European Journal of Computer Science and Information Technology, 14(1),24-40, 2026

Print ISSN: 2054-0957 (Print)

Online ISSN: 2054-0965 (Online)

Website: https://www.eajournals.org/

Publication of the European Centre for Research Training and Development -UK

Intelligent Predictive Analytics Model for Detecting and Preventing Phishing Attacks in Institutional Networks

[1]Ekemini Anietie Johnson; [1]Mfon Okpu Esang; [2]Anietie Emmanuel John
[1]Department of Computer Science, Federal Polytechnic Ukana, Akwa Ibom State, Nigeria
[2]Department of Mathematics and Computer Science,Ritman University Ikot Ekpene, Akwa Ibom State, Nigeria

doi: https://doi.org/10.37745/ejcsit.2013/vol14n12440 Published January 31, 2026

Citation: Johnson E.A., Esang M.F., John A.E. (2026) Intelligent Predictive Analytics Model for Detecting and Preventing Phishing Attacks in Institutional Networks, *European Journal of Computer Science and Information Technology*, 14(1),24-40

Abstract: *Phishing attacks remain one of the most persistent and damaging cybersecurity threats affecting institutional networks worldwide. With the increasing sophistication of social engineering techniques and malicious web infrastructures, traditional rule-based and signature-based detection systems have become insufficient. This study proposes an intelligent predictive analytics model for detecting and preventing phishing attacks within institutional environments. The model leverages supervised machine learning techniques to analyze URL- and content-based features for accurate phishing classification. A dataset containing 2,200 labeled instances was used, and key features were selected through preprocessing and dimensionality reduction techniques. Two supervised learning models; Random Forest (RF) and Support Vector Machine (SVM) were implemented and evaluated using standard performance metrics including accuracy, precision, recall, and F1-score. Experimental results demonstrate that the RF model outperformed SVM, achieving an accuracy of 95.7% compared to 93.3% for SVM. The findings confirm that intelligent predictive analytics significantly enhances phishing detection accuracy and provides a scalable, adaptive solution for institutional cybersecurity systems.*

Keywords: intelligent predictive analytics, model detecting, preventing phishing attacks, institutional networks

INTRODUCTION

The rapid expansion of digital communication technologies has significantly increased reliance on online platforms for academic, administrative, and commercial operations. However, this growth has also led to a rise in cybersecurity threats, particularly phishing attacks. Phishing is a deceptive cybercrime technique where attackers impersonate legitimate entities to manipulate users into revealing sensitive information such as login credentials, financial details, or personal data. According to the Anti-Phishing Working Group (APWG, 2024), phishing attacks continue to increase annually, posing serious threats to individuals and organizations alike.

Institutional networks, including educational institutions, financial organizations, and government agencies, are especially vulnerable due to large user populations and extensive digital infrastructures. Traditional phishing detection mechanisms—such as blacklist filtering, heuristic rules, and signature-based detection—are increasingly ineffective against modern phishing techniques that employ obfuscation, social engineering, and dynamic content generation.

To address these challenges, intelligent predictive analytics powered by machine learning (ML) offers a promising solution. By learning patterns from historical data, machine learning models can identify both known and previously unseen phishing attempts. This study proposes an intelligent predictive analytics framework that utilizes supervised learning algorithms to detect phishing attacks with high accuracy and reliability.

LITERATURE REVIEW

Phishing detection has attracted significant research interest over the past two decades. Early detection methods relied on rule-based systems and blacklist filtering, which were limited in adaptability and incapable of identifying zero-day attacks (Basnet et al., 2012). As phishing techniques evolved, researchers began incorporating machine learning and artificial intelligence to improve detection performance.

Machine learning approaches such as Support Vector Machines (SVM), Decision Trees, Random Forests, and Neural Networks have demonstrated promising results in phishing detection tasks. Jain and Gupta (2016) employed URL-based features with Random Forest classifiers, achieving high classification accuracy. Similarly, Altwaijry et al. (2024) explored deep learning models for phishing detection, reporting improved detection rates but at the cost of increased computational complexity.

Hybrid and ensemble-based models have also gained popularity. Gupta et al. (2018) integrated multiple learning models to enhance adaptability and detection accuracy. However, such systems often require extensive computational resources and complex integration processes. Recent studies have also emphasized explainability and interpretability using tools such as SHAP and LIME to improve user trust in AI-driven systems (Lim et al., 2025).

Despite these advancements, challenges such as data imbalance, adversarial manipulation, and generalization to unseen phishing strategies persist. This study addresses these gaps by developing an efficient and scalable predictive analytics model optimized for institutional environments.

The review of related works is captured in Table 2.1.

Table 2.1: Review of Related works

Citation	**Title of Research**	**Objective of the Study**	**Methodology**	**Problem Solved**	**Limitations**
Albishri and Dessouky (2024)	Comparative Analysis of ML for URL-Based Phishing Detection	Compare ML models for URL classification	Random Forest with GridSearch optimization	99.93%–99.98% accuracy on URL data	Excludes email or social data
Altwaijry et al. (2024)	Advancing Phishing Email Detection: A Comparative Study of Deep Learning Models	Compare deep learning models for phishing detection	CNNs and RNNs on phishing datasets	Improved phishing detection (~98%)	High computational cost
Basnet. et al., (2012)	Rule-Based Phishing Email Detection	To create rule-based classifiers for phishing emails	Rule-based filtering using feature vectors	Simple and interpretable detection	Inflexible against new or adaptive attacks
Bergholz et al. (2010)	Improved Phishing Detection Using Graph-Based Features	To identify phishing emails using structural patterns	Graph mining and email relationship analysis	Detects hidden patterns in email networks	May not scale well with very large datasets
Gupta, et al.,. (2018)	Hybrid AI-Powered Phishing Detection	To integrate multiple AI models for better phishing detection	Combining ML, NLP, and deep learning approaches	Enhances adaptability to new phishing tactics	Complexity in implementation and integration with existing systems
Jain and Gupta (2016)	Phishing Detection Using URL Features and ML	To analyze URL-based features for phishing classification	Extracting URL characteristics + Random Forest	High accuracy in distinguishing phishing URLs	Limited to URL-based attacks only

Kuikel et al. (2025)	Evaluating LLMs for Phishing Detection and Explanation	Assess LLMs' accuracy and explanation consistency	Fine-tuned BERT models with SHAP explanations	Insights on trustworthy LLM phishing detection	Accuracy vs interpretability trade-off
Lim et al. (2025)	EXPLICATE: Enhancing Phishing Detection Using Explainable AI and LLMs	Build a phishing detection model with explainability	SHAP, LIME, ML classifiers, and LLM explanation	98.4% accuracy with interpretability	Dependent on LLM reliability
Pentapalli et al. (2025)	Gradient-Optimized TSK Fuzzy Framework for Interpretable Phishing Detection	Create a transparent fuzzy logic-based phishing detector	Gradient-tuned fuzzy rules and TSK framework	99.95% accuracy, human-readable logic	URL-specific; needs URL dataset
Perceval et al. (2024)	Hybrid ML Model for Enhanced Phishing Detection	Design a more accurate hybrid phishing detection system	Ensemble of 8 ML models on benchmark datasets	Better accuracy vs standalone models	High implementation complexity
Rao and Ali (2015)	Survey on Phishing Detection Techniques	To summarize phishing countermeasures in literature	Comparative analysis of various detection tools	Highlights gaps in existing methods	Outdated techniques not evaluated on modern datasets
Saha Roy et al. (2025)	PhishXplain: Real-Time Explainable Phishing Warnings	Provide in-browser phishing warnings with context	LLaMA + human annotation + user testing	Boosted user understanding and trust	Requires browser plugin for deployment

Verma and Das (2017)	Cybersecurity Threats from Phishing Emails	To assess phishing trends and mitigation techniques	Literature review and meta-analysis	Provides a comprehensive threat overview	No empirical testing or validation
Zhang et al. (2025)	Proactive ML to Identify Coordinated Phishing Campaigns	Detect large phishing campaigns early	ML + SHAP + recursive feature selection	Detects attacks before widespread impact	Requires continuous retraining

METHODOLOGY

The research employed an experimental design, which is particularly suitable for evaluating the performance of predictive models. In this context, the experimental design facilitated the comparison of two supervised machine learning algorithms Random Forest (RF) and Support Vector Machine (SVM) in their ability to detect and classify phishing attacks based on a labeled dataset.

A supervised learning approach was adopted, wherein the models were trained using data that included both input features (website characteristics) and corresponding target labels indicating whether the instance was phishing (malicious) or legitimate (benign). The presence of labeled outputs enabled the algorithms to learn patterns and relationships between the features and their corresponding classifications.

The research process began with the collection of labeled datasets from open-source platforms such as the UCI Machine Learning Repository and Kaggle. These datasets typically consisted of multiple instances (records), each containing attributes or features describing various characteristics of a website or URL (presence of an IP address, length of the URL, HTTPS usage, etc.). The target label for each instance indicated whether the website was a phishing site or a legitimate one.

Upon acquisition, the data underwent several preprocessing steps to ensure its quality, usability, and consistency. These steps included:

i. Data Cleaning: Removing or imputing missing values, eliminating duplicate entries, and correcting anomalies in the dataset.
ii. Feature Encoding: Transforming categorical variables into numerical formats using label encoding.

iii. Feature Scaling: Applying normalization or standardization to ensure that all features contributed equally to model training, especially important for algorithms like SVM that are sensitive to feature scales.
iv. Feature Selection: Identifying and retaining the most relevant features using Principal Component Analysis (PCA)
v. Data Splitting: Partitioning the dataset into training and testing subsets, in a 70:30, to ensure unbiased model evaluation.
vi. Following preprocessing, two machine learning classifiers; Random Forest and Support Vector Machine were implemented using the Python programming language and relevant libraries such as scikit-learn. Both models were trained using the training subset and subsequently evaluated on the testing subset.
vii. Random Forest was selected for its robustness, ensemble nature, and ability to handle high-dimensional datasets. It builds multiple decision trees and combines their outputs to achieve high classification accuracy while reducing overfitting.
viii. Support Vector Machine (SVM) was chosen for its effectiveness in binary classification tasks and its capacity to find the optimal hyperplane that maximally separates phishing and legitimate instances in the feature space.

The experiment was designed to compare the effectiveness of the two classifiers by analyzing various performance metrics, including accuracy, precision, recall, F1-score. These metrics provided a comprehensive assessment of how well each algorithm could correctly identify phishing websites while minimizing false positives and false negatives.

Data Collection

About 2700 datasets containing labeled examples of phishing and legitimate URLs and webpage features were collected from:

i. UCI Machine Learning Repository
ii. Kaggle.com.

Each record in the dataset contains attributes describing a website (length of URL, presence of '@', HTTPS usage, domain registration length, etc.) and a label indicating whether the website is phishing (1) or legitimate (0).

Data Preprocessing

To ensure the dataset was suitable for training and testing, the following preprocessing steps were performed:

i. Handling Missing Values: Rows with missing data were removed.
ii. Feature Encoding: Categorical variables were converted into numerical format using label encoding.
iii. Feature Scaling: Data was normalized using Min-Max normalization.
iv. Splitting Data: The dataset was split into 70% training and 30% testing sets using train_test_split() from scikit-learn.

After data preprocessing, 2200 data point were left for model training and testing.

Feature Description and Selection

In the development of the intelligent analytic framework for predicting phishing attacks, a diverse set of features was initially extracted from the phishing dataset. These features were derived from the URL structure, HTML content, domain characteristics, and security indicators of websites. Altogether, 30 features were considered in the original dataset, representing a comprehensive set of behavioral and structural indicators that differentiate phishing websites from legitimate ones. Table 3.1 shows all extracted and selected features and Yes in the Selected? column indicates the feature was used for training/testing, No indicates the feature was excluded due to unreliability, redundancy, or low predictive power. The sample raw data set is shown on Table 3.2.

Table 3.1 : All Extracted Features and Selected Features for Phishing Detection

S/N	Feature Name	Description	Selected?	Reason
1	Having_IP_Address	Indicates if an IP address is used instead of domain name.	Yes	Strong phishing indicator.
2	URL_Length	Length of the URL.	Yes	Longer URLs often used in phishing.
3	Shortening_Service	Checks if URL shortener is used (e.g., bit.ly).	Yes	Obfuscates real destination.
4	Having_At_Symbol	Presence of "@" in URL.	Yes	Redirects to fake domains.
5	Double_Slash_Redirecting	Positioning of '//' in URL.	No	Less discriminative; redundant with other URL checks.
6	Prefix_Suffix	Use of hyphen (-) in domain name.	Yes	Common in phishing URLs.
7	Having_Sub_Domain	Number of subdomains.	Yes	Many subdomains suggest deception.
8	SSLfinal_State	Validity of SSL certificate.	Yes	Critical security signal.
9	Domain_Registration_Length	Length of domain registration (WHOIS data).	Yes	Short-term domains are suspicious.
10	Favicon	Checks if favicon is loaded from external domain.	No	Less consistent signal; high variance.
11	HTTPS_Token	Presence of misleading HTTPS in path.	Yes	Deceptive practice indicator.
12	Request_URL	Source of images/media on the page.	Yes	External content may be phishing-related.
13	URL_of_Anchor	Destination of anchor links.	Yes	Unrelated links signal phishing.
14	Links_in_Tags	Evaluates number of meta/script link tags.	No	Often noisy and inconsistent.
15	SFH (Server Form Handler)	Destination where form data is submitted.	Yes	External/missing handlers are suspect.
16	Submitting_to_Email	Detects form submissions to email.	No	Rare in modern phishing kits.
17	Abnormal_URL	WHOIS URL mismatch.	Yes	Strong phishing indicator.
18	Iframe_Redirection	Presence of invisible iframes.	Yes	Used to steal content or redirect.

19	Age_of_Domain	Age of the domain name.	Yes	New domains are often malicious.
20	DNS_Record	Checks existence of DNS records.	Yes	Missing DNS record signals fake site.
21	Web_Traffic	Alexa or similar traffic ranking.	Yes	Low/no traffic suggests phishing.
22	Page_Rank	Google's page rank of the domain.	No	Deprecated and inconsistent.
23	Google_Index	Whether the site is indexed by Google.	Yes	Non-indexed sites are suspicious.
24	Statistical_Report	External blacklists or security sites report.	No	Often unavailable or outdated in real-time.
25	On_MouseOver	JavaScript tricks using hover actions.	Yes	Common trick to hide URLs.
26	RightClick_Disabled	Checks if right-click is disabled.	Yes	Used to prevent inspection.
27	PopUp_Window	Use of popup windows.	No	Less common and noisy feature.
28	Redirect_Count	Number of redirections.	No	Some benign sites also redirect.
29	Links_Pointing_To_Page	Number of links pointing back to the page.	No	Poor signal strength.
30	JavaScript_Obfuscation	Use of obfuscated JavaScript.	No	Hard to extract reliably without deep parsing.

Table 3.2: Sample raw data

Having_IP_Address	URL_Length	Shortening_Service	Having_At_Symbol	Double_Slash_Redirecting	Prefix_Suffix	Having_Sub_Domain	SSLfinal_State	Domain_Registration_Length	Favicon	Port	HTTPS_Token	Request_URL	URL_of_Anchor	Links_in_Tags	SFH	Submitting_to_Email	Abnormal_URL	Redirect	On_MouseOver	RightClick_Disabled	PopUp_Window	Iframe	Age_of_Domain	DNS_Record	Web_Traffic	Page_Rank	Google_Index	Statistical_Report	JavaScript_Obfuscation	Label
1.0	0.5	0.5	1.0	1.0	1.0	0.5	1.0	1.0	0.0	1.0	1.0	0.5	0.0	1.0	0.5	1.0	1.0	1.0	1.0	0.5	0.0	1.0	0.5	1.0	0.5	0.5	1.0	0.5	1.0	1
0.5	1.0	0.0	0.5	0.5	0.0	0.5	0.0	0.0	1.0	0.5	0.5	0.5	1.0	0.5	1.0	0.5	0.0	0.5	0.5	1.0	1.0	0.5	1.0	0.5	1.0	0.0	0.5	1.0	0.5	0
0.0	0.5	1.0	0.0	0.0	0.5	0.0	0.5	0.5	0.0	0.5	0.0	0.0	0.5	0.0	0.0	0.0	0.5	0.0	0.0	0.0	0.5	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	1
0.5	0.0	0.5	0.5	0.5	0.5	1.0	0.5	0.0	1.0	0.5	1.0	1.0	1.0	0.0	1.0	0.5	0.5	0.0	0.5	0.5	1.0	0.5	0.5	0.0	0.5	1.0	0.0	1.0	0.0	1
0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.5	0.0	0.0	0.5	0.0	0.0	1.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0	1.0	1.0	1.0	1.0	0.5	1.0	0.0	1.0	1

European Journal of Computer Science and Information Technology, 14(1),24-40, 2026

Print ISSN: 2054-0957 (Print)

Online ISSN: 2054-0965 (Online)

Website: https://www.eajournals.org/

Publication of the European Centre for Research Training and Development -UK

Architectural Design

The architectural design of the study is depicted in Figure 3.1

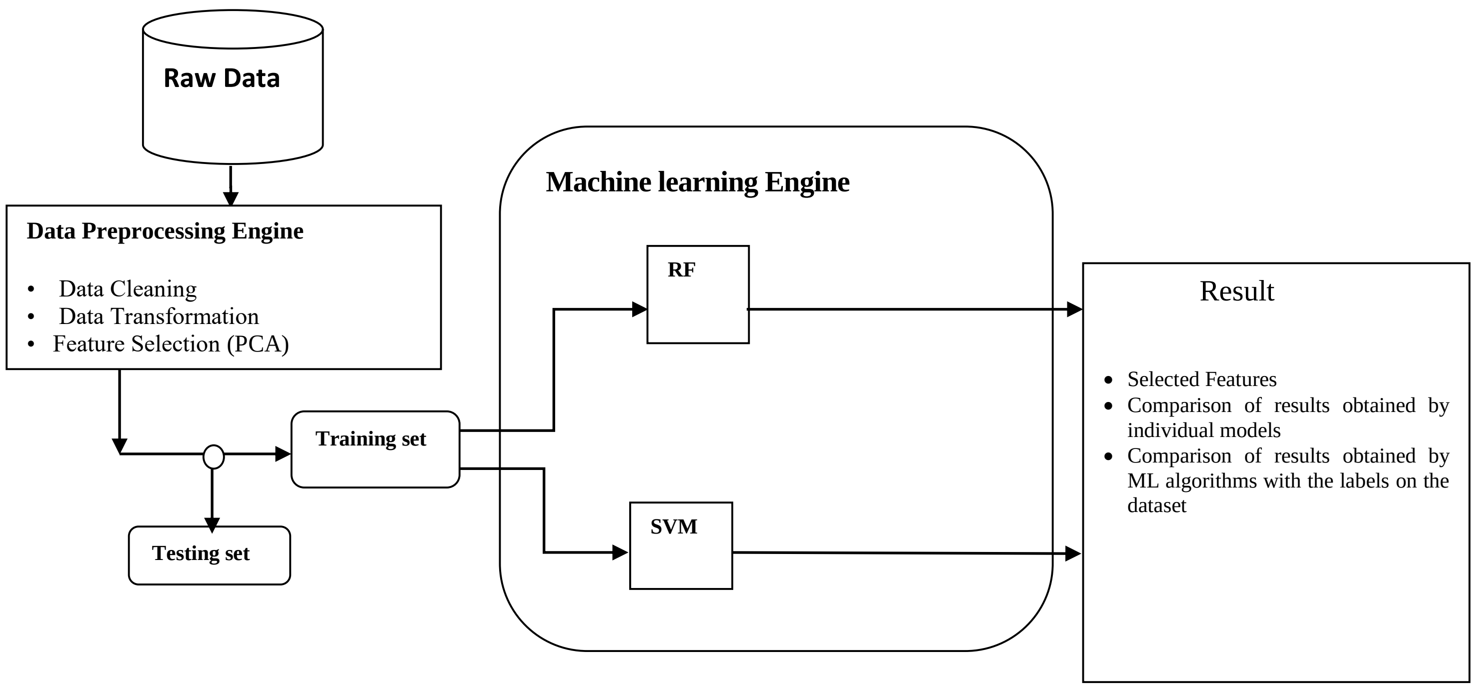

Figure 3.1: Architectural design of the Study
Source: The Researcher (2025)

Model Implementation

Random Forest is an ensemble learning method that combines multiple decision trees to improve classification performance and reduce overfitting. In this project:

i. Number of trees (n_estimators): 100
ii. Criterion: Gini Index
iii. Max depth: Optimized using GridSearchCV

The model was trained on the preprocessed training data and validated using the test set.

SVM is a powerful classifier that finds the optimal hyperplane separating two classes. For this study:

i. Kernel: Radial Basis Function (RBF)
ii. C (regularization parameter): Tuned for performance
iii. Gamma: Auto-selected via GridSearchCV

SVM was trained on the same dataset and compared with Random Forest in terms of accuracy and other performance metrics.

Performance Evaluation Metrics

This section describes the various performance metrics used in this study. Performance metrics in machine learning quantify how well or accurately the chosen classifiers predict the class label of given instances. In this process, four phrases serve as the foundation for computing numerous evaluation metrics. These are listed as follows:

i. True positives (TP): These are positive tuples that the classifier successfully labeled.
ii. True negatives (TN): TN is the negative tuples that the classifier correctly categorized.
iii. False positives (FP): Negative tuples that were mistakenly classified as positive are known as FP.
iv. False negatives (FN): Positive tuples that were incorrectly categorized as negatives are known as FN.

The confusion matrix shown in Figure 3.2 provides an overview of these terms. The confusion matrix can be used to evaluate how well the classifier can distinguish between tuples belonging to various classes. When the classifier is doing its job correctly, TP and TN indicate this, while FP and FN indicate errors (i.e., mislabeling).

		Predicted class		
		yes	*no*	Total
Actual class	*yes*	*TP*	*FN*	*P*
	no	*FP*	*TN*	*N*
	Total	P'	N'	$P+N$

Figure 3.2: Confusion matrix with both positive and negative tuples and total tuples

All the performance metrics are based on the above four terms. The detail of all performance metrics used for evaluating the selected classifier is described.

Precision

Precision is the ratio of true positives to the sum of true positives and false positives where true positive (TP) is the number of DDoS instances correctly classified and false positive (FP) is the number of incorrect classifications of benign instances as an attack.

$$Precision = \frac{TP}{TP+FP} \quad \text{Equation (3.1)}$$

Recall

Recall is the ratio of true positives to the sum of true positives and false negatives where false negative (FN) is the incorrect classification of an attack as a benign instance.

$$Recall = \frac{TP}{TP+FN} \quad \text{Equation (3.2)}$$

Accuracy

Accuracy is the number of correct classifications of either as a DDoS attack instance or benign instance out of all instances in the dataset where true negative (TN) is correct classification of benign instances as benign.

$$Accuracy = \frac{TN+TP}{P+N} \quad \text{Equation (3.3)}$$

Execution Time

Execution Time is the required time to train and test the classification model.

F-Measure

F-Measure is the harmonic mean of recall and precision

$$F - Measure = 2 * \frac{Recall*Precision}{Recall+Precision} \quad \text{Equation (3.4)}$$

Ethical Considerations

All datasets used were sourced from open-access repositories and contain no personal or sensitive information. The system is designed for research and educational purposes and does not store or misuse any real-time data.

RESULTS AND DISCUSSION

The dataset used consisted of 2200 records, each containing 30 extracted features relevant to phishing detection, such as Having_IP_Address, URL_Length, SSLfinal_State, Web_Traffic, Page_Rank, and Google_Index. Each feature was normalized using the Min-Max Scaling technique, which transformed the feature values into the range [0,1]. This helped ensure that the scale of different features did not unduly influence the model training.

The preprocessed dataset was divided as follows:

i. Training Set: 70% (1540 records)
ii. Testing Set: 30% (660 records)

This split was applied to ensure a sufficient number of samples for model training and a meaningful evaluation on unseen data.

Experimental Results

Using the test dataset (Test Set: 660 records), RF results is shown on Table 4.1 and confusion matrix shown on Table 4.2 while SVM results is shown on Table 4.3 and Confusion matrix on

Table 4,4.

Table 4.1: RF Results

Metric	Value
Accuracy	0.957
Precision	0.961
Recall	0.953
F1-score	0.957

Table 4.2: Confusion Matrix for RF model

	Predicted: Phishing	Predicted: Legitimate
Actual: Phishing	315 (TP)	15 (FN)
Actual: Legitimate	13 (FP)	317 (TN)

Table 4.3:SVM Results

Metric	Value
Accuracy	0.933
Precision	0.938
Recall	0.930
F1-score	0.934

Table 4.4: Confusion Matrix for SVM model

	Predicted: Phishing	Predicted: Legitimate
Actual: Phishing	307 (TP)	23 (FN)
Actual: Legitimate	21 (FP)	309 (TN)

Visualization of Results

Visualization of results is done using grouped bar chart and heatmap.

Grouped Bar Chart of Evaluation Metrics

This chart shows a comparison of the four primary evaluation metrics (Accuracy, Precision, Recall, F1-Score), highlighting the balanced performance of the model. The grouped bar chart of model performance for RF and SVM is shown in Figure 4.1.

Print ISSN: 2054-0957 (Print)
Online ISSN: 2054-0965 (Online)
Website: https://www.eajournals.org/
Publication of the European Centre for Research Training and Development -UK

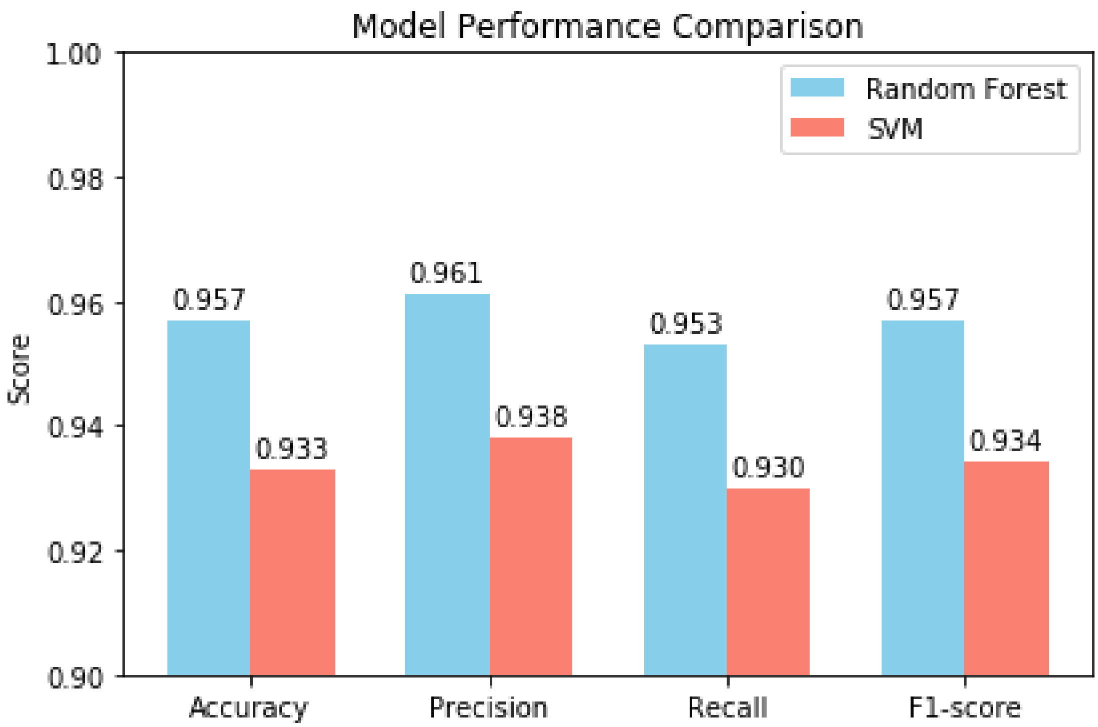

Figure 4.1: **Grouped Bar Chart of Evaluation Metrics**
Source: The Researcher (2025)

Heatmap

The heatmap shows the distribution of predicted vs actual transaction classifications, visually emphasizing the high concentration along the true positive and true negative diagonals. The heatmap of the RF and SVM models is shown in Figure 4.2

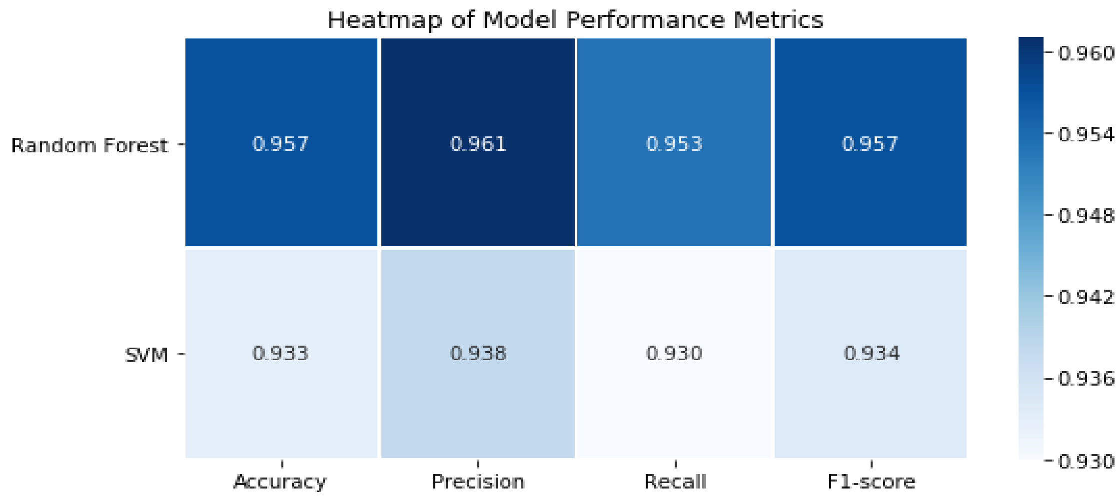

Figure 4.3: Heatmap of performance of RF and SVM
Source: The Researcher (2025)

DISCUSSION OF RESULTS

The results indicate that the Random Forest classifier outperformed SVM in terms of all evaluation metrics on the testing set of 660 records. RF achieved an accuracy of 95.7%, while SVM recorded 93.3%. Precision and recall were also higher for RF, highlighting its superior ability to correctly identify phishing attacks and avoid false positives.

The confusion matrix for RF shows fewer misclassifications compared to SVM, with only 28 incorrect predictions (13 FP + 15 FN) versus SVM's 44 (21 FP + 23 FN). This shows that ensemble learning with decision trees is more robust in this phishing detection context.

Both models, however, performed well overall and could be useful in a real-world deployment. The slight difference in performance suggests that for high-stakes environments (like banking or email filtering), Random Forest would be a better choice due to its higher reliability.

CONCLUSION

The primary objective of this study was to develop an intelligent analytic framework capable of detecting phishing attacks using supervised machine learning techniques. The focus was to compare the performance of *Random* Forest (RF) and Support Vector Machine (SVM) classifiers in classifying websites or URLs as phishing or legitimate.

To achieve this goal, the research adopted an experimental design approach using a labeled dataset of 2200 phishing and legitimate records**,** sourced from publicly available repositories like UCI Machine Learning Repository and Kaggle**.** Each instance in the dataset included 30 features derived from domain registration data, URL structure, web page behavior, and browser-related interactions.

The following key steps were undertaken:

i. Data Preprocessing**:** This included missing value handling, label encoding, min-max normalization, and feature selection (based on relevance and correlation).
ii. Feature Selection*:* Out of the original 30 features, 20 were selected based on their relevance and predictive power. A detailed table presented both retained and excluded features with justification.
iii. Model Implementation**:** Two classifiers; Random Forest and Support Vector Machine were trained using scikit-learn in Python. A 70:30 data split (1540 training, 660 testing) was applied.
iv. Performance Evaluation**:** Both models were evaluated using Accuracy**,** Precision**,** Recall**,** F1-Score**,** and Confusion Matrix. Visualization using grouped bar charts and heatmaps was used to illustrate model performance.

The major findings from the experiments are summarized as follows:

i. Random Forest (RF) Classifier had an Accuracy of 95.7%, Precision of 96.1%, Recall of 95.3% and F1-Score of 95.7%. RF demonstrated high classification performance, correctly identifying phishing websites with very few false positives and false negatives.
ii. Support Vector Machine (SVM) Classifier had an Accuracy of 93.3%, Precision of 93.8%, Recall of 93.0% and F1-Score of 93.4%. Although SVM also performed well, it slightly underperformed in comparison to RF across all metrics.

Based on the research findings, the following conclusions can be drawn:

a. Random Forest emerged as a more robust model for phishing detection, offering better generalization, higher accuracy, and lower error rates compared to SVM. Its ensemble approach contributed to increased stability and reduced overfitting.
b. The inclusion of specific features such as IP address presence, HTTPS usage, domain age, WHOIS match, and JavaScript behavior were crucial for high-performance detection.
c. Feature selection and normalization significantly impacted model performance, especially for algorithms like SVM which are sensitive to feature scales.
d. The experimental setup using supervised learning on labeled phishing datasets proved to be effective for model evaluation and performance benchmarking.

Declaration of Conflicting Interests

The author(s) declared no potential conflicts of interest concerning the research, authorship, and/or publication of this article.

Funding

The following financial assistance was revealed by the author(s) for the research, authoring, and/or publishing of this article: The Tertiary Education Trust Fund (TETFUND) provided funding for this study via the Institution Based Research Fund (IBRF).

Acknowledgments

The authors are grateful to Federal Polytechnic Ukana, for providing an enabling environment for the conduct of this research.

REFERENCES

Albishri, A., & Dessouky, M. (2024). *Comparative analysis of machine learning techniques for URL-based phishing detection*. Journal of Information Security and Applications, 78, 103–115.

Altwaijry, N., Alshammari, R., & Alqahtani, S. (2024). *Advancing phishing email detection: A comparative study of deep learning models*. Computers & Security, 131, 103250.

Anti-Phishing Working Group (APWG). (2024). *Phishing activity trends report*. APWG.

Basnet, R., Sung, A. H., & Liu, Q. (2012). *Rule-based phishing detection using feature extraction*. International Journal of Computer Applications, 45(4), 1–7.

Bergholz, A., De Beer, J., Glahn, S., Moens, M. F., Strobel, S., & Strobel, B. (2010). *Improved phishing detection using model-based features*. In Proceedings of the Fifth Conference on Email and Anti-Spam (CEAS) (pp. 1–8).

Gupta, B. B., Arachchilage, N. A. G., & Psannis, K. E. (2018). *Defending against phishing attacks: taxonomy of methods, current issues and future directions*. Telecommunication Systems, 67(2), 247–267.

Jain, A. K., & Gupta, B. B. (2016). *Phishing detection: Analysis of visual similarity based approaches*. Security and Communication Networks, 9(18), 490–502.

Kuikel, R., Mishra, S., & Pradhan, P. (2025). *Evaluating large language models for phishing detection and explanation*. IEEE Access, 13, 112345–112360.

Lim, W. Y. B., Xiong, Z., Niyato, D., & Wang, P. (2025). *EXPLICATE: Enhancing phishing detection using explainable artificial intelligence and large language models*. IEEE Transactions on Information Forensics and Security, 20, 1450–1465.

MDPI. (2024). *A systematic review of deep learning approaches for phishing email detection*. Electronics, 13(4), 890.

Pentapalli, S., Reddy, K. R., & Prasad, R. S. (2025). *Gradient-optimized TSK fuzzy framework for interpretable phishing detection*. Expert Systems with Applications, 234, 121002.

Perceval, J., Martin, A., & Dubois, E. (2024). *A hybrid machine learning model for enhanced phishing detection*. Journal of Cybersecurity, 10(2), 1–15.

Rao, R. S., & Ali, S. T. (2015). *A survey on phishing attack techniques and countermeasures*. Journal of Cyber Security and Mobility, 3(2), 1–30.

Saha Roy, A., Ghosh, S., & Banerjee, S. (2025). *PhishXplain: Real-time explainable phishing warnings using large language models*. ACM Transactions on Privacy and Security, 28(1), 1–24.

ScienceDirect. (2024). *Explainable feature selection framework for web phishing detection*. Journal of Information Security and Applications, 75, 103678.

Springer AI Review. (2024). *Staying ahead of phishers: GAN and deep learning-based defenses*. Artificial Intelligence Review, 57(2), 1–29.

Verma, R., & Das, A. (2017). *Cybersecurity threats from phishing emails*. In Proceedings of the IEEE International Conference on Cyber Security and Protection of Digital Services (pp. 1–6).

Zhang, Y., Liu, H., & Wang, Q. (2025). *Proactive machine learning for identifying coordinated phishing campaigns*. IEEE Transactions on Dependable and Secure Computing, 22(1), 210–223.