International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

AI-Driven Phishing Detection And Awareness System

Prof. J. N. Kale[1], Aniket Rahane[2], Devang Kolhe[3], Pratibha Pagar[4], Divyani Nale[5]

[1] Computer Engineering Department Sanjivani College of Engineering, Kopargaon Affiliated to: SPPU, Pune Kopargaon, India kalejaydeepcomp@sanjivani.org.in
[2] Computer Engineering Department Sanjivani College of Engineering, Kopargaon Affiliated to: SPPU, Pune Kopargaon, India aniketrahanecomp@sanjivani.org.in
[3] Computer Engineering Department Sanjivani College of Engineering, Kopargaon Affiliated to: SPPU, Pune Kopargaon, India devangkolhecomp@sanjivani.org.in
[4] Computer Engineering Department Sanjivani College of Engineering, Kopargaon Affiliated to: SPPU, Pune Kopargaon, India pratibhapagarcomp@sanjivanicoe.org.in
[5] Computer Engineering Department Sanjivani College of Engineering, Kopargaon Affiliated to: SPPU, Pune Kopargaon, India divyaninalecomp@sanjivani.org.in

ABSTRACT—

Phishing continues to be a major threat to cyberse- curity by tricking consumers into divulging sensitive information via fraudulent websites and erroneous links. Without smart real- time detection systems, people and businesses are vulnerable to financial losses, privacy breaches, and reputational harm. Given the quick changing phishing strategies employed, current security measures sometimes have trouble keeping up, highlighting the need for adaptive, machine learning-based detection systems.

Designed to run over current web browsers, PhishPatrol is introduced here as a thorough phishing detection and awareness solution. The system comprises a real-time Chrome extension for URL analysis, a Flask-based backend for processing, and an educational platform that raises user awareness by way of interactive content like videos and quizzes.

For feature reduction and better model performance, Phish Pa- trol uses several machine learning classifiers—K-Nearest Neigh- bors (KNN), Support Vector Machine (SVM), Random For- est, Decision Tree, and Naive Bayes—enhanced via Principal Component Analysis (PCA). To guarantee constant and precise classification results, a majority voting mechanism is used. From URLs, the backend engine pulls over 70 distinct traits, performs PCA conversion, and categorizes them as either phishing or legitimate.

Along with automated detection, the system encourages secu- rity awareness via user-friendly alerts and educational materials. Results of evaluation reveal that SVM attained the best accuracy rate of 93.79%, thereby demonstrating its usefulness in iden- tifying phishing attacks. PhishPatrol shows great promise as a scalable and practical solution for reducing phishing attacks in real time by combining detection accuracy with user education.

Index Terms—Phishing detection, AI-powered security, Ma- chine learning, PhishPatrol, Real-time detection, Majority voting, K-Nearest Neighbors (KNN), Support Vector Machine (SVM)

Introduction :

Central in modern life, the internet allows for communi- cation, banking, shopping, and access to many of services. Although this connection has provided many advantages, it has also presented major cyber security hazards. Among these, phishing is among the most widespread and destructive danger. Phishing tricks people into divulging sensitive infor- mation—like financial data or passwords—by pretending to be trusted entities. Because the phony websites or emails strongly resemble genuine ones, these attacks are often hard to identify, therefore exposing even cautious users.

Attackers have used sophisticated methods including do- main spoofing, URL obfuscation, and fraudulent SSL certifi- cates to make hostile websites seem legitimate as phishing tactics keep developing. Many phishing websites imitate the appearance of legitimate banking, e-commerce, and social media networks. This rising sophistication questions conventional security measures such blacklists or spam filters, which depend on previous knowledge of threats and cannot match the pace of freshly created scams. Reactive security measures sometimes fall to stop real-time assaults since phishing sites are often short-lived and fast evolving.

The results of phishing might be terrible. Financial losses, identification theft, or unauthorized access to essential ac- counts may be experienced by people. For businesses, one phishing email might cause data breaches, malware infections, or access to secret internal systems. Phishing was the entrance in many well-known cyberattacks. Beyond monetary losses, these violations could undermine public trust and damage a firm's reputation, with long-lasting effects.

Researchers have more and more turned to artificial intel- ligence (AI) and machine learning (ML) to more precisely identify phishing attempts in order to solve these problems. These technologies find malicious patterns that conventional techniques could overlook by means of URL, web content, and domain-level properties analysis. Learning from recent information, machine learning models may identify little in- dications of phishing and can change with new threats. This proactive approach lets systems find phishing websites before they are extensively blacklisted or reported.

Still, AI-based phishing detection has some drawbacks as well. Many models need big, varied datasets for training, and some continue to have difficulty identifying complex or new phishing techniques. Others may get false positives by incorrectly classifying genuine websites as hazardous. Further- more depending not just on technical weaknesses but also on human mistakes, phishing Many users don't understand how phishing works and miss clues of fraud. Therefore, for a more successful defense, user knowledge must be combined with artificial intelligence-based detection.

In response, we created PhishPatrol, a full phishing detec- tion and awareness system that considers both technological and human aspects. Real-time protection is available via a browser extension, strong backend processing utilizing ma- chine learning, and an education platform to educate users in identifying phishing attempts. While users browse the internet, the browser extension searches URLs and page content to deliver instant notifications on a questionable link.

Constructed in Flask, the backend pulls more than 70 features from every URL, encompassing components like structure, content, and domain traits. Five machine learning algorithms—Support Vector Machine (SVM), Random For- est, K-Nearest Neighbors (KNN), Decision Tree, and Na¨ıve Bayes—analyze the input. Their predictions are combined using a majority voting mechanism, which improves the re- liability and consistency of classification results. These char- acteristics are reduced using Principal Component Analysis (PCA) to enhance processing speed.

PhishPatrol encourages security awareness in addition to automatic detection. To assist users grasp typical phishing methods, the educational site provides simulation videos, quizzes, and real-world examples. This element helps con- sumers become better at detecting threats on their own, therefore promoting better security practices over time. Tech- nical detection combined with human training considerably improves the general efficacy of the system.

Our study found PhishPatrol to be very accurate in identi- fying phishing sites. With an accuracy rate of 93.79

Unlike traditional systems based just on blacklists or fixed rules, PhishPatrol uses adaptive machine learning to stay ahead of developing threats. Simultaneously, it gives consumers the knowledge to spot and avoid phishing attacks. This dual strategy— combining advanced detection and proactive educa- tion—provides a potent, scalable approach to lower phishing risks and improve digital security.

Objectives

- To improve cybersecurity by immediately identifying and warning users of phishing risks.
- To lessen the possibility of phishing attack-related finan- cial losses and data breaches.
- To encourage safer online conduct and raise user aware- ness.
- To gradually increase detection accuracy by utilizing AI and machine learning approaches.

Scope

- The system will immediately alert users when it detects a phishing link.
- Features a simple, intuitive interface that is intended to improve overall usability and encourage user involve- ment.
- Works with a variety of browsers.
- The website will offer simulations, advice on safe online conduct, and teaching materials.

Challenges

- Data Availability and Quality: It can be challenging and time-consuming to continuously collect and update pertinent phishing data for training models. Maintaining a current and varied dataset is essential since phishing assaults change quickly [1].
- High False Positive Rate: In order to keep trustworthy websites from being mistakenly identified as phishing sites, it is crucial to reduce false positives. User confi- dence in the system may be weakened by a high false positive rate [6].
- Complexity of Phishing Attacks: Phishing tactics are al- ways evolving, and attackers employ advanced strategies such as URL modification, domain spoofing, and phony security certificates. These constantly evolving strategies must be accommodated by the system [7].
- Model Training and Performance: It's crucial to guarantee steady model performance using varied, often updated training data. To maintain high detection accuracy, the system must steer clear of problems like biases in the training data or overfitting or underfitting [5].
- Security Issues: An attacker could target the system itself. The system as a whole may be compromised via flaws in the backend or the Chrome extension (such as the Flask API). Platform security is crucial [12].
- Response Time: It's critical to provide prompt detection and real-time notifications. Users may fall for frauds if phishing efforts are not detected in a timely manner. Without any discernible lag, the system must process URLs and deliver notifications immediately [3].

Proposed System

The AI-Driven Phishing Detection and Awareness System exists to make users aware of their cybersecurity risks and to proactively detect phishing attempts. Using a complex machine learning model, the system evaluates URLs based on several criteria, including domain type, HTTPS elements, special character usage, and numerical categorization [1].

To maintain accuracy and adaptability to evolving cy- berthreats, the model is continually updated with the most re- cent phishing data [5]. It provides real-time detection through

seamless browser extension integration, with visual alerts that notify users of potential threats—red for phishing and green for safe URLs [7].

The system also includes an interactive awareness com- ponent designed to reinforce safe browsing behavior. This module enhances user understanding of phishing risks through knowledge-assessment questionnaires and video-based educa- tional content [12].

A structured data pipeline underpins the system, beginning with reliable data collection of up-to-date phishing informa- tion. Preprocessing methods are applied to handle missing values, detect outliers, and perform feature scaling. Addi- tionally, Principal Component Analysis (PCA) is used for dimensionality reduction, improving efficiency and reliability [6].

Overall, the system offers a comprehensive cybersecurity solution that not only identifies phishing threats but also empowers users to recognize and avoid such threats indepen- dently by combining automated detection with user awareness [3].

Methodology

A. Data Collection

- A Kaggle dataset is employed with URLs labeled as phishing and legitimate.
- Key attributes include:
  - URL and domain name,
  - HTTPS presence,
  - URL length,
  - Special characters,
  - Numerical digits,
  - Use of an IP address [1],[3].

B. Data Preprocessing

- Missing values are managed using imputation techniques.
- Outliers are handled using the IQR method and Z-score method.
- Features are standardized or scaled to ensure balanced input for the model.
- Dimensionality reduction (PCA) is applied to enhance efficiency.

C. Feature Analysis

- Key features are analyzed, including:
  - HTTPS presence,
  - URL length,
  - Suspicious keywords,
  - Domain name structure.
- The most relevant attributes are selected for phishing detection [3].

D. Model Training

- The dataset is split into training and testing sets.
- The following machine learning algorithms are imple- mented:
  - K-Nearest Neighbors (KNN),
  - Support Vector Machine (SVM),

– Random Forest,

– Decision Tree,

– Na¨ıve Bayes.

- Performance is evaluated based on standard classification metrics [7],[11].

E. Model Testing and Performance Evaluation

Metrics such as Accuracy, Precision, Recall, and F1-score are used to test the trained models and determine how well they detect phishing websites.

F. Real-time Detection in Chrome Extension

An extension for the browser is created to analyze URLs in real time. The model categorizes URLs and warns users about phishing threats. It receives updates from the backend to improve detection accuracy.

G. User Awareness and Alerts

Users are alerted when a phishing website is identified, and the extension offers instructional messages to raise awareness of phishing.

System Architecture

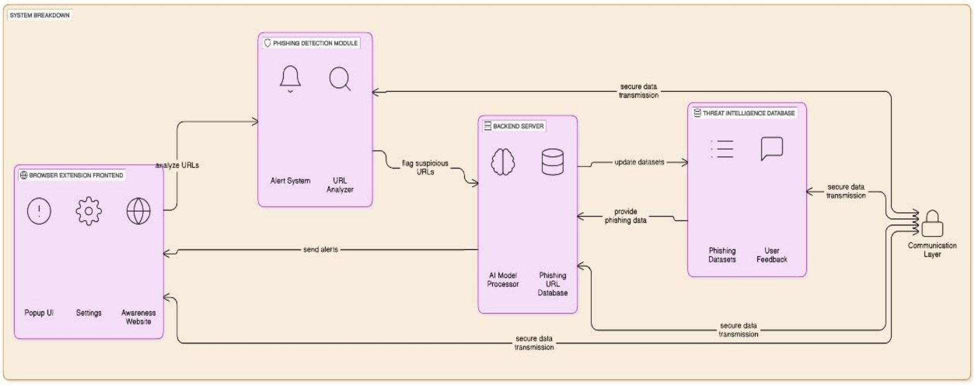


Fig. 1. PhishPatrol System Architecture Diagram

System Architecture

There are three primary parts to the PhishPatrol system architecture:

Client-Side Extension:

- Browser extension for monitoring URLs in real time
- User interface for alerts and instructional materials
- Local caching to improve performance [3]

Backend Service:

- Flask API for handling requests for detection
- A machine learning model that serves
- Database interaction layer
- Model update mechanism

Data Processing Pipeline:

- Module for feature extraction
- Training and assessment of the model
- Method for gathering feedback
- Secure ways of communication [12]

Figure 1 illustrates the architecture's modular design and encrypted communication between all parts. Throughout the detection phase, the system ensures data confidentiality and privacy while maintaining high availability.

Details of Implementation

Overview of the System Architecture

Figure 2 shows the three main components of the distributed architecture used by the PhishPatrol system:

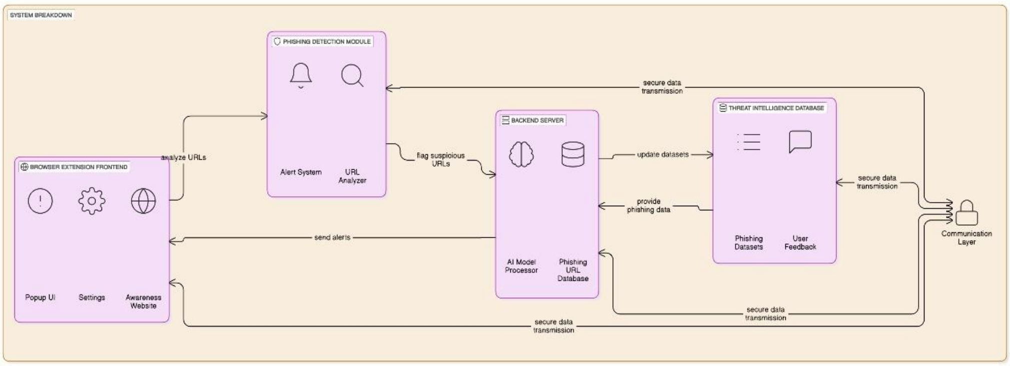


Fig. 2. Architecture of the PhishPatrol system illustrating data flow between components

Implementation of the Backend

Pipeline for Machine Learning: The following process- ing procedure is implemented by the detecting back-end:

Extraction of Features:

- 72 features, comprising lexical information (URL length, special characters), domain features (WHOIS data, TTL), and content features (HTML/JS characteris- tics), were extracted after URL parsing using Python's `urllib.parse` [9].

Training Models:

Ensemble model implementation with Scikit-learn Adjusting hyperparameters with GridSearchCV Five-fold cross-checking [6]

TABLE I

MODEL CONFIGURATION PARAMETERS

Model	Key Parameters
KNN	n neighbors=5, weights='distance'
SVM	C=10, kernel='rbf'
Random Forest	max depth=20, n estimators=150 Decision Tree max depth=10, min samples split=2
Na¨ıve Bayes	default parameters

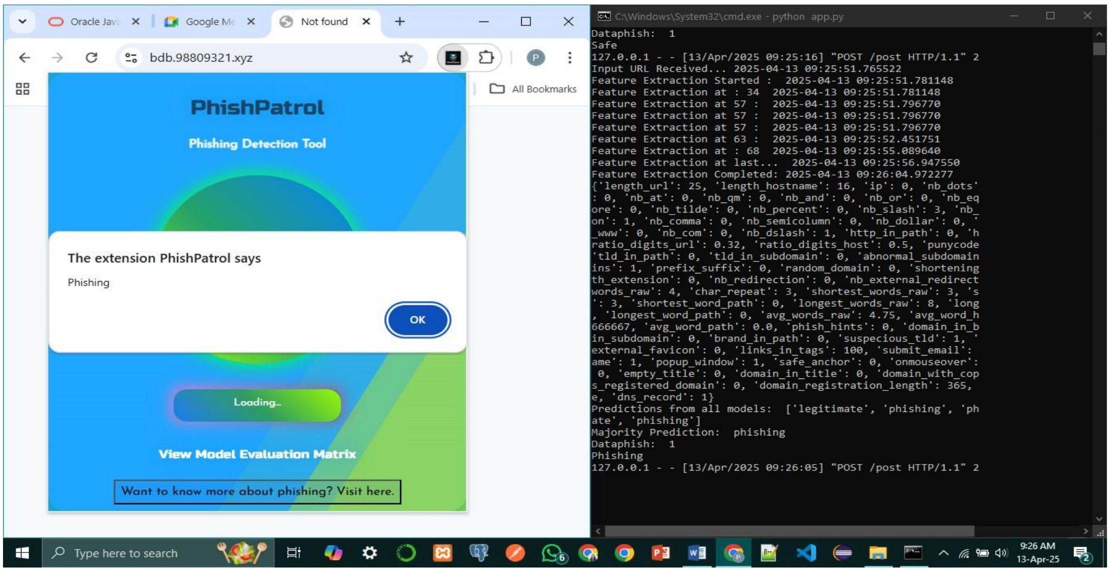
1) *API Service:*

- Three endpoints for the Flask REST API:
  - `/api/v1/model-update` (GET)
  - `/api/v1/feedback` (PUT)
  - `/api/v1/check-url` (POST)
- JWT authentication with 300s token expiration
- Redis caching for regular URL inspections

B. *Front-end Execution*

1) *Chrome Add-on:*

- Implementation of Manifest V3
- Content scripts for DOM monitoring
- Background service worker for API connectivity
- Storage synchronisation for user preferences

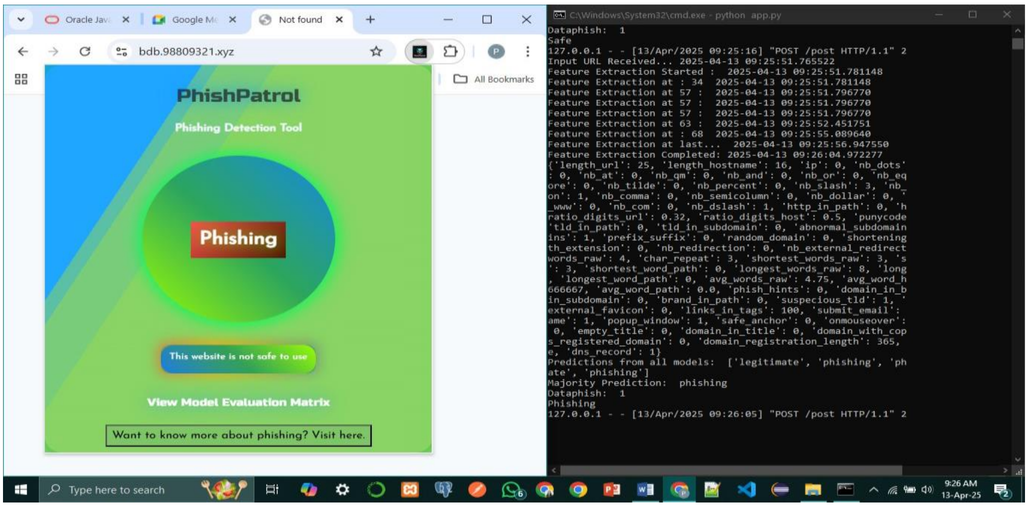


The extension PhishPatrol says

Phishing

```
Dataphish:  1
Safe
127.0.0.1 - - [13/Apr/2025 09:25:16] "POST /post HTTP/1.1" 2
Input URL Received... 2025-04-13 09:25:51.765522
Feature Extraction Started :  2025-04-13 09:25:51.781148
Feature Extraction at : 34  2025-04-13 09:25:51.781148
Feature Extraction at 57 :  2025-04-13 09:25:51.796770
Feature Extraction at 57 :  2025-04-13 09:25:51.796770
Feature Extraction at 57 :  2025-04-13 09:25:51.796770
Feature Extraction at 63 :  2025-04-13 09:25:52.451751
Feature Extraction at : 68  2025-04-13 09:25:55.089640
Feature Extraction at last...  2025-04-13 09:25:56.947550
Feature Extraction Completed: 2025-04-13 09:26:04.972277
{'length_url': 25, 'length_hostname': 16, 'ip': 0, 'nb_dots'
: 0, 'nb_at': 0, 'nb_qm': 0, 'nb_and': 0, 'nb_or': 0, 'nb_eq
ore': 0, 'nb_tilde': 0, 'nb_percent': 0, 'nb_slash': 3, 'nb_
on': 1, 'nb_comma': 0, 'nb_semicolumn': 0, 'nb_dollar': 0, '
_www': 0, 'nb_com': 0, 'nb_dslash': 1, 'http_in_path': 0, 'h
ratio_digits_url': 0.32, 'ratio_digits_host': 0.5, 'punycode
'tld_in_path': 0, 'tld_in_subdomain': 0, 'abnormal_subdomain
ins': 1, 'prefix_suffix': 0, 'random_domain': 0, 'shortening
th_extension': 0, 'nb_redirection': 0, 'nb_external_redirect
words_raw': 4, 'char_repeat': 3, 'shortest_words_raw': 3, 's
': 3, 'shortest_word_path': 0, 'longest_words_raw': 8, 'long
, 'longest_word_path': 0, 'avg_words_raw': 4.75, 'avg_word_h
666667, 'avg_word_path': 0.0, 'phish_hints': 0, 'domain_in_b
in_subdomain': 0, 'brand_in_path': 0, 'suspecious_tld': 1, '
external_favicon': 0, 'links_in_tags': 100, 'submit_email':
ame': 1, 'popup_window': 1, 'safe_anchor': 0, 'onmouseover':
 0, 'empty_title': 0, 'domain_in_title': 0, 'domain_with_cop
s_registered_domain': 0, 'domain_registration_length': 365,
e, 'dns_record': 1}
Predictions from all models:  ['legitimate', 'phishing', 'ph
ate', 'phishing']
Majority Prediction:  phishing
Dataphish:  1
Phishing
127.0.0.1 - - [13/Apr/2025 09:26:05] "POST /post HTTP/1.1" 2
```

Fig. 3. PhishPatrol browser extension Warning interface


Oracle Java
Google M
Not found
bdb.98809321.xyz
All Bookmarks
PhishPatrol
Phishing Detection Tool
Phishing
This website is not safe to use
View Model Evaluation Matrix
Want to know more about phishing? Visit here.
C:\Windows\System32\cmd.exe - python app.py
Type here to search
9:26 AM
13-Apr-25

```
Dataphish:  1
Safe
127.0.0.1 - - [13/Apr/2025 09:25:16] "POST /post HTTP/1.1" 2
Input URL Received... 2025-04-13 09:25:51.765522
Feature Extraction Started :  2025-04-13 09:25:51.781148
Feature Extraction at : 34  2025-04-13 09:25:51.781148
Feature Extraction at 57 :  2025-04-13 09:25:51.796770
Feature Extraction at 57 :  2025-04-13 09:25:51.796770
Feature Extraction at 57 :  2025-04-13 09:25:51.796770
Feature Extraction at 63 :  2025-04-13 09:25:52.451751
Feature Extraction at : 68  2025-04-13 09:25:55.089640
Feature Extraction at last...  2025-04-13 09:25:56.947550
Feature Extraction Completed: 2025-04-13 09:26:04.972277
{'length_url': 25, 'length_hostname': 16, 'ip': 0, 'nb_dots'
: 0, 'nb_at': 0, 'nb_qm': 0, 'nb_and': 0, 'nb_or': 0, 'nb_eq
ore': 0, 'nb_tilde': 0, 'nb_percent': 0, 'nb_slash': 3, 'nb_
on': 1, 'nb_comma': 0, 'nb_semicolumn': 0, 'nb_dollar': 0, '
_www': 0, 'nb_com': 0, 'nb_dslash': 1, 'http_in_path': 0, 'h
ratio_digits_url': 0.32, 'ratio_digits_host': 0.5, 'punycode
'tld_in_path': 0, 'tld_in_subdomain': 0, 'abnormal_subdomain
ins': 1, 'prefix_suffix': 0, 'random_domain': 0, 'shortening
th_extension': 0, 'nb_redirection': 0, 'nb_external_redirect
words_raw': 4, 'char_repeat': 3, 'shortest_words_raw': 3, 's
': 3, 'shortest_word_path': 0, 'longest_words_raw': 8, 'long
, 'longest_word_path': 0, 'avg_words_raw': 4.75, 'avg_word_h
666667, 'avg_word_path': 0.0, 'phish_hints': 0, 'domain_in_b
in_subdomain': 0, 'brand_in_path': 0, 'suspecious_tld': 1, '
external_favicon': 0, 'links_in_tags': 100, 'submit_email':
ame': 1, 'popup_window': 1, 'safe_anchor': 0, 'onmouseover':
 0, 'empty_title': 0, 'domain_in_title': 0, 'domain_with_cop
s_registered_domain': 0, 'domain_registration_length': 365,
e, 'dns_record': 1}
Predictions from all models:  ['legitimate', 'phishing', 'ph
ate', 'phishing']
Majority Prediction:  phishing
Dataphish:  1
Phishing
127.0.0.1 - - [13/Apr/2025 09:26:05] "POST /post HTTP/1.1" 2
```

Fig. 4. PhishPatrol Browser Extension Phishing Result

2) *Awareness Portal:*

- React SPA with the following:
  - React Router for navigating
  - Axios for communication via API
  - Chart.js for visualization
- Authentication with Firebase
- MongoDB for recording user progress

C. Implementation of Data Processing

The following crucial steps are implemented by the prepro- cessing pipeline:

1. **Data Purification:**
   - Normalising URLs
   - Managing missing values
   - Identifying outliers
2. **Engineering Features:**
   - PCA to reduce dimensionality (n=28)

D. Enhancement of Performance

Important optimisation strategies used:

- Feature Selection
- Feature Engineering
- PCA – Principal Component Analysis [12], [14]

As shown in Section VII, the full implementation achieves an accuracy of 93.79%.

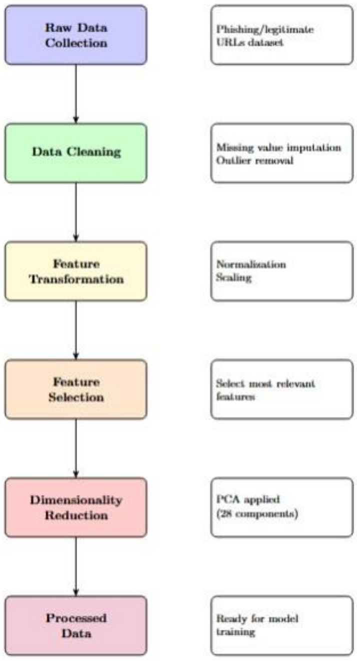


Figure 1: Vertical data preprocessing pipeline for phishing detection system

Fig. 5. Data preprocessing pipeline architecture

Analysis of the Results

The outcomes of the phishing detection model algorithms are thoroughly examined in this section. Accuracy, precision, recall, and F1-score are used to evaluate different machine learning classifiers. The results are backed up by tables and figures that clearly compare the performance of the models.

A. Measures of Performance

The following measures were employed to assess the clas- sifiers' effectiveness:

- **Accuracy**: The proportion of authentic and phishing websites that are accurately classified.
- **Precision**: The percentage of phishing sites that were accurately detected out of all those that were anticipated.
- **Recall**: The model's capacity to identify phishing web- sites with accuracy.
- **F1-Score**: Recall and precision harmonic mean

B. Comparison of Classifier Performance

The project's categorisation models are assessed using im- portant performance indicators. These measurements gave a clear picture of how well each model distinguishes between reputable and fraudulent websites. For phishing detection, a more dependable model is indicated by balanced precision recall values and better accuracy [**?**].

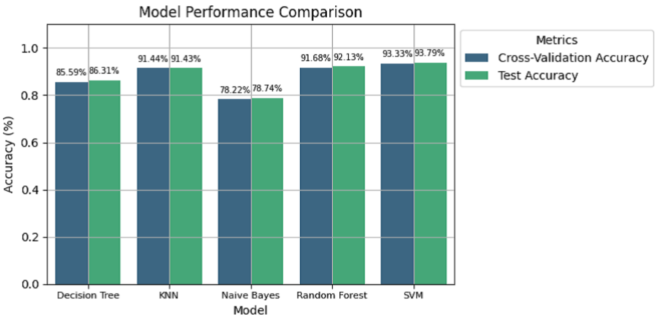
Table II presents a summary and analysis of the performance of several classifiers. Figures 6 and 7 provide the graph representation of the performance.

TABLE II

PERFORMANCE COMPARISON OF CLASSIFICATION MODELS (%)

Model	Accuracy	Precision	Recall	F1-Score
KNN	91.43	91.50	91.50	91.50
SVM	93.79	94.00	93.50	94.00
Random Forest	92.13	92.50	92.00	92.00
Decision Tree	86.31	86.50	86.00	86.50
Naive Bayes	78.74	79.00	78.50	79.00

Fig. 6. Comparative performance of classification models



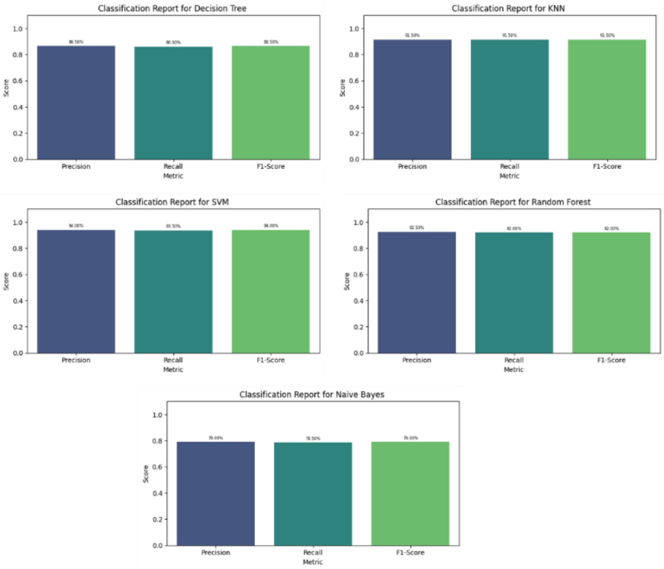


Fig. 7. Performance of classification models

C. Confusion Matrix Analysis

Confusion matrices offer a deeper understanding of each classifier's performance by examining the distribution of var- ious parameters such as true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN) (Figure 8). A lower false-positive rate is essential in phishing detection, as misclassifying legitimate sites as phishing could negatively

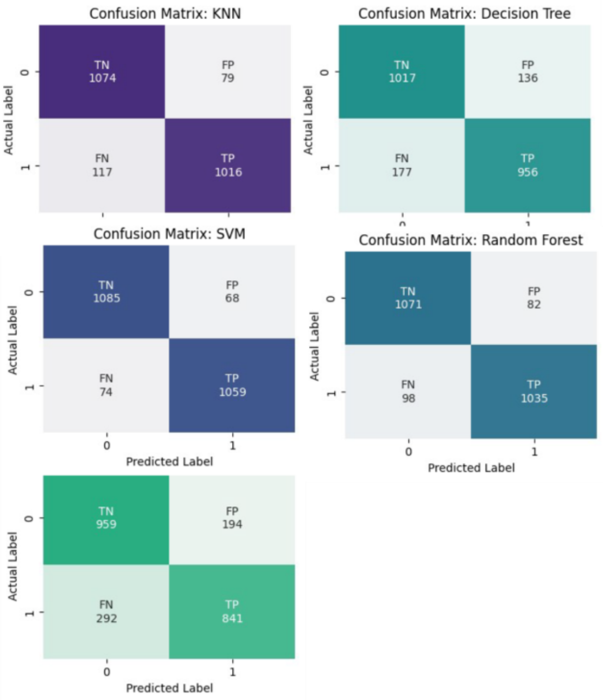
affect user trust. These matrices reveal the misclassification patterns specific to each classifier. Table III presents the TP, FP, TN, and FN values for each model[1].

TABLE III

CONFUSION MATRIX RESULTS

Model	TN	FP	TP	FN
KNN	1074	79	1016	117
SVM	1085	68	1059	74
Random Forest	1071	82	1035	98
Decision Tree	1017	136	956	177
Naive Bayes	959	194	841	292

Fig. 8. Confusion matrices for all classifiers



D. Dimensionality Reduction

To enhance model efficiency Principal Component Analysis (PCA) is applied to reduce feature dimensionality while pre- serving important information. The explained variance ratio demonstrates that selecting the first 28 principal components retains approximately 95% of the variance, ensuring that crucial features contribute to classification while reducing com- putational complexity. To improve computational efficiency of model Principal Component Analysis (PCA) is applied for the dimensionality reduction. The explained variance ratio (Figure 9) shows that the first 28 principal components retain approximately 95% of the variance which ensures minimal loss of information while optimizing model). [3]

E. Cross-Validation Results

Cross-validation is used to validate the robustness of the models by assessing their performances on multiple splits of training and testing of dataset. Here, the consistency between cross-validation accuracy and test set accuracy confirms that the models generalize well and are not prone to overfitting. Additionally GridSearchCV is used for hyper-parameter tuning to optimize model performance, which to ensure the reliability of this models cross-validation is performed alongside test set evaluation. The results indicate that the performance across cross-validation and test sets is consistent, which confirms that the models generalize well without overfitting (Figure 10). GridSearchCV is performed for hyperparameter tuning, and the best parameters for each classifier are determined as follows:

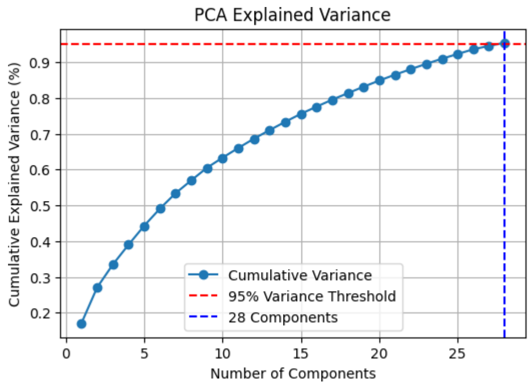


Fig. 9. PCA explained variance ratio (28 components retain 95% variance)

- KNN: n_neighbors=5, weights='distance'
- SVM: C=10, kernel='rbf'
- Random Forest: max_depth=20, n_estimators=150
- Decision Tree: max_depth=10, min_samples_split=2
- Naive Bayes: Default parameters

To ensure the reliability of the models, cross-validation is performed alongside test set evaluation. And the results indi- cates that the performance across cross-validation and test sets remains consistent confirming that the models are generalized well without overfitting. The analysis highlight that SVM achieves the highest accuracy of 93.7% which demonstrates model's phishing detection capability. While due to higher false positive rate Naive Bayes got the lowest accuracy (78.7%).

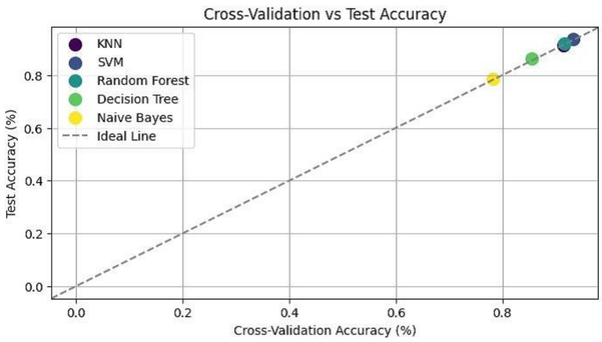


Fig. 10. Cross-validation vs Test Set Performance

E. Key observations from the result

- The Support Vector Machine (SVM) consistently deliv- ered better results than the other algorithms. It strikes a good balance between catching phishing sites and avoiding unnecessary false alarms.
- Na¨ıve Bayes struggled when came to identifying genuine websites correctly. This made a noticeable effect on its overall performance.

- Implementing PCA (Principal Component Analysis) re- ally helped it trimmed down the number of features without sacrificing important information. This made the model faster and more efficient.
- While Random Forest achieved good accuracy still SVM came out on top and offered overall better balance and reliability.

In general, the phishing detection system showed strong accuracy and did a great job of telling phishing websites from legitimate ones. SVM has been shown to be the most effective while managing to minimize both false positives and false negatives.

Conclusion

The PhishPatrol system offers a practical and intelligent solution for detecting phishing attempts in real-time across various digital platforms. At its core, the system uses a combination of machine learning models — including Random Forest, Decision Tree, Support Vector Machine (SVM), Naive Bayes, and K-Nearest Neighbors (KNN) — to analyze and flag suspicious activity. Among these, SVM delivered the highest accuracy at 93.79%, making it the most effective in our tests.

- High-accuracy phishing detection (93.79% for SVM)

But detecting phishing alone is not enough. That's why we also built a Phishing Awareness Website as part of the project. It is designed to help users:

- Understand how phishing works
- Learn key cybersecurity habits
- Watch interactive videos that simulate real phishing sce- narios

By combining automated detection with user education, PhishPatrol creates a more comprehensive and long-term approach to phishing prevention. Our results show strong performance from the technical models, while the awareness site empowers users to make smarter decisions online.

Looking ahead, we plan to:

- Improve browser compatibility
- Use adversarial training to make the system more resilient
- Expand support for mobile devices

In short, PhishPatrol is not just about catching phishing attempts — it is about helping people stay safe online through a smarter, more informed approach to cybersecurity.

REFERENCES

a. Alswailem, N. Alrumayh, B. Alabdullah, and A. Alsedrani, "Detect- ing Phishing Websites Using Machine Learning," *Journal of Computer Science*, 2017, doi: 10.1016/j.jcs.2017.05.006.

2. M. N. Alam, D. Sarma, F. F. Lima, I. Saha, R.-E.-Ulfath, and S. Hossain, "Phishing Attacks Detection Using Machine Learning," in *Proc. IEEE Int. Conf. Commun., Signal Process., Appl.*, 2019, vol. 5, no. 2, pp. 45–51.

3. Y. A. Al-Sariera, V. E. Adeyemo, A. O. Balogun, and A. K. Alazzawi, "AI Meta-Learners and Extra-Trees Algorithm for Phishing Website Detection," *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 5, pp. 134–142, 2020.

a. Karim, M. Shahroz, K. Mustofa, S. Brahim, and S. R. K. Joga, "Hybrid Machine Learning Phishing Detection System Based on URLs," in *Proc. Int. Conf. Comput. Sci. Artif. Intell.*, IEEE, 2018, pp. 321–327.

4. Basit, M. Zafar, X. Liu, A. R. Javed, Z. Jalil, and K. Kifayat, "Survey of AI-Enabled Phishing Detection Techniques," in *Proc. 15th Int. Conf. Artif. Intell. Comput. Sci.*, 2021, pp. 52–60.

5. R. Kadam, G. Kaur, H. Jain, and A. Tiwari, "Machine Learning Approach for Phishing Website Detection: A Literature Survey," *Int.*

6. *J. Comput. Appl.*, vol. 182, no. 23, pp. 1–5, 2021.

7. M. A. Taha, "A Machine Learning Algorithms for Detecting Phishing Websites: A Comparative Study," *Iraqi J. Comput. Sci. Math.*, vol. 5, no. 3, pp. 275–286, 2024.

8. S. Aslam, H. Aslam, A. Manzoor, C. Hui, and A. Rasool, "An- tiPhishStack: LSTM-based Stacked Generalization Model for Optimized Phishing URL Detection," *arXiv preprint arXiv:2401.08947*, 2024.

9. P. Maneriker, J. W. Stokes, E. G. Lazo, D. Carutasu, F. Tajaddodianfar, and A. Gururajan, "URLTran: Improving Phishing URL Detection Using Transformers," *arXiv preprint arXiv:2106.05256*, 2021.

10. Fitzpatrick, X. Liang, and J. Straub, "Fake News and Phishing Detection Using a Machine Learning Trained Expert System," *arXiv preprint arXiv:2108.08264*, 2021.

11. P. Chang, "Multi-Layer Perceptron Neural Network for Improving Detection Performance of Malicious Phishing URLs Without Affecting Other Attack Types Classification," *arXiv preprint arXiv:2203.00774*, 2022.

12. S. Mittal, A. Sharma, and R. Gupta, "Phishing Detection Using Natural Language Processing and Machine Learning," *SMU Data Sci. Rev.*, vol. 4, no. 1, 2022.

13. R. S. Rao and S. T. Ali, "PhishShield: A Desktop Application to Detect Phishing Webpages through Heuristic Approach," *Procedia Comput. Sci.*, vol. 54, pp. 147–156, 2015.

14. Jakobsson and E. Myers, *Phishing and Counter-Measures: Under- standing the Increasing Problem of Electronic Identity Theft*. Wiley, 2006, pp. 2–3.

15. M. Blasi, "Techniques for Detecting Zero Day Phishing Websites," M.A. thesis, Iowa State Univ., USA, 2009.

16. M. A. Adebowale, K. T. Lwin, and M. A. Hossain, "Intelligent Phishing Detection Scheme Using Deep Learning Algorithms," *J. Enterp. Inf. Manage.*, Early Access, p. 20.

17. S. Anwar et al., “Countering Malicious URLs in Internet of Things Using a Knowledge-Based Approach and a Simulated Expert,” *IEEE Internet Things J.*, vol. 7, no. 5, pp. 4497–4504, May 2020.

a. S. Bozkir and M. Aydos, “LogoSENSE: A Companion HOG-Based Logo Detection Scheme for Phishing Web Page and Email Brand Recognition,” *Comput. Secur.*, vol. 95, p. 18, Aug. 2020, Art. no. 101855.

18. Subasi, E. Molah, F. Almkallawi, and T. J. Chaudhery, “Intelligent Phishing Website Detection Using Random Forest Classifier,” in *Proc. IEEE Int. Conf. Electr. Comput. Technol. Appl. (ICECTA)*, 2017, pp. 1–5.

19. M. Dedakia and K. Mistry, “Phishing Detection using Content Based Associative Classification Data Mining,” *J. Eng. Comput. Appl. Sci.*, vol. 4, no. 7, pp. 209–214, 2015.