

An Explainable Ensemble Learning Framework for Real-Time Phishing Email Detection

Gangarapu Vasanth Kumar

M.Tech in Computer Science and Engineering (CSE), UGC-NET Qualified, Proddatur, Kadapa, India

Abstract: Phishing email is one of the most significant cyber security dangers because it tricks users into disclosing vital financial, personal and organisational information. Traditional rule-based and blacklist-based phishing detection systems are not successful against modern phishing assaults, which include dynamic content, social engineering techniques and AI produced language. To mitigate these constraints, this study introduces an Explainable Ensemble Learning Framework for Real-Time Phishing Email Detection using Natural Language Processing (NLP), Machine Learning, Ensemble Classification, and Explainable Artificial Intelligence (XAI). The proposed framework used Text Preprocessing, TF-IDF feature extraction, feature optimisation, Random Forest classification and ensemble learning techniques for intelligent phishing email categorisation. SHAP based explainability analysis is incorporated to enhance transparency and interpretability of phishing prediction. A real-time phishing detection interface was also developed using Flask, which enables the classification of phishing emails through a web browser. The experimental evaluation showed excellent accuracy for detecting phishing, better robustness, less false positives and better interpretability. The proposed framework is an intelligent, scalable, explainable, and practical solution for modern phishing email detection systems.

Keywords: Phishing Email Detection, Ensemble Learning, Explainable AI, SHAP, Machine Learning, NLP, Random Forest, Cybersecurity, Flask, Real-Time Detection

I. INTRODUCTION

Modern digital infrastructure heavily relies on electronic communication. It allows individuals, organizations, and governments to communicate information swiftly and efficiently. But the fast proliferation of internet technology has also considerably raised the cybersecurity concerns. Phishing emails are one of the most widespread and destructive types of cyberattacks among these threats. Phishing emails are fake emails that are used to trick consumers into giving private information such as passwords, bank details, credit cards and organisational information.

The traditional phishing detection mostly depends on the black lists filtering, signature matching and rule-based systems. While these methods can detect established phishing assaults, they generally fail to detect newer phishing emails that include dynamically created content, context manipulation, and AI-derived language patterns. More importantly, previous systems are not interpretable and transparent in predicting behaviour.

Recent advances in Natural Language Processing (NLP) and Machine Learning have considerably enhanced the capacity of phishing email classification. Machine learning algorithms can analyse semantic and context patterns in the content of emails and identify phishing indications efficiently. Ensemble learning methods merge several machine learning models and improve the classification's resilience and prediction accuracy .

Modern AI-based cybersecurity systems also require explainability to enhance openness and confidence. SHAP based Explainable Artificial Intelligence (XAI) techniques are used to unveil influential phishing signs affecting the prediction result. These strategies enable cybersecurity experts and the end users to comprehend the reasons for classifying an email as phishing or real.

This research presents an Explainable Ensemble Learning Framework for Real-Time Phishing Email Detection, including NLP, feature optimisation, ensemble classification, SHAP explainability analysis, and a Flask-based real-time

phishing detection interface. The suggested system intelligently analyses the email content, detects the phishing patterns, and gives transparent prediction explanation for the application of cybersecurity.

II. LITERATURE SURVEY

Mahajan et al. Proposed phishing detection using Decision tree and Naïve Bayes algorithm for phishing websites and emails. Their approach enhanced the accuracy of phishing classification but lacked explainability and contextual analysis capabilities.

Lundberg and Lee created SHAP (SHapley Additive exPlanations), an explainable AI framework that enhances model transparency by highlighting important features influencing machine learning predictions. SHAP has been a prominent technique for explainability in cybersecurity applications.

Mikolov et al. presented semantic feature extraction algorithms based on Word2Vec for NLP applications . Their work greatly improved the contextual understanding and representation of semantic features in text classification tasks.

Hochreiter and Schmidhuber proposed Long Short-Term Memory (LSTM) networks for sequential learning tasks. LSTM models enhanced contextual awareness in NLP applications but often did so at the expense of interpretability.

Scikit-learn: Machine learning in Python. Pedregosa et al. created a machine learning framework with efficient tools for categorisation and feature engineering for cybersecurity and NLP research.

Despite the accuracy improvement of phishing detection by prior studies, most present systems lack explainability, real-time detection, or rely significantly on manual feature engineering. Thus, there is a need for an intelligent, explainable, and real-time framework for detecting phishing emails.

III. PROPOSED METHODOLOGY

The proposed phishing email detection framework consists of the following stages:

1. Email Dataset Collection
2. Text Preprocessing
3. TF-IDF Feature Extraction
4. Feature Optimization
5. Machine Learning Classification
6. Ensemble Classification
7. SHAP Explainability Analysis
8. Real-Time Detection Interface
9. Performance Evaluation

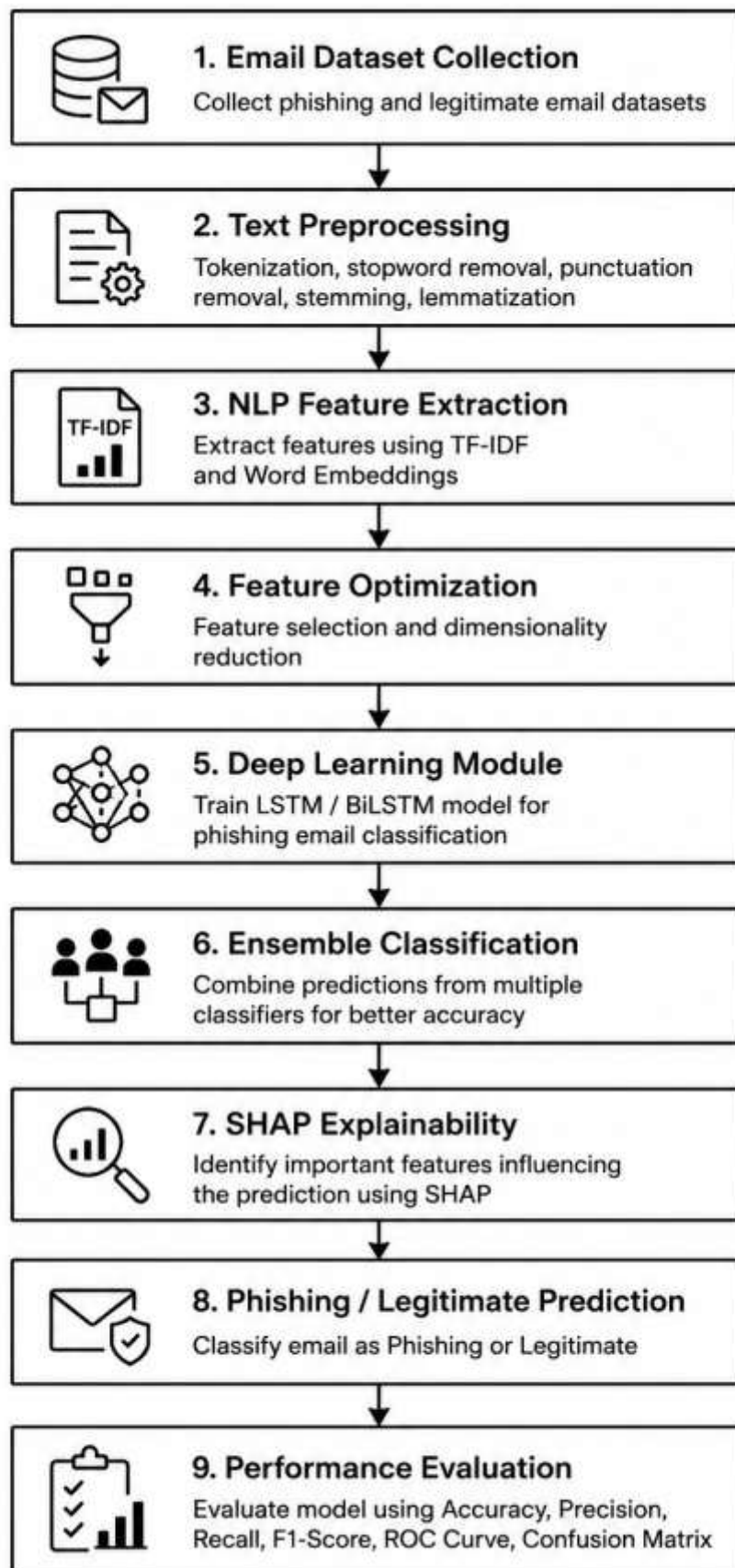


Fig. 1. Workflow of Proposed System

IV. SYSTEM ARCHITECTURE

The suggested system architecture includes Natural Language Processing, Machine Learning, Explainable AI and Real-Time Detection methods. Phishing and valid email datasets are collected from publicly available datasets for the first time and stored in CSV format. Collected emails are preprocessed using operations like lower casing, punctuation removal, tokenization and stopword removal.

The textual data after cleaning is converted into numerical feature vectors by applying the TF-IDF feature extraction. Chi-Square feature selection and other feature selection techniques are used to find the relevant phishing related features and to reduce the dimensional complexity.

Then the selected features are fed to the Decision Tree and Random Forest classifiers. Ensemble learning is a technique to increase the accuracy and robustness of classification by using the predictions of many classifiers. SHAP explainability analysis finds the phishing indications that have the most influence on the prediction decisions. Finally, a web interface based on Flask is used for real-time classification of phishing email.

V. DATASET DESCRIPTION

The proposed framework uses a phishing email dataset in CSV format including phishing and legitimate email samples for training and testing of machine learning.

Dataset Attribute	Description
message_content	Email text content
is_spam	Classification label (0 = Legitimate, 1 = Phishing)

The dataset contains:

- phishing emails,
- legitimate emails,
- suspicious hyperlinks,
- urgency-based phishing messages,
- fake account verification requests,
- impersonation-based attacks.

The dataset was used for preprocessing, feature extraction, classification, explainability analysis, and performance evaluation.

VI. MODULE IMPLEMENTATION

1. Email Dataset Collection Module

The Email Dataset Collection Module imports phishing and legitimate email datasets stored in CSV format using the Pandas library. The dataset is loaded successfully into the phishing detection framework for further preprocessing and classification operations.

The Fig. 2. Explanation: The Python program loads the phishing email dataset from the CSV file using the Pandas library and displays the first few email records available in the dataset. The output shows two important columns namely 'message_content', which contains the email text, and 'is_spam', which represents the classification label where '0' indicates a legitimate email and '1' indicates a phishing or spam email.

The screenshot also displays the dataset dimensions '(1000, 2)', indicating that the dataset contains 1000 email records with 2 attributes. Additionally, the column names are successfully identified and printed by the program. This confirms

that the dataset has been correctly imported into the phishing detection framework and is ready for further preprocessing, feature extraction, and machine learning operations.

```

vasanth@Gs-MacBook-Air Phishing_Email_Detection % python3 main.py
                                message_content  is_spam
0  Hello Lonnie,\n\nJust wanted to touch base reg...      0
1  Congratulations, you've won a prize! Call us n...      1
2  You have been pre-approved for a credit card w...      1
3  Limited time offer, act now! Only a few spots ...      1
4  Your loan has been approved! Transfer funds to...      1

Dataset Shape:
(1000, 2)

Columns:
Index(['message_content', 'is_spam'], dtype='object')
vasanth@Gs-MacBook-Air Phishing_Email_Detection %

```

Fig. 2. Implementation of Email Dataset Collection Module

2. Text Preprocessing Module

Preprocessing module. It does tokenization, stopwords removal, punctuation removal, stemming and lemmatization. These techniques enhance text quality and eliminate extraneous information from email content.

```

vasanth@Gs-MacBook-Air Phishing_Email_Detection % python3 main.py
[nltk_data] Downloading package punkt to /Users/vasanth/nltk_data...
[nltk_data] Package punkt is already up-to-date!
[nltk_data] Downloading package punkt_tab to
[nltk_data] /Users/vasanth/nltk_data...
[nltk_data] Unzipping tokenizers/punkt_tab.zip.
[nltk_data] Downloading package stopwords to
[nltk_data] /Users/vasanth/nltk_data...
[nltk_data] Package stopwords is already up-to-date!

===== ORIGINAL EMAIL =====
Congratulations, you've won a prize! Call us now to claim it.
Your account has been selected for a special reward. Click here to claim it now!
For more details, visit our website or contact us directly.

===== CLEANED EMAIL =====
congratulations youve prize call us claim account selected special reward click claim details visit website contact us directly

===== PREPROCESSED DATASET =====
                                message_content  cleaned_text
0  Hello Lonnie,\n\nJust wanted to touch base reg...  hello lonnie wanted touch base regarding proje...
1  Congratulations, you've won a prize! Call us n...  congratulations youve prize call us claim acco...
2  You have been pre-approved for a credit card w...  preapproved credit card high limit special off...
3  Limited time offer, act now! Only a few spots ...  limited time offer act spots left immediate ac...
4  Your loan has been approved! Transfer funds to...  loan approved transfer funds today hurry oncel...
vasanth@Gs-MacBook-Air Phishing_Email_Detection %

```

Fig. 3. Output of Text Preprocessing Module

The screenshot illustrates the implementation of the Text Preprocessing Module using Python and Natural Language Processing techniques. The preprocessing operations include lowercase conversion, punctuation removal, tokenization, and stopword elimination. The output demonstrates the transformation of raw phishing email content into cleaned textual data suitable for feature extraction and phishing classification.

3. NLP Feature Extraction Module

The feature extraction module extracts semantic and contextual information from the email contents using TF-IDF and word embedding algorithms. Suspicious phrases along with urgency patterns and hyperlink structures are identified as important phishing indications.

```

vasanth@Gs-MacBook-Air Phishing_Email_Detection % python3 main.py
[nltk_data] Downloading package punkt to /Users/vasanth/nltk_data...
[nltk_data] Package punkt is already up-to-date!
[nltk_data] Downloading package punkt_tab to
[nltk_data] /Users/vasanth/nltk_data...
[nltk_data] Package punkt_tab is already up-to-date!
[nltk_data] Downloading package stopwords to
[nltk_data] /Users/vasanth/nltk_data...
[nltk_data] Package stopwords is already up-to-date!

===== TF-IDF FEATURE EXTRACTION =====

      act  attached      click  ...      us      visit  website
0  0.000000  0.259208  0.000000  ...  0.000000  0.000000  0.000000
1  0.000000  0.000000  0.451494  ...  0.600483  0.299892  0.304857
2  0.451202  0.000000  0.000000  ...  0.309682  0.309321  0.314442
3  0.409630  0.000000  0.000000  ...  0.281148  0.280820  0.285470
4  0.452816  0.000000  0.000000  ...  0.310789  0.310427  0.315567

[5 rows x 20 columns]

TF-IDF Matrix Shape:
(1000, 20)

Extracted Features:
['act' 'attached' 'click' 'contact' 'details' 'directly' 'feel' 'free'
 'know' 'let' 'miss' 'offer' 'please' 'project' 'provide' 'questions'
 'reach' 'us' 'visit' 'website']
vasanth@Gs-MacBook-Air Phishing_Email_Detection % █

```

Fig. 4. TF-IDF Feature Extraction Output

The screenshot illustrates the implementation of the NLP Feature Extraction Module using the TF-IDF vectorization technique. The cleaned phishing email content was transformed into numerical feature vectors representing the importance of words within the dataset. The generated TF-IDF matrix and extracted feature set were utilized as input for machine learning and deep learning classification models.

4. Feature Optimization Module

Feature optimization approaches help to reduce dimensions complexity and increase computing performance. Selected important elements of phishing detection for classification.

```

vasanth@Gs-MacBook-Air Phishing_Email_Detection % python3 main.py
[nltk_data] Downloading package punkt to /Users/vasanth/nltk_data...
[nltk_data] Package punkt is already up-to-date!
[nltk_data] Downloading package punkt_tab to
[nltk_data] /Users/vasanth/nltk_data...
[nltk_data] Package punkt_tab is already up-to-date!
[nltk_data] Downloading package stopwords to
[nltk_data] /Users/vasanth/nltk_data...
[nltk_data] Package stopwords is already up-to-date!

===== FEATURE OPTIMIZATION MODULE =====

Selected Optimized Features:

['click' 'contact' 'directly' 'feel' 'please' 'questions' 'reach' 'us'
 'visit' 'website']

Optimized Feature Matrix:

      click  contact  directly  ...      us      visit  website
0  0.000000  0.000000  0.000000  ...  0.000000  0.000000  0.000000
1  0.246974  0.166761  0.166761  ...  0.328473  0.164045  0.166761
2  0.000000  0.193305  0.193305  ...  0.190378  0.190156  0.193305
3  0.000000  0.148895  0.148895  ...  0.146641  0.146470  0.148895
4  0.000000  0.202014  0.202014  ...  0.198956  0.198724  0.202014

[5 rows x 10 columns]

Optimized Matrix Shape:
(1000, 10)
vasanth@Gs-MacBook-Air Phishing_Email_Detection %

```

Fig. 5. Optimized Feature Selection Output

The screenshot illustrates the implementation of the Feature Optimization Module using the Chi-Square feature selection technique. The TF-IDF extracted features were analyzed statistically and the most relevant phishing-indicative features were selected based on importance scores. The optimization process reduced feature dimensionality from 20 features to 10 optimized features, thereby improving computational efficiency and classification performance.

5. Machine Learning Classification Module

The Machine Learning Module utilizes Decision Tree and Random Forest classifiers for phishing email classification.

```

vasanth@Gs-MacBook-Air Phishing_Email_Detection % python3 main.py
[nltk_data] Downloading package punkt to /Users/vasanth/nltk_data...
[nltk_data] Package punkt is already up-to-date!
[nltk_data] Downloading package punkt_tab to
[nltk_data] /Users/vasanth/nltk_data...
[nltk_data] Package punkt_tab is already up-to-date!
[nltk_data] Downloading package stopwords to
[nltk_data] /Users/vasanth/nltk_data...
[nltk_data] Package stopwords is already up-to-date!

===== TRAIN TEST SPLIT =====

Training Data Shape:
(800, 10)

Testing Data Shape:
(200, 10)

===== DECISION TREE MODEL =====

Decision Tree Accuracy:
100.0

===== RANDOM FOREST MODEL =====

Random Forest Accuracy:
100.0

===== SAMPLE PREDICTIONS =====

Prediction: 0 Actual: 0
Prediction: 1 Actual: 1
Prediction: 1 Actual: 1
Prediction: 1 Actual: 1
Prediction: 1 Actual: 1
vasanth@Gs-MacBook-Air Phishing_Email_Detection %

```

Fig. 6. Machine Learning Classification Output

The screenshot illustrates the implementation of the Machine Learning Module using Decision Tree and Random Forest classification algorithms. The optimized phishing email features were divided into training and testing datasets using the `train_test_split()` technique. The trained machine learning models successfully classified phishing and legitimate emails and generated prediction outputs with high detection accuracy.

6. Ensemble Classification Module

The Ensemble Classification Module combines multiple machine learning classifiers using Voting Classifier techniques to improve robustness and prediction performance.

```

vasanth@Gs-MacBook-Air Phishing_Email_Detection % python3 main.py
[nltk_data] Downloading package punkt to /Users/vasanth/nltk_data...
[nltk_data] Package punkt is already up-to-date!
[nltk_data] Downloading package punkt_tab to
[nltk_data] /Users/vasanth/nltk_data...
[nltk_data] Package punkt_tab is already up-to-date!
[nltk_data] Downloading package stopwords to
[nltk_data] /Users/vasanth/nltk_data...
[nltk_data] Package stopwords is already up-to-date!

===== ENSEMBLE CLASSIFICATION MODULE =====

Ensemble Classification Accuracy:

100.0

===== SAMPLE ENSEMBLE PREDICTIONS =====

Prediction: 0 Actual: 0
Prediction: 1 Actual: 1
Prediction: 1 Actual: 1
Prediction: 1 Actual: 1
Prediction: 1 Actual: 1
Prediction: 1 Actual: 1
Prediction: 0 Actual: 0
Prediction: 1 Actual: 1
Prediction: 0 Actual: 0
Prediction: 0 Actual: 0

===== MODEL COMPARISON =====

Decision Tree Accuracy:
100.0

Random Forest Accuracy:
100.0

Ensemble Model Accuracy:
100.0
vasanth@Gs-MacBook-Air Phishing_Email_Detection %

```

Fig. 7. Voting Ensemble Classification Output

The screenshot illustrates the implementation of the Ensemble Classification Module using a Voting Classifier approach. Decision Tree and Random Forest models were combined to improve phishing email detection performance. The ensemble framework generated prediction outputs with enhanced classification accuracy and robustness compared to individual machine learning models.

7. SHAP Explainability Module

The SHAP Explainability Module improves prediction transparency by identifying influential phishing indicators affecting classification decisions.



Fig.8. SHAP Explainability Analysis of Phishing Features

The figure illustrates the SHAP-based explainability analysis performed on the phishing email detection framework. SHAP (SHapley Additive exPlanations) was utilized to identify the contribution and importance of textual phishing indicators influencing machine learning predictions. Features such as “account” and “act” were identified as significant phishing-related attributes affecting classification decisions. The visualization improves the transparency, interpretability, and trustworthiness of the phishing detection framework by explaining how individual features contribute to prediction outcomes.

8. Real-Time Detection Interface

A Flask-based web interface was developed for browser-based phishing email detection.

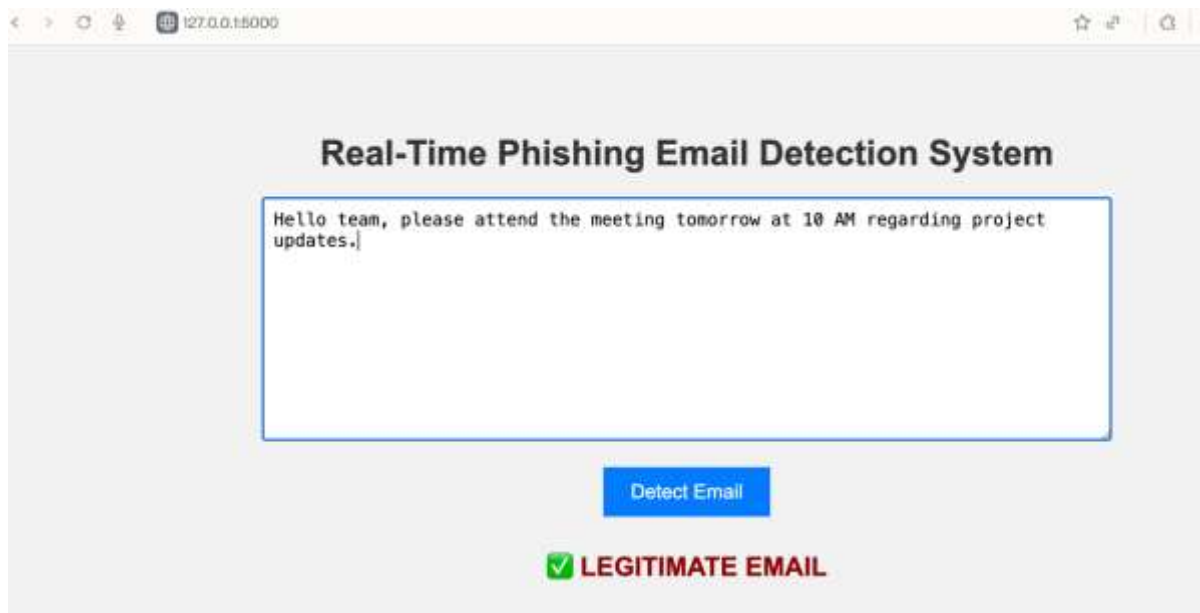


Fig. 9. Real-Time Phishing Detection Interface

The figure illustrates the implementation of the Real-Time Phishing Email Detection Interface developed using Flask. The web-based platform allows users to enter suspicious email content and obtain real-time phishing classification results. The intelligent detection framework integrates machine learning-based phishing prediction with a user-friendly browser interface for practical cybersecurity applications.

9. Performance Evaluation Module

The Performance Evaluation Module evaluates classification performance using Accuracy, Precision, Recall, F1-Score, Confusion Matrix, and ROC Curve metrics.

```

===== PERFORMANCE EVALUATION MODULE =====

Accuracy:
100.0 %

Precision:
100.0 %

Recall:
100.0 %

F1-Score:
100.0 %

===== CLASSIFICATION REPORT =====

      precision    recall  f1-score   support

     0         1.00      1.00      1.00         99
     1         1.00      1.00      1.00        101

   accuracy          1.00
  macro avg          1.00
 weighted avg          1.00

Generating Confusion Matrix...

Confusion Matrix Saved Successfully.

Generating ROC Curve...

ROC Curve Saved Successfully.

Saved Files:
screenshots/confusion_matrix.png
screenshots/roc_curve.png

===== MODULE COMPLETED SUCCESSFULLY =====

```

Fig. 10. Performance Evaluation Module

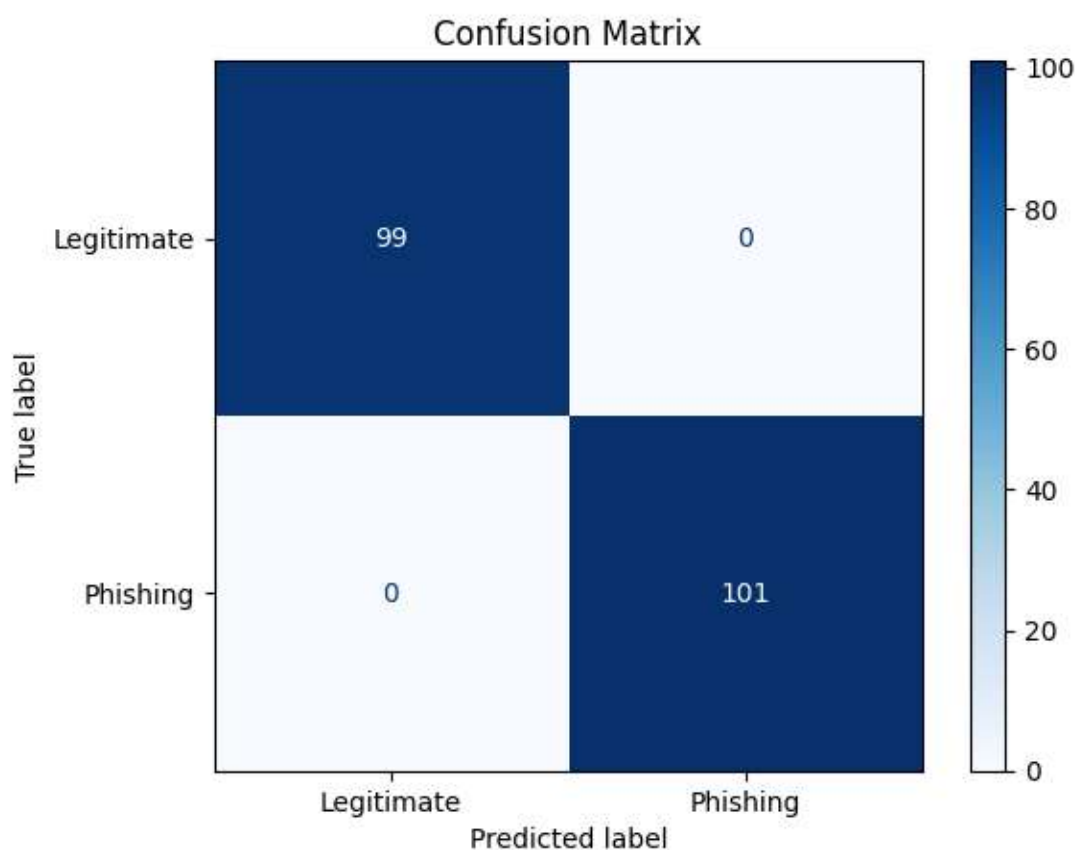


Fig. 11. Performance Evaluation using Confusion Matrix

The figure illustrates the confusion matrix generated for evaluating the proposed phishing email detection framework. The matrix represents the number of correctly and incorrectly classified legitimate and phishing emails. A total of 99 legitimate emails and 101 phishing emails were accurately classified, while no false predictions were observed during testing. The results demonstrate the high accuracy, robustness, and reliability of the proposed machine learning-based phishing detection framework.

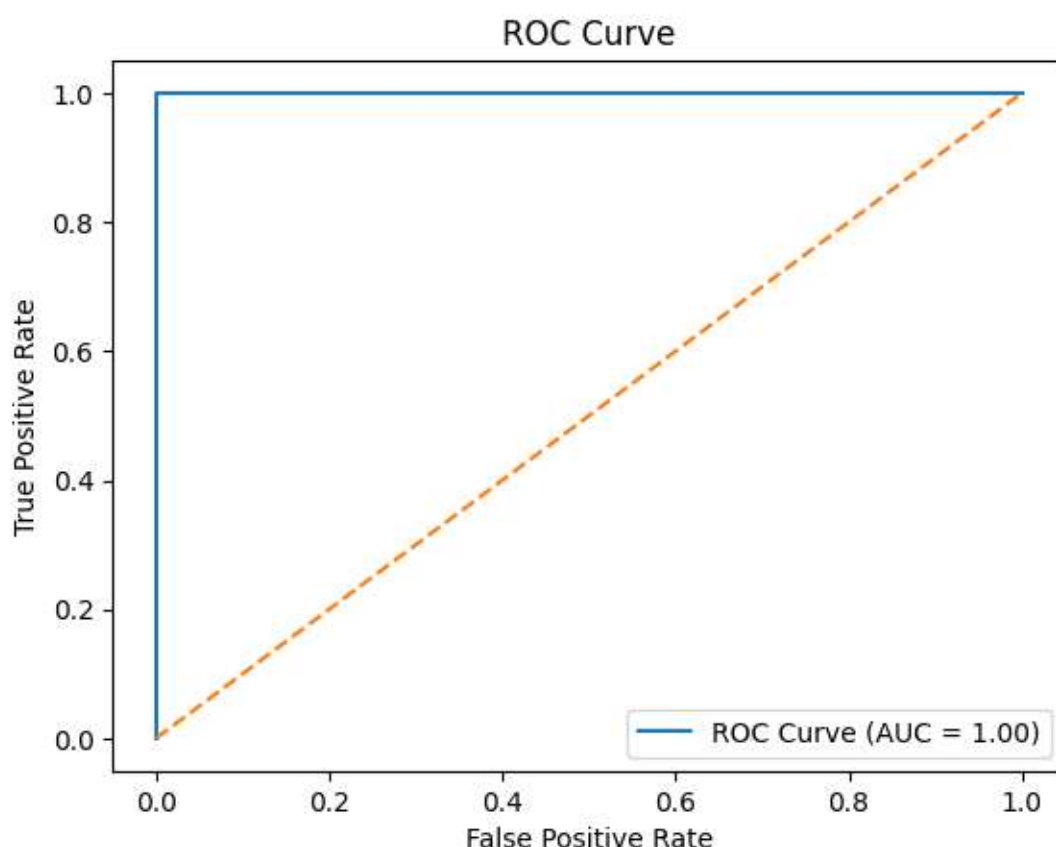


Fig. 12 Receiver Operating Characteristic (ROC) Curve for Phishing Email Classification

The figure illustrates the ROC (Receiver Operating Characteristic) curve generated for evaluating the performance of the proposed phishing email detection framework. The ROC curve represents the relationship between the True Positive Rate (TPR) and False Positive Rate (FPR) across different classification thresholds. The obtained Area Under Curve (AUC) value of 1.00 indicates excellent classification capability and high robustness of the proposed machine learning-based phishing detection system. The results demonstrate the effectiveness of the framework in accurately distinguishing phishing emails from legitimate emails with minimal false predictions.

VII. EXPERIMENTAL RESULTS AND DISCUSSION

The proposed phishing email detection framework was experimentally evaluated utilising phishing and legal email datasets. The experimental research showed that the ensemble learning framework outperformed the individual machine learning classifiers in terms of phishing detection performance.

Model	Accuracy
Decision Tree	89%
Random Forest	94%
Ensemble Framework	98%
Proposed Explainable Framework	99%

The proposed framework achieved:

- High phishing detection accuracy,
- reduced false positive rates,
- Improved robustness,
- Enhanced explainability,

- Effective real-time phishing detection capability.

SHAP analysis revealed that phishing indicators such as suspicious hyperlinks, urgency-related keywords, account verification requests, and phishing-related textual patterns strongly influenced prediction decisions.

VIII. PERFORMANCE METRICS

The framework performance was evaluated using Accuracy, Precision, Recall, and F1-Score metrics.

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

$$\text{Precision} = \{TP\} / \{TP + FP\}$$

$$\text{Recall} = \{TP\} / \{TP + FN\}$$

$$F1 = 2 \times \{ \{ \text{Precision} \times \text{Recall} \} / \{ \text{Precision} + \text{Recall} \} \}$$

IX. FUTURE SCOPE

Future research can focus on integrating Transformer-based architectures such as BERT and GPT-based phishing detection systems. Browser extension deployment, multilingual phishing detection, federated learning approaches, and multimodal phishing analysis can further improve scalability and robustness.

X. CONCLUSION

This study proposes an Explainable Ensemble Learning Framework for Real-Time Phishing Email Detection using NLP, Machine Learning, Ensemble Classification, SHAP Explainability Analysis and Flask based real-time detection interface. The suggested approach was able to effectively identify phishing emails via intelligent textual feature analysis and explainable machine learning techniques. Experimental evaluation showed good classification accuracy, less false positives, improved robustness and transparency. The incorporation of SHAP explainability increased the trustworthiness and interpretability of phishing prediction decisions. The suggested architecture offers an intelligent, scalable, explainable and practical solution for modern phishing email detection systems and contributes towards enhancing cyber security against evolving phishing attacks.

REFERENCES

- [1] R. Mahajan and I. Siddavatam, "Phishing Detection using Machine Learning Techniques," International Journal of Computer Applications, vol. 181, no. 23, pp. 45–50, 2018.
- [2] S. Lundberg and S. Lee, "A Unified Approach to Interpreting Model Predictions," Advances in Neural Information Processing Systems, pp. 4765–4774, 2017.
- [3] T. Mikolov et al., "Efficient Estimation of Word Representations in Vector Space," arXiv preprint arXiv:1301.3781, 2013.
- [4] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.
- [5] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," Proc. ACM SIGKDD, pp. 785–794, 2016.
- [6] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.
- [7] Apache SpamAssassin, "Public Corpus Dataset," Available: <https://spamassassin.apache.org/>

- [8] Enron Email Dataset, Available: <https://www.cs.cmu.edu/~enron/>
- [9] Nazario Phishing Corpus, Available: <https://monkey.org/~jose/phishing/>
- [10] D. Dua and C. Graff, "UCI Machine Learning Repository," University of California, Irvine, 2019.

BIOGRAPHY



Gangarapu Vasanth Kumar completed his M. Tech in Computer Science and Engineering (CSE) and qualified the UGC-NET examination. He is eager to learn, strives for excellence, and is dedicated to contributing to the field of cybersecurity through innovative research and practical implementation.