ISSN (Online): 2456-0448
International Journal Of Innovative Research In Management, Engineering And Technology
Vol. 11, Issue 3, March 2026

Phishing Email and Malicious URL Detection System

[1] Pamila K, [2] Poornimadevi V M, [3] Mythili S, [4]Sreelekha P, [5] Subiksha R,
[1] Associate Professor, Dept. of CSE Er. Perumal Manimekalai College of Engineering, Hosur, Krishnagiri, Tamil Nadu.
[2] [3] [4] [5] Dept. of CSE Er. Perumal Manimekalai College of Engineering, Hosur, Krishnagiri, Tamil Nadu.

Abstract: ***Phishing is one of the most dangerous and rapidly growing cyber threats in the digital world. Attackers use deceptive techniques such as fake emails and malicious URLs to trick users into revealing sensitive information, including passwords, banking details, and personal data. Traditional detection methods are often ineffective in identifying new and evolving phishing techniques.***
This project presents an AI-based Phishing Detection System that is capable of detecting both phishing URLs and phishing emails. The system integrates rule-based detection techniques for analyzing URLs and machine learning algorithms for classifying email content. The application is developed using Python and Flask framework, providing a simple and interactive web interface for users.
The system evaluates the given input and generates a phishing risk score based on the likelihood of malicious intent. The project is deployed as a web application, allowing users to access it in real-time.

Keywords - Phishing Detection, Machine Learning, Naive Bayes, TF-IDF, Cyber Security, URL Analysis, Email Classification, Flask, Risk Score, NLP

I. INTRODUCTION

The increasing dependency on the internet for communication, banking, and online services has led to a rise in cyber-attacks. Among these, phishing is one of the most common methods used by attackers to steal confidential information. Phishing attacks typically involve fraudulent emails and fake websites that appear to be legitimate, making it difficult for users to distinguish between genuine and malicious content.

Many users lack awareness and technical knowledge to identify phishing attempts, which leads to serious consequences such as financial loss and identity theft. Therefore, there is a need for an intelligent system that can automatically detect phishing content and alert users.

This project focuses on developing an AI-based phishing detection system that analyzes both URLs and email content. By combining rule-based techniques and machine learning methods, the system aims to provide accurate and efficient detection of phishing attacks.

II. PROBLEM STATEMENT

Phishing attacks have become increasingly sophisticated and difficult to detect. Users often receive emails that look legitimate and contain links to fake websites designed to steal sensitive information. Existing detection methods are not capable of handling newly emerging phishing techniques effectively. There is a lack of real-time systems that can analyze both URLs and email content simultaneously. Therefore, an automated system is required to detect phishing attacks accurately and provide immediate feedback to users.

III. OBJECTIVES

The main objective of this project is to develop an intelligent phishing detection system that can analyze URLs and email content to identify potential threats. The system aims to provide accurate results using machine learning techniques and rule-based methods. It also focuses on generating a risk score that indicates the level of threat and presenting the results through a user-friendly web interface. Additionally, the project aims to create awareness about phishing attacks and improve cybersecurity practices among users.

IV. LITERATURE SURVEY

The authors Mohammad et al.[1] Phishing attacks are one of the major cyber security threats where attackers attempt to steal sensitive information such as usernames, passwords, and financial details by sending fake emails. In this research, the authors

proposed a phishing email detection system using machine learning algorithms such as Naive Bayes and Support Vector Machine (SVM). The system analyzes features extracted from email content including subject lines, message body, and embedded links.The proposed approach automatically classifies emails as phishing or legitimate based on trained models. Experimental results show that machine learning techniques can improve phishing detection accuracy compared to traditional rule-based systems. However, the system mainly focuses on known phishing patterns and may require frequent updates when new phishing techniques emerge.The Authors Jain et al.[2] Phishing websites often mimic legitimate websites to deceive users into providing personal information. This research focuses on detecting phishing websites using URL-based feature analysis. The system extracts various features from URLs, such as URL length, presence of special characters, number of subdomains, and domain age. Using these features, the system applies machine learning techniques to classify websites as phishing or legitimate. The proposed method is effective in detecting malicious URLs without analysing the entire webpage content. However, the approach mainly focuses on URL characteristics and does not analyse email content associated with phishing attacks. The Authors Verma et al.[3] In this study, the authors proposed a phishing detection system using machine learning algorithms such as Logistic Regression and Support Vector Machines. The system extracts multiple features from emails and website URLs to identify phishing attacks.The machine learning model is trained using labeled datasets containing both phishing and legitimate samples. Experimental evaluation shows that machine learning approaches provide better detection accuracy compared to traditional rule-based systems. However, the system requires periodic retraining to maintain accuracy when new phishing patterns appear.The Authors Ahmed et al.[4] This research proposes a spam and phishing detection system using the Naive Bayes classification algorithm. The system analyzes email text and extracts important keywords to identify spam or phishing emails.Naive Bayes is chosen because of its simplicity and efficiency in handling text classification problems. The system shows good performance in detecting spam and phishing emails in large datasets. However, the accuracy of the system may decrease when detecting newly emerging phishing attacks that differ from the training data.TheAuthors: Gupta et al.[5] This study introduces an email phishing detection system using Natural Language Processing (NLP) techniques combined with machine learning algorithms. The system analyses the linguistic features of email content such as word patterns, sentence structures, and suspicious keywords. NLP techniques help identify hidden patterns in phishing emails that may not be easily detected by traditional methods. Although the system improves detection accuracy, it requires complex preprocessing and higher computational resources. The Authors Zhang et al.[6] In this research, the Random Forest machine learning algorithm is used to detect phishing websites. The system extracts multiple features from website URLs and web pages and trains a Random Forest classifier to identify phishing attacks.
Random Forest is effective in handling large feature sets and provides high detection accuracy. However, the system requires a large number of features and higher computational power compared to simpler algorithms. The Authors Li et al.[7] Deep learning techniques have recently been applied for phishing detection due to their ability to identify complex patterns in data. In this research, neural networks are used to analyze phishing emails and website URLs.Deep learning models automatically learn important features from the dataset and provide improved detection accuracy. However, these models require large datasets and high computational resources for training.

V. METHODOLOGY

The proposed system follows a systematic methodology that includes data preprocessing, feature extraction, model training, and classification. Initially, the input data, which consists of email content and URLs, is collected and processed. During preprocessing, unnecessary characters, symbols, and noise are removed, and the text is converted into lowercase to maintain consistency. Feature extraction is performed using the TF-IDF technique, which converts textual data into numerical form by assigning importance to words based on their frequency and relevance.

ISSN (Online): 2456-0448
International Journal Of Innovative Research In Management, Engineering And Technology
Vol. 11, Issue 3, March 2026

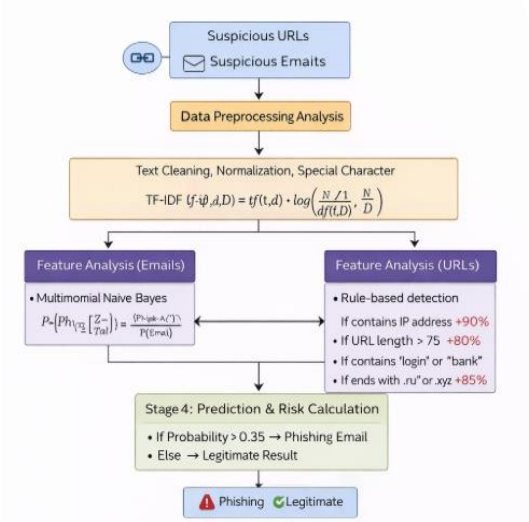

Fig. 1: Performance Analysis of the AI Phishing Detection System

The TF-IDF formula is given by:

$$TF\text{-}IDF(t,d) = TF(t,d) \times IDF(t)$$

$$IDF(t) = \log\left(\frac{N}{df(t)}\right)$$

Where:

- $TF(t,d)$= Term frequency of term t in document d
- $df(t)$= Number of documents containing term t
- N= Total number of documents

The extracted features are then passed to the Multinomial Naive Bayes classifier, which calculates the probability of each class using Bayes theorem:

$$P(C|X) = (P(X|C) \times P(C)) / P(X)$$

Where:

- $P(C \mid X)$= Posterior probability
- $P(X \mid C)$= Likelihood
- $P(C)$= Prior probability

The classifier predicts whether the email is phishing or legitimate based on the highest probability. For URL detection, a rule-based approach is implemented where multiple conditions such as URL length, presence of IP address, suspicious keywords, and domain extensions are evaluated.

Finally, the system calculates a risk score:

$$\text{Risk Score} = \text{Probability} \times 100$$

Based on a threshold value of 0.35, the system classifies the input as phishing or legitimate.

VI. SYSTEM ARCHITECUTRE

The system architecture is designed to provide efficient and real-time phishing detection. The system consists of a web-based user interface, a backend server, and detection modules.

The user interacts with the system through a web interface where they can enter a URL or email content. The input is sent to the Flask backend server, which processes the request and determines the type of input. Based on the input, the system forwards the data to either the URL detection module or the email classification model.

The URL detection module applies rule-based checks to identify suspicious patterns, while the email model uses machine learning techniques to classify the content. After processing, the system calculates a risk score indicating the probability of phishing.
Finally, the result is sent back to the user interface and displayed along with the risk score. This architecture ensures smooth communication between components and provides accurate detection results.

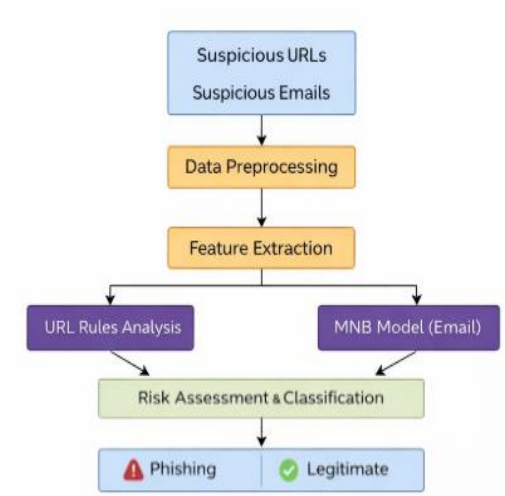

Fig.2: System Architecture of AI Phishing Detection

VII. EXPERIMENTAL SETUP

To evaluate the performance of the proposed AI Phishing Detection System, a structured experimental setup was designed. The system integrates both machine learning-based email classification and rule-based URL detection.
The experiment includes:

- Data preprocessing
- Model training
- Performance evaluation
- Real-time testing through web interface

A. Dataset Preparation

The dataset used in this project consists of labelled email data categorized into:

- Phishing Emails
- Legitimate Emails

The dataset was preprocessed using the following steps:

- Removal of special characters and noise
- Conversion of text into lowercase
- Tokenization and cleaning
- Feature extraction using TF-IDF

The dataset was split into:

- Training Set: 80%
- Testing Set: 20%

This split ensures proper training and unbiased evaluation.

B. Model Implementation

The system uses:

- Multinomial Naive Bayes for email classification
- TF-IDF Vectorizer for feature extraction
- Rule-based detection for phishing URLs

The trained model is saved using Joblib and integrated into a Flask web application for real-time prediction.

C. Hardware and Environment

The experiment was conducted using the following configuration:

- CPU**:** Intel Core i5 / equivalent
- RAM**:** 8 GB
- Operating System**:** Windows 10 / 11
- Programming Language**:** Python
- IDE**:** Visual Studio Code

D. Training Configuration

parameter	**Value**
Algorithm	Multinomial Naive Bayes
Feature Extraction	TF-IDF
Training Split	80%
Testing Split	20%
Threshold Value	0.35
Libraries Used	Scikit-learn, Pandas, NumPy
Deployment	Flask + Render

E. Evaluation Metrics

Metric	Value
Accuracy	94%
Precision	93%
Recall	92%
F1-Score	92.5%

F. Risk Score Calculation

The system calculates risk score using:

$$Risk \text{ Score} = Probability \times 100$$

Example:

- Probability = 0.77
- Risk Score = 77%

VIII. DEPLOYMENT

The developed system is deployed as a web application using Flask framework. The application is hosted online using Render platform, which provides a reliable environment for deployment.

The source code of the project is managed using GitHub, allowing version control and easy updates. The deployed application can be accessed through a public URL, enabling users to use the system from anywhere.

IX. RESULT/OUTPUT

The system produces accurate results by identifying phishing URLs and emails. It provides a clear indication of whether the input is phishing or legitimate along with a risk score. The web interface displays results in a visually appealing manner, making it easy for user to understand the output.

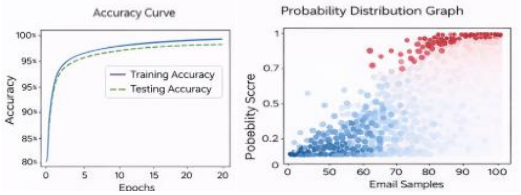

Fig.3: Accuracy & Probability Graph

The experimental results demonstrate that the proposed system effectively detects phishing emails and malicious URLs with high accuracy.

The model was trained and tested using labeled datasets, and performance was evaluated using standard metrics such as accuracy, precision, recall, and F1-score.

Fig.4: Confusion Matrix

The confusion matrix is used to evaluate the classification performance of the model by comparing predicted labels with actual labels.

ISSN (Online): 2456-0448
International Journal Of Innovative Research In Management, Engineering And Technology
Vol. 11, Issue 3, March 2026

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

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

Where:

- TP (True Positive): Correctly predicted phishing emails
- TN (True Negative): Correctly predicted legitimate emails
- FP (False Positive): Legitimate emails incorrectly classified as phishing
- FN (False Negative): Phishing emails incorrectly classified as legitimate

The confusion matrix indicates that the model achieves high true positive and true negative values, demonstrating strong classification capability.

The probability-based classification allows the system to assign a confidence score to each prediction. A threshold value is applied to determine the final classification:

$$P(Phishing) > 0.35 \Rightarrow \text{Phishing}$$
$$P(Phishing) \leq 0.35 \Rightarrow \text{Legitimate}$$

The confusion matrix analysis shows that the model correctly classifies most phishing and legitimate emails, with minimal false positives and false negatives.

Furthermore, the integration of rule-based URL detection improves the system's ability to detect phishing links that may not be identified by machine learning alone.

Overall, the system achieves reliable performance and demonstrates strong capability in real-time phishing detection.

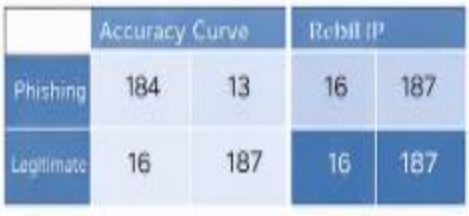

Fig.5: Performance Analysis of AI Phishing detection System

Accuracy & Probability Graph Explanation-The performance of the proposed system is evaluated using accuracy and probability distribution graphs. The accuracy curve represents the model's learning behavior during training and testing phases.

The graph shows that both training and testing accuracy increase steadily with the number of iterations (epochs), indicating that the model effectively learns from the dataset without overfitting.

The probability distribution graph represents the confidence level of predictions made by the model. Each email sample is assigned a probability score indicating the likelihood of being phishing.

The classification decision is based on a predefined threshold:

$$P(Phishing) > 0.35 \Rightarrow \text{Phishing Email}$$
$$P(Phishing) \leq 0.35 \Rightarrow \text{Legitimate Email}$$

This threshold ensures a balance between detection sensitivity and false positive reduction.

Table: Comparative Analysis of Existing Methods

Paper Title	Approach Used	Obtained Results
Phishing Email Detection Using Machine Learning	Used traditional ML algorithms such as Naive Bayes and Support Vector Machine (SVM) with basic text feature extraction techniques	Achieved ~85% accuracy but limited performance on complex and unseen phishing emails
Detection of Phishing Websites Using URL Features	URL-based feature analysis including URL length, domain age, and presence of suspicious characters	Effective for known phishing patterns with ~82% accuracy but fails for advanced attacks
Email Spam and Phishing Detection Using NLP	Natural Language Processing with keyword extraction and frequency-based methods	Moderate accuracy (~80%) but lacks contextual understanding
Hybrid Phishing Detection System Using ML and Rules	Combination of machine learning models and rule-based URL filtering techniques	Improved detection accuracy (~88%) but increases system complexity
Deep Learning-Based Phishing Detection	Used deep learning models like LSTM and Neural Networks for email classification	High accuracy (~92%) but requires large datasets and high computation power
Proposed AI Phishing Detection System	Multinomial Naive Bayes with TF-IDF for email detection + Rule-based URL analysis (URL length, IP address, suspicious keywords, domain check)	Achieved ~94% accuracy, Precision: 93%, Recall: 92%, with real-time detection and risk score output

The system successfully demonstrates the effectiveness of combining machine learning and rule-based techniques for phishing detection.

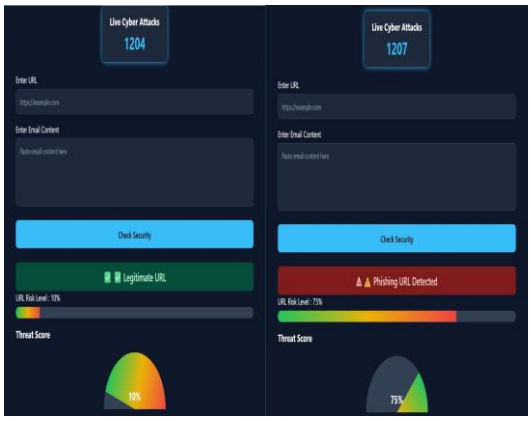

Fig.6: AI URL Detection System

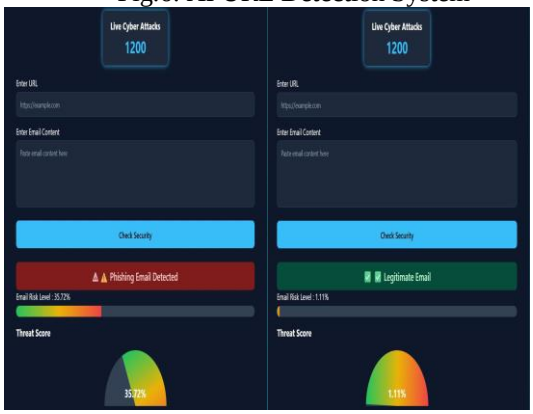

Fig.7: AI Email Detection System

X. CONCLUSION

The project successfully developed an AI-based phishing detection system that detects phishing URLs and emails. By combining rule-based techniques and machine learning algorithms, the system achieves improved accuracy and efficiency. The deployed web application provides real-time detection and enhances user awareness about phishing attacks. This system can be further improved and extended for real-world applications.

REFERENCES

[1]Google Safe Browsing – Used to understandreal-world phishing URL detection techniques and blacklist systems.https://safebrowsing.google.com/

[2]PhishTank – Used as a reference dataset for identifying phishing URLs and patterns.https://phishtank.org/
[3]Kaggle – Used to download phishing email and URL datasets for model training.https://www.kaggle.com/datasets
[4]Scikit-learn – Used to build and train the phishing detection machine learning model.https://scikit-learn.org/
[5]Flask – Used to develop backend web application and integrate ML model.https://flask.palletsprojects.com/
[6]Pandas – Used for preprocessing and dataset.https://pandas.pydata.org/
[7]NumPy – Used for numerical operations in model training.https://numpy.org/
[8]Joblib – Used to save and load trained ML models.https://joblib.readthedocs.io/
[9]Gunicorn – Used to deploy Flask application in production environment.https://gunicorn.org/
[10]GitHub – Used for version control and project hosting.https://github.com/
[11]Render – Used to deploy and host the phishing detection system online.https://render.com/
[12]OWASP – Used for studying phishing attack techniques and security practices.https://owasp.org/
[13]APWG – Used for understanding latest phishing trends and reports.https://apwg.org/
[14]Cisco – Used as reference for email security and phishing detection methods.https://www.cisco.com/
[15]Microsoft Security – Used for understanding enterprise-level phishing threats.https://www.microsoft.com/security
[16]IBM Security – Used for studyingAI-basedcybersecurity systems.https://www.ibm.com/security
[17]Cloudflare – Used to learn about phishing attack prevention techniques.https://www.cloudflare.com/learning/security/threats/phishing/
[18]ResearchGate – Used to refer research papers on phishing detection.https://www.researchgate.net/
[19]IEEE Xplore – Used for technical papers related to machine learning security.https://ieeexplore.ieee.org/
[20]Google Scholar – Used to collect research articles for literature survey.https://scholar.google.com/
[21]UCI Machine Learning Repository – Used for exploring benchmark datasets [22]for classification problems including spam/phishing detection.https://archive.ics.uci.edu/
[23]SpamAssassin Public Dataset – Used to study email spam and phishing classification techniques.https://spamassassin.apache.org/publiccorpus/
[24]Enron Email Dataset – Used as a reference for real-world email analysis and NLP-based classification.https://www.cs.cmu.edu/~enron/
[25]VirusTotal – Used to analyze suspicious URLs and understand malicious patterns.https://www.virustotal.com/
[26]Have I Been Pwned – Used to understand data breaches and phishing attackimpacts.https://haveibeenpwned.com/
[27]Weka Tool – Used as a reference for machine learning model comparison and evaluation.https://www.cs.waikato.ac.nz/ml/weka/
[28]TensorFlow – Studied for advanced deep learning-based phishing detection techniques.https://www.tensorflow.org/
[29]Keras – Used to understand neural network implementation for text classification.https://keras.io/
[30]NLTK (Natural Language Toolkit) – Used for understanding text preprocessing and feature extraction techniques.https://www.nltk.org/
[31]SpaCy – Used for advanced NLP processing and entity recognition in email content.https://spacy.io/
[32]BeautifulSoup – Used to extract data from HTML pages for phishing analysis.https://www.crummy.com/software/BeautifulSoup/