IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

# DETECTION OF PHISHING E-MAIL THROUGH THE USE OF MACHINE LEARNING

**[1]S.Lakshmi Vijetha, [2]M.Yamini Lahari**

[1]Assistant Professor, [2]Assistant Professor
,
[1,2]Sir CRR College of Engineering ,Eluru,India

***Abstract :*** Email phishing is a type of cyber-attack that attempts to steal sensitive information by disguising as legitimate sources. Machine learning has the potential to detect email phishing attacks, and this paper presents an overview of the proposed machine learning-based approach for detection. The proposed approach involves feature extraction from emails, including message content, header information, and is used to train and test machine learning models. It checks several approaches for detecting the phishing mails. It uses supervised learning algorithms like logistic regressions, decision trees, random forests to classify incoming emails as either legitimate or phishing attempts. We calculate the accuracy for all the methods implemented to classify the mails. The results show that the proposed approach achieves high accuracy and outperforms existing approaches, and can be used by organizations and individuals to improve their email security.

***IndexTerms – Phishing email, Random Forest Classifier, Logistic Regression***

## I. INTRODUCTION

### INTRODUCTION

Phishing stands as a profitable form of fraud wherein perpetrators deceive recipients to acquire sensitive information under false pretenses. Phishing emails often prompt users to click on links or attachments, leading them to disclose confidential data such as passwords or credit card details. While these deceptive emails are sent to thousands, only a small fraction succumb to the scam, yielding substantial gains for the sender [1].

In 2006, hackers in the United States utilized emails to lure individuals into divulging usernames and passwords of American Online accounts. Since then, phishing techniques have advanced, rendering fraudulent emails increasingly difficult to detect.

### NEED OF THE STUDY.

According to Verizon's 2016 data breach report, around636,000 phishing emails were dispatched, with merely 3% of recipients flagging them as potential scams [2].

A significant phishing attack targeted millions of Gmail users in May 2017, granting hackers access to users' email histories. Armed with this data, hackers masqueraded as known entities, urging users to inspect attached files. Upon clicking the link, users were prompted to grant permission for a counterfeit application to manage their email accounts.

### APPROACHES

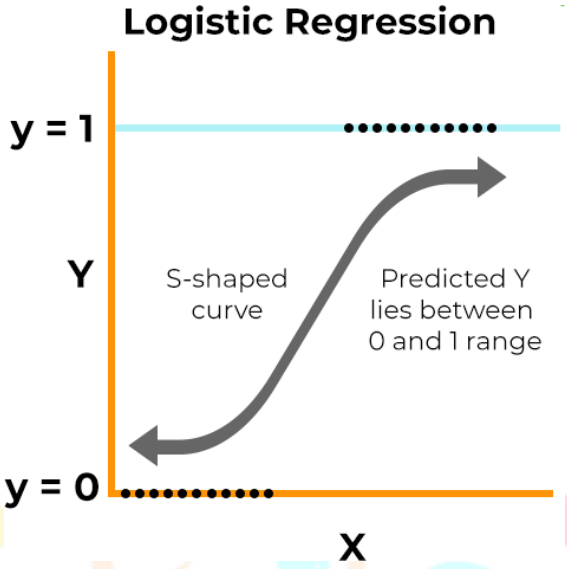
For the analysis purpose we have totally used 3 methods and tried to gather the output based on the result analysis.
The three methods:

1) Logistic Regression
2) Decision Tree Classifier
3) Random Forest Classifier

#### 1.1.1Logistic Regression

Logistic regression [3] is a popular classification algorithm widely used in various fields such as finance, healthcare, and marketing. Unlike linear regression, which predicts continuous outcomes, logistic regression is specifically designed for binary classification tasks, where the outcome is either 1 (positiveclass) or 0 (negative class).The core principle behind logistic regression is

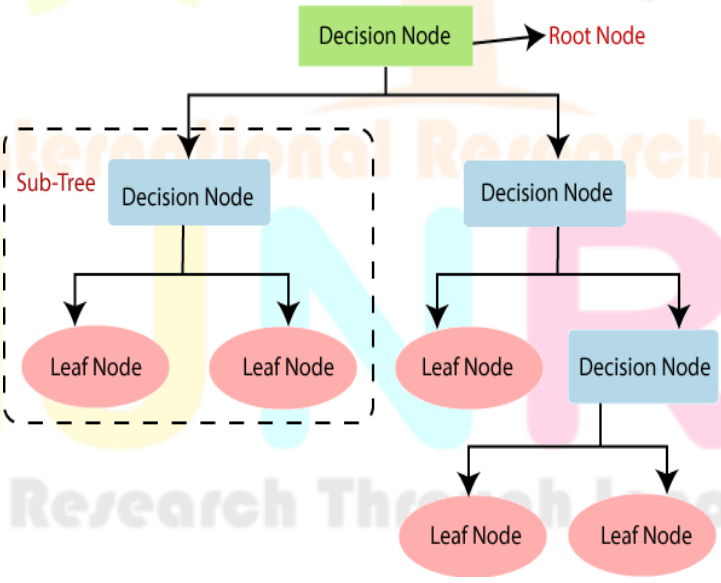
to model the probability that a given input belongsto a particular class. This is achieved by fitting a logistic function (also known as the sigmoid function) to thedata, which maps any real-valued input to a value between 0 and 1. The logistic function has an S-shaped curve, which effectively converts the output of a linear combination ofinput features into a probability score.




**1.1.2 Decision Tree Classifier**

The Decision Tree Classifier [4] is a powerful machine learning algorithm used for both classification and regression tasks. It operates by recursively partitioning the input space into smaller regions, each associated with a particular class or value. This process is guided by a seriesof decision rules, typically represented as a tree-like structure, where each internal node corresponds to a decision based on the value of a specific feature, and each leaf node representsthe predicted class or value.

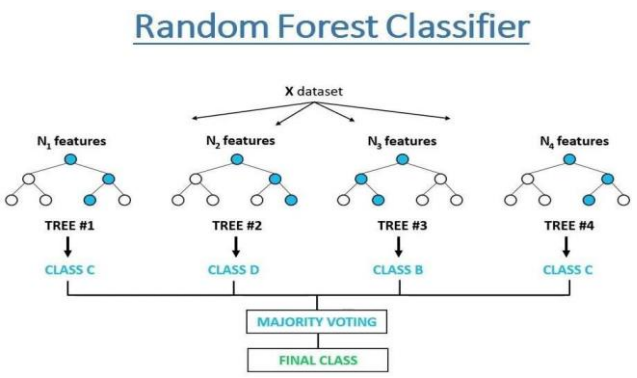
One of the key advantages of decision trees is their simplicity and interpretability. Unlike some other machine learning algorithms, such as neural networks or support vector machines, decisiontrees produce models that are easy to visualize and understand. This makes decision trees particularly useful in domains where interpretability is important, such as healthcare, finance, and customer relationship management.




**1.1.3Random Forest Classifier**

The Random Forest Classifier [5] is a powerful ensemble learning algorithm that combines the strengths ofmultiple decision trees to make robust and accurate predictions. As its name suggests,a random forest iscomposed of a collection of individual decision trees, each trained on a randomsubset of the training data and a random subset of the input features.

One of the key advantages of random forests is their ability to mitigate overfitting, a common problem in individual decision trees. By building multiple trees on different subsets of the data and features, random forests reduce the variance of the model and improve generalization performance. This makes them less sensitive to noise and outliers in the data, resulting in more reliable predictions.

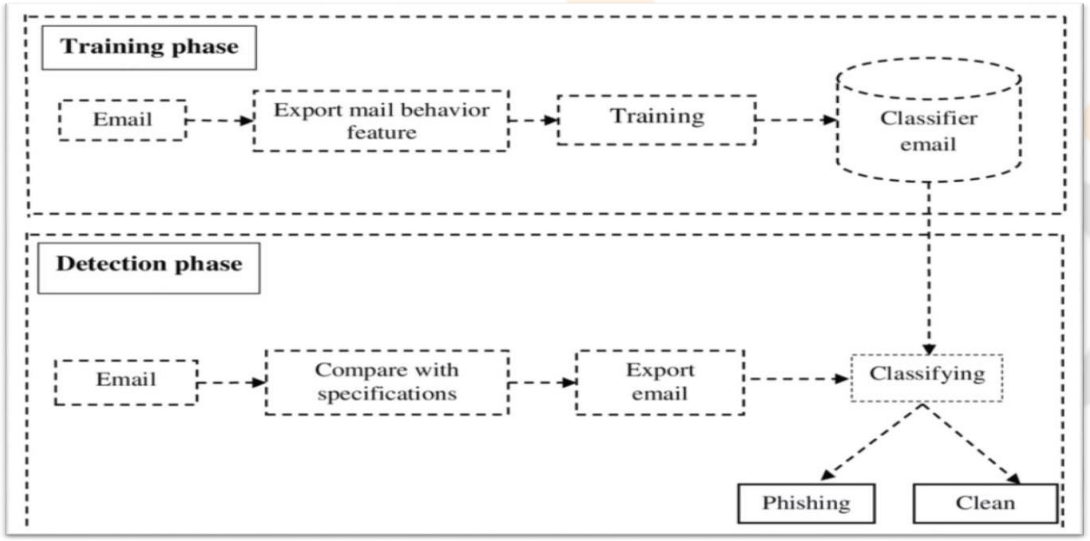



**1.2Global Crime Damage Cost**


All these facts have motivated us to find a solution for this problem using some of the predictiontechniques we have learned till now.

**1.3 Architecture**

The architecture describes the training and detection phase. In the training phase, the Emails areObserved and the features are extracted. These are trained to classify the mails [6].
The Detection phase will take the input mails and compares with the specified features. Aftercomparing these features, the mail will be declared whether its phishing mail or not.




**Design And Methodology**

**2.1Modules**

**2.1.1 Data Collections**
Collecting a diverse dataset of emails [7], including both legitimate and phishing emails.The dataset should be labeled to indicate whether each email is legitimate or malicious**.**

### 2.1.2 Preprocessing

**Tokenization:** Breaking the text of the emails into words or tokens.

**Removal:** Removing common words (e.g., "and", "the") that do not carry.

**Stemming:** Reducing words to their base or root form.
**Feature Extraction:** Converting the textual data into numerical feature vectors [8], which can beMachine Learning Algorithms.

### 2.1.3 Feature Selection

Identifying the most relevant features (words or phrases) that distinguish between legitimate and phishing emails. This step can help improve the efficiency andeffectiveness of the machine learning model.

### 2.1.4 Model Selection

Choosing an appropriate machine learning model for classification. Commonly used models for phishing email detection include:

- Logistic Regression
- Decision Trees
- Random Forest

### 2.1.5 Training

Training the selected machine learning model using the preprocessed data. Thisinvolves feeding the model with the labeled dataset and adjusting its parameters to minimize classification errors**.**

### 2.1.6 Evaluation

Assessing the performance of the trained model using evaluation metrics such as accuracy, precision, recall, F1-score, and ROC-AUC (Receiver Operating Characteristic- Area Under Curve).

## RESEARCH METHODOLOGY

### DATA PREPARATION, CLEANING AND PREPROCESSING

In data preparation, the initial step involves sourcing the requisite data for training the phishing detection model. This dataset may encompass phishing emails, legitimate emails, or a combination thereof, with a crucial emphasis on its representation of email types encountered inreal-world scenarios.

Following data acquisition, the labeling process assumes significance, providing the foundational truthfor the detection model to discern between phishing and legitimate emails effectively.

Subsequently, the dataset undergoes division into distinct subsets for training, validation, and testing purposes. This partitioning facilitates model training, hyperparameter tuning, and performance evaluation on unseen data. Feature extraction constitutes another pivotal phase, involving the retrieval of pertinent attributes such as sender addresses, subject lines, email bodies,and attachments from the email corpus.

Transitioning to data cleaning, the focus shifts towards rectifying any anomalies or inconsistencies within the dataset to bolster the efficacy of email phishing detection. This entailsthe elimination of duplicate emails to prevent bias in model training, alongside the removal of extraneous data and standardization to ensure dataset uniformity. Crucially, email addresses are uniformly converted to lowercase, and missing data is addressed using imputation techniques tocircumvent biases during modeltraining.

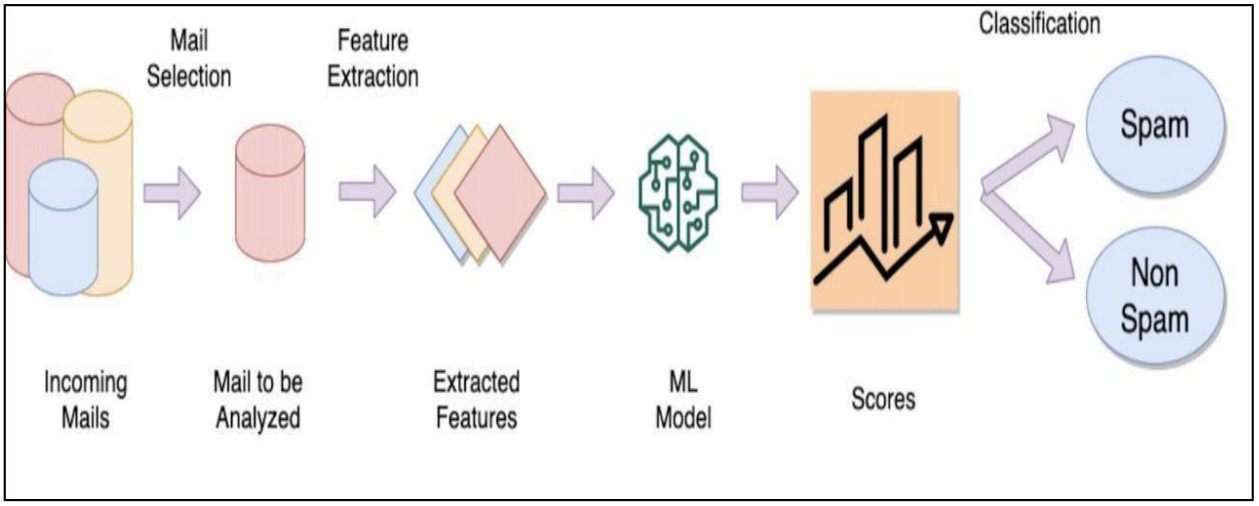
In data preprocessing, email messages undergo requisite transformations to prepare them for analysis. Tasks such as header removal, body extraction, and text formatting are undertaken to render the emails amenable to analysis by machine learning algorithms.

Finally, leveraging labeled email messages, machine learning models are trained and evaluated to detectphishing attempts with precision. Algorithms such as decision trees, random forests, and neural networksare employed to classify incoming email messages as either phishing or legitimate, ensuring robust protection against fraudulent activities.Through meticulous data preparation, cleaning, and preprocessing, the phishing detection modelis primed for training on a high-quality dataset, thereby fostering enhanced accuracy and efficacyin detecting malicious email activities.

.

### DATA FLOW DIAGRAM

A data flow diagram (DFD) provides a graphical representation of the flow of data within a system. In the context of phishing email detection using machine learning, here's a simplified DFD illustrating theflow of data through various modules. A data flow diagram (DFD) is a graphical representation illustrating how data moves within a system. It consists of processes, representing actions or transformations performed on data; data flows, showing the movement of data between processes, data stores, and external entities; data stores, indicating where data is stored within the system; and external entities, representing sources or destinations of data outside the system. DFDs provide a clear and conciseway to understand the flow of information in a system, making them valuable for requirements analysis,system design, and communicationbetween stakeholders.




**TESTING ACCURACY SCORES**

The accuracy of a classifier is given as the percentage of total correct predictions divided by

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN}$$

If the accuracy of the classifier is considered acceptable, the classifier can be used to classifyfuture data tuples for which the class label is not known.

Below are the results from the accuracy score from the trained models.

| Logistic Regression Classifier | Decision Tree Classifier | Random Forest Classifier |
|---|---|---|
| 94.14829659318637 | 97.0741482965932 | 98.11623246492987 |

Descriptive Statics has been used to find the maximum, minimum, standard deviation, mean and normally distribution of the data of all the variables of the study. Normal distribution of data shows the sensitivity of the variables towards the periodic changes and speculation. When the data is not normally distributed it means that the data is sensitive towards periodic changes and speculations which create the chances of arbitrage and the investors have the chance to earn above the normal profit. But the assumption of the APT is that there should not be arbitrage in the market and the investors can earn only normal profit. Jarque bera test is used to test the normality of data.

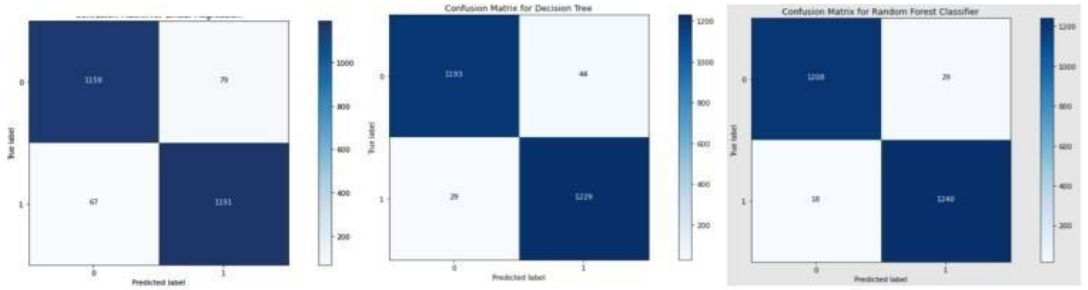
**CREATING CONFUSION MATRIX**

Each prediction will fall into one of these four categories. Let's look at what they are

| | | Predicted | |
|---|---|---|---|
| | | Negative | Positive |
| Actual | False | True Negative (TN) | False Positive (FP) |
| | True | False Negative (FN) | True Positive (TP) |

1. **True Negative (TN):** Data that is labeled false is predicted as false.
2. **True Positive (TP):** Data that is labeled true is predicted as true.
3. **False Positive (FP):** Also called "false alarm", this is a type 1 error in which the test ischecking a single condition and wrongly predicting a positive.
4. **False Negative (FN):** This is a type 2 error in which a single condition is checked and ourclassifier has predicted a true

instance as negative.




*Logistic Rgression Classifier* *Decision Tree Classifier* *Random Forest*

**GIVEN SOME DATA TO THE TRAINED AND ACQUIRED THE OUTPUT**

As we can see that Random Forest classifier have the best results from the list below

| | Class_True | Class_Predicted |
|---|---|---|
| 0 | False | True |
| 1 | False | True |
| 2 | False | False |
| 3 | False | False |
| 4 | False | False |
| 5 | False | True |
| 6 | False | True |
| 7 | False | True |
| 8 | False | False |
| 9 | False | False |
| 10 | False | True |
| 11 | False | True |
| 12 | False | False |
| 13 | False | False |
| 14 | False | False |
| 15 | False | True |
| 16 | False | False |
| 17 | False | True |
| 18 | False | False |
| 19 | False | False |

*Logistic Regression*

| | Class_True | Class_Predicted |
|---|---|---|
| 0 | False | True |
| 1 | False | False |
| 2 | False | False |
| 3 | False | False |
| 4 | False | False |
| 5 | False | False |
| 6 | False | False |
| 7 | False | True |
| 8 | False | False |
| 9 | False | False |
| 10 | False | True |
| 11 | False | False |
| 12 | False | False |
| 13 | False | False |
| 14 | False | False |
| 15 | False | False |
| 16 | False | False |
| 17 | False | False |
| 18 | False | False |
| 19 | False | True |

*Decision Tree Classifier*

| | Class_True | Class_Predicted |
|---|---|---|
| 0 | False | False |
| 1 | False | True |
| 2 | False | False |
| 3 | False | False |
| 4 | False | False |
| 5 | False | False |
| 6 | False | False |
| 7 | False | True |
| 8 | False | False |
| 9 | False | False |
| 10 | False | True |
| 11 | False | False |
| 12 | False | False |
| 13 | False | False |
| 14 | False | False |
| 15 | False | False |
| 16 | False | False |
| 17 | False | False |
| 18 | False | False |
| 19 | False | False |

*Random Forest Classifier*

**EXECUTION TIME FOR EACH MODEL**

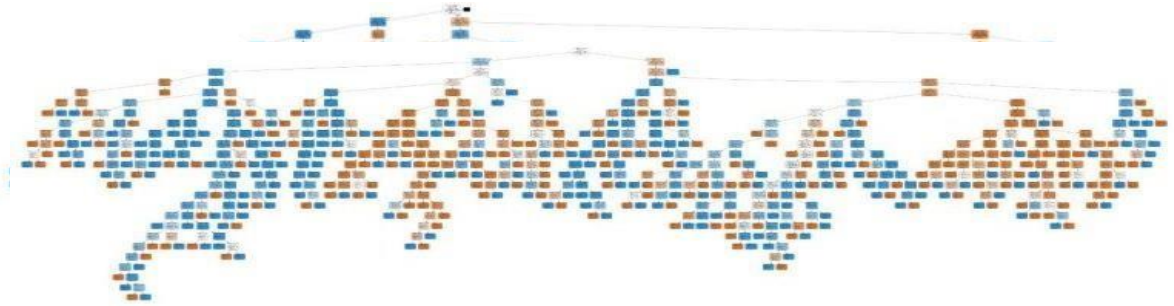
Using the Jupyter Command %time it, we have found out the execution time required to trainand testall three models.

| Logistic Regression Model | Decision Tree Classifier | Random Forest Classifier |
|---|---|---|
| 125 ms | 62.5 ms | 672 ms |

**PLOTTED THE DECISION TREE**

Similarly, we have plotted the decision for our dataset which have 48 predictors from F1 toF48. As thegenerated graph is very huge, it has been scaled by "0.829502" to fit.

Below are the results from the analysis of Random Forest Classifier and Decision TreeClassifier.



Decision Tree Classifier

Random Forest Classifier

**RESULTS**

```
# Logistic Regression

%%time
clf = LogisticRegression()
clf.fit(X_train, Y_train)

predictions_lr = clf.predict(X_test)
score_lr = clf.score(X_test, Y_test) * 100

print("Logistic Regression Score: ", score_lr)
```

```
Logistic Regression Score:  94.14829659318637
CPU times: user 136 ms, sys: 87.4 ms, total: 223 ms
Wall time: 127 ms
```

```
matrix_lr = confusion_matrix(Y_test, predictions_lr.round())
print("Confusion Matrix for Logistic Regression")
print(matrix_lr)
```

```
Confusion Matrix for Logistic Regression
[[1158   79]
 [  67 1191]]
```

Logistic Regression Accuracy and Confusion Matrix Details

Decision Tree Classifier Accuracy and Confusion Matrix Details

```
# Decision Tree Classifier

%%time
dt = DecisionTreeClassifier()
dt.fit(X_train, Y_train)

predictions = dt.predict(X_test)
score = dt.score(X_test, Y_test) * 100

print("Decision Tree Score: ", score)
```

```
Decision Tree Score:  97.11422845691384
CPU times: user 75.7 ms, sys: 1.85 ms, total: 77.5 ms
Wall time: 77.8 ms
```

```
matrix = confusion_matrix(Y_test, predictions.round())
print("Confusion Matrix for Decision Tree")
print(matrix)
```

```
Confusion Matrix for Decision Tree
[[1194   43]
 [  29 1229]]
```

```
[ ] # Random Forest Classifier

    %%time
    clf2 = RandomForestClassifier()
    clf2.fit(X_train, Y_train)

    predictions_fr = clf2.predict(X_test)
    score_fr = clf2.score(X_test, Y_test) * 100

    print("Random Forest Classifier Score: ", score_fr)

    Random Forest Classifier Score:  98.23647294589179
    CPU times: user 901 ms, sys: 6.16 ms, total: 907 ms
    Wall time: 909 ms

[ ] matrix_fr = confusion_matrix(Y_test, predictions_fr.round())
    print("Confusion Matrix for Random Forest Classifier")
    print(matrix_fr)

    Confusion Matrix for Random Forest Classifier
    [[1211   26]
     [  18 1240]]
```

Random Forest Classifier Accuracy and Confusion Matrix Details

```
#Prediction for Unknown Data using Logistic Regression


dfp_lr = pd.DataFrame()

predictionsr_lr = clf.predict(Xr)
dfp_lr["Class_True"] = Yr
predictionsr_lr = pd.DataFrame(predictionsr_lr)
predictionsr_lr = predictionsr_lr.replace({1: True, 0: False})
dfp_lr["Class_Predicted"] = predictionsr_lr
print("The Predicted Values for Logistic Regression is: ")
print(dfp_lr)

The Predicted Values for Logistic Regression is:
    Class_True  Class_Predicted
0        False            True
1        False            True
2        False           False
3        False           False
4        False           False
5        False            True
6        False            True
7        False            True
8        False           False
9        False           False
10       False            True
11       False            True
12       False           False
13       False           False
14       False           False
15       False            True
16       False           False
17       False            True
18       False           False
19       False           False
```

Predicted output for Unseen data using Logistic Regression

```
[ ] #Prediction for Unknown Data using Random Forest Classifier


    dfp_fr = pd.DataFrame()

    predictionsr_fr = clf2.predict(Xr)
    dfp_fr["Class_True"] = Yr
    predictionsr_fr = pd.DataFrame(predictionsr_fr)
    predictionsr_fr = predictionsr_fr.replace({1: True, 0: False})
    dfp_fr["Class_Predicted"] = predictionsr_fr
    print("The Predicted Values for Random Forest Classifier is: ")
    print(dfp_fr)

    The Predicted Values for Random Forest Classifier is:
        Class_True  Class_Predicted
    0        False           False
    1        False            True
    2        False           False
    3        False           False
    4        False           False
    5        False           False
    6        False           False
    7        False            True
    8        False           False
    9        False           False
    10       False            True
    11       False           False
    12       False           False
    13       False           False
    14       False           False
    15       False           False
    16       False           False
    17       False           False
    18       False           False
    19       False           False
```

Predicted output for Unseen data using Decision Tree Classifier

```
#Prediction for Unknown Data using Decision Tree Classifier


dfp_dt = pd.DataFrame()
predictionsr_dt = dt.predict(Xr)
dfp_dt["Class_True"] = Yr
predictionsr_dt = pd.DataFrame(predictionsr_dt)
predictionsr_dt = predictionsr_dt.replace({1: True, 0: False})
dfp_dt["Class_Predicted"] = predictionsr_dt
print("The Predicted Values for Decision Tree Classifier is: ")
print(dfp_dt)
```

```
The Predicted Values for Decision Tree Classifier is:
    Class_True  Class_Predicted
0        False            False
1        False             True
2        False            False
3        False            False
4        False            False
5        False            False
6        False            False
7        False             True
8        False            False
9        False            False
10       False             True
11       False            False
12       False            False
13       False            False
14       False            False
15       False            False
16       False            False
17       False            False
18       False            False
19       False             True
```

Predicted output for Unseen data using Random Forest Classifier

## CONCLUSION

In conclusion, this study presents an approach for classifying emails as either phishing or legitimate (ham) using machine learning algorithms. The dataset underwent preprocessing and feature extraction, facilitated by Python programming and libraries such as regular expressionsand NLTK. These features were then utilized to train various supervised learning classifiers, including Random Forest, Logistic Regression and Decision Trees.

The classification results yielded promising accuracy, with the highest achieving 98.1%. Whilethese outcomes are encouraging, it's essential to acknowledge the limitations inherent in the dataset utilized, which may not fully replicate real-world scenarios. To enhance the robustnessand applicability of the proposed system, future research should focus on expanding the datasetto include a broader range of email samples, encompassing both phishing and legitimate emails. By incorporating diverse samples reflective of evolving phishing techniques, the system can better emulate real-world scenarios, thus bolstering its effectiveness in thwarting fraudulent activities.

## FUTURE ENHANCEMENT

**T**he continued rise of social engineering, exploiting cloud-based infrastructure, IoT devices andmobile apps expanding the threat surface and what the explosion of AI and Machine Learning means for the future of phishing**.** As each year rolls by, phishing and malware attacks continueto be persistent challenges. However, with the monumental technological advancements we have seen recently, the tactics and strategies employed by cybercriminals when conducting these attacks are evolving.

**Implications for organizations:** The continued rise of social engineering in phishing underscores the critical importance of comprehensive employee training and awarenessinitiatives. Organizations must educate staff about the tell-tale signs of phishing attempts, including suspicious sender addresses, grammatical errors, or requests for sensitive information. By fostering a culture of vigilance and scepticism, organizations can empower employees to recognize and report phishing attempts, thereby mitigating the risk of data breaches and financial losses.

**Emerging threats:** Looking ahead, emerging trends in social engineering include the integration of AI and machine learning to automate and optimize phishing campaigns. Additionally, theproliferation of remote work and digital communication platforms presents new opportunities for cybercriminals to exploit human vulnerabilities – and we will touch onthese advancements shortly.

## REFERENCES

### REFERENCES

[1]. K. Zetter, L. Matsakis, I. Lapowsky, G. Graff, E. Dreyfuss, and L. Newman, "Researchers uncoverRSA phishing attack, hiding in plain sight," WIRED, 2018.[Online].Available:https://www.wired.com/2011/ 08/how-rsa-got-hacked.

[2] N. Arachchilage, S. Love, and K. Beznosov, "Phishing threat avoidance behaviour: An empirical investigation," Comput. Hum. Behav.vol. 60, pp. 185 197, 2016.

[3]. M. Alsharnouby, F. Alaca, and S. Chiasson, "Why phishing still works: User strategies for combating phishing attacks," Int. J.Hum.-Comput. Stud., vol. 82, pp.69– 82,2015.[Online]. Available:10.1016/j.ijhcs.2015.05.005.

[4]. T. Jagatic, N. Johnson, M. Jakobsson, and F. Menczer, "Social phishing," Commun. ACM, vol. 50, no.10, pp. 94–100, 2007. 10.1145/1290958.1290968. [Online]

[5] SHARIFF V, CHIRANJEEVI P, &KRISHNA M. A. (2023). AN ANALYSIS ON ADVANCES IN LUNG CANCER DIAGNOSIS WITH MEDICAL IMAGING AND DEEP LEARNING TECHNIQUES: CHALLENGES AND OPPORTUNITIES. Journalof Theoretical and Applied Information Technology 101(17).

[6] S, S., Chandra Shikhi Kodete, Saibaba Velidi, Srikanth Bhyrapuneni, Suresh Babu Satukumati, and Vahiduddin Shariff. "Revolutionizing Healthcare: A Comprehensive Framework for Personalized IoT and Cloud Computing-Driven Healthcare Services with Smart Biometric Identity Management." Journal of Intelligent Systems and Internet of Things 13, no. 1 (January 1, 2024): 31–45. https://doi.org/10.54216/jisiot.130103.

[7] Praveen, S Phani, Veerapaneni Esther Jyothi, Chokka Anuradha, K VenuGopal, Vahiduddin Shariff, and S Sindhura. "Chronic Kidney Disease Prediction Using ML-Based Neuro-Fuzzy Model." International Journal of Image and Graphics, December 15, 2022. https://doi.org/10.1142/s0219467823400132.

[8] K. Arava, C. Paritala, V. Shariff, S. P. Praveen and A. Madhuri, "A Generalized Model for Identifying Fake Digital Images through the Application of Deep Learning," *2022 3rd International Conference on Electronics and Sustainable Communication Systems (ICESC)*, Coimbatore, India, 2022, pp. 1144-1147, doi: 10.1109/ICESC54411.2022.9885341.