Phishing Emails: Analysis and Detection with Comparison of Three Machine Learning Models (LR, NB and MLP)

Theophilus Bamise Ajala

Department of Computer Science, Caleb University, Imota, Lagos, Nigeria

DOI: https://doi.org/10.51583/IJLTEMAS.2025.1411000044

Received: 10 November 2025; Accepted: 20 November 2025; Published: 08 December 2025

ABSTRACT

Phishing attacks are performed by writing and forwarding falsified body of email messages which look legitimate or real from an undisputed origin to a victim or different category of victims. They focus at acquiring the sensitive data of users or by transferring and loading malware on the user's computers. Consequently, this study aims to implement an AI-driven approach to detect emails that seems to be phishing while the features are analyzed. This project leverage three machine learning models namely: MLP, a deep learning algorithm, Naïve Bayes and Logistic Regression. The following performance metrics were obtained -> for Multilayer Perceptron (MLP) model: accuracy: 98.57%, precision: 100.00%, recall: 90.00% while the f1_score metrics was 94.74%. For Naïve Bayes (NB) model: accuracy: 96.95%, precision: 100.00%, recall: 78.75% while the f1_score metrics was 88.11%. For Logistic Regression (LR) model: accuracy: 94.71%, precision: 99.03%, recall: 63.75% while the f1_score metrics was 77.57%. The result shows that MLP Classifier may better capture complex patterns in phishing emails, leading to higher detection rates. Naive Bayes is still a strong choice, especially for simpler or smaller datasets due to its speed and efficiency. Logistic Regression is reliable but slightly less accurate on this particular task. For this project, a phishing email dataset from the Kaggle Machine Learning Repository was utilized. This dataset contains 5000+ instances of phishing and ham emails.

Keywords: Multilayer perceptron Neural Network (MLP), Naïve Bayes (NB), Logistic Regression (LR), Deep Learning, AI, Phishing Email

How to cite: Ajala, T.B. (2025). Phishing Emails: Analysis and Detection with Comparison of Three Machine Learning Models (LR, NB and MLP).

INTRODUCTION

In the year 2021, there is a record of over seven billion registered email accounts in the world and people send more than three million emails per second, for transactions relating to professional and personal matters, emails services is a vital tool to handle such matters in a smooth and stress-free manner. Howbeit, attackers have seized the opportunity to employ the mammoth use of emails services to launch their prosperous and growing attacks. It is nearly impossible for an email account to be compromised due to the End to End (E2E) encrypting strategies integrated into email services by email service provider. Based on the challenge of the foregoing for attackers, attackers, decide to employ social engineering tactics manipulate email accounts by indulging the wisdom of humans to get sensitive and critical information (Salahdine & Kaabouch, 2019).

Phishing attacks are performed by writing and forwarding falsified body of email messages which look legitimate or real from an undisputed origin to a victim or different category of victims (Mohammad, et. al., 2014). They focus at acquiring the sensitive data of users or by transferring and loading malware on the user's computers. For example, the attackers forward an email with a link that will redirect the user to a website that contains malicious content, by so doing the user is asked to proffer some confidential information such as password, login detains, bank account details such as cvv/cvc, expiry date etc. The attacker can equally include a file to the forged email to be loaded by the victim when they click it, which can make the unseen malware attached to the file to be executed. Phishing is a specific form of cybercrime that permits offenders to scheme users and abscond with their sensitive data. Victims of phishing attacks can suffer noteworthy losses and forfeiture of their identity, sensitive information, merchandise and profession (Dinesh, et. al., 2023).

In the year 2006, hackers uses emails to set baits for people to snatch their user and password credentials in America. Since that time, the strategies of phishing has advance, which makes it harder to recognize legitimate and fraudulent emails (Rawal et. al., 2017). In other to solve this evolving phishing issues, there is a need to employ machine learning to help detect phishing attacks.

The field of Artificial Intelligence has evolved lately, indicated by the introduction of deep learning and machine learning. The advancements in these technologies have gathered fame due to the fact that they can dissect complex datasets, formulate models and provide insights that cannot be achieved before (Rezazadeh, 2025). Machine learning can be described as a model that handles the construction of mathematical and analytical-based models automatically. It gives room for computational systems to possess knowledge and improve their performance via experience without the need to explicitly coding them for the required task autonomously. On the other hand, deep learning works as a specialized section of machine learning, which utilizes neural networks to handle critical problems (Rezazadeh, 2025). Machine learning is an AI-based application which allows machines entrance to data and enable them learn without human intervention (Yusoff, 2025). Supervised machine learning is a method where we train or teach the machine employing data that are well labeled; thus, the models keep learning, growing and improving as time goes on (Yusoff, 2025). These models are a part of machine learning model which continuously learn from the given data. When a dataset is annotated, then a column is created that is used for prediction known as the predict class label, classification is a type of data analysis used to extract and derive a model for prediction, this aspect focuses on the supervised learning algorithms (Ajala et. al., 2025). In this machine-learning based envisioned system, the detected phishing emails can be classified into two status such as phishing (spam) and ham (legitimate).

Related Work

This section gives a basic overview of some existing studies conducted on different data to detect phishing emails by other researchers:

Sambare et. al., (2024), introduced a robust and adaptable model by combining unsupervised and supervised algorithms that are machine learning based to analyse the behaviour of user and the properties of email, this enable then the detect the crafty signs in phishing emails. Sasirekha et. al., (2023) explored the strength of machine learning to spot phishing emails. The authors proposed an approach which involves extracting features from emails, this consist of the content of the message, the information header, which are altogether used to train and test the machine learning models. Rawal et. al., (2017) undertook a research to compare the efficacy of two machine learning algorithms to pinpoint phishing emails among many sample emails.

Statement of the Problem

Detecting phishing emails continues to pose significant challenges, necessitating the implementation of effective strategies to accurately distinguish between genuine and harmful emails. The internet is used by people on a daily activity leading to an increment in the number of phishing attacks by hackers. Phishing assaults continue to persist in varying forms in a sophiticated manner, which makes conventional approach measure not to be effective any longer (Fares et. al., 2024). The goal of this project is to propose an efficient machine learning model to enhance phishing emails analysis and detection, by employing the power of machine learning, this system will provide a fast analysis of phishing emails for detection.

Objectives of the Study

The specific objectives are to:

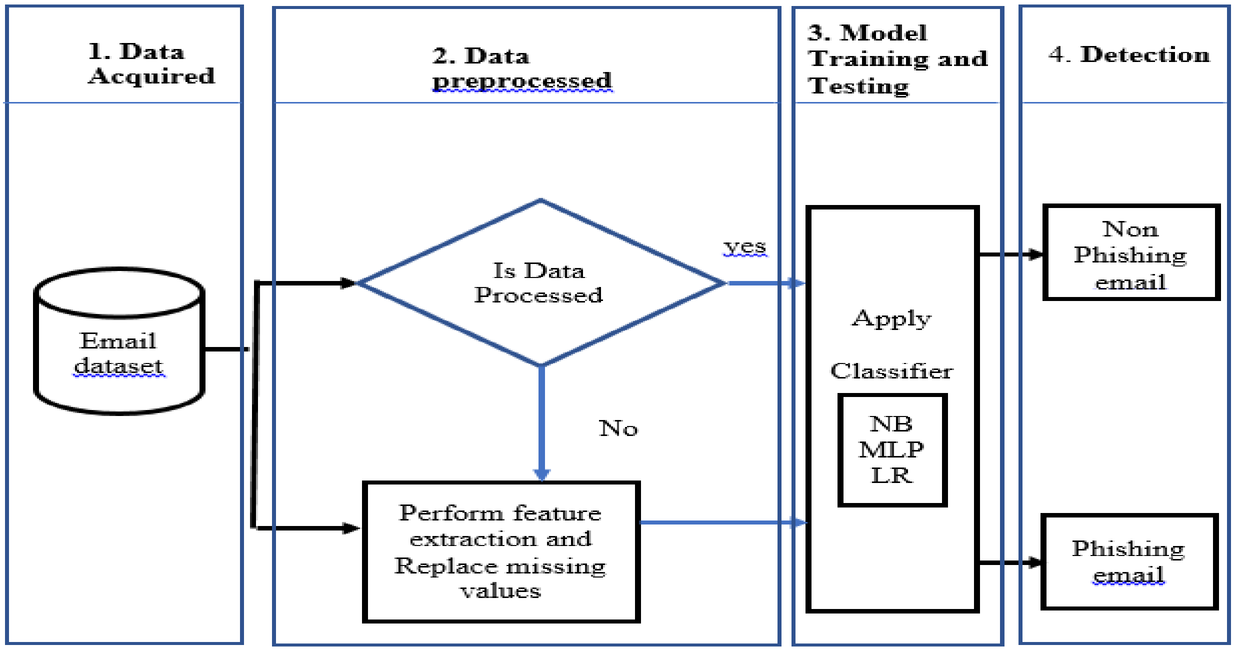
- collect dataset
- implement machine learning algorithm on (i)

- evaluate the experiment of the models on the dataset by focusing on accuracy, f1_score, precision and recall.

MATERIALS AND METHODS

This study adopted an experimental approach, which involves training three (NB, MLP and LR) supervised machine learning models on phishing emails datasets. In the data preprocessing phase, the email messages are well processed to get them ready for phishing and legitimate analysis such as extracting the body and header of the email and also doing text conversion to a format that machine learning can understand and use. For the stage that involves extracting the features, features such as the body, attachments and subject are taken from the emails in the given dataset, which is then handled by the detection algorithm. For the final phase, the email messages that are labelled are used for training and evaluating the detection models. Algorithms like NB, MLP and LR are employed to do the classification of emails and categorize them as phishing or legitimate.

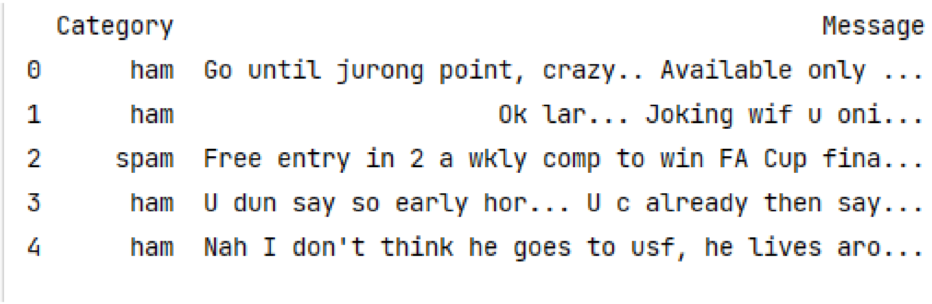
Figure 1: Flow diagram of the system (Author, 2025)



Dataset Description

The dataset have 5571 (rows) items with 2 columns, these are used for training and testing. The dataset is divided into 80% training feature sets and 20% testing feature sets for the identifcation of phishing emails to be performed. A publicly accessible spamham dataset was utilized for this project, which was obtained from Kaggle with this link (https://www.kaggle.com/datasets/uciml/sms-spam-collection-dataset).

Figure 2: Sample of the Dataset (left column) (Kaggle, 2025)



```
  Category                                            Message
0      ham  Go until jurong point, crazy.. Available only ...
1      ham                      Ok lar... Joking wif u oni...
2     spam  Free entry in 2 a wkly comp to win FA Cup fina...
3      ham  U dun say so early hor... U c already then say...
4      ham  Nah I don't think he goes to usf, he lives aro...
```

In the Figure above, The screenshot displays the first five rows of a dataset consisting of text messages annotated as either "ham" (legitimate messages) or "spam" (phishing messages). This dataset is employed for phishing email detection or classification tasks in machine learning.

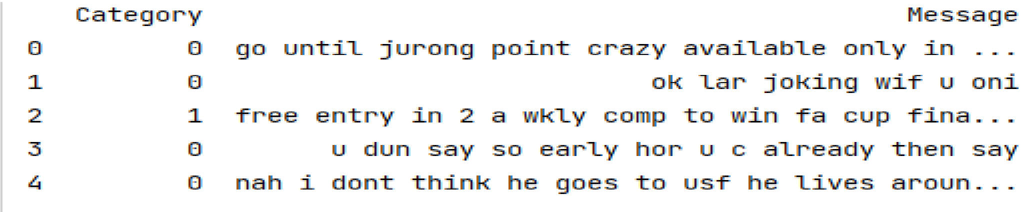
Exploratory Data Analysis (Eda)

Exploratory Data Analysis (EDA) involves the examination and visualization of data to comprehend its key attributes, distributions, and interrelations. Before the dataset can be analyzed, there is a need to transform the data into numerical form for machine learning (ML) prediction because most ML algorithms can only process numbers.

CODE SNIPPET

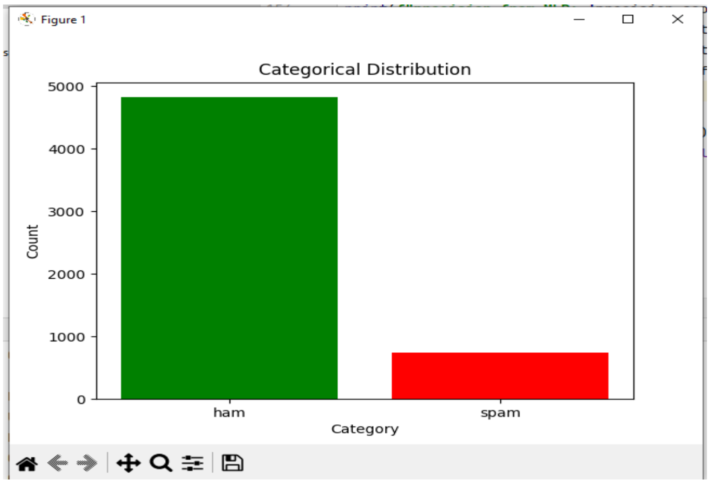
```
#Integer Encoding

#Convert The Categorical Label Into Numerical

Import Re

Import Pandas As Pd

From Sklearn.Preprocessing Import Labelencoder

#Encode Categorical Label (0= Spam/Phishing, 1= Ham/Safe

Label_Encoder = Labelencoder()

Df['Category'] = Label_Encoder.Fit_Transform(Df['Category'])

#Text Preprocessing Function

Def Preprocess_Text(Text: Str) -> Str:

   Text = Re.Sub(R"Http\S+", "", Text)       # Remove Hyperlinks

   Text = Re.Sub(R"[^\W\S]", "", Text)       # Remove Punctuation

   Text = Text.Lower()                       # Remove Hyperlinks

   Text = Re.Sub(R"\S+", "", Text).Strip()   # Remove Hyperlinks

   Return Text

#Apply Preprocessing To The 'Message' Column

Df['Message'] = Df['Message'].Astype(Str).Apply(Preprocess_Text)

#Display Preprocessed Data

Print(Df.Head())
```

Figure 3: Code Snippet for Categorical label into numerical (Author, 2025)



```
   Category                                            Message
0         0  go until jurong point crazy available only in ...
1         0                            ok lar joking wif u oni
2         1  free entry in 2 a wkly comp to win fa cup fina...
3         0        u dun say so early hor u c already then say
4         0  nah i dont think he goes to usf he lives aroun...
```

Figure 4: Screenshot of the transformed dataset using LabelEncoder (Author, 2025)



The figure above indicates that the categorical labels have been converted to numbers. This conversion is typically done using LabelEncoder, this is a function found in Scikit-learn's library.

Figure 5: Screenshot of Ham and Spam Features (Author, 2025) The chart above shows:

Ham emails (green bar): Approximately 4,800 messages Spam emails (red bar): Approximately 750 messages

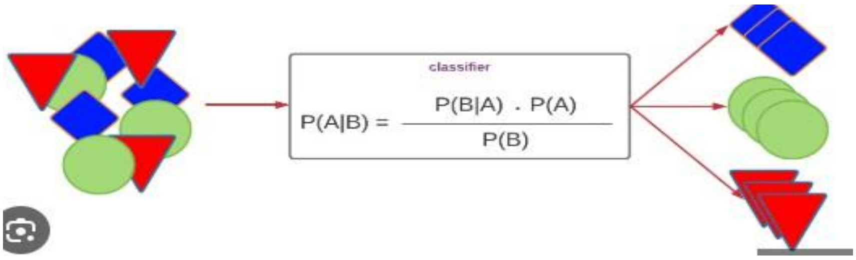
Machine Learning Models

This section briefly presents the machine learning models used for this research:

NAÏVE BAYES (NB)

According to (Alahmar et al., 2023), a Nave Bayes classifier is a classification model that uses probabilistic approach to predict by approximating the likelihood of each given class regarding the data supplied as input for capturing. A Naive Bayes classifier belongs to the category of a classifier that is probabilities oriented in machine learning, which means that it computes the probability of a data point which belongs to a class related to Bayes' theorem, it therefore makes predictions on the likelihood of an event that occurs based on a given specified features; it is regarded to be a less complex and effective classification model, and it is practically handy to handle task related to text classification because of its capability to handle vast amount of data with various features (Vikramkumar et al., 2014).

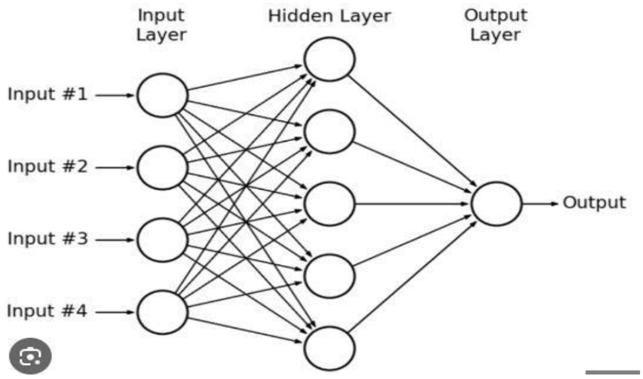
Figure 6: Navie Bayes Classifier (Elzeiny, 2024)



Multi-Layer Perceptron (MLP)

A Multilayer Perceptron (MLP) is a type of supervised, feedforward neural network classifier in machine learning, meaning it learns from labeled data and information flows in a unidirectional manner from the input to output layers, allowing it to handle complex non-linear classification tasks by utilizing multiple layers of interconnected neurons with activation functions (Rashedi et. al., 2024). MLP is one of the neural network that is often used; in this algorithm, loop is not available, the output of one neuron does not change the neuron itself; this structure is known as feed-forward (Popescu et. al., 2009).

Figure 7: Multi-layer Perceptron (https://www.futurelearn.com/)

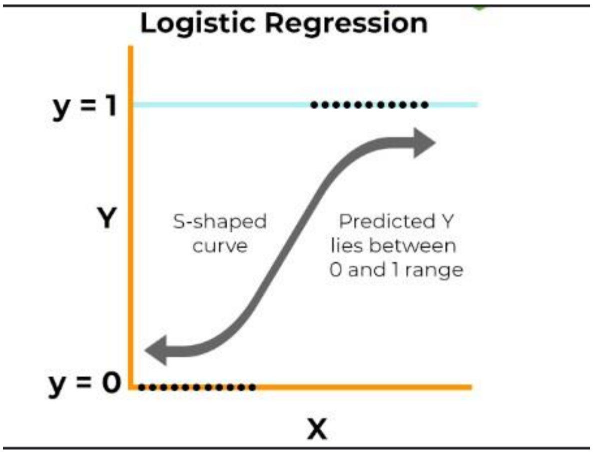


Logistic Regression (Lr)

Logistic regression is a binary classification classifier in machine learning, meaning it is primarily used to predict whether an outcome will fall into one of two categories (e.g., yes/no, spam/not spam) by calculating the probability of an event occurring based on input features; it is considered a supervised learning algorithm. Logistic regression (LR), characterized by its outcome content, is classified as a probabilistic classifier. For example, the negative class is assigned with label 0, whereas the positive class is assigned with label 1.

$$\Pi(x) = \Pi(x1, x2,\ldots,xn), \Pi \text{ € } \{0, 1\}$$

Figure 8: Logistic Regression Classifier (https://www.spiceworks.com/)



Performance Metrics

To evaluate the performance of the models, we used the following metrics:

Accuracy

Accuracy measures the percentage of correctly classified emails.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} * 100\%$$

Precision

Precision focuses on the fraction of phishing emails correctly classified as phishing out of all the samples predicted as phishing emails.

$$Precision = \frac{TP}{TP + FP} \, x \, 100$$

Recall

Recall measures the fraction of phishing emails correctly classified as phishing out of the total number of phishing emails.

$$Recall = \frac{TP}{TP + FN} \, x \, 100$$

F1 Score

F1-score is the harmonic mean of precision and recall. Both false positive and false negative results are considered in the F1 score. F1 is at its best when it is 1, and at its worst when it is 0. It indicates how accurate the classifier is.

$$F1 \, Score = \frac{2 * Recall * Precision}{Recall + Precision}$$

RESULT AND DISCUSSION

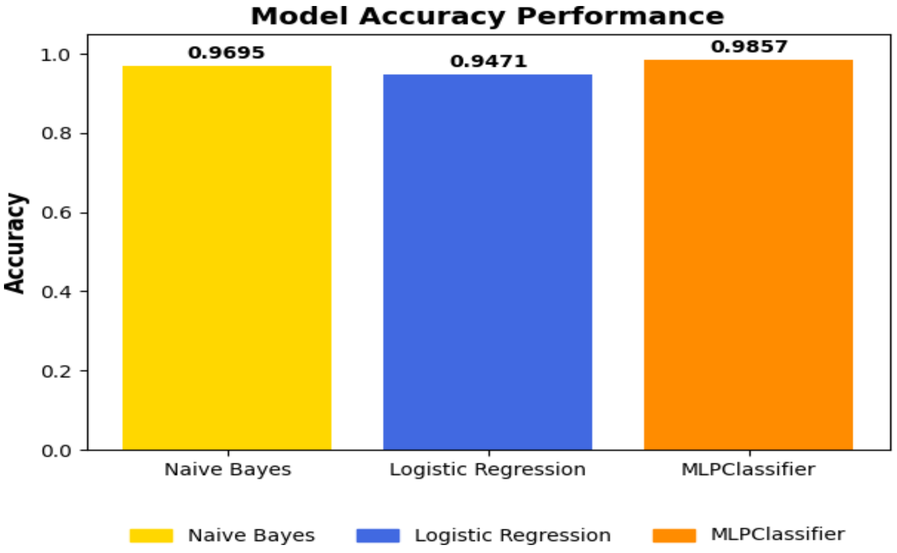
Code Snippet

Table 1: Code Snippet to create Comparison Bar Chart (Author, 2025)

```
import matplotlib.pyplot as plt

import matplotlib.patches as mpatches

#data for the chart

models = ['Naive Bayes', 'Logistic Regression', 'MLPClassifier']

accuracies = [0.9695, 0.9471, 0.9857]

colors = ['gold', 'royalblue', 'darkorange']

$create bar chart

fig, ax = p;t.subplots(figsize = (7, 5))

bars = ax.bar(models, accuracies, colour = colors)

#add text labels on top of bars

for bar, acc in zip(bars, accuracies):

   ax.text(bar.get_x() _ bar.get_width()/2,

   bar.get_height() + 0.01,

   f"{acc: .4f}",
   ha='center', va='bottom',
   fontsize=10, fontweight="bold")
#labels and title
ax.set_ylin(0, 1.05) #extra space not to touch the label top
ax.set_ylabel("Accuracy", fontsize=12, fontweight="bold")
ax.set_title("Model Accuracy Performance", fontsize=14, fontweight="bold")
#add custom legend with square colour as indicators below
patches = [mpatches.Patch(color=col, label=mod) for col, mod in zip(colors, models)]
ax.legend(handles=patches,
   loc= "upper center",
   bbox_to_anchor=(0.5, -0.15),
   ncol=len(models),
   frameon=False
#adjust bottom margin for legend not to cut off
plt.subplots_adjust(bottom=0.2)
#display chart
plt.show()
```

Comparison Using Bar Chart

Figure 9: Screenshot of comparing the three Machine Learning Models (Author, 2025)

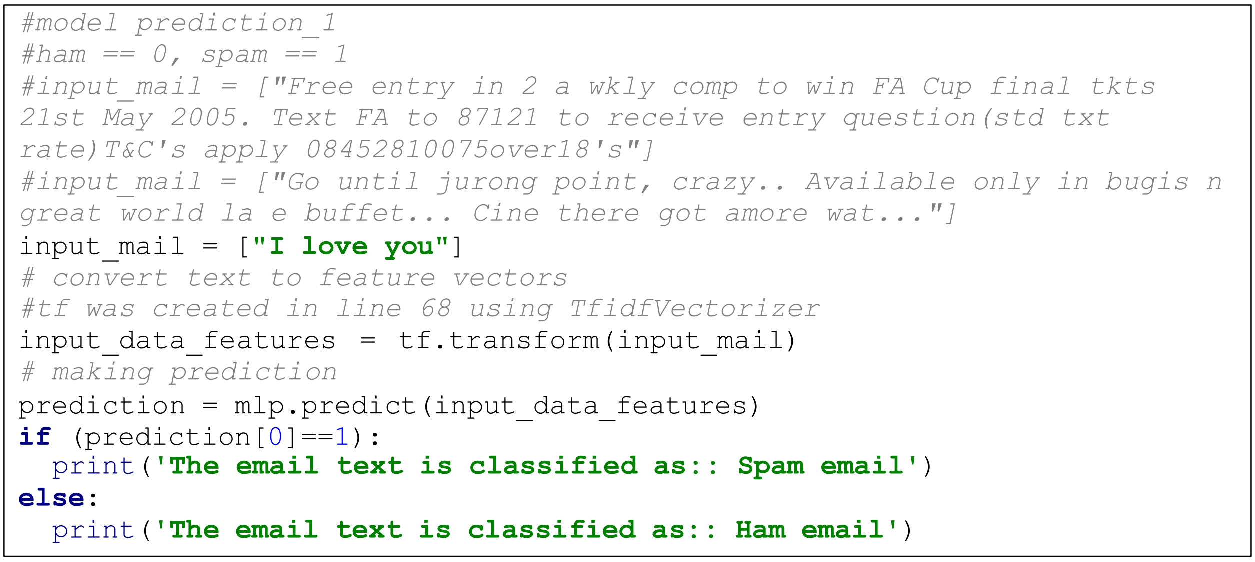


The figure above shows bar chart indicating the comparison of the three models tested in this project, which are Naïve Bayes, Logistic Regression and Multi-Layer Perceptron (MLP) Classifier. The bar chart compared the accuracy of the three different machine learning models for phishing email detection. Accuracy Scores: Naive Bayes: 96.95%; Logistic Regression: 94.71% and MLP Classifier: 98.57%. MLP Classifier achieves the highest accuracy (98.57%), indicating it is the most effective of the three for this dataset. Naive Bayes also performs very well (96.95%), slightly better than Logistic Regression (94.71%). All models perform at a high level, but the neural network model (MLP Classifier) has a clear edge.

In this project, it was set out to address the pressing need to conduct an AI-powered detection of phishing emails by analyzing the text features for more accurate and cost-effective diagnostic tools for phishing email detection. Leveraging machine learning classifiers, the project aim was to develop a classification model capable of accurately detecting between legitimate and phishing emails based on features extracted from a given dataset.

Performing Prediction with Mlp

Figure 10: Code Snippet to test Detection of Phishing Email (Author, 2025)



```
#model prediction_1
#ham == 0, spam == 1
#input_mail = ["Free entry in 2 a wkly comp to win FA Cup final tkts
21st May 2005. Text FA to 87121 to receive entry question(std txt
rate)T&C's apply 08452810075over18's"]
#input_mail = ["Go until jurong point, crazy.. Available only in bugis n
great world la e buffet... Cine there got amore wat..."]
input_mail = ["I love you"]
# convert text to feature vectors
#tf was created in line 68 using TfidfVectorizer
input_data_features = tf.transform(input_mail)
# making prediction
prediction = mlp.predict(input_data_features)
if (prediction[0]==1):
  print('The email text is classified as:: Spam email')
else:
  print('The email text is classified as:: Ham email')
```

Output

The output above represents a non-phishing email while another input serves as input email.

CONCLUSION

The methodology encompassed several key steps, including data preprocessing, exploratory data analysis, feature extraction, model training and testing, and model evaluation. We utilized machine learning classifiers to develop a classification model optimized for phishing email detection. We also demonstrated its practical utility using various Python libraries for implementation. Among the three machine learning classifier considered, the result shows that MLP Classifier may better capture complex patterns in phishing emails, leading to higher detection rates.

REFERENCES

1. Alahmar, M.I., Abdullah, L., Abdullah, H., Fahad, R., & Abdullah, L. (2023). Naïve Bayes Algorithms. DOI:10.13140/RG.2.2.15378.73921
2. Ajala, T.B., Oloko, R.K., & Agboola, A.R. (2025). Developing a Dashboard Embedded with KNN Machine Learning Algorithm for Wine Quality Prediction. International Journal of Innovative Science and Research Technology, 10(11), 1096-1103. DOI: doi.org/10.38124/ijisrt/25nov409.
3. Dinesh, P.M., Mukesh, M., Navaneethan, B., Sabeenian R.S., Paramasivam, M.E., and Manjunathan. A.

(2023). Identification of Phishing Attacks using Machine Learning Algorithm. DOI: https://doi.org/10.1051/e3sconf/202339904010.

4. Elzeiny, M. (2024). The Ultimate Guide to Naive Bayes. Available at: https://mlarchive.com/machine-learning/the-ultimate-guide-to-naive-bayes/.
5. Fares, H., Mouakkal, N., Baddi, Y., and Hajraoui, N. (2024). Robust Email Phishing Detection using Machine Learning and Deep Learning Approach. " International Journal of Communication Networks and Information Security (IJCNIS), vol. 16, no. 3, pp. 19-32.
6. Mohammad, R., Thabtah, F., and McCluskey, L. (2014). "Intelligent rule-based phishing websites classification," IET Inf. Secur., pp. 153–160.
7. Popescu, M.C., Balas, V.E., Popescu, L.P., & Mastorakis, N.E. (2009). Multilayer perceptron and neural networks.
8. Rashedi, K.A., Ismail, M.T., Wadi, S.A., Serroukh, A., Aishammari, T.S., & Jaber, J.J. (2024). Multi-Layer Perceptron-Based Classification with Application to Outlier Detection in Saudi Arabia Stock Returns. DOI: doi.org/10.3390/jrfm17020069.
9. Rawal, S., Rawal, B., Shaheen, A., and Malik, S. (2017). Phishing Detection in E-mails using Machine Learning. International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868. DOI:10.5120/ijais2017451713.
10. Rezazadeh, S. (2025). Review of Machine Learning. TMP Universal Journal of Research and Review Archives 4(2s). DOI:10.69557/ujrra.v4i2s.190
11. Salahdine, F and Kaabouch, N. (2019). "Social Engineering Attacks: A Survey," Future Internet J,, 11, 89, pp. 1-17.
12. Sambare, G.B., Galande, S.B., Kale, S., Nehete, P., Jadhav, V., and Tadavi, N. (2024). Towards Enhanced Security: An improved approach to Phishing Email Detection. J. Electrical Systems 20-2 (2024): 2763-2772.
13. Sasirekha, C., Nandhini, R., Karthiga, M.N., Bhuvaneshwari, R.S., and Chandra, V.S. (2023).Email Phishing Detection Using Machine Learning.
14. Yusoff, M.I.M. (2024) Machine Learning: An Overview. Open Journal of Modelling and Simulation, **12**, 89-99. doi: 10.4236/ojmsi.2024.123006.
15. Vikramkumar, V., Vjaykumar, B., & Tripathy, T. (2014). Bayes and Naive Bayes Classifier.Web linkhttps://www.futurelearn.com/info/courses/machine-learning-for-image-data/0/steps/362737 https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-logistic-regression/