

Email Phishing Attack Detection Using Machine Learning and Deep Learning

Vaishnavi Sunil Jadhav¹, M. B. Jadhav², S. V. More³, H. B. Jadhav⁴
Adsul Technical Campus, Chas, Ahilyanagar, India¹⁻⁴

Abstract: Email phishing is a major cybersecurity threat in which attackers impersonate trusted individuals or organizations to trick users into revealing passwords, banking credentials and personal data. Traditional filters based on blacklists and fixed rules often miss new phishing messages that change their wording, links or sender identity. This paper proposes a comparative framework for phishing email detection using Machine Learning (ML) and Deep Learning (DL). The framework analyzes the sender, subject, message body, URLs and linguistic cues of each email. Logistic Regression, Support Vector Machine, Random Forest and Naive Bayes are compared with Long Short-Term Memory (LSTM) and transformer-based models under one dataset, one preprocessing pipeline and one held-out test set. Performance is to be assessed with accuracy, precision, recall, F1-score and confusion matrices, with particular attention to false negatives and false positives. The system labels an email as legitimate or phishing and reports a risk score together with the indicators that influenced the decision. This paper presents the design of the framework and its evaluation protocol. No experimental results are reported here, and the effectiveness of the approach must be confirmed through experiments on representative data

Keywords: Email Phishing Detection, Machine Learning, Deep Learning, Natural Language Processing, Cybersecurity, Email Classification

I. INTRODUCTION

Email is one of the most widely used means of communication in personal, educational and professional settings. Its popularity also makes it a prime target for cybercriminals. Phishing attacks exploit human trust by sending fraudulent messages that appear to come from legitimate organizations, banks, educational institutions or service providers [1]. Such emails often contain deceptive hyperlinks, fake login pages, malicious attachments, urgent requests and misleading sender identities. A successful attack can lead to identity theft, financial loss, unauthorized account access and exposure of confidential information.

Traditional detection relies on blacklists, predefined rules and manually selected features. These methods catch known threats but may fail on new messages that change their wording, URLs or visual appearance. Machine learning can learn patterns from labeled email data, and deep learning can learn rich textual and contextual representations. Combining them supports a more adaptive classification system. This work designs a framework that compares ML and DL methods for phishing email detection, covering data preprocessing, feature extraction, model training, performance evaluation and the identification of suspicious emails.

A. Problem Statement

Existing email security systems may struggle to detect previously unseen phishing attacks, sophisticated social-engineering messages and deceptive content. A reliable detection framework is needed that analyzes several email features together, separates legitimate communication from phishing attempts and keeps false positives under control.



B. Objectives

1. To study the characteristics and techniques of email phishing attacks.
2. To preprocess email datasets and extract relevant textual and structural features.
3. To implement and compare selected machine learning algorithms.
4. To evaluate deep learning approaches for contextual email analysis.
5. To measure detection performance using standard classification metrics.
6. To investigate false positives and false negatives.
7. To propose an interpretable framework for practical email security applications.

C. Scope of the Study

The study classifies email messages as legitimate or phishing using the email text, sender-related attributes and URL characteristics when available. The framework is intended for research and defensive cybersecurity use. It does not claim to replace existing email security infrastructure.

II. LITERATURE REVIEW

Research on phishing detection has explored URL analysis [2], [3], website and hyperlink characteristics [4], email content classification and machine-learning-based security systems. Early approaches relied on manually defined rules, databases of known malicious URLs and suspicious-keyword detection. These are simple to implement but struggle with attacks that do not match previously identified patterns.

Machine learning offers a data-driven alternative. Algorithms such as Logistic Regression, Support Vector Machine and Random Forest learn relationships between extracted features and phishing labels, and comparative studies have examined their behavior on phishing data [5]. Their effectiveness depends on dataset quality, feature representation, class balance and the diversity of attack examples.

Deep learning goes further by learning representations directly from text. Recurrent networks, including LSTM, model sequential information, while transformer-based language models capture contextual relationships between words and phrases. Open challenges remain: imbalanced datasets, changing phishing tactics, misleading sender information, limited generalization to unseen attacks and the risk of misclassifying legitimate emails.

A. Research Gap

A meaningful comparison requires every model to be evaluated under the same dataset, preprocessing pipeline, train-test separation and metrics. This study organizes these elements into one experimental framework and examines both detection performance and classification errors.

III. PROPOSED METHODOLOGY

The proposed system follows a structured pipeline of dataset collection, preprocessing, feature extraction, model training, evaluation and final classification (Fig. 1).



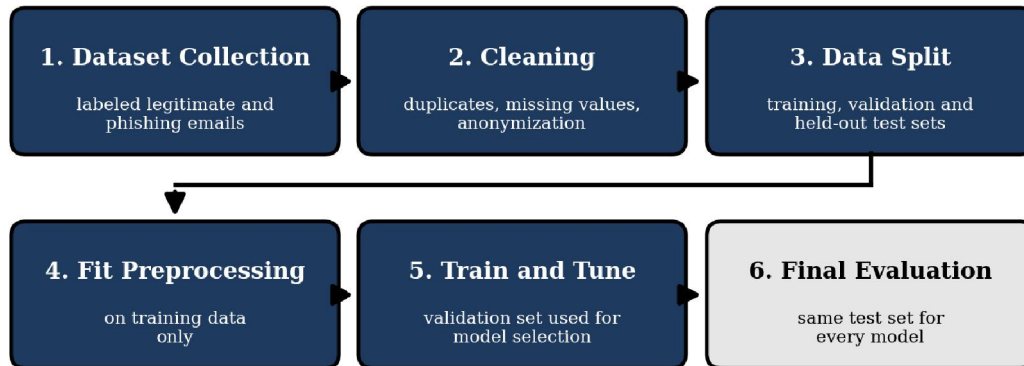


Fig. 1. Experimental pipeline of the proposed framework.

A. Dataset Collection

A labeled dataset of legitimate and phishing emails will be selected from a reputable research dataset or public cybersecurity repository. Each record should contain an email message and its class label. Where available, the sender address, subject, body, hyperlinks and attachment metadata are also used. Personal information must be removed or anonymized before analysis. The dataset is checked for duplicate messages, missing values, mislabeled samples and class imbalance.

B. Data Preprocessing

The following operations prepare the data for training:

- 1) Remove duplicate and invalid records.
- 2) Handle missing values and malformed email fields.
- 3) Normalize text where appropriate.
- 4) Extract the subject and body of each email.
- 5) Identify URLs and suspicious hyperlink patterns.
- 6) Convert text into numerical representations.
- 7) Encode class labels consistently.
- 8) Split the data into training, validation and test sets.

The test set is kept separate from model training and hyperparameter selection to reduce the risk of data leakage.

C. Feature Extraction

The framework investigates the five feature categories in Table I. No single feature is treated as proof of phishing; classification relies on the combined evidence available to the model.

TABLE I: Feature Categories

Category	Examples	Typical source
Textual	Suspicious phrases, urgency, requests for credentials, misleading calls to action	Subject and body
URL	URL length, number of subdomains, use of IP addresses, unusual symbols, suspicious domain patterns	Hyperlinks
Sender	Sender-domain traits, mismatch between displayed name and actual address	Email headers
Structural	Number of hyperlinks, attachments, unusual formatting, metadata	Message structure
Semantic	Contextual relationships captured by text embeddings or pretrained language models	Body text



D. Machine Learning Models

Four traditional algorithms are proposed for comparison. *Logistic Regression* is a baseline that estimates the probability that an email is phishing. The *Support Vector Machine* (SVM) [6] finds a decision boundary between the two classes in a numerical feature space. *Random Forest* [7] combines many decision trees and can model nonlinear relationships among features. *Naive Bayes* is a probabilistic baseline that can work well for text classification with suitable text representations.

E. Deep Learning Models

The *Long Short-Term Memory* (LSTM) network [8] learns sequential relationships in email text and can pick up contextual patterns associated with phishing. A pretrained *transformer-based* language model [9], [10] can be fine-tuned for binary email classification so that it analyzes context and relationships between words. Deep models usually need more computing resources and labeled data than traditional ones, and their actual performance must be established experimentally.

F. Model Training

The dataset is divided into training, validation and test subsets. Models are trained on the training subset, while the validation subset supports model selection and hyperparameter tuning. For a fair comparison, all models use the same held-out test set. Class imbalance can be handled with class weights or training-only resampling. Preprocessing and feature selection are fitted on the training data, not on the complete dataset.

G. Email Classification

After training, the system accepts an email, processes its available features and outputs one of two classes: *Legitimate Email* (the model produced no phishing classification) or *Phishing Email* (the model detected patterns associated with phishing). The application may also show a risk score and the features that contributed to the prediction. A prediction is a risk indicator, not a guarantee that an email is safe or malicious.

IV. SYSTEM ARCHITECTURE

The architecture has six stages, shown in Fig. 2. Stage 1 receives an email message or a permitted sample. Stage 2 extracts and cleans the subject, body, sender information, URLs and available metadata. Stage 3 generates numerical and textual representations. Stage 4 passes them to the selected ML or DL classifier. Stage 5 produces a class label, a confidence or risk score and relevant warning indicators. Stage 6 presents the result and recommends caution when suspicious characteristics are found.

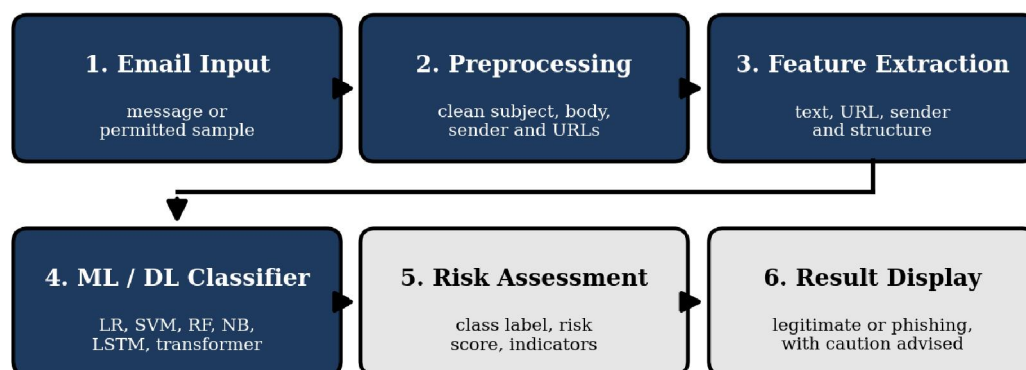


Fig. 2. Proposed system architecture.

DOI: 10.48175/IJARSCT-40129



A. Proposed Technology Stack

The proposed implementation uses Python with Pandas and NumPy for data processing, Scikit-learn for traditional machine learning, TensorFlow/Keras or PyTorch for deep learning, NLTK, spaCy or transformer tokenizers for text processing, and Matplotlib and Seaborn for visualization. Streamlit or a web interface may serve as an optional front end. These are proposed implementation choices, not a claim that a completed application already uses them.

V. PERFORMANCE EVALUATION

Models are evaluated with standard classification metrics computed on the held-out test set (Table II). TP, TN, FP and FN denote true positives, true negatives, false positives and false negatives, with phishing as the positive class.

TABLE II: Evaluation Metrics

Metric	Formula	Meaning
Accuracy	$(TP + TN) / (TP + TN + FP + FN)$	Share of emails classified correctly
Precision	$TP / (TP + FP)$	Share of predicted phishing emails that are truly phishing
Recall	$TP / (TP + FN)$	Share of actual phishing emails that are detected
F1-score	$2 \times (Precision \times Recall) / (Precision + Recall)$	Harmonic mean of precision and recall

A confusion matrix summarizes TP, TN, FP and FN. False negatives are especially important because they are phishing emails classified as legitimate. False positives must also be monitored because they cause legitimate messages to be flagged.

The final experiments will compare all selected algorithms on the same test set and report accuracy, precision, recall, F1-score, training time and inference time (Table III). The table below is the reporting template; its cells are marked TBD (to be determined) because no experiment has been completed.

TABLE III: Planned Comparison (To Be Completed After Experiments)

Model	Accuracy	Precision	Recall	F1	Train time	Infer. time
Logistic Regression	TBD	TBD	TBD	TBD	TBD	TBD
SVM	TBD	TBD	TBD	TBD	TBD	TBD
Random Forest	TBD	TBD	TBD	TBD	TBD	TBD
Naive Bayes	TBD	TBD	TBD	TBD	TBD	TBD
LSTM	TBD	TBD	TBD	TBD	TBD	TBD
Transformer	TBD	TBD	TBD	TBD	TBD	TBD

VI. RESULTS AND DISCUSSION

This paper presents the framework and evaluation protocol only. No numerical results are reported because no experiment on a measured dataset has been completed, and no accuracy values are assumed. The results section of the final study will present the measured performance of each implemented model and discuss the observed differences.

That analysis should address whether deep learning improves recall or F1-score over traditional machine learning, whether the added complexity brings meaningful benefits and how the models behave on difficult or previously unseen phishing messages. A confusion matrix will be given for the selected model, and a comparison table will report the same metrics for every algorithm. If resources permit, further experiments can examine different datasets or phishing categories.

The discussion must also consider dataset bias, changes in phishing techniques, misleading labels, class imbalance, computational cost and how well the models generalize to real-world emails. Conclusions about effectiveness must rest on measured results; until those experiments are done, the effectiveness of the framework remains to be verified.



VII. ADVANTAGES AND LIMITATIONS

A. Advantages

1. Supports automated email classification.
2. Combines textual and structural email characteristics.
3. Allows traditional and deep learning methods to be compared fairly.
4. Can be updated by retraining on appropriately labeled data.
5. Provides measurable performance through standard metrics.
6. Can be integrated into a research prototype or email-security workflow.

B. Limitations

1. Performance depends on dataset quality and diversity.
2. New phishing strategies may not resemble the training examples.
3. False positives and false negatives cannot be eliminated completely.
4. Deep learning models may need additional computational resources.
5. Sender and URL features may be missing from some datasets.
6. A classification result cannot by itself guarantee email safety.

VIII. CONCLUSION

Email phishing remains a significant cybersecurity challenge because attackers keep changing message content, sender identities and malicious links. This paper proposed a comparative framework for detecting phishing emails with machine learning and deep learning. The framework covers dataset preparation, preprocessing, feature extraction, model training, binary classification and performance evaluation. Logistic Regression, SVM, Random Forest and Naive Bayes can be compared with LSTM and transformer-based models using accuracy, precision, recall and F1-score. The approach aims to make phishing detection more systematic and measurable. Its practical value must be established through reproducible experiments on representative datasets, with particular attention to false negatives, false positives and performance on previously unseen emails. Future work may explore explainable artificial intelligence, multilingual phishing detection, continuous learning, real-time email integration and robust evaluation against evolving phishing techniques.

ACKNOWLEDGMENT

The authors thank the International Centre of Excellence in Engineering and Management (ICEEM), Chhatrapati Sambhajanagar, and the faculty of the Department of Computer Science and Engineering for their guidance and support. They also acknowledge the researchers and dataset providers whose publicly available work supports the study.

REFERENCES

- [1] M. Khonji, Y. Iraqi, and A. Jones, "Phishing detection: A literature survey," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 4, pp. 2091–2121, 2013.
- [2] S. Marchal, J. François, R. State, and T. Engel, "PhishStorm: Detecting phishing with streaming analytics," *IEEE Trans. Netw. Service Manag.*, vol. 11, no. 4, pp. 458–471, 2014.
- [3] O. K. Sahingoz, E. Buber, O. Demir, and B. Diri, "Machine learning based phishing detection from URLs," *Expert Syst. Appl.*, vol. 117, pp. 345–357, 2019.
- [4] A. K. Jain and B. B. Gupta, "A machine learning based approach for phishing detection using hyperlinks information," *J. Ambient Intell. Humaniz. Comput.*, vol. 10, pp. 2015–2028, 2019.
- [5] S. Abu-Nimeh, D. Nappa, X. Wang, and S. Nair, "A comparison of machine learning techniques for phishing detection," in *Proc. Anti-Phishing Working Groups 2nd Annu. eCrime Researchers Summit*, 2007, pp. 60–69.



- [6] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.
- [7] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
- [8] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [9] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.
- [10] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL-HLT*, 2019, pp. 4171–4186.

