International Conference on (pp. 596-601). IEEE.

Barraclough, P.A., Hossain, M.A., Tahir, M.A., Sexton, G. and Aslam, N., 2013. Intelligent phishing detection and protection scheme for online transactions. Expert Systems with Applications, 40(11), pp.4697-4706.

Toolan, F. and Carthy, J., 2010, October. Feature selection for spam and phishing detection. In eCrime Researchers Summit (eCrime), 2010 (pp. 1-12). IEEE.

Fette, I., Sadeh, N. and Tomasic, A., 2007, May. Learning to detect phishing emails. In Proceedings of the 16th international conference on World Wide Web (pp. 649-656). ACM.

Rodriguez-Galiano, V. F., Ghimire, B., Rogan, J., Chica-Olmo, M., & Rigol-Sanchez, J. P. 2012. An assessment of the effectiveness of a random forest classifier for land-cover classification. ISPRS Journal of Photogrammetry and Remote Sensing, 67, 93-104.

Albon, C. (2017). Precision, Recall, and F1 Scores - Machine Learning. [online] Chrisalbon.com. Available at: https://chrisalbon.com/machine-learning/precision_recall_and_F1_scores.html [Accessed 5 Sep. 2017].

Appendices

Appendix A – Main.java

```
import java.lang.reflect.InvocationTargetException;
import javax.swing.SwingUtilities;
public class Main
{
    // References to both objects required by our program
    private static MainFrame mf;
    private static MailReader mr;

    // Main function of our program: the entry point
    public static void main(String[] args)
    {
        mr = new MailReader();

        /* Create our main window, SwingUtilities is used to run code in GUI thread, it's a must here */
        try {
            SwingUtilities.invokeAndWait(new Runnable()
            {
                public void run()
                {
                    mf = new MainFrame();
                    mf.setMailReader(mr); // Give MainFrame a reference to MailReader
object so it can pass messages.
                    mf.setSize(700, 600);
                    mf.setVisible(true);
                }
            });
        }
        catch (InvocationTargetException e) {
            e.printStackTrace();
        }
        catch (InterruptedException e) {
            // Do nothing
        }

        mr.setMainFrame(mf); // Give MailReader a reference to MainFrame object so it can pass messages
and use GUI features

        long lastTime = -1;

        /* An infinite loop which periodically checks for new emails */
        while (true)
        {
            long currentTime = System.currentTimeMillis();

            // Check mails when first run or once 15 seconds have passed
            if (lastTime == -1 || currentTime - lastTime > 15000)
            {
                mr.loop();

                lastTime = currentTime;
            }
```

```
			// Obtain exclusive access to mutex (thread synchronization)
			synchronized (MailReader.mutexObj) {
				int mailId = mr.getMailId();

				if (mailId != -1)
				{
					mr.switchToMail(mr.getMailId()); // Send MailReader a message to
retrieve contents for the selected e-mail
					mr.SetMailId(-1); // Remove selection
				}
			}

			try {
				// Wait a bit each time so this project don't use cpu 99%
				Thread.sleep(100);
			}
			catch (Exception e) {
				// Do nothing
			}
		}
	}
}
```

Appendix B – MailReader.java

```
import java.io.BufferedReader;
import java.io.File;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.HashSet;
import java.util.LinkedList;
import java.util.List;
import java.util.Properties;
import java.util.Set;

import javax.mail.Flags;
import javax.mail.Flags.Flag;
import javax.mail.Folder;
import javax.mail.Message;
import javax.mail.MessagingException;
import javax.mail.Session;
import javax.mail.internet.MimeMultipart;
import javax.mail.search.FlagTerm;
import javax.swing.SwingUtilities;

import jsat.ARFFLoader;
import jsat.DataSet;
import jsat.classifiers.CategoricalResults;
import jsat.classifiers.ClassificationDataSet;
import jsat.classifiers.Classifier;
import jsat.classifiers.DataPoint;
import jsat.classifiers.neuralnetwork.LVQ;
import jsat.linear.DenseVector;
import jsat.linear.Vec;
import jsat.linear.distancemetrics.EuclideanDistance;

import org.apache.commons.io.IOUtils;

import com.sun.mail.imap.IMAPStore;

/* Main class of our program */
```

```
public class MailReader
{
        private static final String username = "Testemailsec2@gmail.com"; // enter username
        private static final String password = "testemail"; // enter password

        /* Some variables for holding IMAP related objects */
        private IMAPStore mailStore = null;
        private Folder inbox = null;
        private Message[] messages = null;


        private MainFrame mf = null; // Reference to MainFrame (GUI) object

        /* Variables used for Machine learning algorithm */
        @SuppressWarnings("rawtypes")
        private DataSet dataSet;
        private ClassificationDataSet cDataSet;
        private Classifier classifier;

        // Used to receive message from another object
        private int idToRead = -1;

        // Used for synchronization
        public static final Object mutexObj = new Object();

        // List of important words this project care about (the same list that this project used to create feature vectors in
ARFF files)
        private static final String[] impWords = { "please", "account", "information", "thank", "click", "link", "security",
"user", "protect", "help", "update", "produced", "member", "access", "agreement", "rights", "reserved", "customer",
"notification", "service", "verify", "fraud", "notice", "attention", "recently", "ensure", "process", "department", "assistance",
"included", "review", "activity", "need", "site", "visit", "personal", "preferences", "confirm", "secure", "matter", "apologize",
"choose", "possible", "complete", "verification", "provide", "updated", "fraudulent", "password", "center", "protection",
"inconvenience", "report", "request", "safety", "safe", "changes", "accordance", "problems", "suspended", "unusual",
"change", "confidence", "suspension", "billing", "compromised" };

        // This projectights for each word (respectively)
        private static final double[] wordDec = { 2.6487941126, 1.0186156782, 2.5636789424, 4.6394942967,
3.5824815112, 3.1556343439, 3.4994349682, 2.1777406549, 5.3891913218, 2.0206898588, 2.9744649079,
2.6239087810, 5.4350670702, 2.9785105770, 7.1147834472, 2.6492900758, 2.7332425081, 5.0604115167,
5.7233765294, 2.1850562701, 1.4319963264, 6.5774288280, 3.8267202469, 4.1699250014, 2.7583530216,
4.6133071264, 2.1699250014, 5.2910821279, 5.9829935747, 3.1670596677, 2.8134039667, 5.6786979390,
0.4245812719, 1.9872889483, 3.1667154450, 2.5219527032, 3.0435690149, 5.1732607140, 1.6584246627,
2.8384561393, 8.7108064337, 2.9759831189, 1.5106151593, 2.8675497040, 7.0407463423, 2.9216121013,
2.8524428116, 8.5887146356, 3.6232325176, 2.4443641705, 3.6667565919, 8.5077946402, 1.3992701834,
3.0834160082, 5.0122435958, 3.7671658319, 2.1018796140, 8.4594316186, 0.9344116584, 7.3353903547,
4.6349506366, 0.6898048666, 4.6903155009, 8.0714623626, 8.0223678130, 5.0056245492 }; // new

        public MailReader()
        {
                /* Load our training data */
                dataSet = ARFFLoader.loadArffFile(new File("model.arff"));
                cDataSet = new ClassificationDataSet(dataSet, 0);

                /* Create classifier object for LVQ algorithm */
                /* 5 and 30 are some parameters to LVQ which this project found by trial and error */
                /* These parameter seem to yield max success rate */
    classifier = new LVQ(new EuclideanDistance(), 5);
    LVQ instance = (LVQ)classifier;
    instance.setRepresentativesPerClass(30);
    instance.setLVQMethod(LVQ.LVQVersion.LVQ1);
    classifier.trainC(cDataSet); // Train model!
        }

        // Set reference to GUI window
        public void setMainFrame(MainFrame mf)
        {
                this.mf = mf;
        }

        /* public functions for message passing */
        public int getMailId()
        {
                return idToRead;
        }

        public void SetMailId(int newId)
        {
                idToRead = newId;
        }

        // Removed unneeded stuff from words. For example makes "please", "please.", "please," same by removing .
and ,.
        // Uses regular expressions to do this
        public static String sanitizeWord(String text)
        {
                String newText = text.toLothis projectrCase();

                newText = newText.replaceFirst("^[^a-zA-Z]*([a-zA-Z]+?.*?[a-zA-Z]+?)[^a-zA-Z]*$", "$1");

                return newText;
        }

        // Strip HTML tags from mail content using regular expressions
        public static String sanitizeSentence(String text)
        {
                String newSentence = text.replaceAll("\\<[^>]*?\\>", " ");

                return newSentence;
        }

        // Function which checks for new mails
        public void checkMails()
        {
                /* Properties required for GMAIL IMAP connection using JavaMail API */
                Properties props = new Properties();
                props.put("mail.imap.host", "imap.gmail.com");
                props.put("mail.imap.port", "993");
                props.put("mail.imap.socketFactory.class", "javax.net.ssl.SSLSocketFactory");
                props.put("mail.imap.socketFactory.fallback", "false");
                props.put("mail.imap.socketFactory.port", "993");

                // Open a mail transfer session
                Session recvSession = Session.getInstance(props);

                // Close first if there are objects already open
                try {
                    if (inbox != null && inbox.isOpen())
                        inbox.close(true);
                    if (mailStore != null && mailStore.isConnected())
                        mailStore.close();
                }
                catch (MessagingException e) {
                    // do nothing
                }

                try {
                    mailStore = (IMAPStore)recvSession.getStore("imap"); // use imap to access mails
                    mailStore.connect(username, password); // Connect using username and password
                    inbox = mailStore.getFolder("Inbox"); // Retrieve contents of Inbox folder
                    inbox.open(Folder.READ_ONLY); // This project won't change anything so read only

                    /* This means to filter the mails so that this project see only unread mails */
                    messages = inbox.search(new FlagTerm(new Flags(Flag.SEEN), false));
                }
                catch (MessagingException e) {
                    e.printStackTrace();
                }
        }

        // Extracts text data from MIME multipart type mails
        public boolean getContent(MimeMultipart mmp, StringBuilder content)
        {
                BufferedReader br = null;

                try {
                    if (mmp.getCount() < 1)
                        return false;

                    br = new BufferedReader(new
InputStreamReader(mmp.getBodyPart(0).getInputStream()));

                    String line;

                    while ((line = br.readLine()) != null)
                    {
                        content.append(line);
                        content.append("\r\n");
                    }
                }
                catch (Exception e) {
                    e.printStackTrace();
                    return false;
                }
                finally {
                    IOUtils.closeQuietly(br);
                }

                return true;
        }

        // This function does the classification. It needs a DataPoint to work
        // A data point is a single feature vector just like one line in our .arff files
        public ClsLabel calculatePhishingProb(DataPoint dp)
        {
                CategoricalResults predictionResults = classifier.classify(dp); // Clasify
          int predicted = predictionResults.mostLikely(); // Get most likely category (phishing or clean)
          double prob = predictionResults.getProb(predicted); // Confidence

                return new ClsLabel(predicted, prob);
        }

        // Converts text data from mail content to a feature vector
        public DataPoint mailToVector(String mailData)
        {
                List<Double> vecItems = new LinkedList<Double>();

                // Some sanitization like removing some whitespace
                mailData = mailData.replace("\r\n", " ");
                mailData = mailData.replace("\n", " ");
                mailData = sanitizeSentence(mailData);

                String[] words = mailData.split(" ");
                Set<String> wordSet = new HashSet<String>();

                // Find unique words using Set data structure
                for (int j = 0; j < words.length; j++)
                {
                    String realWord = sanitizeWord(words[j]);
                    wordSet.add(realWord);
                }

                // Fill feature vector
                for (int i = 0; i < impWords.length; i++)
                {
                    /* If next important word exists in text value of this index of feature vector is its this
projectight (1*this projectight) */
                    if (wordSet.contains(impWords[i]))
                        vecItems.add(wordDec[i]);
                    else
                        vecItems.add(0.0); // If word doesn't exist value at this index is 0
                }

                /* Return vector in the format JSAT library demands */
                Vec vector = new DenseVector(vecItems);
                return new DataPoint(vector);
        }

        // Retrieve and show mail content
        public void switchToMail(int id)
        {
                if (id < 0)
                    return;

                try {
                    final String fullData;
                    Object rawContent = messages[id].getContent();

                    /* If mail content is consisting of only a string no problem, if its MIME Multipart this project
need to extract data from it */
                    if (rawContent instanceof String)
                    {
                        fullData = (String)rawContent;
                    }
                    else if (rawContent instanceof MimeMultipart)
                    {
                        MimeMultipart mmp = (MimeMultipart)rawContent;
                        StringBuilder sb = new StringBuilder();

                        // Extract text from MIME Multipart data
                        if (!getContent(mmp, sb))
                        {
                            throw new MessagingException("Cannot parse MIME Multipart
message!");
                        }
```

```
                        fullData = sb.toString();
                }
                else
                {
                        return;
                }

                // Show mail content in GUI (works in GUI thread by using invokeLater)
                SwingUtilities.invokeLater(new Runnable()
                {
                        public void run()
                        {
                                mf.setMailContent(fullData);
                        }
                });

                // Conver raw mai data to vector form
                DataPoint curVec = mailToVector(fullData);
                ClsLabel cls = calculatePhishingProb(curVec); // Classify the mail

                final String statusText;

                /* If mail is phishing show it in GUI and play a sound alarm */
                if (cls.clsId == ClsLabel.CLS_PHISHING)
                {
                        statusText = "Caution: This message probably contains a phishing link!";

                        File alarmSound = new File("ring.wav");

                        if (alarmSound.exists())
                                PlaySound.play(alarmSound);
                }
                else
                {
                        // Show in GUI mail is clean
                        statusText = "Info: This message seems clean.";
                }

                // Set status textarea to tell user whether the mail is phishing or not
                SwingUtilities.invokeLater(new Runnable()
                {
                        public void run()
                        {
                                mf.setStatusText(statusText);
                        }
                });
        }
        catch (MessagingException e) {
                e.printStackTrace();
        }
        catch (IOException e) {
                e.printStackTrace();
        }
}

// Called from main every 15 seconds to check for new mails
public void loop()
{
        checkMails();
        // If no messages in the mailbox, don't do anything
        if (messages == null)
            return;

        // set unread mail count on GUI
        SwingUtilities.invokeLater(new Runnable()
        {
            public void run()
            {
                mf.setUnreadMailCount(messages.length);
            }
        });

        if (messages.length > 0)
        {
            /* If there are messages (mailbox is not empty) pass list of mails to GUI window */
            final MailMessage[] ms = new MailMessage[messages.length];
            for (int i = 0; i < ms.length; i++)
            {
                try {
                    ms[i] = new MailMessage(i, messages[i].getSubject());
                }
                catch (MessagingException e) {
                    e.printStackTrace();
                }
            }

            SwingUtilities.invokeLater(new Runnable()
            {
                public void run()
                {
                    mf.setMailList(ms);
                }
            });
        }
    }
}
```

Appendix C – MainFrame.java

```
import javax.swing.DefaultListModel;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JList;
import javax.swing.ListSelectionModel;

import java.awt.Font;

import javax.swing.JTextArea;
import javax.swing.event.ListSelectionEvent;
import javax.swing.event.ListSelectionListener;
import javax.swing.JScrollPane;

/* This class is a window (JFrame) in Swing GUI framework */
```

```
@SuppressWarnings("serial")
public class MainFrame extends JFrame
{
        /* Variables for GUI components */
        private JLabel lblUnreadMails;
        private JList<String> list;
        private JTextArea textArea;
        private JTextArea textAreaStatus;
        private MailReader mr;

        // Public function to set our reference for MailReader object so this project can pass messages.
        public void setMailReader(MailReader mr)
        {
        this.mr = mr;
        }

        // Public function used by MailReader class to update unread mail count on GUI
        public void setUnreadMailCount(int count)
        {
        lblUnreadMails.setText(String.format("Unread Mails: %d", count));
        }

        // Public function used by MailReader class to update list of mails on GUI
        public void setMailList(MailMessage[] messages)
        {
        list.setSelectedIndex(-1);
        DefaultListModel<String> dlm = new DefaultListModel<String>();

        for (int i = 0; i < messages.length; i++)
        {
        if (messages[i] != null)
        dlm.addElement(messages[i].subject);
        else
        dlm.addElement(" ");
        }

        list.setModel(dlm);
        }

        // Public function to set mail contents on the text area on the GUI
        public void setMailContent(String content)
        {
        textArea.setText(content);
        }

        // Public function which is used to set status bar text
        public void setStatusText(String status)
```

```
{
textAreaStatus.setText(status);
}

// Mainframe class constructor
public MainFrame()
{
/* Prepare our window by setting its title, size, etc */
setTitle("Secure Mail Reader");
getContentPane().setLayout(null);

/* Initialize various GUI components in the following lines */
lblUnreadMails = new JLabel("Unread Mails: 0");
lblUnreadMails.setBounds(564, 38, 99, 14);
getContentPane().add(lblUnreadMails);

DefaultListModel<String> dlm = new DefaultListModel<String>();

list = new JList<String>();
list.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);
list.setModel(dlm);
JScrollPane jsp1 = new JScrollPane(list);
jsp1.setBounds(25, 81, 249, 285);
getContentPane().add(jsp1);

// Add an event listener to the selection change event of listbox (mail list)
list.addListSelectionListener(new ListSelectionListener()
{
@Override
public void valueChanged(ListSelectionEvent e)
{
// Obtain access to mutex (used for thread synchronization
// Mutexes are used to synchronize access so something can't be accessed from more than one place at the
same time
synchronized (MailReader.mutexObj) {
// Pass MailReader object selected mail index as a message so it shows our content
mr.SetMailId(list.getSelectedIndex());
}
}
});

/* Various GUI component initializations follow */
JLabel lblSecureMailReader = new JLabel("SECURE MAIL READER v0.5");
lblSecureMailReader.setFont(new Font("Tahoma", Font.BOLD, 26));
lblSecureMailReader.setBounds(25, 26, 396, 27);
getContentPane().add(lblSecureMailReader);
```

```
		textArea = new JTextArea();
		textArea.setLineWrap(true);
		JScrollPane jsp2 = new JScrollPane(textArea);
		jsp2.setBounds(295, 81, 368, 285);
		getContentPane().add(jsp2);

		textAreaStatus = new JTextArea();
		textAreaStatus.setLineWrap(true);
		textAreaStatus.setBounds(25, 402, 638, 118);
		getContentPane().add(textAreaStatus);

		JLabel lblStatus = new JLabel("Status:");
		lblStatus.setBounds(25, 377, 46, 14);
		getContentPane().add(lblStatus);
	}
}
```