**CHAPTER 1**

# **INTRODUCTION**

Machine learning models have been utilized for multiple purposes in the field of computer science from resolving a network traffic issue to detecting a malware. Emails are used regularly by many people for communication and for socialising. Security breaches that compromises customer data allows ‘spammers’ to spoof a compromised Email address to send illegitimate (spam) Emails. This is also exploited to gain unauthorized access to their device by tricking the user into clicking the spam link within the spam Email, that constitutes a phishing attack. Many tools and techniques are offered by companies in order to detect spam Emails in a network. Organisations have set up filtering mechanisms to detect unsolicited Emails by setting up rules and configuring the firewall settings. Google is one of the top companies that offers 99.9% success in detecting such Emails. There are different areas for deploying the spam filters such as on the gate way (router), on the cloud hosted applications or on the user’s computer. In order to overcome the detection problem of spam Emails, methods such as content-based filtering, rule-based filtering or Bayesian filtering have been applied. Unlike the ‘knowledge engineering’ where spam detection rules are set up and are in constant need of manual updating thus consuming time and resources, Machine learning makes it easier because it learns to recognise the unsolicited Emails (spam) and legitimate Emails (ham) automatically and then applies those learned instructions to unknown incoming Emails. The proposed spam detection to resolve the issue of the spam classification problem can be further experimented by feature selection or automated parameter selection for the models. This research conducts experiments involving five different machine learning models with Genetic Algorithm (GA). This will be compared with the base models to conclude whether the proposed models have improved the performance with parameter tuning. The user interface to detect the Spam Email is implemented using the modules Service provider, View and Authorize users, Remote users. The accuracy of all the algorithms used is represented using pie chart and line graph, also the accuracy of the machine learning algorithms with Bio inspired algorithm is represented using the Bar graph.

* 1. **Literature Review**

**W. Feng, J. Sun, L. Zhang, C. Cao, and Q. Yang, ‘‘A support vector machine based Naive Bayes algorithm for spam filtering,’’ in Proc. IEEE 35th Int. Perform. Comput. Commun. Conf. (IPCCC), Dec. 2016, pp. 1–8, doi: 10.1109/pccc.2016.7820655.**

Naive Bayes classifiers are widely used to filter spam Emails, however, the strong independence assumptions between features limit their performance in accurately identifying spams. To address this issue, we proposed a support machine vector based naive Bayes - SVM-NB - filtering system. The SVM-NB first constructs an optimal separating hyperplane that divides samples in the training set into two categories. For samples located nearby the hyperplane, if they are in different categories, one of them will be eliminated from the training set. In this way, the dependence between samples is reduced and the entire training sample space is simplified. With the trimmed training set, the naive Bayes algorithm is applied to classify Emails in the test set. The SVM-NB system is evaluated with the dataset obtained from DATAMALL. Experiment results demonstrate that SVM-NB can achieve a higher spam-detection accuracy and a faster classification speed.

**E. G. Dada, J. S. Bassi, H. Chiroma, S. M. Abdulhamid, A. O. Adetunmbi, and O. E. Ajibuwa, ‘‘Machine learning for Email spam filtering: Review, approaches and open research problems,’’ Heliyon, vol. 5, no. 6, Jun. 2019, Art. no. e01802, doi: 10.1016/j.heliyon.2019.e01802.**

The upsurge in the volume of unwanted Emails called spam has created an intense need for the development of more dependable and robust antispam filters. [Machine learning methods](https://www.sciencedirect.com/topics/engineering/machine-learning-method) of recent are being used to successfully detect and filter spam Emails. We present a [systematic review](https://www.sciencedirect.com/topics/psychology/systematic-review) of some of the popular machine learning based Email spam filtering approaches. Our review covers survey of the important concepts, attempts, efficiency, and the research trend in spam filtering. The preliminary discussion in the study background examines the applications of [machine learning techniques](https://www.sciencedirect.com/topics/computer-science/machine-learning-technique) to the Email spam filtering process of the leading [internet service providers](https://www.sciencedirect.com/topics/computer-science/internet-service-provider) (ISPs) like Gmail, Yahoo and [Outlook](https://www.sciencedirect.com/topics/nursing-and-health-professions/angiographic-catheter) Emails spam filters. Discussion on general Email spam filtering process, and the various efforts by different researchers in combating spam through the use machine learning techniques was done. Our review compares the strengths and drawbacks of existing  [machine learning approaches](https://www.sciencedirect.com/topics/computer-science/machine-learning-approach) and the open research problems in spam filtering. We recommended deep leaning and deep adversarial learning as the future techniques that can effectively handle the menace of spam Emails.

**W. Awad and S. ELseuofi, ‘‘Machine learning methods for spam E-Mail classification,’’ Int. J. Comput. Sci. Inf. Technol., vol. 3, no. 1, pp. 173–184, Feb. 2011, doi: 10.5121/ijcsit.2011.3112.**

Recently unsolicited commercial / bulk e-mail also known as spam, become a big trouble over the internet. Spam is waste of time, storage space and communication bandwidth. The problem of spam e-mail has been increasing for years. In recent statistics, 40% of all Emails are spam which about 15.4 billion Email per day and that cost internet users about $355 million per year. Automatic e-mail filtering seems to be the most effective method for countering spam at the moment and a tight competition between spammers and spam-filtering methods is going on. Only several years ago most of the spam could be reliably dealt with by blocking e-mails coming from certain addresses or filtering out messages with certain subject lines. Spammers began to use several tricky methods to overcome the filtering methods like using random sender addresses and/or append random characters to the beginning or the end of the message subject line. Knowledge engineering and machine learning are the two general approaches used in e-mail filtering. In knowledge engineering approach a set of rules has to be specified according to which Emails are categorized as spam or ham. A set of such rules should be created either by the user of the filter, or by some other authority (e.g. the software company that provides a particular rule-based spam-filtering tool). By applying this method, no promising results shows because the rules must be constantly updated and maintained, which is a waste of time and it is not convenient for most users. Machine learning approach is more efficient than knowledge engineering approach; it does not require specifying any rules. Instead, a set of training samples, these samples is a set of pre classified e-mail messages. A specific algorithm is then used to learn the classification rules from these e-mail messages. Machine learning approach has been widely studied and there are lots of algorithms can be used in e-mail filtering.

* 1. **Motivation**

Emails are used regularly by many people for communication and for socialising. Security breaches that compromises customer data allows ‘spammers’ to spoof a compromised Email address to send illegitimate (spam) Emails. This is also exploited to gain unauthorized access to their device by tricking the user into clicking the spam link within the spam Email, that constitutes a phishing attack.Therefore this paper aims to present a method for detection of spam Emails with machine learning algorithms that are optimized with bio-inspired methods with the following.

1. Finding the highest accuracy for detecting the Emails correctly as ham and spam.
2. To enhance the performance efficiency of the algorithm.
   1. **Objectives**
3. To explore machine learning algorithms for the spam detection problem.
4. To investigate the workings of the algorithms with the acquired datasets.
5. To implement the bio-inspired algorithms.
6. To test and compare the accuracy of base models with bio-inspired implementation.
7. To implement the framework using Python.
   1. **Problem Statement**

Security breaches that compromises customer data allows ‘spammers’ to spoof a compromised Email address to send illegitimate (spam) Emails. This is also exploited to gain unauthorized access to their device by tricking the user into clicking the spam link within the spam Email, that constitutes a phishing attack. Where spam detection rules are set up and are in constant need of manual updating thus consuming time and resources. In the existing system, the system is not accurate on large data sets due to lack of bio-inspired algorithms. This system is less performance due to lack of accurate spam detection on large set of Emails.

**1.5 Organization of the Report**

The report is classified into five chapters. Each chapter deals with different aspects associated with project work.

The first chapter deals with introduction, literature review, problem statement and aim of the project.

Second chapter is based on theory and concept such as existing system, proposed system, advantages and disadvantages of existing system.

Third chapter presents Block diagram, Methodology, Data flow diagram and System design are discussed in this chapter.

Fourth chapter explains the implementation which includes modules, user interface, outcomes and system testing in brief.

Fifth chapter summarizes the conclusion and future scope of the project along with references.

**CHAPTER 2**

# **THEORY AND CONCEPT**

**2.1 MACHINE LEARNING**

Researchers have taken a lead to implement machine learning models to detect spam Emails. In the paper , the authors have conducted experiments with six different machine learning algorithms: Naïve Bayes (NB) classification, K-Nearest Neighbour (K-NN), Artificial Neural Network (ANN), Support Vector Machine (SVM), Artificial Immune System and Rough Sets. Their aim of the experiment was to imitate the detecting and recognising ability of humans. Tokenisation was explored and the concept provided two stages: Training and Filtering. Their algorithm consisted of four steps: Email Pre-Processing, Description of the feature, Spam Classification and Performance Evaluation. It concluded that the Naïve Bayes provided the highest accuracy, precision and recall. Feng et al describes a hybrid system between two machine learning algorithms i.e. SVM-NB. Their proposed method is to apply the SVM algorithm and generate the hyperplane between the given dimensions and reduce the training set by eliminating datapoints. This set will then be implemented with NB algorithm to predict the probability of the outcome. This experiment was conducted on Chinese text corpus. They successfully implemented their proposed algorithm and there was an increase in accuracy when compared to NB and SVM on their own. Mohammed et al.aimed to detect the unsolicited Emails by experimenting with different classifiers such as: NB, SVM, KNN, Tree and Rule based algorithms. They generated a vocabulary of Spam and Ham Emails which is then used to filter through the training and testing data. Their experiment was conducted with Python programming language on Email-1431 dataset. They concluded that NB was the best working classifier followed by Support Vector Machine. Wijaya and Bisri proposes a hybrid-based algorithm, which is integrating Decision Tree with Logistic Regression along with False Negative threshold. They were successful in increasing the performance of DT. The results were compared with the prior research. The experiment was conducted on the SpamBase dataset. The proposed method presented a 91.67% accuracy.

**2.2 BIO-INSPIRED METHODS**

Agarwal and Kumar experimented with NB along with Particle Swarm Optimisation (PSO) technique. The paper used the Emails from Ling-Spam corpus and aimed to acquire an improvement in F1-score, Precision, Recall and Accuracy. The paper used Correlation Feature Selection (CFS) to select appropriate features from the dataset. The dataset was split into 60:40 ratio. Particle Swarm Optimisation was integrated along with Naïve Bayes. They concluded a success when their proposed integrated method increased the accuracy of the detection compared to NB alone. Belkebir and Guessoum reviewed the SVM algorithm along with Bee Swarm Optimization (BSO) and Chi-Squared on Arabic Text. Since there have been plenty of research conducted for text mining on English and some European languages, the authors considered to review the algorithms work on Arabic language. They experimented with three different approaches to categorise automatic text – Neural networks, Support Vector Machine (SVM) and SVM optimizing with Bee Swarm Algorithm (BSO) along with Chi-Squared. Bee Swarming Optimization algorithm is inspired by the behaviour of swarm of bees to achieve global solution. A search area is divided and each area within the divided section is assigned to other bees to explore. Every solution is distributed amongst the bees and the best solution is accepted and the process is repeated until the solution meets the criteria of the problem. The main problem advertised is: ‘‘The problem of selecting the set of attributes is NP-hard’’. The research explains the problem dealing with the feature selection due to the computation time. A vocabulary is generated and fed into the Chi2-BSO algorithm to acquire the features and finally the achieved result is loaded within the SVM algorithm. The experiment was carried on OSAC dataset which included 22,429 text records. The study randomly selected 100 texts from each category distributed by 70:30 ratio. The program performed removal of digits, Latin alphabets, isolated letters, punctuation marks and stopwords. The document representation step was conducted with different modes for all approaches – SVM, BSO-CHI-SVM and artificial neural network (ANN). The SVM outperformed the ANN execution time. The proposed algorithm BSO-CHI-SVM exceeds the learning time but it is still identified as effective. The paper concluded that the proposed algorithm provides an accuracy rate of 95.67%. They have also stated that SVM approach outperformed ANN. A further development is to evaluate the approach of this article on other datasets and use modes such as n-gram or concept representation.

**2.3 EXISTING SYSTEM**

Where spam detection rules are set up and are in constant need of manual updating thus consuming time and resources, Machine learning makes it easier because it learns to recognize the unsolicited Emails(spam)and legitimate Emails(ham)automatically and then applies those learned instructions to unknown incoming Emails.

**Disadvantages**

* In the existing work, the system is not accurate on large data sets due to lack of bio-inspired algorithms.
* This system is less performance due to lack of accurate spam detection on large set of Emails.

**2.4 PROPOSED SYSTEM**

The proposed spam detection to resolve the issue of the spam classification problem can be further experimented by feature selection or automated parameter selection for the models. This research conducts experiments involving five different machine learning models with Genetic Algorithm (GA). This will be compared with the base models to conclude whether the proposed models have improved the performance with parameter tuning.

1) To explore machine learning algorithms for the spam detection problem.

2) To investigate the workings of the algorithms with the acquired datasets.

3) To implement the bio-inspired algorithms.

4) To test and compare the accuracy of base models with bio-inspired implementation.

5) To implement the framework using Python.

**Advantages**

* The proposed spam detection to resolve the issue of the spam classification problem can be further experimented by feature selection or automated parameter selection for the models.
* The system is more effective since spam detection problem is solving by GADT proposed algorithm.
  1. **TOOLS AND TECHNIQUES**
     1. **Weka**
* WEKA is a GUI tool that allows to load a dataset and apply different functions/rules upon an algorithm .
* The application allows to apply the classification, regression, clustering algorithms and enable to visualise the data and the performance of the algorithm.
  + 1. **Scikit-Learn**
* Scikit-Learn (SKLearn) is an environment that is incorporated with Python programming language.
* The library offers high-level implementation to train with the ‘Fit’ methods and ’predict’ from an estimator (Classifier).
* It also offers to perform the cross validation, feature selection, feature extraction and parameter tuning.
  + 1. **Python IDLE**
* It has efficient high-level data structures and a simple but effective approach to object-oriented programming.
* The Python interpreter is easily extended with new functions and data types implemented in C or C++ (or other languages callable from C).
  + 1. **Open CV-Python**
* OpenCV is a library of programming functions mainly aimed at real-time computer vision.
* It enables the programmer to express his ideas in fewer lines of code without reducing any readability.
* Compared to other languages like C/C++, Python is slower. But another important feature of Python is that it can be easily extended with C/C++.
* This feature helps us to write computationally intensive codes in C/C++ and create a Python wrapper for it so that we can use these wrappers as Python modules.
  + 1. **Jupyter Notebook**
* This is an open source tool that provides a Python framework.
* This is similar to ‘Spyder’ IDE, except this tool lets a user run the source code via a web browser .
* Anaconda framework also offers ‘Jupyter’ to be utilised by the user through the local server.
  + 1. **HTML**

HTML stands for Hyper Text Markup Language. Not to be mistaken, HTML is not a programming language but it is a markup language for web pages. HTML documents are the web pages and the HTML tags and plain text describes the web pages. A web browser will read the documents by interpreting the tags and then displays the content of the web pages. In between the HTML opening and closing tags, the author may create the pages by adding images and objects or designing text structure by putting headings, paragraphs. Even though HTML language seems like the most simple and basic, it can embed scripts in languages that can affect the behavior of the web pages such as JavaScript. To enhance the appearance and layout of the pages, the author can use Cascading Style Sheets (CSS) on the HTML web pages.

* + 1. **PHP**

PHP stands for Hypertext Preprocessor. PHP is developed as open-source software. It is one of the server-side scripting languages among Perl, Python and Java. PHP scripts are executed on the server. Therefore, a PHP page will not display the PHP codes on the client interface because it is executed on the server. One of the advantages of PHP is that it can run on different platforms or operating systems (Windows, Linux, Unix, etc.). PHP also supports many databases (MySQL, Informix, Oracle, Sybase, etc.). It is also compatible with almost all web servers used today (Apache, IIS, etc.). As open-source software, PHP is free to download and use. It is also easy to learn and runs efficiently on the server side.

* + 1. **MySQL**

MySQL is an open-source database server system. MySQL supports to use on many platforms such as Windows, Linux and Mac OS. MySQL is one of the world's top databases as it is used by the world's largest organizations including Facebook, Google, Adobe, Twitter and Wikipedia. The advantages of MySQL that attract authors to use are its high performance, high reliability and its ease of use. MySQL also has the function of indexes which improves SELECT queries speed tremendously. MySQL is open-source software so it is free to download and use.

# **MACHINE LEARNING ALGORITHMS**

## **Naïve BAYES - MULTINOMIAL**

Naïve Bayes model is used to resolve classification problems by using probability techniques. The Naïve Bayes algorithm for this article can be denoted as equation :

P(Class|WORD) = (P(WORD|Class) × P(Class)) (P(WORD)) where WORD is (word1,word2, . . .wordn) from within an uploaded Email and ‘Class’ is either ‘Spam’ or ‘Ham’.The algorithm calculates the probability of a class from the bag of words provided by the program. Where P(Class | WORD) is a posterior probability, P(WORD | Class) is likelihood and P(Class) is the prior probability. If ‘Class’ = Spam, the equation could be rewritten to find the spam Email from the given words. Multinomial Naïve Bayes algorithm has been selected to perform the spam Email identification because it is text related and outperforms Gaussian and Bernoulli. Multinomial Naïve Bayes (MNB) classifier uses Multinomial Distribution for each given feature, focusing on term frequency.

* + 1. **DECISION TREE CLASSIFIER**

The Decision Tree model is based on the predictive method. The model creates a category which is further distributed into sub-categories and so on. The algorithm runs until the user has terminated or the program has reached its end decision. The model predicts the value of the data by learning from the provided training data. The longer and deeper the tree implies it has more complicated rules to be executed. The Decision Tree - CART Algorithm shows the pseudo-code for Decision Tree, where it terminates

at the end of the node for each split of the tree depth. Similar to MNB and SGD, Decision Tree (DT) algorithm was loaded from the Scikit-learn library and it is executed on the default parameters which are ‘Gini’ for Criterion and ‘best’ for Splitter. The advantage of Gini is that it calculates the incorrectly labelled data that was selected randomly.

# **PROGRAM STRUCTURE, DATASETS AND REQUIREMENTS**

* + 1. **TOKENIZATION**

Tokenization is the method where the sentences within an Email are broken into individual words (tokens). These tokens are saved into an array and used towards the testing data to identify the occurrence of every word in an Email. This will help the algorithms in predicting whether the Email should be considered as spam or ham.

* + 1. **FEATURE EXTRACTION AND STOP WORDS**

This was used to remove the unnecessary words and characters within each Email, and creates a bag of words for the algorithms to compare against. The module ‘Count Vectorizer’ from Scikit-learn assigns numbers to each word/token while counting and provides its occurrence within an Email. The instance is invoked to exclude the English stopwords, and these are the words such as: A, In, The, Are, As, Is etc., as they are not very useful to classify whether the Email is spam or not.

* + 1. **MODEL TRAINING AND TESTING PHASE**

Supervised learning methods are used and the model is trained with known data and tested with unknown data to predict the accuracy and other performance measures. To acquire the reliable results K-Fold cross validation is applied. This method does have its disadvantages such as, there is a chance that the testing data could be all spam Emails, or the training set could include the majority of spam Emails. This is resolved by Stratified K-fold cross validation, which separates the data while making sure to have a good range of Spam and Ham into the distributed set. Lastly, parameter tuning is conducted with the Scikit-Learn and bio-inspired algorithms approach to try and improve the accuracy of ML models. This provides a platform to compare the Scikit-learn library with the bio-inspired algorithms.

* + 1. **DATASETS**

The project access the publicly available datasets and included each Email as an individual text file. The text files were string based. A list of the few spam Email datasets from the public repository are explained below:

1. Ling-Spam dataset is divided into 10 parts from the ‘bare’ distribution that includes individual Emails as a text file (.txt). This data is not pre-processed, and it includes numbers, alphabets and characters. Each part was trained and tested to acquire the average accuracy.
2. The PUA dataset is a numerical dataset that includes sets/combination of numbers characterised as a string. PU1, PU2 and PU3 are similar to PUA dataset but include different weights of spam and ham Emails and they are extracted from different users.

# **PERFORMANCE MEASURES**

* + 1. **CONFUSED MATRIX**

The detection of spam Emails can be evaluated by different performance measures. Confusion Matrix is being used to visualise the detection of the Emails for models. Confusion matrix can be defined as below:

1) TN = True Negative – Ham Email predicted as ham

2) TP = True Positive – Spam Email predicted as spam

3) FP = False Positive – Spam Email predicted as ham

4) FN = False Negative – Ham Email predicted as spam

* + 1. **ACCURACY**

The research was aimed at finding the highest accuracy for detecting the Emails correctly as ham and spam. The module from the Scikit-learn library called ‘Accuracy’ helped analyse the correct number of Emails classified as ‘Spam’ and ‘Ham’. This can be measured by equation below: Accuracy = (TN + TP) (TP + FN + FP + TN) where the denominator of the equation is the total number of Emails within the testing data.

# **BIO-INSPIRED OPTIMIZATION ALGORITHMS**

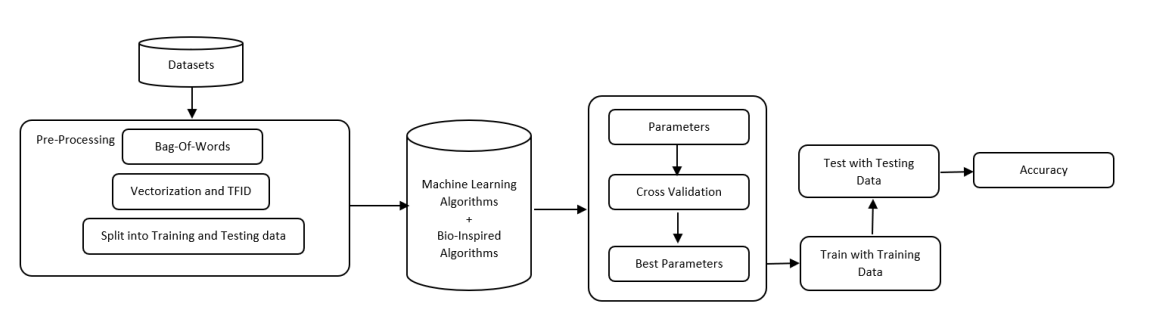
* + 1. **GENETIC ALGORITHM**

Genetic Algorithm is known to be an evolutionary process wherein a population of solutions evolves over a sequence of new generations. The process of Genetic Algorithm starts and the cross over takes place once the chromosomes are constructed for all the mails. Cross-over operation can be performed in various ways. Crossover is only allowed for a bit of gene in a particular category only. In this algorithm, both multi-point and single point crossover is done and the positions of the bits are selected randomly. In each generation of chromosomes only 12% of the bits are crossed. Next follows the process of mutation. so as to recover some of the lost genes. Assume an example of only 3 % of genes are mutated. The process starts by comparing the weight of the words of the gene in the test mail with those of the gene in the spam mail prototype so as to find the matching gene. If number of matched genes is greater than or equal to the numerical value three, than spam mail prototype receives a single score point. If the score point happens to be greater than say some threshold score point than the mail is considered as a spam mail.

**CHAPTER 3**

**BLOCKDIAGRAM, METHODOLOGY, SPECIFICATIONS, DESIGN**

**3.1 BLOCK DIAGRAM**

****

**Figure 1: Spam detection block diagram.**

**WEKA**

WEKA is a GUI tool that allows to load a dataset and apply different functions/rules upon an algorithm. The application allows to apply the classification, regression, clustering algorithms and enable to visualise the data and the performance of the algorithm. In this experiment WEKA acted as a black box and provided the better performing algorithms which were Naïve Bayes, Logistic Regression and Decision Tree. Since spam Email detection falls into classification category, supervised learning method will be used. Supervised learning is a concept where the dataset is split into two parts:1) Training data and 2) Testing data. The main aim of this learning method is to train a classifier with a given data and parameters and then predict the outcome with the testing dataset which will not be known to the program or classifier. The models will be trained with a training dataset of 60%, 70%, 75% and 80%. Once the model is trained, model will be provided with the testing dataset which is distributed as 40%, 30%, 25% and 20% respectively with training dataset. This will provide a better knowledge of what percentage split is best suited and thus be more efficient to work with majority of the datasets. This will provide results on classifiers working best with more or less training data.

**SCIKIT-LEARN**

Scikit-Learn (SKLearn) is an environment that is incorporated with Python programming language. The library offers a wide range of supervised algorithms that will be suitable for this project. The library offers high-level implementation to train with the ’Fit’ methods and ’predict’ from an estimator (Classifier). It also offers to perform the cross validation, feature selection, feature extraction and parameter tuning.

**KERAS**

Keras is an API that supports Neural Networks. The API supports other deep learning algorithms for easy and fast approach. It offers CPU and GPU running capabilities in order to simultaneously process the models. Online tutorials are available for neural network for learning and development. Their guide demonstrates the performance optimization techniques to utilize GPU and ways to work with RNN algorithm and other deep learning algorithms .

**TENSORFLOW**

Tensorflow is an end-to-end ML platform that is developed by Google. The architecture lets a user run the program on multiple CPUs and it also has access to GPUs. The website also provides a learning platform for both beginners and experts. TensorFlow can also be incorporated with Keras to perform deep learning experiments

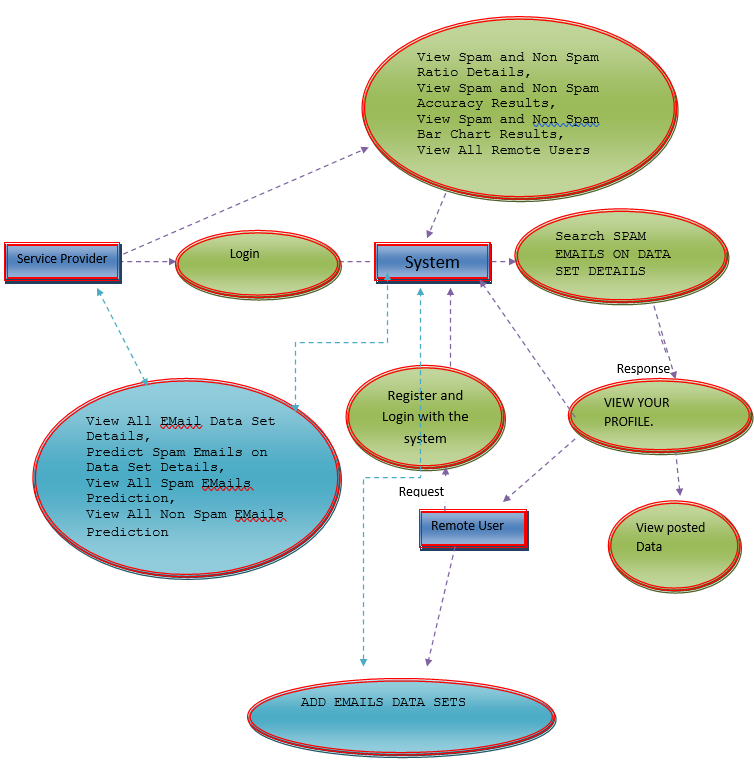
**PYTHON PLATFORMS**

Research was conducted into the different platforms that could be used for ML program implementation in Python.

**JUPYTER NOTEBOOK**

This is an open source tool that provides a Python framework. This is similar to ‘Spyder’ IDE, except this tool lets a user run the source code via a web browser [18]. Anaconda framework also offers ‘Jupyter’ to be utilised by the user through the local server. F. ONLINE PLATFORMS Along with the desktop-based platforms, other online platforms that offers additional support are: Google Collaboratory and Kaggle. Both platforms are the top ML and DL based that also offers TPU (Tensor Processing Unit) along with CPU and GPU. Multiple core servers can also be accessed. The platforms are cloud-based, and the user’s program is run until the ‘Runtime’ is ended.

**3.2 METHODOLOGY**

****

**Figure 2: Data Flow View**

**3.3 SPECIFICATIONS**

#### **SYSTEM CONFIGURATION:**

**HARDWARE:**

* System : Pentium IV 2.4 GHz.
* Hard Disk : 40 GB.
* Monitor : 15 VGA Colour.
* Mouse : Logitech.
* Ram : 512 Mb

#### **SOFTWARE:**

* Operating system : Windows XP/ Windows 7 or More
* Software Tool : Open CV.
* Coding Language : Python.
* Toolbox : Image processing toolbox.

#### **Python IDE**

Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python’s elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms. The Python interpreter is easily extended with new functions and data types implemented in C or C++ (or other languages callable from C). Python is also suitable as an extension language for customizable applications.

#### **OpenCV**

OpenCV is a library of programming functions mainly aimed at real-time computer vision. It has a modular structure, which means that the package includes several shared or static libraries. We are using image processing module that includes linear and non-linear image filtering, geometrical image transformations (resize, affine and perspective warping, and generic table-based remapping), color space conversion, histograms, and so on. Our project includes libraries such as Viola-Jones or Haar classifier, LBPH (Lower Binary Pattern histogram) face recognizer, Histogram of oriented gradients (HOG).

* + 1. **OpenCV-Python**

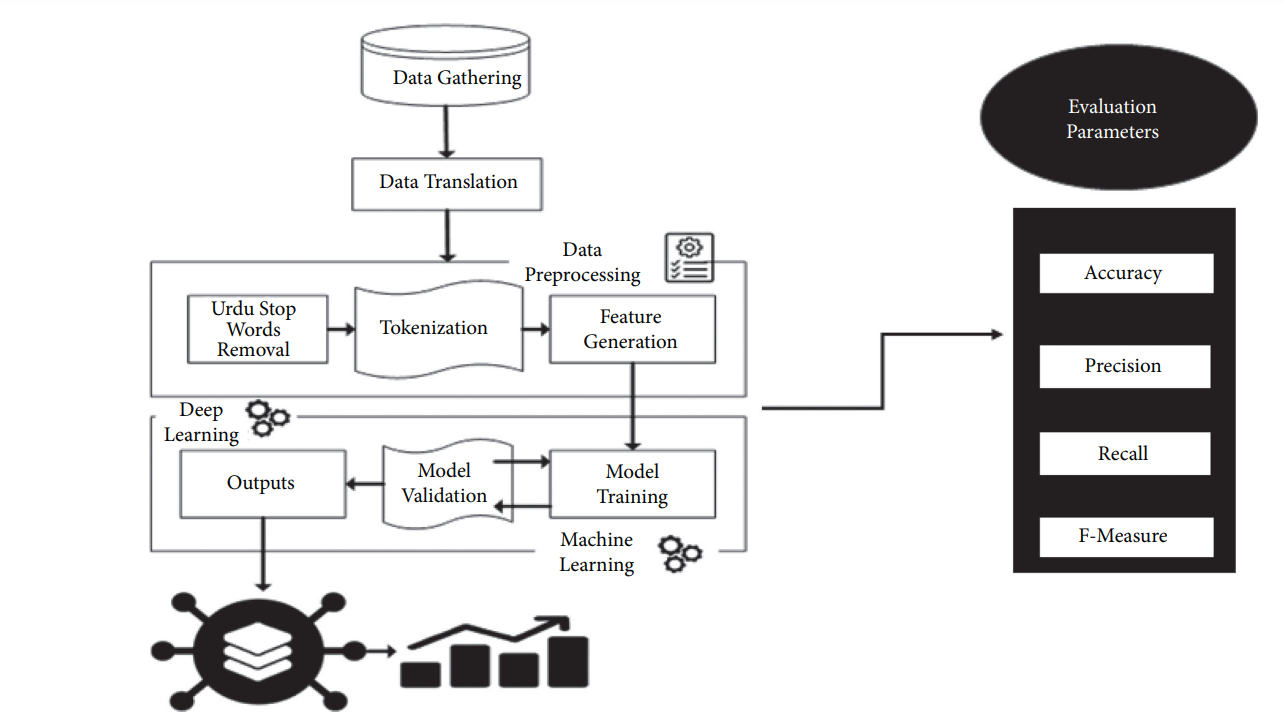
Python is a general purpose programming language started by Guido van Rossum, which became very popular in short time mainly because of its simplicity and code readability. It enables the programmer to express his ideas in fewer lines of code without reducing any readability.Compared to other languages like C/C++, Python is slower. But another important feature of Python is that it can be easily extended with C/C++. This feature helps us to write computationally intensive codes in C/C++ and create a Python wrapper for it so that we can use these wrappers as Python modules. This gives us two advantages: first, our code is as fast as original C/C++ code (since it is the actual C++ code working in background) and second, it is very easy to code in Python. This is how OpenCV-Python works, it is a Python wrapper around original C++ implementation. And the support of Numpy makes the task more easier. Numpy is a highly optimized library for numerical operations. It gives a MATLAB- style syntax. All the OpenCV array structures are converted to-and-from Numpy arrays. So whatever operations you can do in Numpy, you can combine it with OpenCV, which increases number of weapons in your arsenal. Besides that, several other libraries like SciPy, Matplotlib which supports Numpy can be used with this. So OpenCV-Python is an appropriate tool for fast prototyping of computer vision problems.

**3.3.4 PYTHON**

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

* **Python is Interpreted:** Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
* **Python is Interactive**: You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
* **Python is Object-Oriented**: Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
* **Python is a Beginner's Language**: Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

**3.4 SYSTEM DESIGN**

****

**Figure 3: Working Procedure**

Step 1. Pick a random mail from the collection for testing purposes.

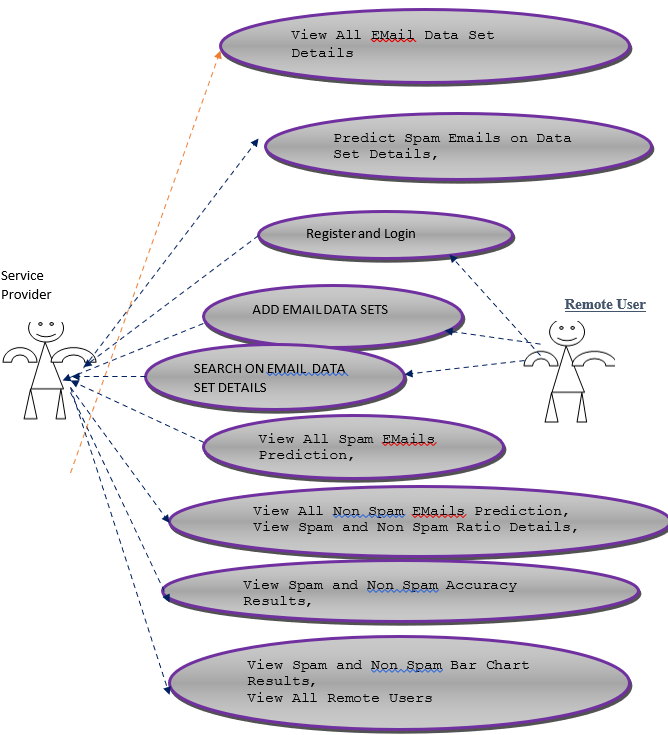
Step 2. The e-mail in question is in its unprocessed state. E-mail must be preprocessed before the feature extraction and classification procedure can begin. Tokenization,

Step3. To use the feature extraction technique, select suitable attribute words from the validation set. Just the set of features that is most nearly connected to the category is selected.

Step4. Use extracted features and created tokens to train ML and DL models. 1at model can easily distinguish between spam and ham Emails.

Step5. Tokens are classified as spam or ham based on their feature similarity as ML models determines.

**USE CASE**



**Figure 4: Use case diagram**

**CHAPTER 4**

# **IMPLEMENTATION**

* 1. **MODULES**
     1. **SERVICE PROVIDER**

In this module, the Service Provider has to login by using valid user name and password. After login successful he can do some operations such as View All Email Data Set Details, Predict Spam Emails on Data Set Details, View All Spam Emails Prediction, View All Non Spam Emails Prediction, View Spam and Non Spam Ratio Details, View Spam and Non Spam Accuracy Results, View Spam and Non Spam Bar Chart Results, View All Remote Users.

* + 1. **VIEW AND AUTHORIZE USERS**

In this module, the admin can view the list of users who all registered. In this, the admin can view the user’s details such as User Name, Email, Address And Admin Authorizes The Users.

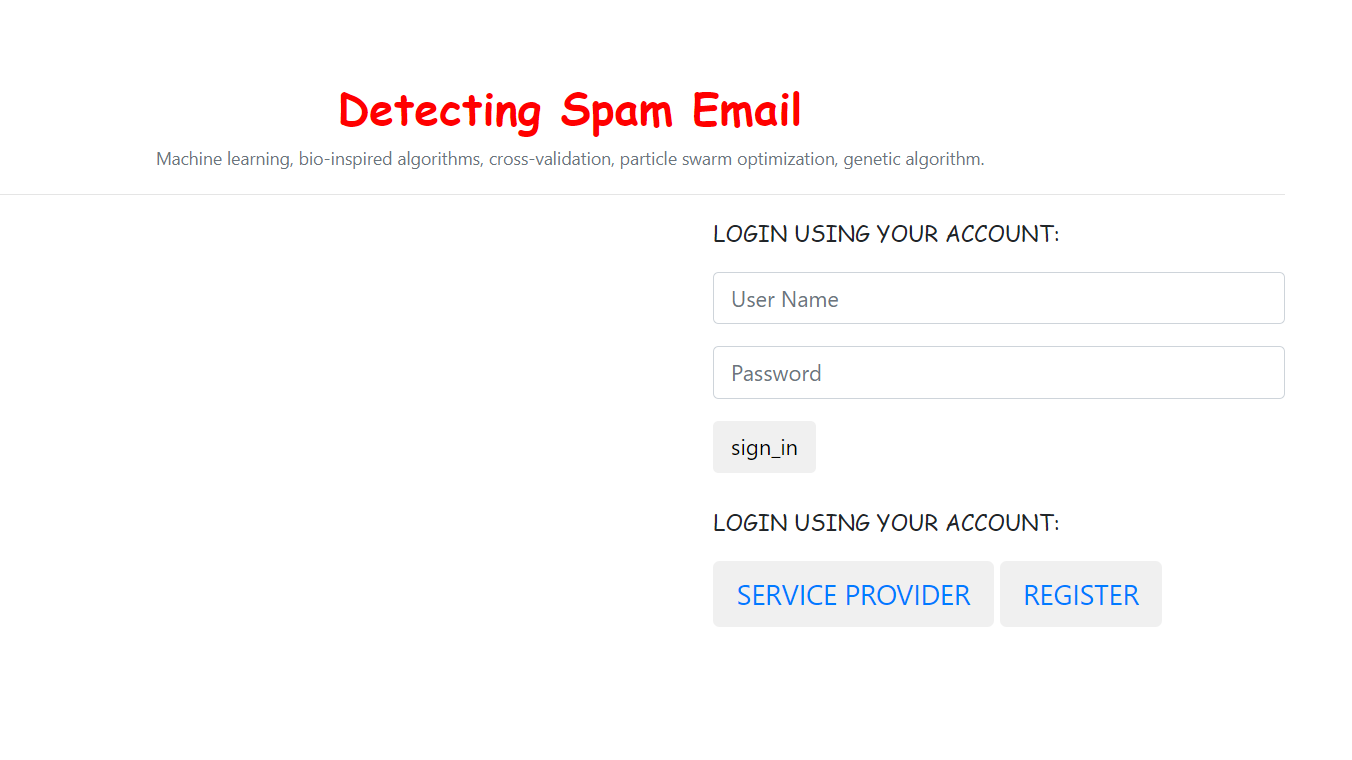
* + 1. **REMOTE USERS**

In this module, there are n numbers of users are present. User should register before doing any operations. Once user registers, their details will be stored to the database. After registration successful, he has to login by using authorized user name and password. Once login is successful user will do some operations Like Register and Login, Add Emails Data Sets, Search Spam Emails On Data Set Details, View Your Profile.

* 1. **TEST CASES**

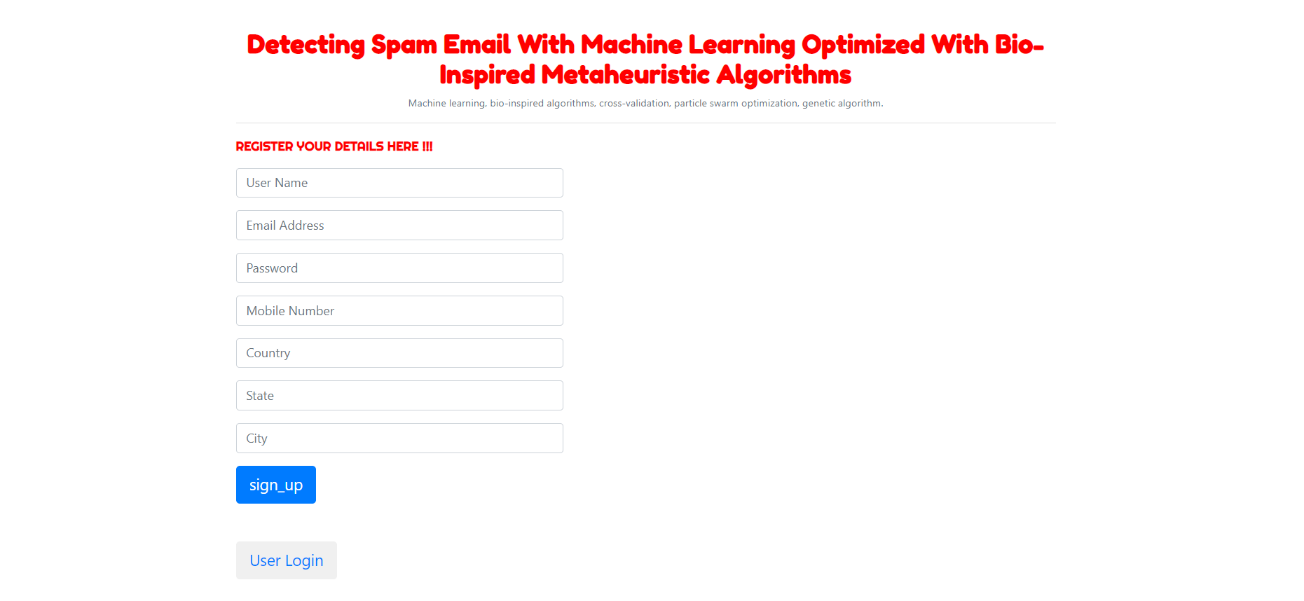
|  |  |  |  |
| --- | --- | --- | --- |
| **S. No** | **Action** | **Input** | **Output** |
| 1 | Login page | SERVICE PROVIDER, REMOTE USER.  Requirements: Username, password. | Login to particular welcome pages of Service provider, Remote user. |
| 2 | Add Remote user | User name, Email address, password, mobile number, country, state, city. | Remote user is successfully added to the database. |
| 3 | Service provider | Enter the username and password to login successfully. | View All Email Data Set Details, Predict Spam Emails on Data Set Details, View All Spam Emails Prediction, View All Non Spam Emails Prediction, View Spam and Non Spam Ratio Details, View Spam and Non Spam Accuracy Results, View Spam and Non Spam Bar Chart Results, View All Remote Users, Test Email, Bio graph. |
| 4 | Remote user | Enter the remote user credentials | Register And Login, Add Emails Data Sets, Search Spam Emails On Data Set Details, View Your Profile. |
| 5 | View And Authorize Users | View All Remote Users | User’s details such as, user name, Email, mobile number, country, state, city. |
| 6 | Predict Spam Emails on Data Set Details | Enter the keyword | Display the Email type. |
| 7 | Test Email | Enter the message | Detection of Spam and Non spam Emails. |
| 8 | View Spam and NonSpam Ratio Details |  | Ratio details of spam and non-spam Emails in percentage. |
| 9 | Bio graph |  | Graphical representation of Bio-inspired and machine learning algorithm accuracy. |
| 10 | Log out |  | Exit from the login pages. |

**4.3 USER INTERFACE**

****

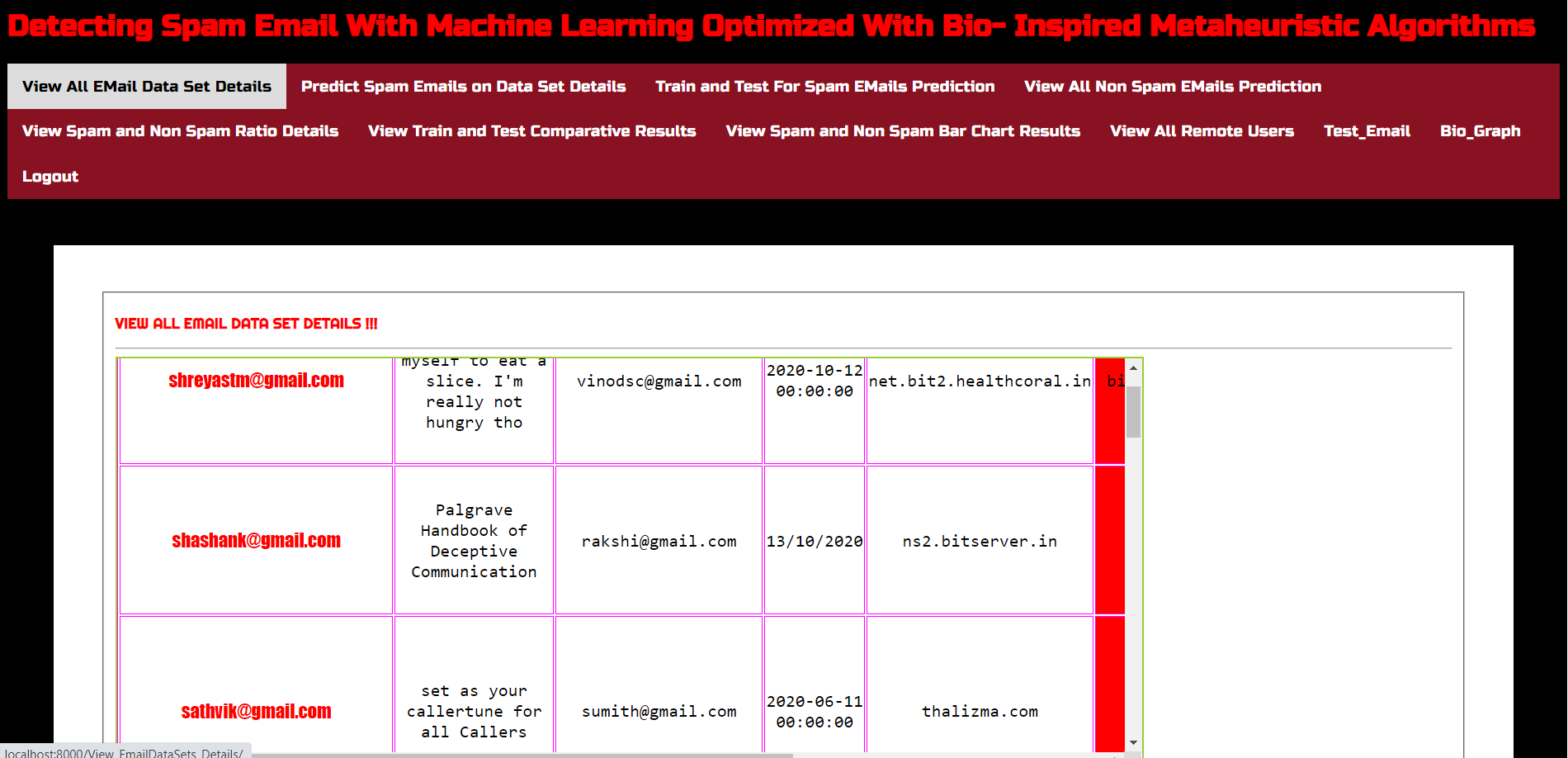
**Figure 5: Login Page to access account**

The Login page consists of two login options one as Service provider and the other as Remote user. The remote user can login using the valid login credentials such as username and password. If their account has not been setup then they have to create an account via register.



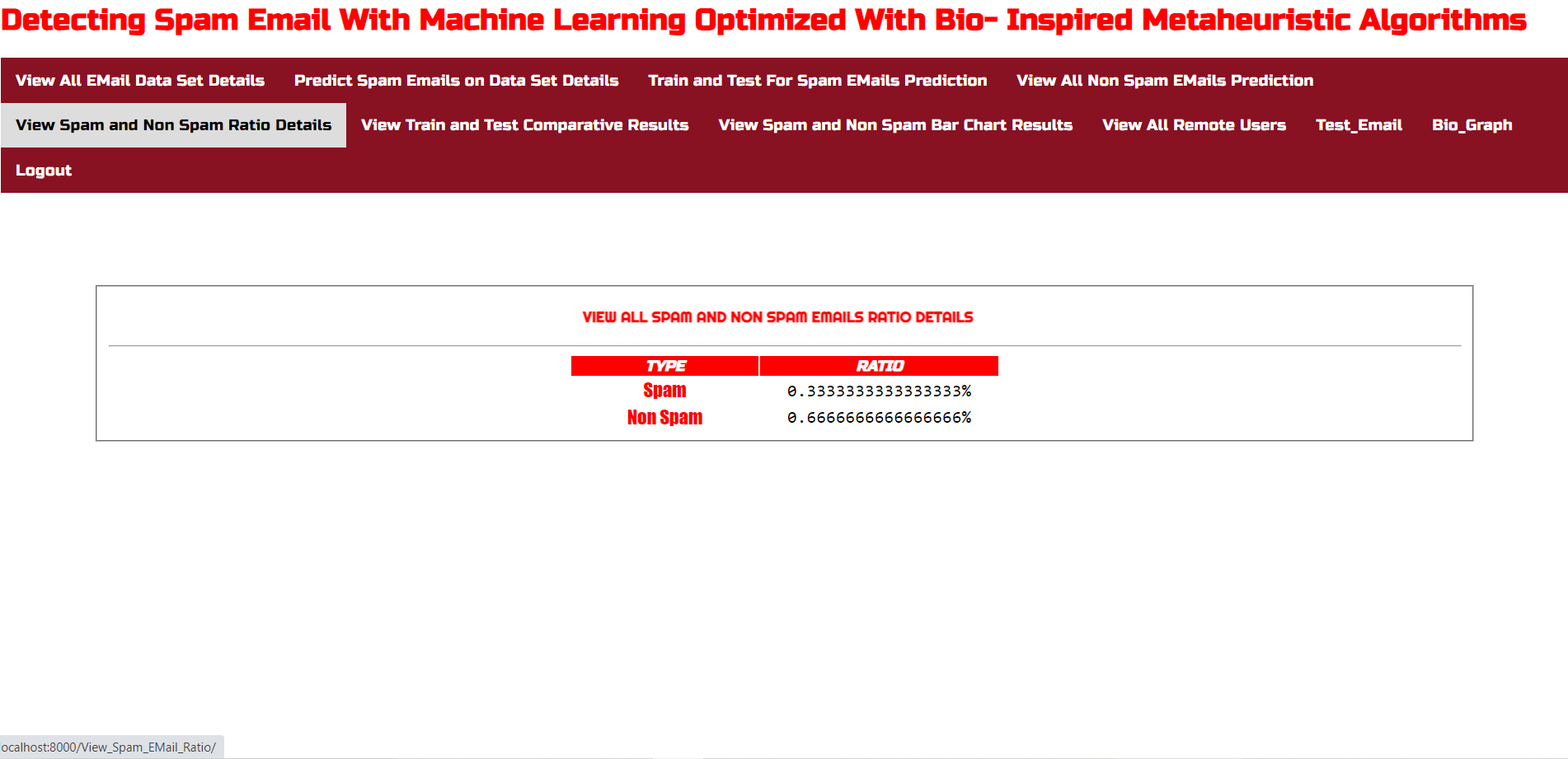
**Figure 6: Register page for Remote Users**

Necessary details required to register are as follows: User name, Email address, password, mobile number, country, state, city. After filling the details signup will redirect the user to login page.



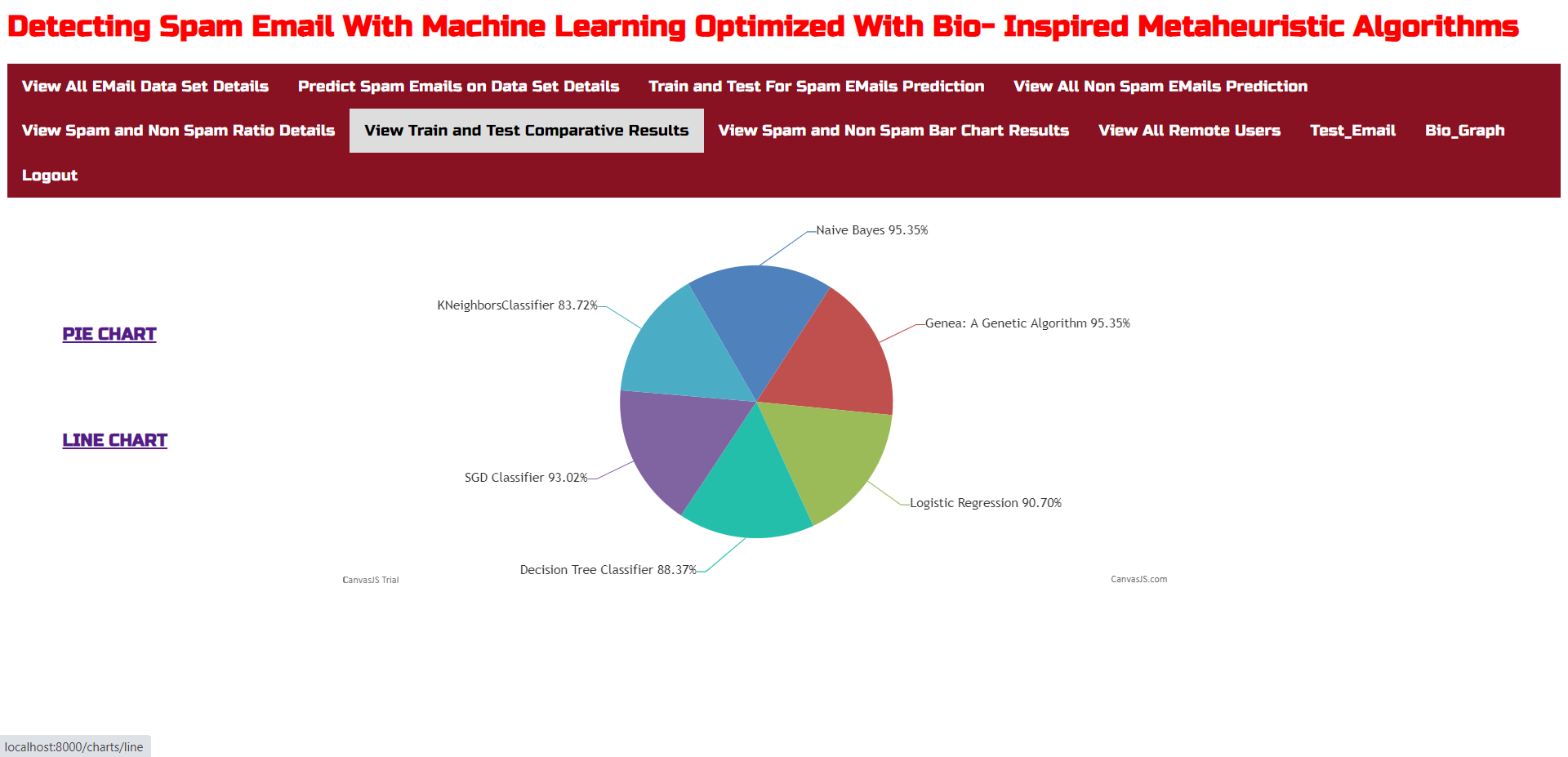
**Figure 7: Service Provider View**

The Service Provider has to login by using valid user name and password. After successful login service provider can view all the Email data set details. The Email data set will be containing the details such as from address, to address, subject, Email date and the contents.



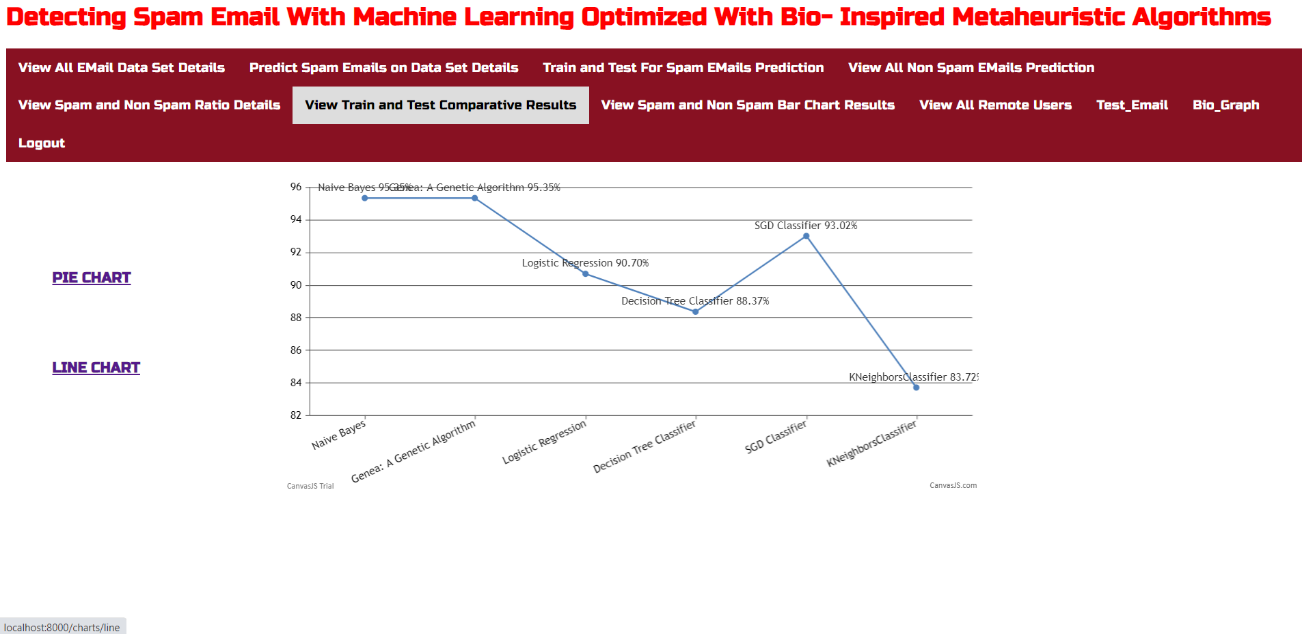
**Figure 8: Estimation of Spam and Non-Spam Emails**

After the training and testing for spam and non-spam Emails they are classified separately and the ratio details of spam and non-spam Emails are displayed in percentage.



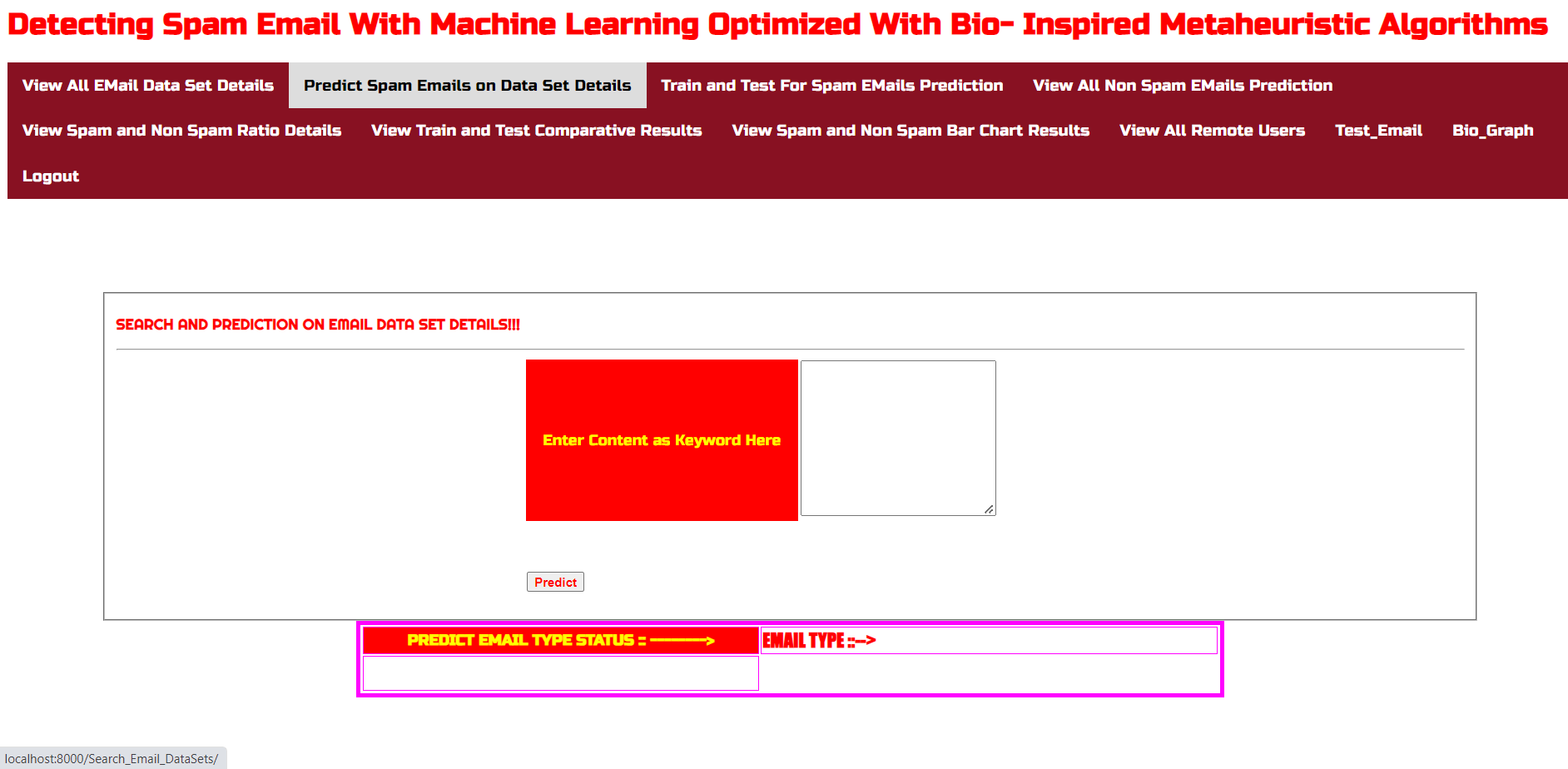
**Figure 9: Representation of Accuracy of Algorithms in Detecting Spam Emails**

The comparision of training and testing results of Naïve bayes, Genetic, Logistic regression, Decision tree classifier, SGD classifier and KNN classifier algorithms in terms of accuracy is represented using pie chart in the above figure.



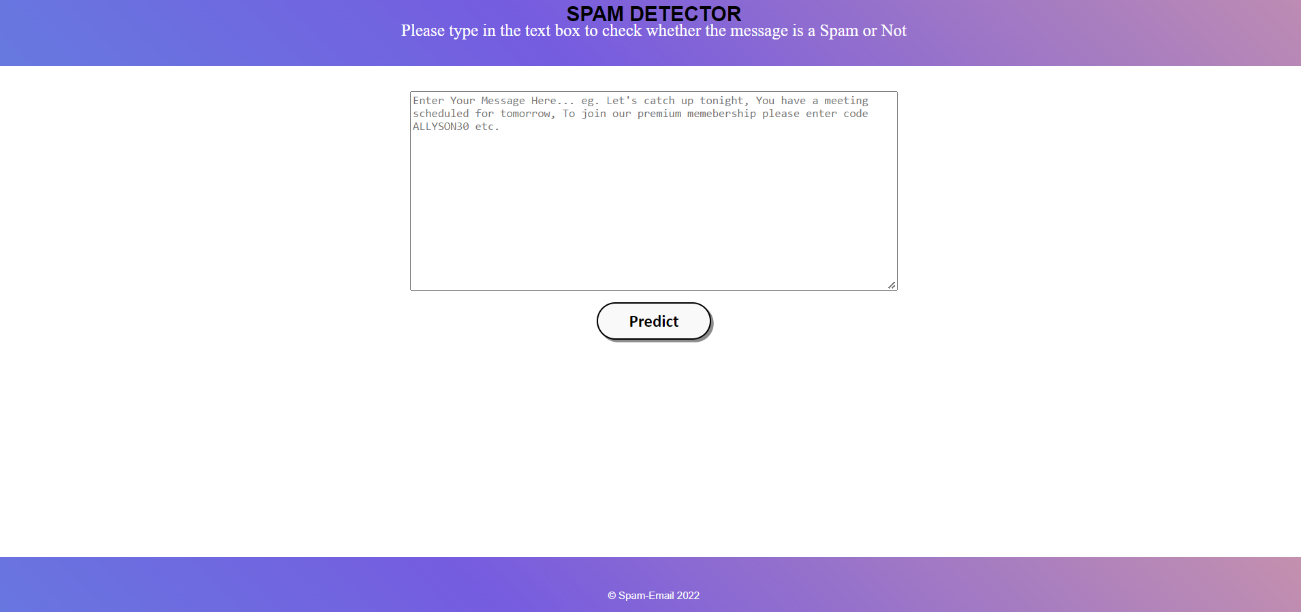
**Figure 10: Graphical representation of accuracy (line graph)**

The results of accuracy comparision of the algorithms are represented using line chart in the above figure.



**Figure 11: Interface to predict Email type using keyword**

On the basis of the keyword entered the Email type status will be predicted as spam or not. Spam Emails are predicted by keywords using the feature extraction and stop words.



**Figure 12: Interface to Predict the Spam Emails**

To check whether the message is spam or not enter the message (contents present in an Email) in the text box and prediction will be done and the result will be displayed as the outcomes.

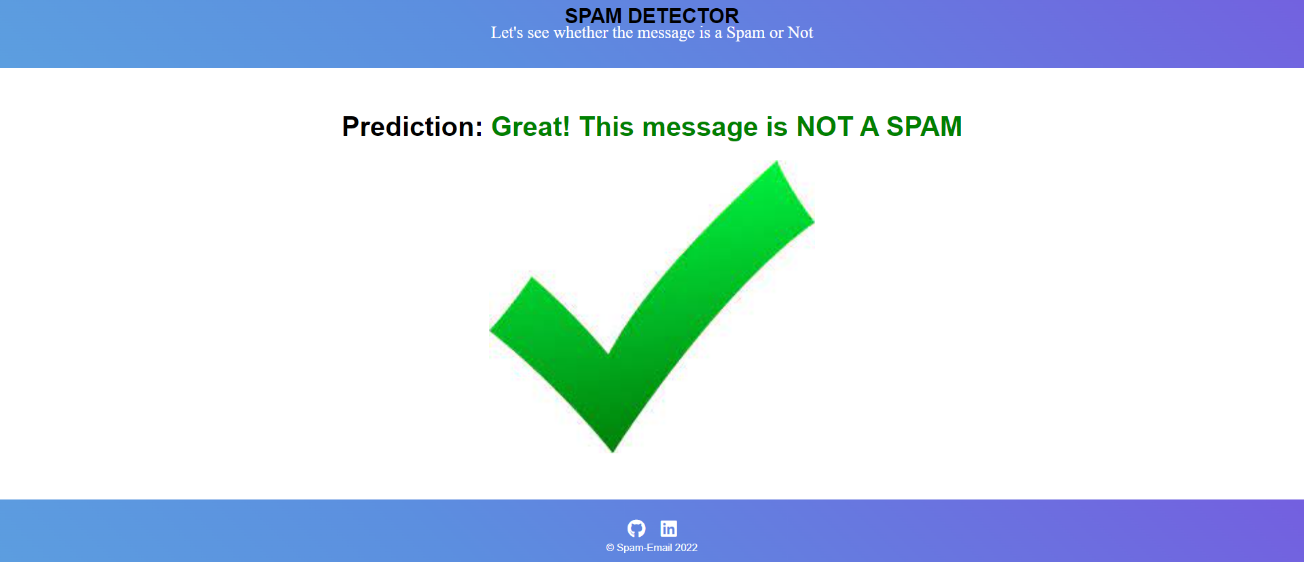
**4.4 OUTCOMES**

Detection of Spam and Non spam Emails.

****

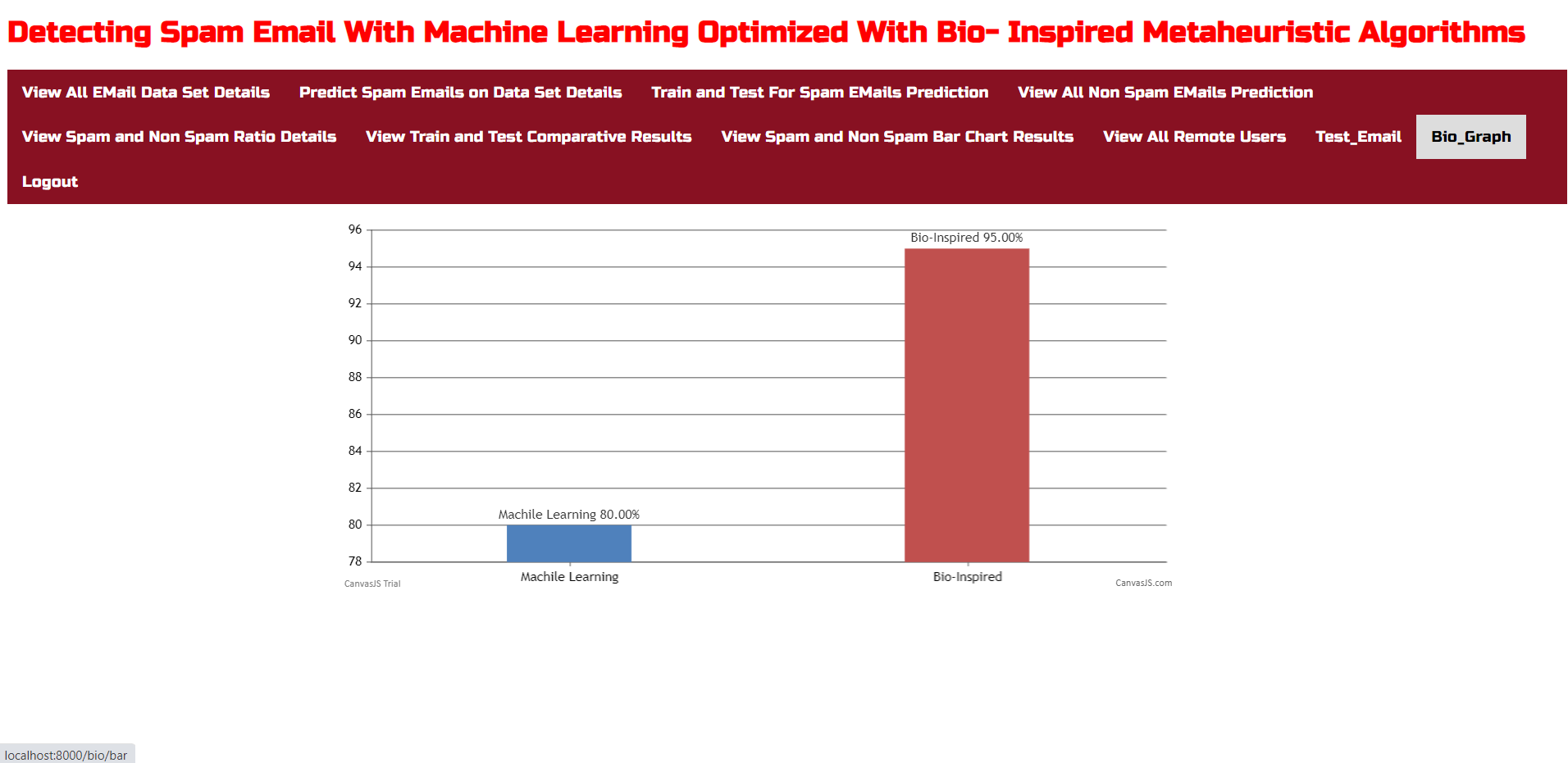
**Figure 13: Representing Spam Email is detected**

After the prediction if the message is a spam then a message will pop-up as SPAM as the above figure.



**Figure 14: Representing Non Spam Email is detected**

If it is a legitimate message then the message will pop-up as NOT a Spam as the above figure.



**Figure 15: Bar graph representation of accuracy of Bio-inspired and Machine learning algorithm.**

As the accuracy of machine learning algorithm is 80.00% and bio-inspired algorithm is 95.00% both are represented as bar graph in the above figure. So it can be concluded as the accuracy of bio-inspired algorithm is high compared to machine learning algorithm.

**4.5 SYSTEM TESTING**

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

**SYSTEM TESTING**

### TESTING METHODOLOGIES:

* Unit Testing.
* Integration Testing.
* User Acceptance Testing.
* Output Testing.
* Validation Testing.

**UNIT TESTING**

Unit testing focuses verification effort on the smallest unit of Software design that is the module. Unit testing exercises specific paths in a module’s control structure to ensure complete coverage and maximum error detection. This test focuses on each module individually, ensuring that it functions properly as a unit. Hence, the naming is Unit Testing.

During this testing, each module is tested individually and the module interfaces are verified for the consistency with design specification. All important processing path are tested for the expected results. All error handling paths are also tested.

**INTEGRATION TESTING**

Integration testing addresses the issues associated with the dual problems of verification and program construction. After the software has been integrated a set of high order tests are conducted. The main objective in this testing process is to take unit tested modules and builds a program structure that has been dictated by design.

**THE FOLLOWING ARE THE TYPES OF INTEGRATION TESTING:**

1. **TOP DOWN INTEGRATION**

This method is an incremental approach to the construction of program structure. Modules are integrated by moving downward through the control hierarchy, beginning with the main program module. The module subordinates to the main program module are incorporated into the structure in either a depth first or breadth first manner.

In this method, the software is tested from main module and individual stubs are replaced when the test proceeds downwards.

**2. BOTTOM-UP INTEGRATION**

This method begins the construction and testing with the modules at the lowest level in the program structure. Since the modules are integrated from the bottom up, processing required for modules subordinate to a given level is always available and the need for stubs is eliminated.

The bottom up integration strategy may be implemented with the following steps:

* The low-level modules are combined into clusters into clusters that perform a specific Software sub-function.
* A driver (i.e.) the control program for testing is written to coordinate test case input and output.
* The cluster is tested.
* Drivers are removed and clusters are combined moving upward in the program structure

The bottom up approaches tests each module individually and then each module is module is integrated with a main module and tested for functionality.

**OTHER TESTING METHODOLOGIES**

**USER ACCEPTANCE TESTING**

User Acceptance of a system is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes wherever required. The system developed provides a friendly user interface that can easily be understood even by a person who is new to the system.

**OUTPUT TESTING**

After performing the validation testing, the next step is output testing of the proposed system, since no system could be useful if it does not produce the required output in the specified format. Asking the users about the format required by them tests the outputs generated or displayed by the system under consideration. Hence the output format is considered in 2 ways – one is on screen and another in printed format.

**VALIDATION CHECKING**

Validation checks are performed on the following fields.

* Text Field
* Number field

**TEXT FIELD:**

The text field can contain only the number of characters lesser than or equal to its size. The text fields are alphanumeric in some tables and alphabetic in other tables. Incorrect entry always flashes and error message.

**NUMERIC FIELD:**

The numeric field can contain only numbers from 0 to 9. An entry of any character flashes an error messages. The individual modules are checked for accuracy and what it has to perform. Each module is subjected to test run along with sample data. The individually tested modules are integrated into a single system. Testing involves executing the real data information is used in the program the existence of any program defect is inferred from the output. The testing should be planned so that all the requirements are individually tested. A successful test is one that gives out the defects for the inappropriate data and produces and output revealing the errors in the system.

**PREPARATION OF TEST DATA**

Taking various kinds of test data does the above testing. Preparation of test data plays a vital role in the system testing. After preparing the test data the system under study is tested using that test data. While testing the system by using test data errors are again uncovered and corrected by using above testing steps and corrections are also noted for future use.

**USING LIVE TEST DATA:**

Live test data are those that are actually extracted from organization files. After a system is partially constructed, programmers or analysts often ask users to key in a set of data from their normal activities. Then, the systems person uses this data as a way to partially test the system. In other instances, programmers or analysts extract a set of live data from the files and have them entered themselves.

It is difficult to obtain live data in sufficient amounts to conduct extensive testing. And, although it is realistic data that will show how the system will perform for the typical processing requirement, assuming that the live data entered are in fact typical, such data generally will not test all combinations or formats that can enter the system. This bias toward typical values then does not provide a true systems test and in fact ignores the cases most likely to cause system failure.

**USING ARTIFICIAL TEST DATA:**

Artificial test data are created solely for test purposes, since they can be generated to test all combinations of formats and values. In other words, the artificial data, which can quickly be prepared by a data generating utility program in the information systems department, make possible the testing of all login and control paths through the program.

The most effective test programs use artificial test data generated by persons other than those who wrote the programs. Often, an independent team of testers formulates a testing plan, using the systems specifications.The package “Virtual Private Network” has satisfied all the requirements specified as per software requirement specification and was accepted.

**USER TRAINING**

Whenever a new system is developed, user training is required to educate them about the working of the system so that it can be put to efficient use by those for whom the system has been primarily designed. For this purpose the normal working of the project was demonstrated to the prospective users. Its working is easily understandable and since the expected users are people who have good knowledge of computers, the use of this system is very easy.

**MAINTAINENCE**

This covers a wide range of activities including correcting code and design errors. To reduce the need for maintenance in the long run, we have more accurately defined the user’s requirements during the process of system development. Depending on the requirements, this system has been developed to satisfy the needs to the largest possible extent. With development in technology, it may be possible to add many more features based on the requirements in future. The coding and designing is simple and easy to understand which will make maintenance easier.

**TESTING STRATEGY :**

A strategy for system testing integrates system test cases and design techniques into a well planned series of steps that results in the successful construction of software. The testing strategy must co-operate test planning, test case design, test execution, and the resultant data collection and evaluation .A strategy for software testing must accommodate low-level tests that are necessary to verify that a small source code segment has been correctly implemented as well as high level tests that validate major system functions against user requirements.

Software testing is a critical element of software quality assurance and represents the ultimate review of specification design and coding. Testing represents an interesting anomaly for the software. Thus, a series of testing are performed for the proposed system before the system is ready for user acceptance testing.

**SYSTEM TESTING:**

Software once validated must be combined with other system elements (e.g. Hardware, people, database). System testing verifies that all the elements are proper and that overall system function performance is achieved. It also tests to find discrepancies between the system and its original objective, current specifications and system documentation.

**UNIT TESTING:**

In unit testing different are modules are tested against the specifications produced during the design for the modules. Unit testing is essential for verification of the code produced during the coding phase, and hence the goals to test the internal logic of the modules. Using the detailed design description as a guide, important Conrail paths are tested to uncover errors within the boundary of the modules. This testing is carried out during the programming stage itself. In this type of testing step, each module was found to be working satisfactorily as regards to the expected output from the module.

In Due Course, latest technology advancements will be taken into consideration. As part of technical build-up many components of the networking system will be generic in nature so that future projects can either use or interact with this.The future holds a lot to offer to the development and refinement of this project.

**CHAPTER 5**

# **CONCLUSION AND FUTURE SCOPE**

**5.1 CONCLUSION AND FUTURE SCOPE**

The project successfully implemented models combined with bio-inspired algorithms. The spam Email corpus used within the project were both numerical as well as alphabetical. Approximately 5,000 Emails were tested with the proposed models. Initially, WEKA acted as a black box that ran the datasets on different classification algorithms and provided the top algorithms: Multinomial Naïve Bayes, Logistic regression and Decision Tree. These algorithms were then tested and experimented with Scikit-learns library and its modules. Genetic algorithm (GA) was implemented using Python and experimented with feature extraction and stop words removal along with converting the tokens for the algorithms to process. GA worked better overall for both text-based datasets and numerical-based datasets than other algorithms. The combination of GA and Multinomial Naïve Bayes worked well and was proved to have been the best accuracy for spam detection. This was concluded by evaluating the results for both numerical and alphabetical based dataset. The highest accuracy provided was 98% with GA optimization on randomised data distribution for 80:20 train and test split set on Spam Email dataset.

We plan to further carry out the machine learning algorithms to optimize and compare with different bio-inspired algorithms such as Firefly, Bee Colony and Ant Colony Optimization as researched in the previous sections. We could also explore the Deep learning Neural Network with PSO and GA by exploring different libraries such as TensorFlow’s DNN Classifier or similar. We found that the Neural Network algorithm could have worked better with more dimension like providing broader range of values for learning rate, activation, solver, and alpha. If this project is taken further, implementation for MLP could be done through Keras or TensorFlow with GPU application. This will allow the user to input other parameters and a range of possibilities as their key values.

* 1. **REFERENCES**

[1] W. Feng, J. Sun, L. Zhang, C. Cao, and Q. Yang, ‘‘A support vector machine based Naive Bayes algorithm for spam filtering,’’ in Proc. IEEE 35th Int. Perform. Comput. Commun. Conf. (IPCCC), Dec. 2016, pp. 1–8, doi: 10.1109/pccc.2016.7820655.

[2] E. G. Dada, J. S. Bassi, H. Chiroma, S. M. Abdulhamid, A. O. Adetunmbi, and O. E. Ajibuwa, ‘‘Machine learning for Email spam filtering: Review, approaches and open research problems,’’ Heliyon, vol. 5, no. 6, Jun. 2019, Art. no. e01802, doi: 10.1016/j.heliyon.2019.e01802.

[3] W. Awad and S. ELseuofi, ‘‘Machine learning methods for spam E-Mail classification,’’ Int. J. Comput. Sci. Inf. Technol., vol. 3, no. 1, pp. 173–184, Feb. 2011, doi: 10.5121/ijcsit.2011.3112.

[4] S. Mohammed, O. Mohammed, and J. Fiaidhi, ‘‘Classifying unsolicited bulk Email (UBE) using Python machine learning techniques,’’ Int. J. Hybrid Inf. Technol., vol. 6, no.1,pp.4355,2013.[Online].Available:https://www.researchgate.net/publication/236970412\_Classifying\_Unsolicited\_Bulk\_Email\_UBE\_using\_Python\_Machine\_Learning\_ Techniques.

[5] A. Wijaya and A. Bisri, ‘‘Hybrid decision tree and logistic regression classifier for Email spam detection,’’ in Proc. 8th Int. Conf. Inf. Technol. Electr. Eng