A Hybrid Heuristic-Machine Learning Framework for Phishing Detection Using Multi-Domain Feature Analysis

Ashvini Jadhav
Department of Computer Science and Engineering, MIT School of Computing, MIT Art Design and Technology University, Pune, India
ashvinigjadhav@gmail.com (corresponding author)

Pankaj Chandre
Department of Computer Science and Engineering, MIT School of Computing, MIT Art Design and Technology University, Pune, India
pankajchandre30@gmail.com

Received: 16 April 2025 | Revised: 29 May 2025 and 4 June 2025 | Accepted: 7 June 2025

 DOI: https://doi.org/10.48084/etasr.11548

ABSTRACT

This study introduces a hybrid phishing detection framework that combines machine learning with heuristic rule-based techniques to provide accurate, scalable, and policy-compliant detection across a variety of phishing types. The proposed method uses diverse datasets, including URL patterns, email headers, and HTML content, organized in a layered manner, allowing flexible analysis even when some features are missing. Feature selection techniques, such as variance thresholding and Recursive Feature Elimination (RFE), are applied to improve learning efficiency and reduce noise. Several classifiers, including Random Forest (RF), XGBoost, Gradient Boosting (GB), and CatBoost, are trained on optimized features, and their outputs are combined using voting to boost overall reliability. The system also includes a rule-based engine aligned with India's national Email Policy, incorporating heuristic checks such as non-government domains, missing authentication (SPF/DKIM/DMARC), use of insecure protocols, foreign IPs, phishing URLs, and other threat indicators. Each rule is weighted and contributes to a composite suspicion score, which is explainable and policy-mapped. These heuristic signals are used both directly and as features for the machine learning models, allowing for layered, interpretable AI. The final phishing score balances the contribution of both heuristic and ML predictions and is compared against an optimized threshold to determine whether an input is phishing or safe. Experimental results on benchmark datasets demonstrate that heuristic-guided feature selection, combined with hybrid data integration, significantly improves performance, achieving an average accuracy exceeding 95% in real-world datasets. Individual models, including CatBoost and XGBoost, demonstrated outstanding performance, achieving training accuracies of up to 100% and testing accuracies of 96.7% and 96.4%, respectively, for URL datasets. For email header analysis, RF achieved the highest accuracy at 99.85%. The findings underscore the significance of feature engineering in developing scalable and reliable phishing detection systems.

Keywords-machine learning; feature analysis; random forest; phishing detection

I. INTRODUCTION

In Q2 2024, the Anti Phishing Working Group (APWG) observed a modest drop in phishing incidents from 963,994 to 877,536, partly due to reporting challenges caused by email provider restrictions [1]. Social media platforms have become the main target (32.9%), while phishing attempts on financial institutions and payment services have declined. Business Email Compromise (BEC) scams remained prevalent, with average transfer requests around $89,520, and Google Gmail leading as the primary free webmail platform used in 72.4% of these BEC attacks.

Traditional email phishing has become increasingly difficult as defenses have improved. The growing sophistication of phishing, particularly BEC and social media scams, has caused a pressing need for adaptive detection methods. To address this, the study leverages three datasets, preprocessing through normalization and heuristic scoring, and feature refinement through variance threshold and Recursive Feature Elimination (RFE). Finally, it compares the performance of the Random Forest (RF) and XGBoost classifiers trained on the selected features to improve detection accuracy.

II. BACKGROUND ON PHISHING ATTACKS

In [2], a hybrid metaheuristic algorithm was proposed for feature selection, introducing the Krill Herd (KH) algorithm combined with Tabu Search (TS) for spam email detection, achieving an accuracy of 97.8 on seven benchmark datasets. In [3], a comprehensive survey of deepfake technologies, including face swapping, reenactment, talking-face generation, and attribute editing, demonstrated significant advancements beyond earlier character-based models. KnowPhish [4] is a large-scale multimodal Brand Knowledge Base (BKB) using existing Reference-Based Phishing Detectors (RBPDs) to analyze logos and textual content through Large Language Models (LLMs), reliably detecting phishing pages even in the absence of logos. In [5], URLNet, a convolutional neural network model, was trained on character- and word-level URL embeddings and achieved over 99% accuracy. Hybrid models, integrating URL and hyperlink features, further increase accuracy to 99.17%, significantly surpassing URL-only systems. In [6], a K-Nearest Neighbors (KNN) model trained on a Kaggle dataset of 5,171 emails processed using Word2Vec embeddings, achieved 97% testing accuracy. More advanced frameworks blend KNN, LSTM, and clustering with features extracted from URLs, HTML, and behavioral data.

Traditional detection methods include Naïve Bayes (NB) classifiers, rule-based filters (e.g., domain heuristics, SSL checks), and behavior-based scoring systems. Models use features such as HTML tags, authentication mechanisms (DKIM, SPF, DMARC), sender credibility, and email headers. Advanced ensemble methods, such as FMPED/FMMPED and LDA-based clustering, are employed to address class imbalance and contextual nuance. Stacking various classifiers (DT, GB, LR, SVM) under an RF meta-classifier, enhanced with oversampling and TF-IDF vectorization, can achieve high accuracy and benefit from SHAP-based interpretability. Despite their performance, existing models suffer from latency, lack of explainability, and inconsistent results across domains. Therefore, lightweight, interpretable, and generalized phishing detection models built on diverse, real-world datasets are necessary.

In [7, 8], hybrid feature sets combined URL and hyperlink information to achieve high accuracy and outperform models that relied on singular feature types. In [9], it was shown that mobile browser UI designs often hide phishing indicators, making users more vulnerable. Poorly designed alerts and a lack of explanatory details on warnings further diminish their effectiveness. Overly complex or disengaging training materials also hinder user preparedness, especially against spear-phishing attacks. In [10], theoretical models such as the Heuristic-Systematic Model (HSM) and the Social-Cognitive Analytical Model (SCAM) suggest that people with high self-efficacy process information more systematically. In [11], a contradiction to the theory of routine activity was observed, as users with high Internet usage during leisure-time showed a lower susceptibility to phishing. Eye-tracking studies underscore the value of targeted training. In [12], the collaboration between industries and institutions was emphasized to share best practices in combating phishing.

In [13], a review of 80 studies showed that Machine Learning (ML), especially RF (used in 31 studies), is the most widely applied technique in the detection of phishing websites. However, the highest reported accuracy (99.98%) was achieved by a CNN in detecting phishing websites. In [14], Support Vector Machines (SVM), Naive Bayes (NB), and LSTM were used to classify phished emails. Among them, SVM achieved the highest accuracy of 99.62%, followed by LSTM at 98% and NB at 97%. In [15, 16], phishing susceptibility was shown to be higher among employees with shorter tenure and lower satisfaction, underlining the need for targeted cybersecurity training. In [17], behavioral traits, such as self-control and job satisfaction, were identified as predictors of phishing risk. During the COVID-19 pandemic, phishing campaigns exploited fear-related terms such as "COVID-19," "Wuhan," "lockdown," and "facemask" to deceive users. In [18], more than 1100 domains used pandemic-related keywords, such as "coronavirus," "stayathome," and "PPE shortage", to bypass email filters and trick recipients.

In [19], Latent Dirichlet Allocation (LDA) outperformed Doc2Vec with k-means clustering in capturing contextual phishing patterns. In [20], a robust phishing detection framework was built using content, behavior, and security features, including sender reputation, IP address, and email header anomalies. In [21], the link between users' online browsing hours, spending habits, and vulnerability to phishing attacks was explored. This study provided practical insights for individuals and cybersecurity professionals to strengthen phishing defense strategies. In [22], a comprehensive review of phishing website detection techniques focused on list-based, similarity-based, and ML-based approaches. In [23], behavior-based detection evaluated suspicious links and interactions, while FMPED/FMMPED effectively handled class imbalance. In [24], detection was enhanced by analyzing content and HTML features such as the use of JavaScript and embedded links. In [25], a deep learning model achieved 98% accuracy and 97.5% F1-score in spam filtering. In [26], an ensemble model used DT, GB, LR, and SVM to achieve 97% accuracy with SHAP-based transparency. In [27], comparative analysis highlighted the need for interpretable, time-efficient, and generalizable phishing detection systems trained on diverse datasets.

III. PROPOSED METHOD

The proposed method, shown in Figure 1, was evaluated through a comprehensive experimental setup, designed to assess the effectiveness of ML models in detecting phishing across three distinct data types: URLs, email headers, and web-based HTML features. The experiment involved the use of three separate datasets.

A. Purpose and Scope

The primary purpose was to explore and evaluate the use of ML algorithms, focusing on Web-, URL-, and email header-based phishing detection tasks on distinct datasets: one for URLs, , another for email headers, including attributes related to email composition, path, and domain matching, and another with Web-based HTML features.

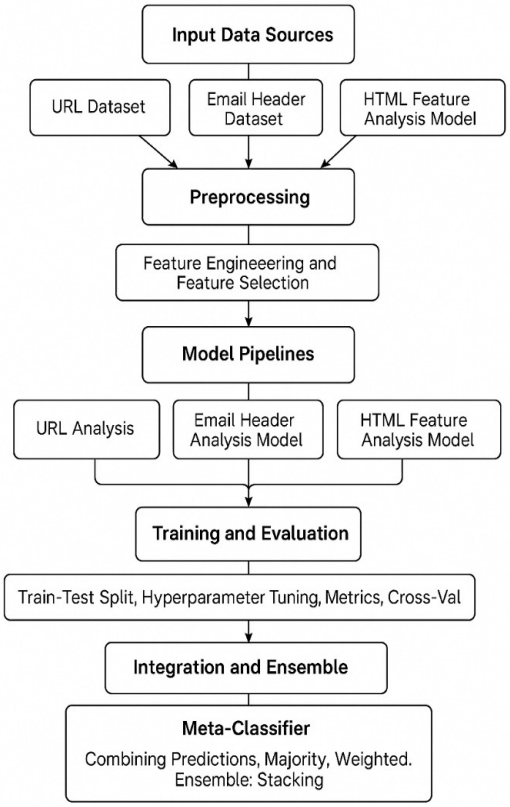

Fig. 1. The proposed hybrid heuristic-ML framework for phishing detection using multi-domain features.

B. Research Objectives

1) Optimize Feature Selection Techniques

Filter methods (variance thresholding) are applied to remove low-variance features, while wrapper methods, such as RFE, are applied iteratively to refine feature subsets. Embedded methods, using feature importance from RF and XGBoost, can enhance the performance of the model and reduce the dimensionality. Integrating metaheuristic-selected feature sets, ensemble classifiers, and contextual organizational variables can deliver a phishing detection system that is accurate, interpretable, and scalable for real-world deployment.

2) Address Security Awareness and Behavior

Using insights from behavioral science and security awareness training studies can help the development of adaptive training interventions to reduce the susceptibility to phishing by up to 30% according to empirical evidence.

C. Mathematical Model

1) Feature Representation

Let the combined feature set from all datasets be represented as a feature vector:

$X = [x_1, x_2, x_3, \ldots, x_n]$

where each $x_i \in \mathbb{R}$ or $\{0,1\}$ represents a normalized, engineered, or heuristic feature from:

- Dataset 1: URL features $F1$: $X_1 = [\ldots]$
- Dataset 2: Email header features $F2$ $X_2 = [\ldots]$
- Dataset 3: HTML content-based features $F3$ $X3 = [\ldots]$

Let $Y \in \{0,1\}$ be the binary target variable, where $Y = 1$ denotes phishing and $Y = 0$ denotes legitimate.

2) Heuristic Rule-Based Indicator Function

The heuristic rule-based indicator function plays a foundational role in phishing detection by encoding domain knowledge into a set of manually designed rules. Heuristic rules act as binary signals with output $h_i \in \{0,1\}$.

In a policy framework established by the Indian Ministry of Communication & Information Technology, Department of Electronics & Information Technology, the sample feature-based heuristic rules (h_i) offer a structured, policy-aligned method for detecting phishing activities in government email communications and user behavior. Each heuristic rule is assigned a weight (α_i) based on its criticality and the overall heuristic score $S_H = \sum(\alpha_i \times h_i)$ integrates multiple indicators, all mapped to specific clauses of the Email Policy of the Government of India, such as emails not sent from designated government domains (h_1), insecure POP3/IMAP access (h_2), and auto-forwarding to personal domains (h_3). Other rules address browser auto-login (h_4), foreign IP access (h_5), absence of digital signatures for sensitive content (h_6), repeated ignored alerts (h_7), expired account activity (h_8), lack of SPF/DKIM/DMARC (h_9), and device/IP anomalies (h_{10}). Additional heuristics capture phishing links (h_{11}), malicious attachments (h_{12}), misuse of official designations (h_{13}), threat intelligence flags (h_{14}), urgency keywords in BEC emails (h_{15}), brute-force login attempts (h_{16}), suspicious BCC use (h_{17}), lack of MFA/OTP (h_{18}), anonymized access via VPNs/TOR (h_{19}), and header spoofing (h_{20}).

These rules can be applied in three key ways: calculating phishing risk scores, serving as features in ML/DL-based detection models, and contributing to explainable AI layers for analyst insight and compliance verification, thereby offering a scalable, interpretable, and policy-compliant phishing detection framework. $H = [h_1, h_2, \ldots, h_k]$ represents the complete heuristic signal for an input instance, where $h_i = 1$ if rule i is triggered, and $h_i = 0$ means the condition is not met. These individual rule outputs are collected into a vector H. Each rule is assigned a weight α_i based on its importance, reliability, or historical effectiveness. The combined heuristic score S_H is calculated as a weighted sum $S_H = \Sigma(\alpha_i * h_i)$, where α_i are the heuristic weights. This produces a scalar value indicating the overall heuristic suspicion level of the input. A higher S_H implies stronger evidence of phishing based on heuristic analysis alone.

D. ML Prediction Function

An ML model trained on labeled data provides a probabilistic prediction. Let $f_\theta(X)$ be the output of a trained ML model, with:

$f_\theta(X) = P^{ML}(Y = 1 \mid X) \in [0,1]$

being the likelihood that the input X is a phishing instance. In ensemble settings, multiple models $f_{\theta j}$ can be averaged using:

$$f_{ensemble}(X) = (1/M)\ \Sigma\ f_{\theta j}(X),\ \text{for } j = 1\ to\ M$$

E. Hybrid Model Combining Heuristic and ML-Based Scores

In the hybrid model fusion phase, the system combines both heuristic and ML-based insights into a unified phishing probability score

$$P_{phish}(X) = \lambda * S_H + (1 - \lambda) * f_\theta(X)$$

where $\lambda \in [0, 1]$ balances heuristic versus ML contributions. A higher λ prioritizes heuristics, while a lower λ relies more on ML outputs. The final classification rule uses an optimized decision threshold τ, optimized (e.g., using F1-score on a validation set), to make a binary decision:

If $P_{phish}(X) \geq \tau \rightarrow Predict\ Phishing\ (Y = 1)$

Else $Predict\ Legitimate\ (Y = 0)$.

F. Parameter Initialization Strategy in OptimizeParameters Procedure

1) Heuristic Weights α_{init}

- Uniform initialization: Assign equal weight to each heuristic rule (e.g., $\alpha_i = 1/n$ for n rules).
- Random initialization: Use random values to promote exploration of different weight combinations.
- Importance-based initialization: Initialize based on prior knowledge such as feature importance or rule contribution scores**.**

2) Hybrid blending weight λ_{init}

- Fixed initialization: Commonly set to 0.5 to give equal importance to heuristic and ML outputs.
- Trust-based initialization**:** Adjust λ based on the reliability or performance of heuristic rules versus the ML model.

3) Decision Threshold τ_{init}

- Default initialization set at 0.5 for balanced classification.
- ROC-based tuning: τ is based on ROC curve analysis to optimize F1-score or achieve a preferred precision-recall trade-off.

IV. HYBRID HEURISTIC-ML FRAMEWORK WITH MULTI-DOMAIN FEATURES

A. Dataset Collection

To support the development of a robust phishing detection system, five diverse datasets were collected, each offering distinct characteristics in terms of size, feature diversity, and data type. The first dataset, Phishing URL Dataset, was sourced from the UCI Machine Learning Repository and contains 235,795 entries, with 134,850 legitimate and 100,945 phishing URLs. It includes 56 features capturing URL structure, HTML characteristics, redirection behavior, and content-based signals, such as CharContinuationRate and URLTitleMatchScore [28].

The second dataset, titled Phishing_Legitimate_fullurl, was collected using Selenium WebDriver and is hosted on Mendeley Data. It contains 10,000 balanced entries from the period 2015-2017 and includes 50 well-defined features focused on URL composition and behavior, making it suitable for benchmarking ML models [29]. The third dataset [30] has 7,658 entries with 89 features and works as a compact balanced dataset, ideal for early-stage experimentation and model prototyping. The fourth dataset, URL_with_feature.csv, contains 11,430 entries and 89 features, mainly highlighting the lexical structure and redirection behaviors in URLs [31]. The fifth and largest dataset, protocol.csv, contains 146,160 entries and 197 features having is over 28 million data points related to multi-layered attributes such as SSL/TLS metadata, domain reputation, and email server configurations, making it valuable for these phishing detection models that involve both network and content analysis. In the process of creating a consistent input format appropriate for multi-layered detection, the raw datasets were grouped and integrated based on feature domains.

1) Dataset 1 – URL Dataset

The first grouped dataset on URL-based phishing detection was created by concatenating the Training.csv and URL_with_feature.csv datasets. It has 7,658 entries and 89 features, capturing a wide array of structural, compositional, and behavioral characteristics of URLs. Notable features include URL length, hostname length, and the presence of special characters, which are instrumental in identifying suspicious patterns often associated with phishing. The dataset also incorporates domain and IP-based indicators such as domain age, DNS record validity, and the presence of IP addresses in place of domain names. In addition, it includes content and behavioral signals, such as the presence of login forms, email submission fields, and suspicious JavaScript events, all of which contribute to detecting phishing activities. Additional external validity signals, such as Google indexing status and page rank, serve as reputation-based features to reinforce classification accuracy. The target variable is binary, phishing or legitimate, allowing supervised learning for phishing URL detection.

2) Dataset 2: Email Header Dataset

The second dataset highlights email header analysis and is derived from protocol.csv, specifically focusing on phishing indicators in email communication. It contains 95 features spanning both numerical variables, such as the number of SMTP hops, content length, and line count, and categorical indicators, such as missing Subject or To fields, and mismatches between sending and display domains. Preprocessing steps include missing value imputation, normalization, and the application of feature engineering techniques to generate authentication indicators and anomaly scores**.** These improvements aim to find understated patterns that differentiate phishing emails from legitimate ones. The dataset is labeled with a binary target variable, where 1 indicates a phishing email and 0 represents a legitimate one. It is partitioned into 80% training and 20% testing sets to support model training and evaluation.

3) Dataset 3: URL Dataset with Web-Based HTML features

The third dataset combines URL structure with web-based HTML content features, leveraging the PhiUSIIL dataset to enrich phishing detection capabilities. This dataset includes attributes such as URL and domain length, along with a binary indicator (IsDomainIP) that flags whether the domain is an IP address, an indicator commonly associated with malicious sites. Additionally, it includes Top-Level Domain (TLD) classifications and various HTML content indicators, such as HTML tags and CSS tags, as well as the number of external and self-referencing links. These features are designed to capture the underlying web content structure and behavior of phishing pages. The dataset's target label distinguishes between phishing and legitimate samples and exhibits a slight class imbalance, containing 124,421 phishing and 93,049 legitimate entries, and is particularly useful for deep content-level phishing analysis.

B. Preprocessing Pipeline - Feature Selection Using Hybrid Metaheuristic Algorithm

Correlations between numeric features visualize relationships to determine which ones are closely related. The top features most correlated with phishing were: google_index, page_rank, nb_www, ratio_digits_url, nb_hyperlink, domain_in_title, phish_hints, domain_age, ip, and nb_qm. These features show the highest absolute correlations with the phishing status.

XGBoost and CatBoost were the top-performing models across most feature selection methods, with variance threshold and Lasso providing the highest accuracy scores. ANOVA performed slightly better than mutual information, making it a preferred choice for select best-based selection.

C. Models

The RF model's performance in phishing detection was analyzed using varying values of $n\_clusters$ (number of clusters for KMeans) and max_depth (maximum tree depth). Training accuracy consistently increased with higher tree depths, reaching perfect accuracy (1.0) at $max\_depth \geq 20$ across all cluster configurations. However, test accuracy demonstrated a more nuanced trend, as it initially improved with increasing depth, peaking around $max\_depth = 20$ for most settings, before stabilizing or slightly declining, indicating diminishing returns and potential overfitting at higher depths The effect of $n\_clusters$ on model performance revealed that simpler clustering configurations achieved the highest test accuracies, with $n\_clusters = 2$ and $max\_depth = 20$ yielding the best overall test accuracy of 0.958. In contrast, higher cluster counts (e.g., $n\_clusters = 3$ or more) slightly reduced test accuracy, due to redundancy or noise introduced by additional clusters.

The optimal configuration for this task with $n\_clusters = 2$ and $max\_depth = 20$ balances high test accuracy with generalization. Limiting max_depth to 20 - 25 prevents overfitting while maintaining strong performance. The analysis underscores the importance of balancing model complexity and data representation for effective phishing detection, making the identified optimal settings a reliable choice for deployment.

TABLE I. N CLUSTERS ON MODEL PERFORMANCE

n_clusters	max_depth	Train	Test
1	10	0.98	0.96
1	15	1.00	0.96
1	20	1.00	0.96
1	25	1.00	0.96
2	5	0.95	0.94
2	10	0.98	0.96
2	15	1.00	0.96
2	20	1.00	0.97
2	25	1.00	0.96
3	5	0.94	0.94
3	10	0.98	0.96
3	15	1.00	0.96
3	20	1.00	0.96
3	25	1.00	0.96
4	5	0.94	0.94
4	10	0.98	0.96
4	15	1.00	0.96
4	20	1.00	0.96
4	25	1.00	0.96
5	5	0.94	0.94
5	10	0.98	0.96
5	15	1.00	0.96
5	20	1.00	0.96
5	25	1.00	0.96

Experiments were conducted to identify the optimal feature selection methods and their impact on model performance. Variance threshold, select best (using mutual information), RFE, RFE with Cross-Validation (RFECV), SelectFromModel, and Lasso regression were applied. Among these, variance threshold and SelectFromModel yielded consistently high results, with RF emerging as the most robust model in each feature selection scenario. GB, XGBoost, and CatBoost also demonstrated competitive performance, particularly with the variance threshold approach. The analysis reveals several key insights into detecting phishing emails through content and feature analysis. One significant finding is that the absence of standard list-related fields, missing list-subscribe, missing list-help, and missing list-id, strongly correlates with phishing attempts. Phishing emails tend to omit these fields, which are typically present in legitimate mailing lists. However, features str_precedence_list and str_return-path_bounce are more indicative of legitimate emails, as these attributes are often inconsistent or absent in phishing emails. The missing-spam-status feature, while important, might be influenced by pre-existing spam filters and is most effective when used in conjunction with other features for better detection accuracy.

D. Model Training and Testing Evaluation

DT achieved perfect training results but saw a decline in testing performance. RF mirrored the perfect training results and maintained high testing performance. CatBoost and XGBoost exhibited the most consistent and robust performance across both training and testing phases, demonstrating their effectiveness in phishing detection.

TABLE II. PERFORMANCE OF VARIOUS PHISHING DETECTION MODELS

Model	Training accuracy	Testing accuracy
Logistic Regression (LR)	0.951	0.944
k-Nearest Neighbors (KNN)	0.964	0.944
SVM	0.946	0.935
NB	0.744	0.745
Decision Tree (DT)	0.99	0.918
RF	1.000	0.954
GB	0.967	0.949
CatBoost	0.993	0.954
XGBoost	1.000	0.954

For HTML content classification, the proposed model achieved perfect accuracy (1.0) and ROC-AUC (1.0) on the test data, suggesting that it predicts phishing and legitimate websites with complete precision. The classification report further confirms this with precision, recall, and F1-scores of 1.0 for both classes. The test set had 18,610 legitimate samples (label=0) and 24,884 phishing samples (label=1). The feature selection process identified as relevant attributes the URLSimilarityIndex, LineOfCode, HasSocialNet, and content-based metrics, such as the number of images, CSS, JavaScript files, self-references, and external references. These features are highly aligned with phishing indicators.

TABLE III. IMPACT OF FEATURES ON MODEL PREDICTIONS

Estimators	D1	D2	D3
100	0.9561	0.9889	1
200	0.9641	0.9989	1
300	0.9654	0.999	1

E. Integration of Hybrid Phishing Detection Using Heuristic Rules and ML

As shown in Table IV, the hybrid phishing detection algorithm integrates heuristic rules with ML predictions to improve detection accuracy and adaptability across diverse datasets. This integration is governed by three key parameters: λ, τ, and α_i. The parameter λ controls the balance between the heuristic score (S_H) and the ML model's output probability (P_{ML}), while τ acts as a classification threshold that determines whether an email or URL is flagged as phishing. The weights α_i define the relative importance of specific heuristic rules, which may be manually assigned based on domain knowledge or learned from data. Heuristic rules are designed around common phishing characteristics such as suspicious URL structures, SPF or DKIM failures, and abnormal header fields. Feature-specific thresholds are extracted through empirical dataset analysis, and both λ and τ are fine-tuned using performance metrics, such as the F1-score and ROC-AUC, during validation. The ML component of the system leverages RF as a baseline due to its strong interpretability and accuracy in phishing URL detection. To enhance performance, feature selection techniques such as mutual information, variance threshold, and RFE were applied to retain the most relevant ones and reduce dimensionality. Ensemble methods, including model stacking and soft voting, can be used to combine predictions from multiple classifiers. A voting classifier, comprising three optimized models, achieved an accuracy of 99.87% and an F1-score of 99.91%, demonstrating excellent precision and recall. These results confirm that combining heuristic reasoning with ensemble-based ML models leads to a highly robust and accurate phishing detection system.

TABLE IV. HEURISTIC RULES AND ML TESTING MODEL

Operation	Details
Extract email features	Header (From, SPF, DKIM) URL (links in email body) HTML content (script, iframe, form tags).
Feature vector (X)	F1 (URL Dataset): X_1 = [….] F2 (Email Header Dataset): X_2 = [...] F3 (HTML Dataset): X_3 = [...]
Heuristic rules (H)	Evaluate heuristic thresholds: → Output $h_i \in \{0,1\}$
Calculate heuristic weighted score:	$S\_H = \Sigma\ (\alpha_i \times h_i)$ where α_i is the weight of rule h_i. Rules come from domain reputation, email validation, as per policy govt policy.
Predict with ML model	Input $[X_1 + X_2 + X_3]$ into the trained model Output: $P_{ML} = f_\theta(X) \in [0,1]$
Heuristic+ML compute hybrid detection score:	$P_{phish} = \lambda \times S_H + (1-\lambda) \times PML$ $\lambda \in [0,1]$ balances rule-based and ML-based predictions.
Decision threshold (τ)	If $P_{phish} \geq \tau \rightarrow$ Phishing If $P_{phish} < \tau \rightarrow$ Legitimate τ optimized using F1, ROC-AUC, or precision-recall tradeoff

The hybrid heuristic-ML framework for phishing detection integrates insights from existing literature with extensive experimental evaluations across multiple diverse phishing datasets. Consistent with previous research, the results demonstrate that phishing detection performance varies significantly with dataset characteristics, feature selection, and model architectures, with ensemble models such as RF and GB showing strong results. To overcome the limitations of purely data-driven methods, such as limited explainability and policy alignment, the framework incorporates a heuristic rule engine based on phishing-specific security policies, including domain reputation, email header verification, behavioral heuristics, and structural content inspection. These heuristics act both as independent indicators and complementary features, enhancing detection accuracy and interpretability. Leveraging multi-domain features from URLs, email headers, and web HTML content, combined with advanced preprocessing, feature engineering, and ensemble modeling, the framework achieves robust phishing detection. A key contribution is a mathematical fusion model that uses a tunable parameter λ to balance heuristic scores with probabilistic outputs from classifiers such as RF, XGBoost, and CatBoost, enabling adaptive sensitivity based on threat context and policy priorities. The heuristics, each weighted by criticality and aligned with the Government of India Email Policy, cover signals such as unauthorized domain use, insecure protocols, phishing links, malicious attachments, BEC urgency keywords, and header spoofing. These rules support phishing risk scoring and feature augmentation for ML/DL models.

V. CONCLUSION

This study presented a robust hybrid framework for phishing detection that incorporates ML models with heuristic

rule-based analysis. Designed to be scalable, accurate, and aligned with national cybersecurity policies, the system processes diverse types of data, including URL structures, email headers, and HTML content. Organized in a layered format, the framework ensures adaptability even when parts of the data are missing. It employs feature selection methods, such as variance threshold and RFE, to filter out irrelevant or noisy input, improving model performance. Core classifiers, such as RF, XGBoost, GB, and CatBoost, are trained on these optimized features, and their outputs can be combined using voting for enhanced reliability. A key component is the policy-compliant heuristic engine, built around guidelines from the Ministry of Communication & Information Technology, India, which assigns weighted scores to features such as non-government domains, insecure links, foreign IPs, and missing SPF/DKIM/DMARC records. These heuristic signals not only help generate explainable scores but can also be integrated into ML workflows. The final phishing score balances both heuristic and ML predictions, using a threshold optimized for the best F1-score to determine whether a message is phishing or legitimate. Experiments on multiple real-world datasets showed that this hybrid approach achieved over 95% accuracy on average. Individual models, such as CatBoost and XGBoost, delivered training accuracies up to 100% and testing accuracies of 96.7% and 96.4% for URL-based data, while RF reached 99.85% accuracy in email header analysis. These results highlight the importance of combining engineered features, intelligent model tuning, and policy-aware rule sets for building effective phishing detection systems.

REFERENCES

[1] R. Tanti, "Study of Phishing Attack and their Prevention Techniques," *International Journal of Scientific Research in Engineering and Management*, vol. 08, no. 10, pp. 1–8, Oct. 2024, https://doi.org/10.55041/IJSREM38042.

[2] G. H. Al-Rawashdeh, O. A. Khashan, J. Al-Rawashde, J. A. Al-Gasawneh, A. Alsokkar, and M. Alshinwa, "Feature Selection Using Hybrid Metaheuristic Algorithm for Email Spam Detection," *Cybernetics and Information Technologies*, vol. 24, no. 2, pp. 156–171, Jun. 2024, https://doi.org/10.2478/cait-2024-0021.

[3] G. Pei *et al.*, "Deepfake Generation and Detection: A Benchmark and Survey." arXiv, May 16, 2024, https://doi.org/10.48550/arXiv.2403.17881.

[4] Y. Li *et al.*, "KnowPhish: Large Language Models Meet Multimodal Knowledge Graphs for Enhancing {Reference-Based} Phishing Detection," presented at the 33rd USENIX Security Symposium (USENIX Security 24), 2024, pp. 793–810.

[5] R. J. Van Geest, G. Cascavilla, J. Hulstijn, and N. Zannone, "The applicability of a hybrid framework for automated phishing detection," *Computers & Security*, vol. 139, Apr. 2024, Art. no. 103736, https://doi.org/10.1016/j.cose.2024.103736.

[6] J. Gikandi, J. Kamau, D. Njuguna, and L. Sawe, "Sentence Level Analysis Model for Phishing Detection Using KNN," *Journal of Cyber Security*, vol. 6, no. 1, pp. 25–39, 2024, https://doi.org/10.32604/jcs.2023.045859.

[7] S. Das Guptta, K. T. Shahriar, H. Alqahtani, D. Alsalman, and I. H. Sarker, "Modeling Hybrid Feature-Based Phishing Websites Detection Using Machine Learning Techniques," *Annals of Data Science*, vol. 11, no. 1, pp. 217–242, Feb. 2024, https://doi.org/10.1007/s40745-022-00379-8.

[8] P. Maturure, A. Ali, and A. Gegov, "Hybrid Machine Learning Model for Phishing Detection," in *2024 IEEE 12th International Conference on Intelligent Systems (IS)*, Varna, Bulgaria, Aug. 2024, pp. 1–7, https://doi.org/10.1109/IS61756.2024.10705257.

[9] L. Tang and Q. H. Mahmoud, "A Survey of Machine Learning-Based Solutions for Phishing Website Detection," *Machine Learning and Knowledge Extraction*, vol. 3, no. 3, pp. 672–694, Sep. 2021, https://doi.org/10.3390/make3030034.

[10] C. Opara, Y. Chen, and B. Wei, "Look before you leap: Detecting phishing web pages by exploiting raw URL and HTML characteristics," *Expert Systems with Applications*, vol. 236, Feb. 2024, Art. no. 121183, https://doi.org/10.1016/j.eswa.2023.121183.

[11] A. Yasin, R. Fatima, Z. JiangBin, W. Afzal, and S. Raza, "Can serious gaming tactics bolster spear-phishing and phishing resilience? : Securing the human hacking in Information Security," *Information and Software Technology*, vol. 170, Jun. 2024, Art. no. 107426, https://doi.org/10.1016/j.infsof.2024.107426.

[12] B. Montaruli, L. Demetrio, M. Pintor, L. Compagna, D. Balzarotti, and B. Biggio, "Raze to the Ground: Query-Efficient Adversarial HTML Attacks on Machine-Learning Phishing Webpage Detectors," in *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security*, Copenhagen, Denmark, Nov. 2023, pp. 233–244, https://doi.org/10.1145/3605764.3623920.

[13] A. Safi and S. Singh, "A systematic literature review on phishing website detection techniques," *Journal of King Saud University - Computer and Information Sciences*, vol. 35, no. 2, pp. 590–611, Feb. 2023, https://doi.org/10.1016/j.jksuci.2023.01.004.

[14] 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, no. 3, pp. 3043–3070, Jun. 2023, https://doi.org/10.1007/s40747-022-00760-3.

[15] N. Beu *et al.*, "Falling for phishing attempts: An investigation of individual differences that are associated with behavior in a naturalistic phishing simulation," *Computers & Security*, vol. 131, Aug. 2023, Art. no. 103313, https://doi.org/10.1016/j.cose.2023.103313.

[16] A. Jayatilaka, N. A. G. Arachchilage, and M. A. Babar, "Why People Still Fall for Phishing Emails: An Empirical Investigation into How Users Make Email Response Decisions." arXiv, Jan. 24, 2024, https://doi.org/10.48550/arXiv.2401.13199.

[17] B. Naqvi, K. Perova, A. Farooq, I. Makhdoom, S. Oyedeji, and J. Porras, "Mitigation strategies against the phishing attacks: A systematic literature review," *Computers & Security*, vol. 132, Sep. 2023, Art. no. 103387, https://doi.org/10.1016/j.cose.2023.103387.

[18] R. Hoheisel, G. Van Capelleveen, D. K. Sarmah, and M. Junger, "The development of phishing during the COVID-19 pandemic: An analysis of over 1100 targeted domains," *Computers & Security*, vol. 128, May 2023, Art. no. 103158, https://doi.org/10.1016/j.cose.2023.103158.

[19] A. Wu, Z. Feng, X. Li, and J. Xiao, "ZTWeb: Cross site scripting detection based on zero trust," *Computers & Security*, vol. 134, Nov. 2023, Art. no. 103434, https://doi.org/10.1016/j.cose.2023.103434.

[20] R. Brindha, S. Nandagopal, H. Azath, V. Sathana, G. Prasad Joshi, and S. Won Kim, "Intelligent Deep Learning Based Cybersecurity Phishing Email Detection and Classification," *Computers, Materials & Continua*, vol. 74, no. 3, pp. 5901–5914, 2023, https://doi.org/10.32604/cmc.2023.030784.

[21] S. Kuraku and D. Kalla, "Impact of phishing on users with different online browsing hours and spending habits," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 12, no. 10, Oct. 2023, https://doi.org/10.17148/ijarcce.2023.121005.

[22] R. Zieni, L. Massari, and M. C. Calzarossa, "Phishing or Not Phishing? A Survey on the Detection of Phishing Websites," *IEEE Access*, vol. 11, pp. 18499–18519, 2023, https://doi.org/10.1109/access.2023.3247135.

[23] Q. Qi, Z. Wang, Y. Xu, Y. Fang, and C. Wang, "Enhancing Phishing Email Detection through Ensemble Learning and Undersampling," *Applied Sciences*, vol. 13, no. 15, Jul. 2023, Art. no. 8756, https://doi.org/10.3390/app13158756.

[24] S. Zhuo *et al.*, "What You See is Not What You Get: The Role of Email Presentation in Phishing Susceptibility." arXiv, Apr. 03, 2023, https://doi.org/10.48550/arXiv.2304.00664.

[25] M. H. Alsuwit, M. A. Haq, and M. A. Aleisa, "Advancing Email Spam Classification using Machine Learning and Deep Learning Techniques,"

Engineering, Technology & Applied Science Research, vol. 14, no. 4, pp. 14994–15001, Aug. 2024, https://doi.org/10.48084/etasr.7631.

[26] A. Alzahrani, "Explainable AI-based Framework for Efficient Detection of Spam from Text using an Enhanced Ensemble Technique," *Engineering, Technology & Applied Science Research*, vol. 14, no. 4, pp. 15596–15601, Aug. 2024, https://doi.org/10.48084/etasr.7901.

[27] A. Jadhav and P. R. Chandre, "Survey and comparative analysis of phishing detection techniques: current trends, challenges, and future directions," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 14, no. 2, Apr. 2025, Art. no. 853, https://doi.org/10.11591/ijai.v14.i2.pp853-866.

[28] A. Prasad and S. Chandra, "PhiUSIIL: A diverse security profile empowered phishing URL detection framework based on similarity index and incremental learning," *Computers & Security*, vol. 136, Jan. 2024, Art. no. 103545, https://doi.org/10.1016/j.cose.2023.103545.

[29] C. L. Tan, "Phishing Dataset for Machine Learning: Feature Evaluation." Mendeley, Mar. 24, 2018, https://doi.org/10.17632/H3CGNJ8HFT.1.

[30] S. Marchal, J. François, R. State, and T. Engel, "PhishStorm: Detecting Phishing With Streaming Analytics," *IEEE Transactions on Network and Service Management*, vol. 11, no. 4, pp. 458–471, Sep. 2014, https://doi.org/10.1109/TNSM.2014.2377295.

[31] A. Hannousse, "Web page phishing detection." Mendeley, Jun. 25, 2021, https://doi.org/10.17632/C2GW7FY2J4.3.