Advances in Artificial Intelligence and Machine Learning; Research 6 (4) 6108-6121 Received 23-02-2026; Accepted 21-08-2026; Published online 28-08-2026

OFSPUD: Optimizing Feature Selection for Phishing URL Detection through Statistical Tests and Ensemble Methods

Rachana Sandeep Potpelwar rspotpelwar@sggs.ac.in
Information Technology Department
Shri Guru Gobind Singhji Institute of Engineering and Technology,
Vishnupuri, India

U. V. Kulkarni uvkulkarni@sggs.ac.in
Department of Computer Science and Engineering
Shri Guru Gobind Singhji Institute of Engineering and Technology,
Vishnupuri, India.

J. M. Waghmare jmwaghmare@sggs.ac.in
Department of Computer Science and Engineering
Shri Guru Gobind Singhji Institute of Engineering and Technology,
Vishnupuri, India.

Corresponding Author: Rachana Sandeep Potpelwar

Abstract

The surge in online users utilizing cloud-based platforms—particularly for financial and retail services—is fueled by the convenience and flexibility these services provide. To effectively counter evolving URL phishing threats, machine learning classifiers should analyze all components of URLs, including domain structures, path patterns, and query parameters, to enhance threat detection. The proposed framework, OFSPUD (Optimizing Feature Selection for Phishing URL Detection through Statistical Tests and Ensemble Methods), was evaluated using a large, imbalanced dataset containing 11,055 benign and phishing URLs with over 31 features. Using ANOVA (Analysis of Variance), 25 of the 31 features were selected, significantly improving the method's performance and correctness. The RBWSV (Rank-Based Weighted Soft Voting) ensemble method outperformed prominent machine learning techniques, achieving test accuracy of 97.47% and F1 and validation accuracies of 99% after applying ANOVA-based feature selection. Thus, the proposed method, combining ANOVA-based feature selection with the RBWSV ensemble approach, achieved near-perfect accuracy while improving efficiency.

Citation: Rachana Sandeep Potpelwar, et al. OFSPUD: Optimizing Feature Selection for Phishing URL Detection through Statistical Tests and Ensemble Methods. **Advances in Artificial Intelligence and Machine Learning**. 2026;6(4):338. https://dx.doi.org/10.54364/AAIML.2026.64338

Keywords: Phishing URLs, Legitimate URLs, Classification algorithms, Ensemble learning, Statistical testing

1. INTRODUCTION

Every day, a large number of websites are launched which include authentic portals to obtain users' login details [1]. The massive scale of these websites makes it difficult to verify their credibility [2]. Therefore, overcoming these existing systems is a second challenge. Phishing attacks occur in various forms, including email phishing [3], and common phishing attacks [4, 5]. Consequently, there is a critical need to effectively distinguish between phishing and legitimate websites to mitigate such attacks [6, 7].

In this context, we introduce the OFSPUD framework, which addresses these challenges through a dual approach: (1) ANOVA-based feature selection to optimize the feature set and reduce dimensionality, and (2) a ranking-based weighted soft vote ensemble method that combines multiple machine learning models for more robust predictions. By integrating statistical feature selection with an ensemble approach, OFSUPD aims to maximize detection accuracy while minimizing computational complexity.

Our contributions, OFSPUD framework was evaluated using a large-scale, imbalanced dataset comprising 11,055 legitimate and phishing URLs and 30 features, thereby providing a challenging and realistic evaluation environment. The findings show that the ANOVA-based feature selection method successfully pinpoints 25 key features, leading to a notable improvement in classification performance. Combined with a weighted ensemble learning strategy, the OFSPUD model achieved a test accuracy of 97.47%, validation accuracy of 99%, and a strong F1 score, demonstrating robust predictive power. These results shows the proposed work achieved high accuracy and efficiency [8].

The key contributions of our study are as follows:

- We propose a hybrid URLs feature that provides a comprehensive and accurate representation of both benign and phishing URLs.
- We show that combining statistical testing with the ranking-based weighted soft-computing ensemble method improves phishing detection accuracy.

Our work is organized as follows. Section II reviews on the literature work. Section III presents our contributions in detail. Section IV outlines the framework which includes the proposed assessment guidelines, and presents and discusses the experimental results. Section V, summarizes the key findings and conclusions of our study.

2. RELATED WORK

In [5], the study is proposed on feature weight selection. In this work, the training process was conducted using dataset that contain both legitimate and phishing websites. The Particle Swarm Optimization (PSO) algorithm was used in this study because it achieves fast convergence in numerous optimization problems. After feature weighting based on PSO, many machine learning algorithms were used. The classification accuracy was good when applied the PSO-based feature weighting method [9, 10].

Chiew et al. (2019) [11], proposed a hybrid ensemble feature selection framework for machine learning-based phishing detection systems. The framework combines multiple feature selection techniques to identify the most discriminative phishing website features while reducing dimensionality and computational overhead. Using a diverse set of URL, domain, and webpage-based attributes, the authors evaluated several machine learning classifiers and demonstrated that ensemble feature selection can improve detection performance while reducing training complexity. Their results highlighted the importance of selecting informative features for building efficient and scalable phishing detection systems [12, 13].

2.1 Comparison with State-of-the-Art Methods

The proposed OFSPUD framework shown in Figure 1 achieved superior overall performance, obtaining a test accuracy of 97.47%, precision with other metrics 99.0%, as shown in TABLE 1. The improvements are the combination of ANOVA-based feature selection and the proposed Rank-Based Weighted Soft Voting (RBWSV) ensemble strategy, which effectively reduces feature redundancy while leveraging complementary strengths of multiple classifiers. Although direct comparison should be interpreted carefully due to differences in datasets and experimental settings, the results indicate that OFSPUD achieves competitive and, in several cases, superior detection performance while maintaining low computational complexity.

Table 1: Comparison with Methods

method	accuracy (%)	precision (%)	recall (%)	F1-score (%)
PSO [14]	96.43	96.11	97.12	97.13
Hybrid Ensemble Model [11]	97.30	97.20	97.30	97.20
OFSPUD (Proposed)	97.47	99.00	99.00	99.00

3. METHODOLOGY

3.1 Url Parsing

All URLs undergo the parsing stage, in which the corresponding fields based on address bar(A), domain(D), and HTML and JavaScript(H) based features are extracted and stored in arrays A_{nxi}, where i is several features extracted for address-based features (Eq.(1)). D_{nxj}, where j is the number of features extracted for domain-based features (Eq. 2) and H_{nxk}, where k is the number of features extracted for HTML and JavaScript-based features (Eq.(3)). The respective class labels are maintained in the array L, which identifies each URL as either phishing or legitimate. Here, n denotes the total number of URL samples included in the dataset. The last step of the URL parsing creates a joint array E(final combined feature matrix) (Eq.(4)) by merging arrays A, D, H, and L. The resultant matrix prepares the data for subsequent analysis and classification tasks [15].

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1i} \\ a_{21} & a_{22} & \dots & a_{2i} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{ni} \end{bmatrix} \tag{1}$$

$$D = \begin{bmatrix} d_{11} & d_{12} & \dots & d_{1j} \\ d_{21} & d_{22} & \dots & d_{2j} \\ \vdots & \vdots & \ddots & \vdots \\ d_{n1} & d_{n2} & \dots & d_{nj} \end{bmatrix} \tag{2}$$

$$H = \begin{bmatrix} h_{11} & h_{12} & \dots & h_{1k} \\ h_{21} & h_{22} & \dots & h_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ h_{n1} & h_{n2} & \dots & h_{nk} \end{bmatrix} \tag{3}$$

$$E = \begin{bmatrix} a_{11} \dots a_{1i} & d_{11} \dots d_{1j} & h_{11} \dots h_{1k} \\ a_{22} \dots a_{2i} & d_{22} \dots d_{2j} & h_{22} \dots h_{2k} \\ \vdots \quad \vdots & \ddots & \vdots \\ a_{n1} \dots a_{ni} & d_{n1} \dots d_{nj} & h_{n1} \dots h_{nk} \end{bmatrix} \tag{4}$$

3.2 Pre-Processing

At this stage, the correlation of the features is identified using ANOVA test [16]. We formulated our hypotheses as follows and the notations are shown in Table 2: the null hypothesis (H_0) states that the means of the features are equal for both phishing and legitimate URLs, whereas the alternative hypothesis (H_1) posits that at least one feature mean is significantly different, to calculate the Total Sum of Squares (SST) and Between-Group Sum of Squares (SSB), is given by

Table 2: The Key Notations Used in the ANOVA testing for the calculation of the OFSPUD model

Notation	Description
α_j	Represents intra-group variability adjustment for the j-th feature group. It can be derived from the entropy or variance within the feature group to account for feature consistency.
β_j	Between-group weighting factor for the jth feature group. It reflects the relevance or importance of the feature category (e.g., domain features vs. address bar features) to the classification task.
γ_{ij}	Confidence weight for the i-th sample in the j-th group. It could be derived from a model's prediction confidence or the feature's contribution to a decision boundary.
k	Number of groups in the dataset. For phishing datasets, this is often 2 (phishing and legitimate), but it could be extended for multiclass problems (e.g., different phishing attack types).
N	Number of records in the dataset.
x_{ij}	Value of a specific feature for a sample. For phishing datasets, features might include URL length, dot count, presence of suspicious keywords, and domain age.
n_j	Denotes the samples of j-th group.
$\bar{x}_j$	Average of the j-th group.
$\bar{x}$	Overall mean across all groups.

$$SST = \sum_{j=1}^{k} \sum_{i=1}^{n_j} \gamma_{ij} \cdot (x_{ij} - \bar{x})^2 \tag{5}$$

$$SSB = \sum_{j=1}^{k} \beta_j \cdot n_j \cdot (\bar{x}_j - \bar{x})^2 \tag{6}$$

The Within-Group Sum of Squares (SSW) is defined as:

$$SSW = \sum_{j=1}^{k} \alpha_j \cdot \sum_{i=1}^{n_j} \gamma_{ij} \cdot (x_{ij} - \bar{x}_j)^2 \tag{7}$$

The F-statistic is calculated as:

$$F = \frac{MSB}{MSW}, \quad \text{where} \quad MSB = \frac{SSB}{k-1}, \quad MSW = \frac{SSW}{N-k} \tag{8}$$

Table 3: ANOVA Testing Results for Phishing Dataset Features

Feature	Sum of Squares (SS)	Mean Square (MS)	F-Ratio	P-Value
having_IP_Address	9967.347928	0.901614	98.873891	3.367×10^{-23}
URL_Length	6488.203365	0.586902	36.575219	1.516×10^{-9}
Shortining_Service	5021.995296	0.454274	51.294764	8.454×10^{-13}
having_At_Symbol	5629.455401	0.509223	31.073851	2.542×10^{-8}
double_slash_redirecting	4977.584223	0.450256	16.499617	4.900×10^{-5}
Prefix_Suffix	5083.897232	0.459873	1529.043445	2.269×10^{-313}
having_Sub_Domain	7388.453591	0.668336	1079.778531	4.918×10^{-226}
SSLfinal_State	9192.759544	0.831548	11543.570464	0
Domain_registeration_length	9802.089379	0.886666	593.762202	8.626×10^{-128}
Favicon	6687.572282	0.604936	0.000864	0.976
port	5192.185996	0.469669	14.679425	1.281×10^{-4}
HTTPS_token	6017.428623	0.544317	17.583773	2.771×10^{-5}
Request_URL	10670.237018	0.965196	758.253010	1.680×10^{-161}
URL_of_Anchor	5653.770038	0.511422	10209.239894	0
Links_in_tags	6452.297268	0.583654	725.777602	6.959×10^{-155}
SFH	6370.976479	0.576298	569.825002	7.626×10^{-123}
Submitting_to_email	6588.953863	0.596016	3.682166	0.055
Abnormal_URL	5556.343043	0.502609	40.588721	1.954×10^{-10}
Redirect	1131.129365	0.102318	4.473316	0.034
on_mouseover	4634.738556	0.419244	19.381707	1.080×10^{-5}
RightClick	1822.183463	0.164829	1.769917	0.183
popUpWidnow	6896.242446	0.623812	0.000082	0.993
Iframe	3677.769495	0.332679	0.127288	0.721
age_of_domain	11014.537362	0.996340	165.602039	1.258×10^{-37}
DNSRecord	9483.667813	0.857862	63.734211	1.565×10^{-15}
web_traffic	7574.249502	0.685142	1504.193018	1.271×10^{-308}
Page_Rank	8469.569025	0.766130	122.376589	2.687×10^{-28}
Google_Index	5299.483988	0.479374	186.899510	3.327×10^{-42}
Links_pointing_to_page	3591.065316	0.324836	11.740341	6.138×10^{-4}
Statistical_report	5331.192329	0.482243	70.938416	4.141×10^{-17}

MSB is the mean square between groups, and MSW is the mean square within groups.

$$p = \begin{cases} > 0.05 & \text{No significance (feature is not selected)} \\ \leq 0.05 & \text{Feature is selected} \end{cases}$$

The contribution of features to classification performance is determined by assessing the F-statistic against critical values as shown in Table 3. Features with p-values below this threshold (p[1] = 0.05) were considered significant and were retained for model training, whereas those with p-values above the threshold were deemed less relevant and excluded.

[1] In our analysis, a p-value of 0.05 was employed to determine whether there is statistically significant difference.

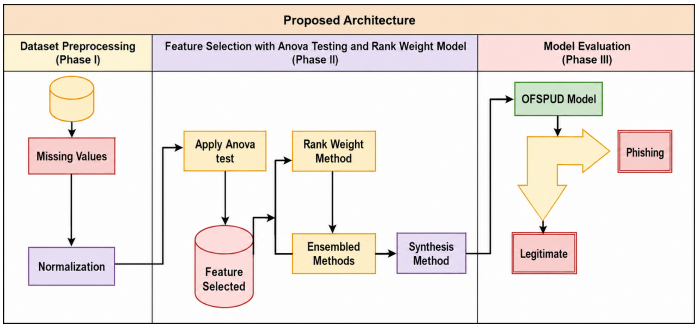

Figure 1: Proposed Architecture

3.3 Ensemble Classification

As in Algorithm 1, M_i predicts the probability scores for each class in a classification problem. The predicted probability scores for a particular instance by model M_i can be denoted as $\text{Scores}_{M_i} = [s_1, s_2, \ldots, s_c]$, where c is the number of classes. Ranking R_i can be obtained by sorting the indices of predicted scores in descending order [17].

$$\text{Rank}_{M_i}(j) = \text{argsort}(\text{Scores}_{M_i})[j] \tag{9}$$

Here, $\text{argsort}(\text{Scores}_{M_i})$ returns the indices that would sort the predicted scores in ascending order. The $[j]$ notation extracts the index at position j after sorting in descending order. The ranking function orders predicted probabilities generated by each base learner. This ranking information is subsequently used to assign weights during ensemble aggregation. Consequently, classifiers exhibiting stronger confidence in their predictions contribute more significantly to the final decision, thereby improving robustness and reducing the impact of uncertain predictions.

4. EXPERIMENTAL RESULTS

4.1 Dataset and Feature Evaluation

TABLE 4 summarises the key characteristics of each dataset that includes a combination of lexical, structural, HTML/JavaScript, and domain-based features.

Table 4: Characteristics of phishing URL datasets

Dataset	Features	Instances	Phishing (%)	Source
Dataset1	31	11,055	45.6	UCI Repository
Dataset2	111	129,698	52.3	Kaggle
Dataset3	89	11,430	48.7	Kaggle
Dataset4	77	32,157	41.2	UNB CIC

Algorithm 1: Overall algorithm for ranking-based Weighted Soft Voting ensemble method.

Data: Dataset D, Base models $M_1, M_2, \ldots, M_k$
Result: Ensemble model E

Step 1: Train base models;
for $i = 1$ **to** k **do**
    Train base model M_i on dataset D;

Step 2: Generate rankings;
for $i = 1$ **to** k **do**
    Generate ranking R_i based on the rank of predicted classes by M_i on a validation set;
This ranking, R_i, represents the order of the predicted classes by model M_i for a particular instance in the validation set.
In sorted order, the indices of array A can be obtained using $\mathrm{argsort}(A)$.

Step 3: Assign weights based on rankings;
for $i = 1$ **to** k **do**
    Compute weight w_i based on the ranking R_i;

Step 4: Combine predictions;
Initialize empty ensemble predictions array P;
for $i = 1$ **to** k **do**
    Get predictions P_i from base model M_i on the test set;
    Combine predictions using weights: $P = P + w_i \cdot P_i$;

Step 5: Normalize ensemble predictions;
Normalize predictions in P to get the final ensemble predictions E;
return *Ensemble model E*

4.2 Experimental Environment

The work was conducted on a personal computer with an Intel Core i5 processor, 16 GB of RAM, and windows 11. The OFSPUD framework was implemented in Python 3.11 using Scikit-learn, NumPy, Pandas, and Matplotlib. Feature selection used ANOVA. Model training and evaluation used Scikit-learn. The setup allowed thorough evaluation of OFSPUD's detection performance, efficiency, and readiness for real-time phishing detection.

4.3 URL-based feature extraction

URLs exhibit over 31 distinct features, broadly categorized into four types that are given in the TABLE 5.

Table 5: Features and Values for Phishing Item Detection

Category	No.	Phishing Item	Abbreviation	Value
Address bar features	F_1	URL contains an IP Address	having IP_Address	-1, 0, 1
	F_2	length of URL	URL_Length	-1, 0, 1
	F_3	URL shortening service	Shortening Service	-1, 1
	F_4	URLs containg @ symbol	having _ At_Symbol	-1, 1
	F_5	URL uses redirect symbol '//'	double slash redirecting	-1, 1
	F_6	URL contains '_' symbol	prefix_suffix	-1, 1
	F_7	Number of '.' in the URL	having _sub_Domain	-1, 0, 1
	F_8	Protocol and SSL certificate status	SSL final State	-1, 0, 1
	F_9	Domain registration length	Domain register length	-1, 1
	F_10	Favicon presence	favicon	-1, 1
	F_11	Uses non-standard port	port	-1, 1
	F_12	Tokens of HTTPS	HTTPS_token	-1, 1
Abnormal features	F_13	URL request	URL request	-1, 1
	F_14	URL Anchor tag	URL_of_Anchor tag	-1, 0, 1
	F_15	Links in the tags	Tags_link	-1, 0, 1
	F_16	server from handler	sfh	-1, 0, 1
	F_17	Information of submitting Email	submitting email	-1, 1
	F_18	abnormal URL	Abnormal URL	-1, 1
HTML and JavaScript features	F_19	Website forwarding	Redirect	-1, 0, 1
	F_20	Status bar customization	on_mouseover	-1, 1
	F_21	Disables right-click	RIght Click	-1, 1
	F_22	popup window	popUpWindow	-1, 1
	F_23	IFrame redirections	Iframe	-1, 1
Domain features	F_24	domain's age	Age_of_Domain	-1, 1
	F_25	DNS records	DNSRecords	-1, 1
	F_26	Website traffic	web_traffic	-1, 0, 1
	F_27	PageRank	page_rank	-1, 1
	F_28	google index status	google_index	1, -1
	F_29	total number of links pointing to the page	links_pointing_to_page	1, 0 , -1
	F_30	statistical based features	Statistical_report	-1, 1

4.4 OFSPUD(Proposed) Evaluation

The performance evaluation confirms the robustness and effectiveness of the proposed OFSPUD framework. By integrating ANOVA-based feature selection with the RBWSV ensemble strategy, the proposed model achieved a test accuracy of 97.47% and 99% for validation accuracy as shown in FIGURE 2 and F1-score, respectively.

These results as shown in Table 6 demonstrate that the proposed framework can accurately identify phishing URLs while maintaining low computational overhead, thereby offering a practical solution for real-time phishing detection systems. As shown in FIGURE 3 and FIGURE 4, the experimental results of the ROC curve and confusion matrix. Furthermore, the dual approach of ANOVA-based feature optimization and RBWSV ensemble ensured robustness against feature redundancy and model complexity, providing a scalable solution for evolving cyber threats.

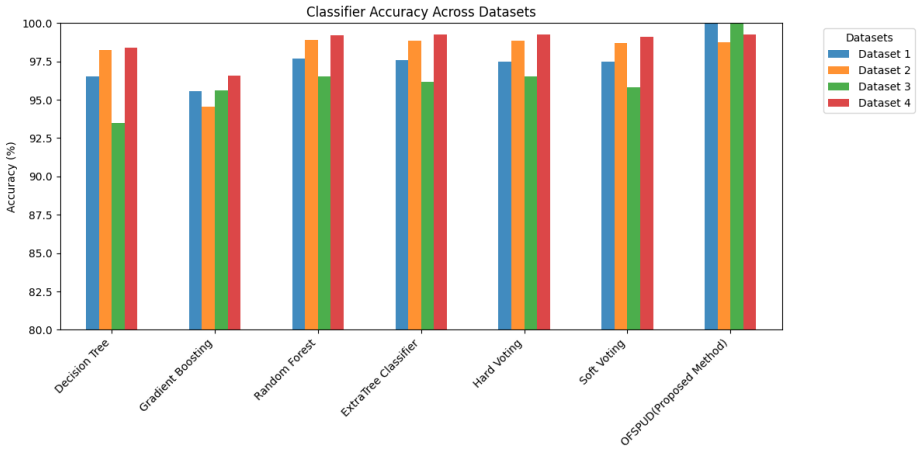

Figure 2: Accuracy

Table 6: Performance Comparison of Classifiers Across Four Datasets

Classifier	Metric	Dataset 1	Dataset 2	Dataset 3	Dataset 4
Decision Tree	Accuracy	96.52%	98.23%	93.48%	98.40%
	Precision	97.00%	99.00%	94.00%	98.00%
	Recall	97.00%	99.00%	94.00%	98.00%
	F1-Score	96.50%	99.00%	93.50%	98.00%
Gradient Boosting	Accuracy	95.57%	94.54%	95.63%	96.58%
	Precision	96.00%	96.00%	96.00%	97.00%
	Recall	96.00%	95.00%	96.00%	98.00%
	F1-Score	96.00%	95.00%	96.00%	97.00%
Random Forest	Accuracy	97.69%	98.92%	96.50%	99.22%
	Precision	98.00%	99.00%	97.00%	99.00%
	Recall	98.00%	99.00%	97.00%	99.00%
	F1-Score	97.50%	99.00%	96.50%	99.00%
ExtraTree Classifier	Accuracy	97.60%	98.84%	96.15%	99.28%
	Precision	98.00%	99.00%	96.00%	99.00%
	Recall	98.00%	99.00%	96.00%	99.00%
	F1-Score	97.50%	99.00%	96.00%	99.00%
Hard Voting	Accuracy	97.51%	98.87%	96.50%	99.27%
	Precision	98.00%	99.00%	96.00%	99.00%
	Recall	98.00%	99.00%	97.00%	99.00%
	F1-Score	97.50%	99.00%	96.50%	99.00%
Soft Voting	Accuracy	97.47%	98.72%	95.84%	99.13%
	Precision	98.00%	99.00%	96.00%	99.00%
	Recall	98.00%	99.00%	96.00%	99.00%
	F1-Score	97.50%	99.00%	96.00%	99.00%
OFSPUD (Proposed)	Accuracy	**98.00%**	98.75%	95.89%	99.24%
	Precision	**98.00%**	99.00%	96.00%	99.00%
	Recall	**99.00%**	99.08%	96.00%	99.00%
	F1-Score	**99.00%**	98.90%	96.00%	99.00%

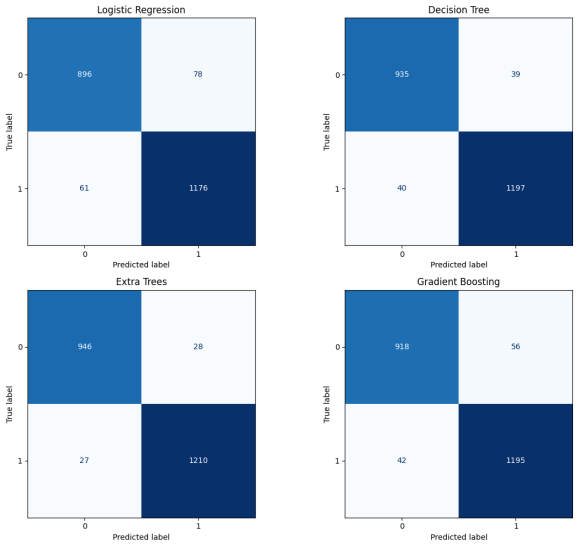

Figure 3: Confusion Matrix

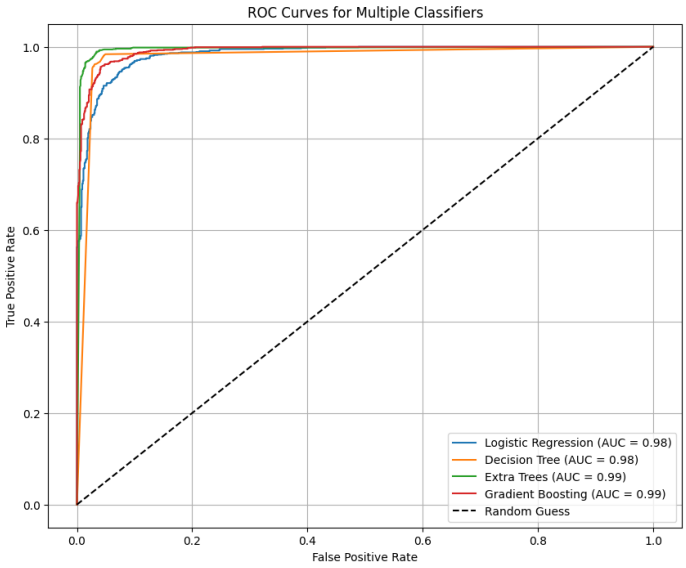

Figure 4: ROC Curve

5. CONCLUSION

The proposed method consists of primary models that process different features individually, but in parallel. The results indicate that, using statistical testing, the best performance is achieved by extracting the importance of the characteristic. The experimental results illustrate that OFSPUD,

using hybrid features, achieves superior performance with Rank-Based weighted soft vote ensemble learning, achieving a 99% F1 score and precision/recall, along with a low training time of 0.0786 s as given in Table 7.

Table 7: Time Efficiency Estimation of Ensemble Classifiers Before and After ANOVA Testing

Classifier	**Learning Time (s)**	**Test Time (s)**	**Learning Time (s)**	**Test Time (s)**
	Before ANOVA Testing		**After ANOVA Testing**	
Logistic Regression(LR)	0.0557	0.0009	0.0503	0.0012
Support Vector Machine (SVM)	2.8451	0.6489	1.2273	0.2683
Decision Tree	0.0298	0.0015	0.0258	0.0009
Gradient Boosting Classifier(GBC)	0.9927	0.0060	1.0937	0.0070
Random Forest(RF)	0.9099	0.0333	0.6301	0.0356
Extra Tree Classifier	0.6901	0.0396	0.8242	0.0512
Hard Voting Classifier	3.0082	0.4536	2.5767	0.3988
Soft Voting Classifier	5.4452	0.4142	3.0515	0.1125
OFSPUD (Proposed Method)	4.8343	0.0768	2.5559	0.0786

The observed improvement is due to the combination of diverse feature representations and ensemble classification techniques guided by statistical analysis. Moreover, OFSPUD is likely to promote further research in hybrid feature modeling and enhance the adoption of ensemble learning in phishing detection.

Although OFSPUD demonstrates strong phishing URL detection performance, several opportunities remain for further enhancement. In future, our work may focus on developing adaptive feature selection mechanisms that are capable of responding to the growth of these phishing attacks [18]. The incorporation of Explainable Artificial Intelligence (XAI) techniques, that includes LIME, SHAP, and counterfactual explanations, will be investigated to enhance model transparency. In the future, the integration of lightweight ensemble methods, deep learning models, and graph-based learning architectures [13, 19–21], may further enhance detection accuracy, generalization capability, and real-time deployment performance.

References

[1] Moghimi M, Varjani AY. New Rule-Based Phishing Detection Method. Expert Syst Appl. 2016;53:231-42.

[2] Abdelhamid N, Ayesh A, Thabtah F. Phishing Detection Based Associative Classification Data Mining. Expert Syst Appl. 2014;41:5948-5959.

[3] Fang Y, Zhang C, Huang C, Liu L, Yang Y. Phishing Email Detection Using Improved Rcnn Model With Multilevel Vectors and Attention Mechanism. IEEE Access. 2019;7:56329-40.

[4] Khonji M, Iraqi Y, Jones A. Phishing Detection: A Literature Survey. IEEE Commun Surv Tutorials. 2013;15:2091-121.

[5] Ali W, Malebary S. Particle Swarm Optimization-Based Feature Weighting for Improving Intelligent Phishing Website Detection. IEEE Access. 2020;8:116766-116780.

[6] Gupta BB, Yadav K, Razzak I, Psannis K, Castiglione A, Chang X. A Novel Approach for Phishing URL Detection Using Lexical-Based Machine Learning in a Realtime Environment. Comput Commun. 2021;175:47-57.

[7] Lin G, Sun N, Nepal S, Zhang J, Xiang Y, Hassan H. Statistical Twitter Spam Detection Demystified: Performance, Stability and Scalability. IEEE Access. 2017;5:11142-54.

[8] Mandadi A, Boppana S, Ravella V, Kavitha R. Phishing Website Detection Using Machine Learning. In 2022 IEEE 7th International Conference for Convergence in Technology (I2CT). IEEE. 2022:1-4.

[9] Orunsolu AA, Sodiya AS, Akinwale AT. A Predictive Model for Phishing Detection. J King Saud Univ Comput Inf Sci. 2022;34:232-247.

[10] Motiur Rahman SS, Islam T, Jabiullah MI.Phishstack: Evaluation of Stacked Generalization in Phishing URLs Detection. Procedia Comput Sci. 2020;167:2410-8.

[11] Chiew KL, Tan CL, Wong K, Yong KS, Tiong WK. A New Hybrid Ensemble Feature Selection Framework for Machine Learning-Based Phishing Detection System. Inf Sci. 2019;484:153-66.

[12] Alsariera YA, Adeyemo VE, Balogun AO, Alazzawi AK. AI Metalearners and Extra-Trees Algorithm for the Detection of Phishing Websites. IEEE Access. 2020;8:142532-142542.

[13] Zhu E, Chen Y, Ye C, Li X, Liu F. OFS-NN: An Effective Phishing Websites Detection Model Based on Optimal Feature Selection and Neural Network. IEEE Access. 2019;7:73271-84.

[14] Feng J, Zou L, Ye O, Han J. Web2Vec: Phishing Webpage Detection Method Based on Multidimensional Features Driven by Deep Learning. IEEE Access. 2020;8:221214-221224.

[15] Tan CL, Chiew KL, Yong KS, Sze SN, Abdullah J, Sebastian Y. A Graph-Theoretic Approach for the Detection of Phishing Webpages. Comput Secur. 2020;95:101793.

[16] Hannousse A, Yahiouche S. Towards Benchmark Datasets for Machine Learning Based Website Phishing Detection: An Experimental Study. Eng Appl Artif Intell. 2021;104:104347.

[17] Sahingoz OK, Buber E, Demir O, Diri B. Machine Learning Based Phishing Detection From URLs. Expert Syst Appl. 2019;117:345-357.

[18] Uzel VN, Essiz ES, Ozel SA. Using Fuzzy Sets for Detecting Cyber Terrorism and Extremism in the Text. In 2018 Innovations in Intelligent Systems and Applications Conference (ASYU). IEEE. 2018:1-4.

[19] Yuan J, Chen G, Tian S, Pei X. Malicious URL Detection Based on a Parallel Neural Joint Model. IEEE Access. 2021;9:9464-9472.

[20] Yang P, Zhao G, Zeng P. Phishing Website Detection Based on Multidimensional Features Driven by Deep Learning. IEEE Access. 2019;7:15196-15209.

[21] Adebowale MA, Lwin KT, Sánchez E, Hossain MA. Intelligent Webphishing Detection and Protection Scheme Using Integrated Features of Images, Frames and Text. Expert Syst Appl. 2019;115:300-13.