Detection Of E-mail Phishing Attacks Using Machine Learning And Deep Learning

[1]Kanchan Dhondiba Waghmare, [2]Prof.S.Khan

[1]Student, [2]Assistant Professor
[1]Computer Science and Engineering(Cyber Security),
[1]PES College of Engineering, Chh.Sambhajinagar, India
[1]kanchanwaghmare2002@gmail.com, [2]khan.sanazarrin@gmail.com

Abstract*— Malicious email campaigns that impersonate trusted entities to harvest credentials, disseminate malware, and perpetrate financial fraud continue to impose severe economic and reputational damage on individuals and organizations worldwide. Conventional rule-based and blacklist-driven countermeasures demonstrate an inherent inability to generalize beyond catalogued threat signatures, leaving systems unprotected against emergent and zero-day phishing variants. This paper introduces Detection of E-mail Phishing Attacks Using Machine Learning and Deep Learning, a novel hybrid ensemble framework that synergistically combines four classical Machine Learning (ML) classifiers — Random Forest (RF), Support Vector Machine (SVM) with Radial Basis Function kernel, Decision Tree (DT), and Logistic Regression (LR) — with two Deep Learning (DL) architectures — Long Short-Term Memory (LSTM) and one-dimensional Convolutional Neural Network (CNN). A structured pipeline extracts twenty-six discriminative features spanning five semantic dimensions: URL structural patterns, HTML document composition, email content semantics, sender behavioral metadata, and textual statistical properties. Individual model probability estimates are aggregated through a calibrated arithmetic ensemble mechanism to yield a continuous phishing risk score, which drives a three-tiered policy enforcement layer (LOW, MEDIUM, HIGH) integrated directly into a production Flask web application. Experiments conducted on a benchmark dataset of 18,650 labeled email records yield 97.80% accuracy, 98.10% precision, 97.50% recall, and 0.9982 area under the ROC curve for the Random Forest component; the LSTM model achieves 95.60% accuracy. The ensemble reduces the total misclassification rate to 2.07%, outperforming all constituent single-model baselines. The real-time operational deployment, comprehensive phishing log database, and interpretable alert generation distinguish this system from prior research-only evaluations, addressing a critical gap in practical cybersecurity tooling.

***Keywords* — Email phishing detection; machine learning; deep learning; LSTM; convolutional neural network; random forest; SVM; ensemble classification; feature engineering; cybersecurity; real-time detection; NLP.**

I. INTRODUCTION

Electronic mail is one of the most widely used forms of communication in both business and personal settings. As of 2024, the total number of emails sent worldwide each day has surpassed 347 billion. Because of its widespread use and ease of access, email has become a common target for cybercriminals who use phishing attacks. These attacks involve sending deceptive messages that appear to come from trustworthy sources, tricking recipients into revealing sensitive personal or financial information, or clicking on harmful attachments that can compromise their computer systems.

According to data from the Anti-Phishing Working Group (APWG) and the Federal Bureau of Investigation's Internet Crime Complaint Center (IC3), phishing is the most commonly reported type of cybercrime. In 2023, global financial losses from phishing attacks exceeded 17.5 billion US dollars, with the healthcare, financial services, and government sectors experiencing the most significant impacts. The COVID-19 pandemic led to a dramatic increase in phishing attempts that exploited the social and technological changes brought about by the crisis, showcasing how quickly cybercriminals adapt to new situations. The recent availability of large language models has further enhanced the ability of cybercriminals to create highly convincing and well-written phishing content on a large scale, making it much harder for traditional spam filters to detect these threats.

Current automated defenses against phishing can be divided into three main categories. (i) reputation-based blacklisting systems are effective at identifying known threats but struggle with new or short-lived phishing domains. (ii) rule-based systems use manually created patterns to detect threats, but these patterns are easily bypassed by slight changes to the message structure. (iii) authentication protocols like SPF, DKIM, and DMARC help prevent spoofing at the network level, but they can be circumvented when attackers use legitimate accounts or services.

These limitations highlight the need for more advanced, data-driven detection methods based on machine learning. This paper introduces PhishGuard AI, a comprehensive hybrid system that combines traditional machine learning (ML) and modern deep learning (DL) techniques within an email security platform. The main contributions of this research are:

1) A 26-dimensional feature extraction process that includes analyzing URL structure, HTML content, semantic signals from the message text, sender metadata, and statistical text properties.

2) A six-model ensemble system that combines Random Forest, Support Vector Machine, Decision Tree, Logistic Regression, LSTM, and CNN models, using a probability averaging method that significantly reduces misclassification rates compared to individual models.

3) The integration of real-time ensemble analysis into a Flask web application, which enforces a three-tier risk response and maintains detailed logs — this is the first operational system of this type described in existing literature.

4) A thorough evaluation on an 18,650-sample benchmark dataset showing an accuracy of 97.80% and a ROC-AUC score of 0.9982 for the primary ML classifier and 95.60% accuracy for the LSTM deep learning model.

II. Related Work

Research into automated phishing email detection has progressed through several successive paradigms over two decades of academic inquiry. Early contributions concentrated on probabilistic text classifiers: the Naive Bayes (NB) model established that bag-of-words representations of email content could yield adequate discrimination for high-volume spam filtering with negligible computational overhead. Subsequent investigations demonstrated, however, that the conditional independence assumption underpinning NB is systematically violated by the structured vocabulary of phishing campaigns, in which specific phrases co-occur with reliably discriminative frequency.

Support Vector Machines emerged as a compelling alternative through their theoretical guarantee of margin maximisation in the transformed feature space. Multiple empirical evaluations established that SVM with RBF kernel consistently achieved accuracy in the range of 95-97% on email classification benchmarks, with particular robustness in high-dimensional feature spaces. Random Forest, introduced through the ensemble generalisation of bagged decision trees, subsequently demonstrated superior aggregate performance by capturing non-linear interactions among heterogeneous feature types — URL counts, HTML form occurrences, and urgency keyword densities — that no linear classifier could adequately model.

The deep learning era introduced Recurrent Neural Networks (RNNs) and their Long Short-Term Memory variant as natural architectures for sequential text modelling. LSTM networks, by maintaining explicit memory cells governed by input, forget, and output gating mechanisms, are capable of capturing long-range co-occurrence dependencies across email paragraphs that bag-of-words models entirely discard. Systematic reviews of DL-based phishing detection document LSTM accuracy figures ranging from 91% to 99.1% across diverse datasets and training configurations, with pre-trained contextual embeddings consistently elevating performance above TF-IDF baselines.

One-dimensional Convolutional Neural Networks provide a complementary deep learning perspective by applying learned filter banks as n-gram detectors across token sequences, capturing local phrasal patterns — "click here", "verify account", "act now" — with translation invariance that RNN formulations do not inherently possess. Hybrid CNN-LSTM architectures, reported in recent literature, leverage the local pattern sensitivity of CNN layers feeding into the sequential integration capacity of LSTM cells, achieving marginal improvements over single-architecture DL models.

Transformer-based pre-trained language models, particularly BERT and its derivatives, represent the current performance frontier, with fine-tuned models achieving accuracy above 99% on select benchmark datasets. However, the substantial inference latency and computational resource requirements of transformer models — absent GPU acceleration — render direct real-time deployment in standard email server environments economically and technically impractical. The body of existing literature is further characterised by a persistent gap: virtually all published systems evaluate models in isolated offline batch settings without integration into operational email workflows. PhishGuard AI is specifically designed to address this deployment gap.

III. Proposed Methodology

3.1 System Architecture

PhishGuard AI is architected across three functionally decoupled tiers. The Presentation Tier comprises Flask-rendered Jinja2 HTML templates styled with Bootstrap 5, providing six user-facing interfaces: secure authentication (login and registration with Bcrypt password hashing), email composition with automatic pre-send phishing analysis, standalone on-demand email analyser, real-time analytics dashboard with Chart.js visualisations, sent email history, and template management. The Application Tier houses the Flask route handlers and the PhishGuard AI detection engine (phishing_detector.py), which orchestrates feature extraction, model inference, ensemble aggregation, and risk classification. The Data Tier implements a SQLAlchemy ORM layer over SQLite (development) or MySQL (production), with four relational tables: User, EmailTemplate, SentEmail, and PhishingLog.

The phishing detection pipeline is invoked synchronously within the email sending route, ensuring every outbound email passes through ML+DL analysis before transmission. Detection results are logged to the PhishingLog table with all twenty-six extracted feature values, individual model probability scores, ensemble risk score, risk level classification, and human-readable alert descriptions — constituting a continuously growing research dataset that supports longitudinal threat analysis and periodic model retraining.

3.2 Feature Extraction

Feature engineering bridges raw email text and the numerical input required by ML and DL classifiers. PhishGuard AI extracts twenty-six features across five semantic categories, as enumerated in Table I.

TABLE I — PHISHGUARD AI: TWENTY-SIX EXTRACTED FEATURES

No.	Feature Name	Category	Phishing Discriminative Signal
1-5	**url_count, has_ip_url, has_suspicious_shortener, has_at_in_url, has_double_slash_redirect**	URL	Presence of IP-based URLs, link shorteners, and redirect chains concealing destination domains

6-10	**has_html, html_form_count, html_input_count, html_script_count, img_count**	HTML	Embedded forms and input fields indicative of credential harvesting; scripts enabling malicious code execution
11-17	**phishing_keyword_count, has_urgency, has_financial_terms, has_credential_request, exclamation_count, question_count, capital_ratio**	Content	Semantic signals: urgency language, financial terminology, credential solicitation, and capitalisation anomalies
18-21	**sender_has_numbers, sender_is_free_mail, sender_is_known_legit, sender_domain_length**	Sender	Structural anomalies in sender address — numeric sequences, free-mail providers, domain length extremes
22-26	**body_length, subject_length, word_count, special_char_ratio, has_obfuscation**	Text Stat.	Statistical outliers in length and character distributions; presence of URL-encoded obfuscation sequences

3.3 Machine Learning Classifiers

Four ML classifiers are trained on the twenty-six-dimensional handcrafted feature vector, preceded by StandardScaler normalisation to eliminate scale asymmetries that would otherwise distort kernel- and margin-based algorithms.

Random Forest (RF) constructs an ensemble of one hundred decision trees through bootstrap aggregation, with random feature subset selection ($k = \sqrt{26} \approx 5$) at each split and max_depth constrained to fifteen to prevent memorisation. Majority voting aggregates tree predictions; probability estimates are the proportion of trees voting for each class. SVM with RBF kernel ($C = 10$, γ = scale) identifies the maximum-margin separating hyperplane in the implicitly high-dimensional kernel-induced space. The regularisation parameter $C = 10$ was selected through five-fold cross-validated grid search. Decision Tree (DT) employs Gini impurity-based recursive partitioning with max_depth = 10 and min_samples_split = 5, providing fully interpretable decision rules. Logistic Regression (LR) models posterior phishing probability through the sigmoid link function with L2 regularisation ($C = 1.0$), serving as a linear baseline and probability calibration reference.

3.4 Deep Learning Architectures

The DL models operate on TF-IDF text representations (max_features = 3,000; ngram_range = (1,2)) of combined subject-body text, capturing both unigram and bigram discriminative vocabulary. LSTM: TF-IDF vectors are reshaped into 50 timestep sequences of 60 features per step, then processed by a two-layer LSTM stack (64 → 32 units) with intervening Dropout regularisation (0.3, 0.2), followed by a Dense(16, ReLU) layer and a sigmoid output neuron. CNN: A complementary 1D-CNN applies two successive Conv1D layers (64 filters, then 128 filters; kernel_size = 3; ReLU activation) to a 100-dimensional subsequence of the TF-IDF vector, extracts the globally maximal activation per filter through GlobalMaxPooling1D, and feeds a Dense(64, ReLU) → Dropout(0.3) → Dense(1, sigmoid) head. Both DL models are trained with the Adam optimiser, binary cross-entropy loss, batch_size = 32, and EarlyStopping (patience = 5) monitoring validation loss with best-weight restoration.

3.5 Ensemble Aggregation and Risk Policy

Let $P_m(x)$ denote the phishing probability produced by model m for email x, where $m \in \{RF, SVM, DT, LR, LSTM, CNN\}$. The ensemble risk score is defined as:

$$R(x) = (1/|M|) \Sigma P_m(x), \quad m \in M$$

where M is the set of available trained models. Risk classification follows three ordered intervals: $R(x) < 0.30$ → LOW (email dispatched automatically); $0.30 \leq R(x) < 0.60$ → MEDIUM (user warning with optional override); $R(x) \geq 0.60$ → HIGH (email blocked, detailed alerts surfaced). This calibrated policy prioritises recall over precision — false negatives (undetected phishing) carry substantially higher operational cost than false positives (blocked legitimate emails recoverable through override) — while preserving usability for borderline cases.

IV EXPERIMENTAL EVALUATION

4.1 Dataset and Setup

Experiments were conducted on the Kaggle Phishing Email Dataset, comprising 18,650 labelled email records: 11,322 phishing samples (60.7%) and 7,328 legitimate samples (39.3%). The dataset spans diverse phishing campaign categories including financial institution impersonation, e-commerce credential harvesting, healthcare data theft, and government agency spoofing. An 80:20 stratified train-test partition (training: 14,920; test: 3,730) preserved the natural class distribution. All random seeds were fixed at 42 for complete reproducibility. ML models were evaluated on the twenty-six-dimensional handcrafted feature vector post-StandardScaler normalisation; DL models utilised TF-IDF text representations. Experimental hardware: Intel Core i7-10750H, 16 GB RAM, TensorFlow 2.15 CPU mode.

4.2 Performance Metrics

Model evaluation employed five complementary metrics: Accuracy = (TP + TN)/(TP + TN + FP + FN); Precision = TP/(TP + FP); Recall = TP/(TP + FN); F1-Score = 2 × (Precision × Recall)/(Precision + Recall); and the Area Under the Receiver Operating Characteristic Curve (ROC-AUC), which measures discriminative capacity across all classification thresholds independently of any particular decision boundary. In the phishing detection security context, Recall (True Positive Rate) is the primary metric of operational interest, as undetected phishing emails (false negatives) impose direct financial and security harm, whereas falsely flagged legitimate emails (false positives) incur usability costs recoverable through user override.

4.3 Results

TABLE II — COMPREHENSIVE PERFORMANCE COMPARISON OF ALL SIX MODELS

Model	Type	Acc.(%)	Prec.(%)	Rec.(%)	F1(%)	AUC	Rank
Random Forest	ML Ensemble	97.80	98.10	97.50	97.80	0.9982	1st
SVM (RBF)	ML Kernel	96.50	96.80	96.20	96.50	0.9876	2nd
LSTM	Deep Learning	95.60	96.10	95.10	95.60	0.9854	3rd
Decision Tree	ML Tree	94.30	94.70	93.90	94.30	0.9631	4th
Logistic Reg.	ML Linear	93.10	93.40	92.80	93.10	0.9712	5th
CNN (1D)	Deep Learning	83.75	84.20	83.30	83.75	0.8941	6th
Ensemble (Proposed)	6-Model Avg.	97.93	98.02	97.82	97.92	0.9989	Best

TABLE III — CONFUSION MATRIX: RANDOM FOREST (Test Set n = 3,730)

	Pred: Phishing	Pred: Legitimate
Actual: Phishing	2,197 (TP)	57 (FN)
Actual: Legitimate	24 (FP)	1,452 (TN)

Derived from Table III: Sensitivity (Recall) = 97.47%; Specificity = 98.37%; False Positive Rate = 1.63%; False Negative Rate = 2.53%; Matthews Correlation Coefficient (MCC) = 0.9556. The low false negative rate is of primary importance: only 57 of 2,254 phishing emails (2.53%) evaded Random Forest detection; within the ensemble framework, the LSTM and SVM components recover a proportion of these misclassified instances, reducing ensemble false negatives further.

TABLE IV — TOP 8 FEATURE IMPORTANCE SCORES (RANDOM FOREST, GINI CRITERION)

Rank	Feature	Importance Score	Category
1	phishing_keyword_count	0.1823	Content Semantic
2	has_credential_request	0.1456	Content Semantic
3	has_urgency	0.1234	Content Semantic
4	url_count	0.1102	URL Structural
5	html_form_count	0.0943	HTML Composition
6	has_financial_terms	0.0891	Content Semantic
7	capital_ratio	0.0712	Content Semantic
8	has_ip_url	0.0634	URL Structural

4.4 Comparative Analysis Against Prior Art

TABLE V — COMPARISON WITH REPRESENTATIVE PUBLISHED METHODS

Study	Methodology	Dataset	Acc.(%)	F1
Faya & Nagar (2022)	5 ML Classifiers	Kaggle	97.80	0.977
Butt et al. (2023)	Cloud ML+DL Hybrid	APWG	96.10	0.960
Thakur et al. (2023)	DL Systematic Review	Multiple	98.30	0.982
Fares et al. (2024)	ML Approach	Custom	96.80	0.967
Kyaw et al. (2024)	LSTM (Review Best)	Multiple	99.10	0.990
PhishGuard AI (Ours)	6-Model Ensemble + Real-Time App	Kaggle	97.93 (ens.)	0.979

PhishGuard AI achieves performance comparable to the best ML-based benchmarks (97.93% ensemble accuracy vs. 97.80% for the top single-model baseline). The primary differentiation is not marginal accuracy improvement — which would require transformer architectures — but the architectural advancement of operational real-time deployment: all reviewed comparison systems are offline research tools with no integration into functional email workflows.

V DISCUSSION

5.1 Ensemble Synergy

The ensemble risk score reduction to 2.07% total error rate, versus 2.17% for the best individual model (RF), may appear modest in percentage terms but corresponds to a meaningful improvement in operational security. In a deployment processing 10,000 emails per day, the ensemble recovers approximately 100 additional phishing detections daily compared to RF alone. The mechanism of improvement is complementary error correction: phishing emails misclassified by RF — which operates on handcrafted features — are partially recovered by LSTM, which captures sequential textual patterns that are feature-agnostic. This architectural complementarity, spanning hand-engineered and representation-learned feature spaces, is the core theoretical justification for the hybrid ML+DL ensemble design.

5.2 Deep Learning Performance Analysis

The LSTM model's 95.60% accuracy using TF-IDF representations validates the sequential text modelling hypothesis for phishing detection without requiring pre-trained word embeddings, though a performance ceiling is evident from the sparse, order-insensitive nature of TF-IDF. The CNN's comparatively lower accuracy (83.75%) corroborates the theoretical expectation that convolutional local pattern detection is better served by dense, positionally meaningful embeddings (Word2Vec, GloVe) than by sparse TF-IDF vectors where the spatial arrangement of terms is arbitrary. Replacing TF-IDF with BERT contextual embeddings is projected to elevate both DL models above the RF baseline.

5.3 Feature Importance Insights

The dominance of content-semantic features in Random Forest importance rankings — phishing_keyword_count (18.23%), has_credential_request (14.56%), has_urgency (12.34%) — quantitatively confirms that the semantic intent of the message is the primary differentiator between phishing and legitimate correspondence, outweighing structural URL signals. This finding has a practical implication: detection systems that analyse URL patterns exclusively, without semantic content analysis, are structurally incomplete. The complementarity between content-semantic ML features and sequential text patterns captured by LSTM further justifies the multi-architecture ensemble design.

5.4 Operational Deployment Significance

The operational integration of PhishGuard AI into the email dispatch workflow represents a qualitative advancement beyond the research-only evaluations that characterise the published literature. The three-tier risk enforcement policy — automatic approval for LOW-risk, user notification for MEDIUM-risk, and hard blocking for HIGH-risk — implements a security-usability trade-off that accommodates the operational reality that no detection system achieves perfect accuracy. User override capability for MEDIUM-risk emails prevents false positives from disrupting legitimate communication workflows, while maintaining security audit trails through the PhishingLog database. This design reflects defence-in-depth principles appropriate for enterprise email security.

5.5 Limitations

Three primary limitations constrain the current system. First, TF-IDF representations for DL models forfeit the semantic contextual richness available from pre-trained transformer embeddings, constraining achievable DL accuracy. Second, the absence of live URL reputation checking through external threat intelligence APIs (VirusTotal, PhishTank) limits detection of zero-day phishing domains with no prior catalogued malicious activity. Third, the training dataset, while large and diverse, predates the widespread deployment of LLM-generated phishing content; systematic evaluation against AI-crafted phishing emails is required to assess detection robustness against this emergent threat class.

VI Conclusion

This paper presented Detection of Email Phishing Attacks Using Machine learning and Deep Learning , a hybrid ensemble framework combining four classical ML classifiers and two DL architectures for real-time email phishing detection, operationally integrated within a Flask web application. The framework extracts twenty-six discriminative features across five semantic dimensions and aggregates individual model probabilities into a calibrated ensemble risk score. Experimental evaluation on 18,650 labelled emails demonstrates that Random Forest achieves 97.80% accuracy and 0.9982 ROC-AUC; LSTM achieves 95.60% accuracy; and the six-model ensemble reduces total misclassification to 2.07%, outperforming all constituent models. The system's real-time deployment, three-tier risk enforcement, and persistent research logging address the critical gap between academic model evaluation and practical email security tooling.

Future research directions encompass: (i) replacing TF-IDF with BERT or RoBERTa embeddings to elevate DL model accuracy to the transformer performance frontier; (ii) integrating live URL threat intelligence via VirusTotal and PhishTank APIs; (iii) adversarial training against LLM-generated phishing content; (iv) extending multi-modal analysis to email attachments (PDF, Office documents) and embedded images; and (v) implementing federated learning to enable privacy-preserving distributed model improvement across organisational deployments

References

[1] K. D. Waghmare and S. Khan, "Detection of E-mail Phishing Attacks Using Machine Learning and Deep Learning: A Review," International Journal of Scientific Development and Research (IJSDR), vol. 10, no. 12, pp. b13-b17, Dec. 2025.

[2] I. Faya and Z. A. Nagar, "Email Phishing Detection Using Machine Learning," International Journal of Emerging Technologies and Innovative Research, ISSN 2349-5162, 2022.

[3] U. A. Butt, R. Amin, H. Aldabbas, S. Mohan, B. Alouffi, and A. Ahmadian, "Cloud-based email phishing attack using machine and deep learning algorithm," Complex & Intelligent Systems, vol. 9, pp. 3043-3070, 2023.

[4] P. H. Kyaw, J. Gutierrez, and A. Ghobakhlou, "A Systematic Review of Deep Learning Techniques for Phishing Email Detection," Electronics, vol. 13, no. 19, p. 3823, 2024. doi: 10.3390/electronics13193823.

[5] K. Thakur, M. L. Ali, M. A. Obaidat, and A. Kamruzzaman, "A Systematic Review on Deep-Learning-Based Phishing Email Detection," Electronics, vol. 12, no. 21, p. 4545, 2023.

[6] H. F. Atlam and O. Oluwatimilehin, "Business Email Compromise Phishing Detection Based on Machine Learning: A Systematic Literature Review," Electronics, vol. 12, no. 1, p. 42, 2023.

[7] H. Fares, J. Kilani, F. Fagroud, H. Toumi, F. Lakrami, Y. Baddi, and N. Aknin, "Machine Learning Approach for Email Phishing Detection," Procedia Computer Science, vol. 251, pp. 746-751, 2024.

[8] S. Salloum, T. Gaber, S. Vadera, and K. Shaalan, "Phishing Email Detection Using Natural Language Processing Techniques: A Literature Survey," Procedia Computer Science, vol. 189, pp. 19-28, 2021.

[9] D. Rathee and S. Mann, "Detection of E-Mail Phishing Attacks Using Machine Learning and Deep Learning," International Journal of Computer Applications, vol. 183, no. 47, 2022.

[10] L. Breiman, "Random Forests," Machine Learning, vol. 45, no. 1, pp. 5-32, 2001.

[11] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Computation, vol. 9, no. 8, pp. 1735-1780, 1997.

[12] Y. Kim, "Convolutional Neural Networks for Sentence Classification," in Proc. EMNLP 2014, pp. 1746-1751, 2014.

[13] 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, 2019.

[14] O. Kolade, S. R. Olatunji, and A. Adeyemi, "Phishing Detection Using an Optimised Stacking Ensemble Classifier," IEEE Access, vol. 11, pp. 27313-27324, 2023.

[15] N. Yadav, S. Singh, and S. Siddiqui, "AI Powered Mail Server for Smart and Secure Email Management," International Journal of Creative Research Thoughts, ISSN 2349-5162, 2024.