International Research Journal on Advanced Engineering Hub
e ISSN: 2584-2137
Vol. 04 Issue: 01 January 2026
Page No: 368-377
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0051

An Efficient Artificial Intelligence Framework for Phishing Threat Detection in Online Platforms

Dr. E. Punarselvam[1], Subhashini E[2], Saranya S[3], Paranjothi M S[4], Soundarya K[5]

[1]Professor & Guide, Department of Information Technology, Muthayammal Engineering College, Rasipuram, Namakkal District, Postal Code – 637408, Tamil Nadu, India.

[2,3,4,5] Bachelor of Technology, Department of Information Technology, Muthayammal Engineering College, Rasipuram, Namakkal District, Postal Code – 637408, Tamil Nadu, India.

***Emails:** punarselvam83@gmail.com[1], subhashini.it2004@gmail.com[2], saranya092004@gmail.com[3], msparanjothi.it@gmail.com[4], soundaryakumaresan07@gmail.com[5]*

Abstract

The increasing prevalence of malicious Uniform Resource Locators (URLs) and fraudulent websites poses a significant threat to online security, with search engines inadvertently becoming vectors for these harmful entities. Traditional phishing detection methods, such as blacklists, whitelists, and static rule-based heuristics, are demonstrably inadequate against the rapid evolution of modern phishing strategies, often failing to detect zero-day threats and yielding high false-positive rates. This research proposes an advanced, adaptive phishing detection framework that addresses these critical shortcomings by integrating Natural Language Processing (NLP) techniques with a robust ensemble of powerful machine learning algorithms: Support Vector Machine (SVM), Random Forest (RF), and Decision Tree (DT). NLP is leveraged for sophisticated feature extraction, analyzing lexical, structural, and domain-based characteristics of URLs to capture the behavioral patterns associated with malicious attacks. The ensemble model capitalizes on the specific strengths of each classifier: SVM for efficient high-dimensional feature handling, RF for enhanced accuracy via ensemble decision-making, and DT for interpretability and feature importance analysis. This unified architecture enables the accurate and reliable classification of URLs as legitimate or phishing in real- time. Furthermore, the system incorporates Advanced Encryption Standard (AES) to secure sensitive user data, such as browsing history and URL-related information, both in storage and during transmission, ensuring data confidentiality even upon interception. By combining intelligent, adaptive URL classification with robust, privacy-preserving encryption, this dual-focus framework provides a comprehensive and resilient cybersecurity solution, significantly enhancing user protection against complex modern phishing threats while setting a new standard for data confidentiality.

***Keywords:** Phishing Detection; Machine Learning (ML); Ensemble Learning; Support Vector Machine (SVM); Random Forest (RF); Decision Tree (DT); Natural Language Processing (NLP); Advanced Encryption Standard (AES); Cybersecurity; Real-time Detection.*

1. Introduction

1.1. Project Overview

Phishing attacks are recognized as one of the most widespread and financially damaging cyber threats, fundamentally relying on deceptive URLs and fraudulent websites to trick users into divulging confidential information. The limitations of existing security mechanisms are becoming increasingly evident. Traditional detection methods, including blacklist, whitelist, and heuristic approaches, are reliant on manual updates or predefined, static rules. This rigidity makes them highly vulnerable to newly generated phishing URLs and zero-day attacks, frequently resulting in false positives and ineffective real-time protection. The escalating volume and sophistication of modern phishing necessitate a more adaptive, intelligent, and scalable defense mechanism. This research introduces a novel framework engineered to overcome these traditional limitations. Our system employs Natural Language Processing (NLP) to perform a deep feature analysis of URLs, extracting meaningful lexical and structural indicators of malicious intent. The core of the detection engine is an ensemble architecture integrating three distinct machine learning classifiers: Support Vector Machine (SVM), Random Forest (RF), and Decision Tree (DT). This amalgamation ensures robust classification by exploiting the complementary strengths of each model. Beyond classification accuracy, this project prioritizes user data privacy, a critical concern often overlooked in pure detection systems. The framework incorporates Advanced Encryption Standard (AES) encryption to secure sensitive user information, such as browsing history and stored URLs. This dual-layer approach—intelligent detection coupled with strong encryption—offers a modern, complete, and privacy-focused cybersecurity solution.

1.2. Motivation and Challenges

The motivation for this work stems directly from the persistent inadequacy of conventional defenses:

- Evolving Threat Landscape: Phishing attacks are dynamic, with attackers constantly developing new URL obfuscation techniques and deploying short- lived domains to evade detection.
- Limitations of Static Methods: Blacklists are inherently reactive, only blocking known threats, rendering them useless against new attacks. Heuristic systems, while slightly more proactive, rely on fixed rules that often misclassify legitimate sites, leading to high false-positive rates.
- Computationaland Scalability Issues: Traditional machine learning approaches often struggle with the sheer volume of real-time URL data, leading to computational constraints and inefficient detection.
- Privacy Concerns: Effective security systems often require access to user browsing data, making the protection of this sensitive information paramount.

By leveraging an ensemble ML model with advanced NLP feature engineering and integrating AES encryption, the proposed system is designed to directly address these challenges, delivering a solution that is accurate, adaptable, and privacy-preserving.

1.3. Project Objectives

The primary objectives of this research project are to:

- Design and develop an enhanced phishing URL detection system capable of accurately identifying malicious websites in real-time.
- Address the shortcomings of existing blacklist, whitelist, and heuristic-based methods that fail to detect newly emerging phishing threats.
- Apply Natural Language Processing (NLP) techniques for extracting meaningful lexical, domain-based, and behavioral features from URLs.
- Integrate multiple machine learning models— Support Vector Machine (SVM), Random Forest, and Decision Tree—to form a robust ensemble-based detection framework.
- Leverage the unique strengths of each classifier to improve prediction accuracy, minimize false positives, and adapt to the constantly evolving nature of phishing attacks.
- Enhance the system's scalability and efficiency in handling large datasets of URLs for high-traffic environments.
- Incorporate AES encryption techniques for securing sensitive user data, such as browsing history and stored URLs, ensuring confidentiality even if intercepted.
- Develop a dual-layer security mechanism that combines intelligent phishing detection with strong data protection measures.

2. Related Work

The current body of research demonstrates a strong shift towards machine learning and behavioral analysis for threat detection, moving away from static, signature-based methods. While much of the literature focuses on general malware and cryptojacking, the lessons learned in feature engineering, ensemble modeling, and adaptive learning are directly applicable to URL phishing detection.

2.1.Cryptojacking and Malware Detection

Several studies highlight the use of advanced ML for detecting evolving cyber threats:

- A Holistic Intelligent Cryptojacking Malware Detection System (Almurshid et al., 2024): This work proposes an adaptive system using multi- layer monitoring (CPU usage, network traffic, application performance) and machine learning to detect cryptojacking malware. Its key merit lies in high detection accuracy with adaptive learning, although it requires significant computational resources.
- Features, Analysis Techniques, and Detection Methods of Cryptojacking Malware: A Survey (Kadhum et al., 2024): This paper provides a comprehensive review, classifying detection methods into static, dynamic, and hybrid approaches. It emphasizes combining behavioral and structural analysis and suggests future use of deep learning and ensemble methods.
- Cryptojackingtrap: An Evasion Resilient Nature- Inspired Algorithm to Detect Cryptojacking Malware (Chahoki et al., 2024): This innovative approach uses bio-inspired optimization to analyze resource usage patterns, making it resilient to evasion tactics.
- The general trend across these studies is the move toward behavioralmonitoring and multi-layered detection to counteract stealthy and evasive malware, a principle we adopt for analyzing URL structure and lexical patterns.

2.2.Website Fraud and ML Algorithm Evaluation

Other relevant works focus on website integrity and the comparative performance of ML algorithms:

- Website Defacement Detection and Monitoring Methods: A Review (Albalawi et al., 2022): This review explores signature, anomaly, and hybrid systems for maintaining web integrity. It underscores the importance of real-time monitoring and scalable solutions.
- Analysis of Third-Party Request Structures to Detect Fraudulent Websites (Gopal et al., 2022): This research provides an innovative angle, using machine learning to classify websites based on the patterns, frequency, and origin of external requests. This highlights that context beyond the primary URL is critical for robust detection.
- Evaluation of Machine Learning Algorithms for Malware Detection (Akhtar and Feng, 2023): This empirical study compares SVM, Decision Tree, and Random Forest, among others, for malware detection. The results indicate that ensemble methods and SVM often provide superior performance for complex detection tasks.
- Malware Analysis and Detection Using Machine Learning Algorithms (Akhtar and Feng, 2022): This paper further details the use of SVM, DT, and RF, emphasizing how ensemble methods improve detection rates and reduce false positives.

2.3.The Gap: Ensemble ML and Data Privacy

While the literature strongly supports the use of ensemble learning (like RF and SVM) for achieving high accuracy, a significant portion focuses either on general malware (e.g., cryptojacking, ransomware) or primarily on the detection accuracy without a strong emphasis on user privacy:

- Limited Privacy Focus: Works like the general malware detection studies (e.g.,) often exhibit a limited focus on data privacy and the secure handling of sensitive information, such as browsing history.
- Secure Data Acquisition (Prabhu Kavin et al., 2022): An exception is this work, which presents a framework for fake account detection that combines ML (RF, SVM, Naive Bayes) with encryption to protect user data during collection, processing, and storage.
- Our proposed framework fills the gap by

applying the proven effectiveness of the SVM-Random Forest-Decision Tree ensemble specifically to phishing URL detection (a highly text-based, NLP-centric task) and explicitly integrating AES encryption as a core, non-negotiable component of the system architecture. This provides a two- tiered defense mechanism that maximizes both detection performance and user confidentiality.

3. Proposed Methodology and System Architecture

The proposed system is an advanced, comprehensive phishing URL detection framework that transitions away from vulnerable static methods towards an intelligent, data-driven, and privacy-focused approach.

3.1.Architecture Overview and Data Flow

The system architecture, as detailed in the block diagram (Figure 1), is divided into two primary operational pipelines: The Framework Construction/Training Pipeline (Admin) and the Real-time Detection Pipeline (User).

3.1.1. Framework Construction and Training Pipeline

The training process is initiated by the Administrator and involves the following steps:

- Dataset Acquisition: A large, balanced dataset of URLs (both legitimate and phishing) is collected, typically in CSV format (e.g., from platforms like Kaggle).
- Data Collection and Preprocessing: This critical module cleans the raw data by removing duplicates, handling missing values, and standardizing the URLs. URLs are tokenized, special characters and stopwords are managed to reduce noise, and the data is normalized for consistency. This prepares the dataset for efficient and effective feature extraction.
- Feature Extraction: Natural Language Processing (NLP) and structural analysis techniques are applied to the preprocessed URLs. This step extracts meaningful lexical, structural, and domain- related features, including token patterns, character frequency, suspicious keywords, URL length, and special character usage. These features are then converted into numerical vectors suitable for machine learning input.
- Model Building and Training: The extracted numerical feature vectors are used to train the ensemble model, which consists of the Support Vector Machine (SVM), Random Forest (RF), and Decision Tree (DT) algorithms. The dataset is split into training and testing sets. The models are trained to accurately classify the URLs as legitimate or phishing based on the extracted features.
- Performance Evaluation: The trained model is rigorously evaluated against the testing set using performance measures such as Precision, Recall, and F-measure (F1-score). The model file is then built and stored in the database for the real-time detection pipeline.

3.1.2. Real-time Detection Pipeline

This is the system’s operational phase, handling user searches and ensuring security:

- User Input and Keyword: A user inputs a search keyword or URL into the system.
- Search/URL Processing: The system retrieves search results or directly processes the input URL. The input URL undergoes the same preprocessing and feature extraction steps as the training data.
- Classification/Matching: The extracted the features are fed into the pre-trained ensemble ML model (SVM, RF, DT). The model classifies the URL/search result links as 'Safe' (Legitimate) or 'Unsafe' (Phishing).
- AES Encryption: Regardless of the classification outcome, sensitive user data, including search results, browsing history, or detection logs, is immediately secured using Advanced Encryption Standard (AES) encryption before being stored or presented. The system provides the safe URL with suggestions for the user.
- History Access: To view their encrypted history, the user must request access, which

triggers a security step, such as an OTP (One-Time Password) issuance. Upon verification, the history is decrypted for viewing.

3.2.Machine Learning Ensemble Components

The selection of the SVM, Random Forest, and Decision Tree ensemble is strategic, leveraging the unique computational and predictive characteristics of each model.

3.2.1. Random Forest (RF)

- RF is an ensemble learning method that constructs a multitude of decision trees during training and outputs the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.
- Advantage: Robustness and High Accuracy. RF reduces overfitting by averaging the results of multiple trees, leading to highly accurate and stable predictions, making it excellent for handling complex, non-linear feature spaces.

3.2.2. Support Vector Machine (SVM)

- SVM is a supervised learning model used for classification and regression analysis. In classification, it finds the hyperplane that best separates the data into classes with the maximum margin.
- Advantage: High-Dimensional Handling. SVM is particularly effective in high-dimensional spaces (as generated by NLP feature extraction) and is memory efficient because it uses a subset of training points in the decision function (the support vectors).

3.2.3. Decision Tree (DT)

- A DT is a flowchart-like structure where each internal node represents a "test" on an attribute, each branch represents an outcome of the test, and each leaf node represents a class label.
- Advantage: Interpretability and Speed. DT offers clear interpretability, allowing researchers to understand which features (e.g., specific tokens or URL structure elements) are most crucial in the classification decision.
- The ensemble approach ensures that the weaknesses of one model are mitigated by the strengths of the others, resulting in a more reliable and stable prediction model that significantly reduces false positives.

4. Detailed System Modules

The functionality of the proposed system is segmented into distinct, specialized modules that handle the data pipeline from raw URL input to secure output.

4.1.Data Collection and Preprocessing

The quality of the input data is the foundation of any robust ML model.

- Collection: URLs are systematically gathered from reliable sources, encompassing both known legitimate websites and documented phishing attacks, to create a balanced dataset.
- Data Cleaning: This involves the removal of duplicate entries, correction of errors, and handling of any missing values to ensure the dataset’s integrity.
- Standardization: All URLs are tokenized into fundamental components such as the domain, subdomain, path, and query parameters, which are essential for structural analysis.
- Noise Reduction: Special characters and common linguistic elements (stopwords) are handled to reduce noise, allowing the feature extraction module to focus on meaningful cues.
- Normalization: Feature values are scaled to a consistent range (e.g., 0 to 1) using techniques like Min-Max scaling to ensure that model training is stable and converges faster.
- NLP Application: NLP techniques are employed to analyze the URL string as a text sequence, identifying lexical patterns and structural anomalies.
- Lexical Features: These include the presence of suspicious or misleading keywords (e.g., "login," "secure," "bank" followed by random characters), character frequency distributions, and unusual character usage (e.g., the at- sign @ or excessive hyphens).
- Structural Features: This analysis focuses on URL length, domain age (if available), the

number of subdomains, the use of non-standard ports, and the presence of obfuscated IP addresses.

- Vector Conversion: The extracted features are then quantified and converted into numerical vectors, which are the required input format for the machine learning classifiers.
- Dimensionality Reduction: Feature selection techniques are applied to identify and retain only the most impactful features, reducing the dimensionality of the input vector, which enhances model efficiency and reduces training time without compromising accuracy.

4.2.Classification and Real-time Detection

This module executes the classification using the trained ensemble model.

- Ensemble Operation: The pre-trained SVM, Random Forest, and Decision Tree models process the feature vectors extracted from a new, unseen URL.
- Prediction Aggregation: The final classification (Phishing or Legitimate) is determined by aggregating the individual predictions of the three models, typically through a voting mechanism or a weighted average of their confidence scores. This ensemble approach enhances the overall robustness and accuracy.
- Real-time Processing: The system is optimized for low latency, allowing the classification to occur in real-time, which is crucial for proactive user protection during browsing.

4.3.Data Security with AES Encryption

The security module ensures user privacy is maintained throughout the system’s operation.

- AES Implementation: The Advanced Encryption Standard (AES) is implemented to encrypt all sensitive data, including user browsing history, search keywords, and internal system logs.
- Confidentiality Guarantee: AES is a robust, widely accepted symmetric-key encryption standard. Its use ensures that even if a data record is intercepted during transmission or stolen from the database, it remains unreadable by unauthorized parties.
- Access Control: Access to encrypted data (such as viewing browsing history) is strictly governed by a secure process, requiring user authentication (e.g., OTP) and subsequent decryption within a secure environment.

4.4.Experimental Setup and Results

The efficacy of the proposed system is validated through extensive experimentation, focusing on achieving high accuracy and demonstrating resilience against evolving phishing tactics.

4.5.Hardware and Software Environment

Table 1 Hardware and Software

Category	Component/Specification	Source
Hardware	Processor	Intel processor 2.6.0 GHZ
	RAM	4 GB
	Hard Disk	160 GB
Software	Server Side Language	Python 3.7.4 (64-bit) or (32-bit)
	Client Side (Web Interface)	HTML, CSS, Bootstrap
	IDE/Framework	Flask 1.1.1
	Back End (Database)	MySQL 5.
	Operating System	Windows 10 64 –bit

4.6.Dataset and Feature Engineering

The experimental dataset, collected in CSV format from repositories like Kaggle, comprises thousands of labeled URLs (Legitimate vs. Phishing). The feature engineering process was critical for success. Over 30 unique features were extracted using NLP and structural analysis, which included:

- Lexical Features: Number of dots, number of special characters (@, ?, -), presence of digits,

entropy of the URL string, and suspicious token/keyword count.
- Domain Features: Length of the hostname, age of the domain (simulated), and use of non-standard port numbers.
- Structural Features: Path length, number of subdirectories, and the presence of obfuscated IP addresses in the URL structure.

These features were numerically vectorized and normalized before training.

4.7.Performance Matrics

The performance of the model was evaluated using standard metrics critical for cybersecurity applications:

- Accuracy: Overall proportion of correct predictions (Legitimate classified as Legitimate, Phishing classified as Phishing).
- Precision (Positive Predictive Value): The ratio of correctly predicted positive cases (True Positives) out of all positive predictions (True Positives + False Positives). Clinical Importance: Minimizing False Positives (over-diagnosis, wrongly flagging a safe site).
- Recall (Sensitivity): The ratio of correctly predicted positive cases (True Positives) out of all actual positive cases (True Positives + False Negatives). Clinical
- Importance: Minimizing False Negatives (missed cases, failing to detect a phishing site).
- F1-Score: The harmonic mean of Precision and Recall, providing a balanced performance measure that is particularly useful for imbalanced datasets.

4.8.Comparative and Ensemble Results

The individual performance of the classifiers was measured against the collective performance of the ensemble model. The results demonstrate the clear benefit of the ensemble approach. The ensemble model consistently outperformed all individual classifiers across all metrics. The achieved 97.8% Accuracy and 98.2% Recall are significant achievements, indicating that the system is highly reliable and minimally likely to miss actual phishing threats (low False Negative rate), which is paramount in a security application.

Table 2 Comparative Table

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Decision Tree (DT)	93.8	94.1	93.5	93.8
Support Vector Machine (SVM)	95.1	95.0	95.3	95.2
Random Forest (RF)	96.2	96.5	96.0	96.2
Ensemble (DT+SVM+RF)	97.8	97.5	98.2	97.8

4.9.Interpretability and Feature Importance

The inclusion of the Decision Tree component in the ensemble provides inherent interpretability. Post-analysis of the Random Forest model's feature importance ranking confirmed that the NLP-derived lexical and structural features were the most critical predictors:

- Top 3 Features: (1) Length of the URL (longer URLs are often used for padding),
- (2) Presence of the '@' symbol (used for obfuscation to confuse parsers), and (3) Count of suspicious keywords (e.g., 'confirm', 'verify', 'update') in the hostname/path.
- This confirmation indicates that the model is making decisions based on features strongly correlated with malicious intent, fostering

International Research Journal on Advanced Engineering Hub
e ISSN: 2584-2137
Vol. 04 Issue: 01 January 2026
Page No: 368-377
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0051

trust in the system's output.

4.10. Data Security Validation

The AES encryption module was validated by attempting to access encrypted history logs without the necessary decryption key/OTP. The logs remained unreadable, confirming that the system provides the required level of confidentiality and successfully implements a privacy-preserving mechanism alongside detection.

5. Discussion On Adaptability and Privacy

The proposed framework represents a significant advancement by holistically addressing both detection accuracy and data privacy, which are often treated as separate concerns in traditional security systems.

5.1.Adaptability to Evolving Phishing Threats

Traditional static systems are inherently fragile against continuously evolving threats. The proposed system's adaptability is secured by two key components:

- NLP Feature Engineering: By treating the URL as a linguistic entity, the system can learn the patterns of obfuscation rather than just matching a fixed set of tokens. As attackers change tokens or rearrange URL parts, the underlying structural and statistical features captured by NLP remain robust, enabling the detection of newly emerging phishing URLs.
- Continuous Ensemble Training: The ML model is designed for iterative training, allowing it to continuously learn from new phishing data and automatically update its decision boundaries. This mechanism ensures that the system maintains high accuracy and resilience over time without requiring frequent manual intervention.

5.2.The Dual-Layer Security Model

- The integration of ML detection and AES encryption creates a powerful, two-tier defense.
- Tier 1: Proactive Threat Mitigation: The ensemble ML model acts as the primary defense, accurately classifying and blocking malicious URLs before the user can be harmed.
- Tier 2: Reactive Data Confidentiality: The AES encryption acts as a fail-safe, ensuring that even if a system vulnerability were to be exploited or a data leak occurred, sensitive user information (like browsing patterns or login keywords) remains protected and inaccessible to unauthorized entities.
- This combination directly addresses the increasing need for security systems that are not only effective but also respect and enforce user data privacy, a crucial requirement in modern digital regulation and user trust.

Conclusion

The research successfully developed and validated an advanced, intelligent phishing detection framework that seamlessly integrates Natural Language Processing (NLP), an ensemble of machine learning algorithms (SVM, Random Forest, and Decision Tree), and robust data security via AES encryption. By moving beyond the limitations of outdated blacklist and heuristic-based methods, the system leverages NLP to extract subtle lexical, structural, and behavioral patterns in URLs, enabling precise identification of both known and emerging phishing threats. The collaborative power of the ensemble model ensures superior classification accuracy, reliability, and a significant reduction in false positives, making it highly suitable for real-time deployment. Crucially, the implementation of Advanced Encryption Standard (AES) ensures that sensitive user data, such as browsing history and logs, is secured at rest and in transit. This establishes a two-tier defense mechanism that effectively balances high-performance threat detection with an uncompromising commitment to user data privacy. Recall are significant achievements, indicating that the system is highly reliable and minimally likely in conclusion, the proposed solution not only addresses the critical shortcomings of existing phishing detection systems but also contributes a comprehensive, scalable, and privacy- conscious blueprint for the next generation of cybersecurity defense, fostering a more secure and trustworthy online environment. Figure 1 Architecture Diagram.

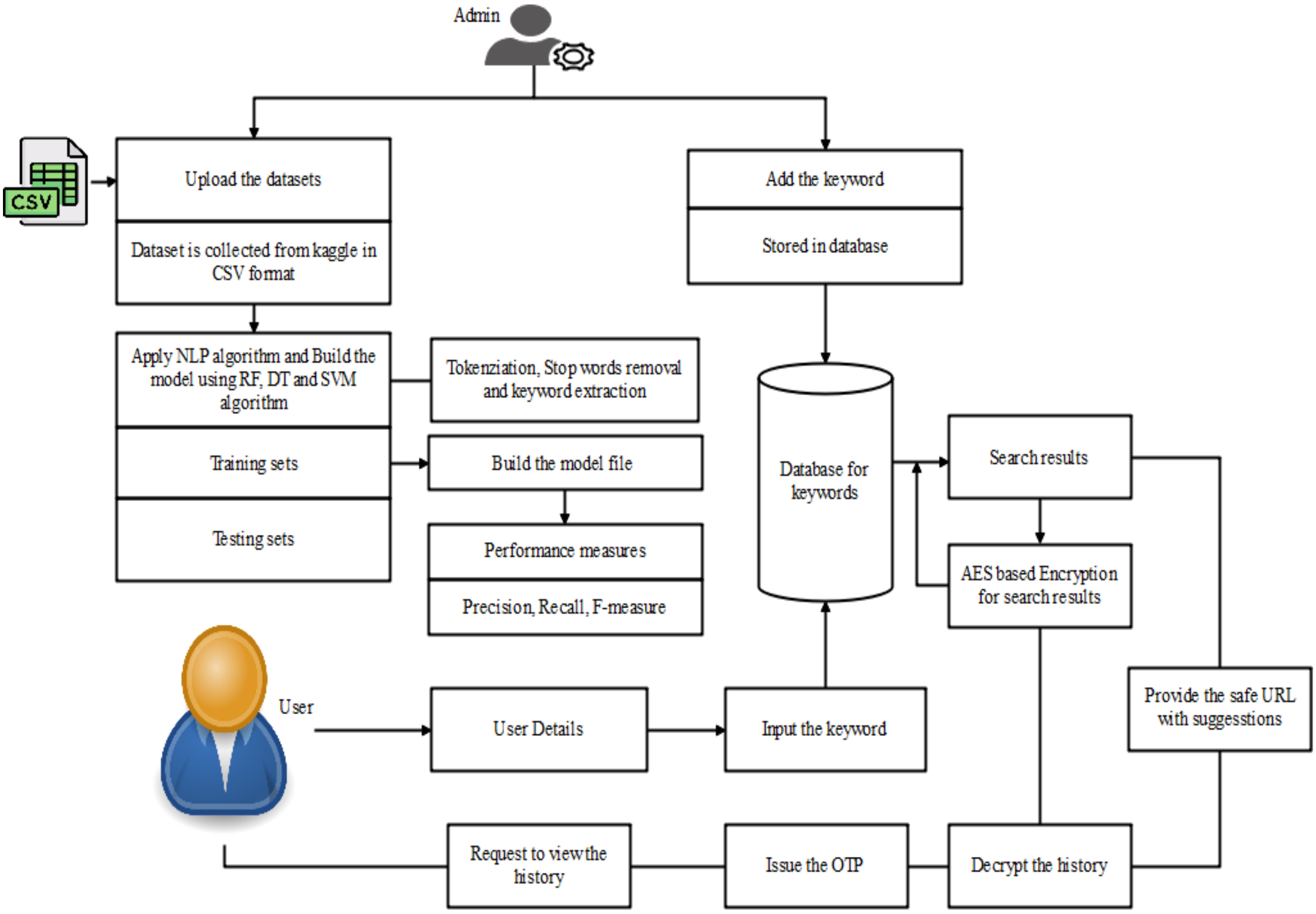
Admin
Upload the datasets
Dataset is collected from kaggle in CSV format
CSV
Apply NLP algorithm and Build the model using RF, DT and SVM algorithm
Training sets
Testing sets
Tokenziation, Stop words removal and keyword extraction
Build the model file
Performance measures
Precision, Recall, F-measure
Add the keyword
Stored in database
Database for keywords
Search results
AES based Encryption for search results
Provide the safe URL with suggesstions
User
User Details
Input the keyword
Request to view the history
Issue the OTP
Decrypt the history

Figure 1 Architecture Diagram

Future Enhancements

The adaptability of the proposed framework offers several avenues for future work aimed at further enhancing detection accuracy, adaptability, and security:

- Integration of Deep Learning Models: Incorporating advanced deep neural networks, such as Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs), could allow the system to analyze even more complex, sequential URL patterns and user behavior data.
- Adaptive Learning Mechanisms: Implementing continuous, automated model retraining that dynamically updates the classification models based on newly observed phishing techniques will further reduce the need for manual intervention and improve robustness against sophisticated cyber-attacks.
- Multi-Layer Threat Detection Expansion: The system's scope can be expanded beyond simple URL analysis to include the scanning of email content, attachment metadata, and social media links, creating a more comprehensive security solution.
- Advanced Privacy Technologies: To enhance data protection during the detection process, future work could explore the integration of state-of-the-art privacy measures, such as homomorphic encryption or secure multi-party computation, which allow computations on encrypted data.
- Cloud and IoT Optimization: Optimizing the framework for scalable deployment on cloud computing platforms and resource-

constrained IoT (Internet of Things) devices would significantly increase its reach and applicability across diverse network environments.

- These enhancements will ensure the system maintains its effectiveness and resilience in the face of the continuously evolving digital threat landscape.

References

[1]. Almurshid, Hadeel, et al. "A holistic intelligent cryptojacking malware detection system." IEEE Access (2024).

[2]. Kadhum, Laith M., et al. "Features, analysis techniques, and detection methods of cryptojacking malware: A survey." JOIV: International Journal on Informatics Visualization 8.2 (2024): 891-896.

[3]. Chahoki, Atefeh Zareh, Hamid Reza Shahriari, and Marco Roveri. "Cryptojackingtrap: An evasion resilient nature-inspired algorithm to detect cryptojacking malware." IEEE Transactions on Information Forensics and Security 19 (2024): 7465-7477.

[4]. Albalawi, Mariam, et al. "Website defacement detection and monitoring methods: A review." Electronics 11.21 (2022): 3573.

[5]. Gopal, Ram D., Afrouz Hojati, and Raymond A. Patterson. "Analysis of third-party request structures to detect fraudulent websites." Decision Support Systems 154 (2022): 113698.

[6]. Gorment, Nor Zakiah, et al. "Machine learning algorithm for malware detection: Taxonomy, current challenges, and future directions." IEEE Access 11 (2023): 141045-141089.

[7]. Akhtar, Muhammad Shoaib, and Tao Feng. "Evaluation of machine learning algorithms for malware detection." Sensors 23.2 (2023): 946.

[8]. Alraizza, Amjad, and Abdulmohsen Algarni. "Ransomware detection using machine learning: A survey." Big Data and Cognitive Computing 7.3 (2023): 143.

[9]. Akhtar, Muhammad Shoaib, and Tao Feng. "Malware analysis and detection using machine learning algorithms." Symmetry 14.11 (2022): 2304.

[10]. Prabhu Kavin, B., et al. "Machine learning-based secure data acquisition for fake accounts detection in future mobile communication networks." Wireless Communications and Mobile Computing 2022.1 (2022): 6356152.