Cyber-Sentinel: A Smart AI-Driven Cyber Guard for Continuous Threat Monitoring and Mitigation

Prof. Mohammed Juned Shaikh[1], Mohtaseem Khan[2], Rehan Khan[3],
Husain Ansari[4], Saad Shaikh[5]
[1,2,3,4,5]*Department of Computer Engineering, Rizvi College of Engineering, University of Mumbai*

***Abstract*—With the exponential growth of digital systems, cybersecurity threats have become increasingly sophisticated and dynamic. Traditional rule-based security mechanisms fail to detect novel and evolving attacks in real time. This paper proposes Cyber Sentinel, an AI-driven cybersecurity framework that integrates Machine Learning techniques with real-time monitoring systems to detect, analyse, and mitigate cyber threats proactively. The system leverages anomaly detection, behavioural analysis, and predictive modelling to enhance threat detection accuracy. The proposed architecture incorporates Python-based machine learning models with a scalable Node.js backend for continuous monitoring and automated response. Experimental evaluation using benchmark datasets demonstrates improved detection rates and reduced false positives compared to traditional systems The system also incorporates QR code URL extraction and analysis, allowing detection of malicious links embedded within QR codes**

***Index Terms*—Machine Learning, Intrusion Detection System, AI Security, Threat Detection, Anomaly Detection, Cyber Sentinel**

I. INTRODUCTION

The rapid expansion of the internet and digital services has significantly increased the attack surface for cyber threats, particularly phishing and malicious websites. These threats exploit human vulnerabilities by deceiving users into interacting with fraudulent links, often resulting in the compromise of sensitive information such as login credentials, financial data, and personal identities. Despite growing awareness, users continue to fall victim to such attacks due to the increasing sophistication and realism of malicious URLs. Traditional security mechanisms, including blacklist-based filtering and rule-based detection systems, are often insufficient in addressing modern phishing techniques [1]. These approaches struggle to identify newly generated or obfuscated URLs, commonly referred to as zero-day threats. Furthermore, they frequently generate high false positive rates, leading to user distrust and reduced system effectiveness. To address these challenges, there is a growing need for intelligent, automated, and real-time detection systems capable of identifying malicious URLs with high accuracy and minimal user intervention. In this context, Machine Learning (ML) techniques offer a promising solution by enabling systems to learn patterns and characteristics of malicious behaviour from data rather than relying solely on predefined rules. This paper presents a machine learning–based URL detection system as a core component of the proposed CyberSentinel framework. The system analyses structural and lexical features of URLs to classify them as either safe or malicious. By leveraging feature-driven classification models, the approach enhances detection capabilities while maintaining computational efficiency. Additionally, the system is designed for real-world applicability through integration with a browser extension and backend API, enabling real-time threat detection and user alerts. This practical deployment ensures that users receive immediate feedback when interacting with potentially harmful links, thereby reducing the risk of successful phishing attacks. In addition to URL-based threat detection Additionally, the system extends its detection capability by extracting embedded URLs from QR codes and analysing them using the same machine learning model.

Overall, the proposed approach aims to bridge the gap between theoretical detection models and deployable cybersecurity solutions, contributing toward a smarter and more proactive defence mechanism in modern digital environments.

II. PROBLEM STATEMENT AND OBJECTIVES

Existing URL protection mechanisms face significant limitations in effectively combating modern cyber threats. Traditional blacklist-based systems are unable to detect newly emerging phishing websites, while heuristic and rule-based approaches often generate a high number of false positives. Although machine learning–based solutions [3] have shown promise, many lack integration into real-time, user facing applications, limiting their practical usability.
Therefore, the key challenge addressed in this work is the development of an accurate, efficient, and real-time URL detection system that:

- Identifies both known and previously unseen malicious URLs
- Operates seamlessly in real-world environments such as browser extensions or lightweight APIs

2.1 Objectives

The primary objectives of this research are:

- To design and train a robust machine learning model capable of classifying URLs as safe or malicious based on discriminative URL-based features
- To enhance detection accuracy while minimizing false positives through effective feature engineering and validation techniques
- To integrate the trained model into a real-time detection pipeline using a browser extension or backend API
- To incorporate threat intelligence enrichment (e.g., optional verification using public threat feeds) for improved reliability
- To implement a feedback mechanism that allows continuous learning and performance improvement over time

III. LITERATURE REVIEW

Recent advancements in cybersecurity research have emphasized the use of machine learning techniques for detecting phishing and malicious URLs. Various studies have explored classification models such as Support Vector Machines (SVM), Decision Trees, and Random Forests to identify malicious patterns within URL structures.

A common approach involves extracting lexical and host-based features from URLs [3], including length, use of special characters, domain age, and subdomain patterns. These features are then used to train supervised learning models capable of distinguishing between legitimate and malicious links. While such methods have demonstrated promising accuracy, they often suffer from limitations such as overfitting, lack of generalization to unseen attacks, and dependency on static datasets. However, most existing systems lack real-time deployment capabilities and do not incorporate user-driven feedback mechanisms or multi-modal threat detection such as QR-based attacks. More recent research has introduced hybrid and ensemble models to improve detection performance. Additionally, some studies have explored deep learning techniques for capturing complex patterns in URL sequences. However, these approaches often require significant computational resources, making them less suitable for real-time deployment in lightweight environments such as browser extensions.

Another major limitation identified in existing literature is the lack of integration between detection models and user-facing applications. Many proposed systems [2] remain confined to experimental settings without practical implementation for end users. In this project, approximately 60,000 URLs comprising both legitimate and deceptive links were collected and utilized for model training and evaluation. Comprehensive preprocessing and feature extraction techniques were applied, including analysis of URL length, presence of special characters, domain structure, and other relevant attributes.

A machine learning classification model was then developed and evaluated based on performance metrics such as accuracy, precision, and recall. Furthermore, a prototype Chrome extension was implemented to demonstrate the practical applicability of the system, enabling users to manually verify URLs. The proposed system also outlines a future integration pipeline in which the browser extension communicates

with the machine learning model through a backend service, providing real-time classification results (Safe/Suspicious). This approach ensures both scalability and usability, addressing key gaps identified in existing research.

IV. METHODOLOGY

The proposed methodology for the CyberSentinel URL detection system focuses on developing a practical and efficient machine learning [2] pipeline for identifying malicious URLs. The system is designed to operate in both offline (training) and real-time (deployment) environments. The overall workflow consists of data collection, preprocessing, feature extraction, model training, evaluation, and system integration.

4.1 Data Collection

The initial stage of the proposed CyberSentinel system involves the collection of a comprehensive dataset consisting of approximately 60,000 URLs obtained from publicly available sources. The dataset includes a balanced combination of legitimate and malicious URLs, such as phishing, spam, and deceptive links. This diversity is essential to ensure that the machine learning model can generalize effectively across different types of web threats. By incorporating both safe and harmful examples, the system is trained to recognize subtle variations in URL structures and patterns, which is critical for accurate classification.

4.2 Data Preprocessing

Following data collection, preprocessing is performed to enhance the quality and consistency of the dataset. This stage involves the removal of duplicate and invalid URLs, normalization of URL formats, and handling of incomplete or missing entries. Additionally, labels are assigned to each URL, where legitimate links are categorized as safe and malicious links are marked accordingly. These preprocessing steps ensure that the dataset is well-structured and suitable for supervised learning, ultimately improving the reliability and performance of the trained model.

4.3 Feature Extraction

Feature extraction is a crucial step in the methodology, as it determines how effectively the model can distinguish between safe and malicious URLs. In this system, emphasis is placed on extracting lightweight lexical features directly from the URL, avoiding the need to analyse webpage content, which can increase computational complexity. The extracted features include URL length, the presence of special characters, usage of IP addresses instead of domain names, number of subdomains, protocol type (HTTP or HTTPS), and the occurrence of suspicious keywords such as “login” or “verify.” These features are widely recognized indicators of phishing behaviour and enable efficient detection with minimal processing overhead.

4.4 Model Selection

The classification of URLs is performed using a supervised machine learning approach. Algorithms such as Random Forest, Logistic Regression, and Decision Trees are considered due to their balance between predictive performance and computational efficiency. The dataset is divided into training and testing subsets, typically following an 80:20 ratio, to ensure proper evaluation of the model’s generalization capability. During training, the selected algorithm learns patterns and relationships between extracted features and their corresponding labels, enabling it to classify unseen URLs effectively.

4.5 Model Evaluation

To assess the effectiveness of the trained model, standard evaluation metrics such as accuracy, precision, recall, and F1-score are utilized. Accuracy measures the overall correctness of predictions, while precision and recall evaluate the model’s ability to correctly identify malicious URLs and detect all potential threats, respectively. The F1-score provides a balanced measure of both precision and recall. These metrics are particularly important in cybersecurity applications, as they help ensure that the system minimizes both false positives and false negatives, thereby maintaining reliability and user trust.

4.6 System Integration

To ensure practical applicability, the trained model is integrated into a real-time system architecture consisting of a Node.js-based backend, a Python-based machine learning module, and a Chrome extension interface. The backend is responsible for handling API requests and facilitating communication between the frontend and the machine learning model. The Python

module processes incoming URLs, performs feature extraction, and generates predictions. The Chrome extension serves as the user interface, allowing users to interact with the system and receive immediate feedback regarding the safety of URLs.

4.7 Real-Time Detection Flow

In the real-time detection process, when a user interacts with or inputs a URL, the Chrome extension captures the link and sends it to the backend API. The backend then forwards the request to the machine learning module, where the URL is analysed and classified based on the trained model. The prediction result, indicating whether the URL is safe or suspicious, is returned to the backend and subsequently displayed to the user through the extension interface. This streamlined workflow ensures minimal latency and enables continuous monitoring of user activity.

4.8 Threat Enrichment and Feedback Loop

To further enhance system performance, the proposed framework includes provisions for threat enrichment and continuous improvement. This involves optional verification of URLs using public threat intelligence sources to increase detection reliability. Additionally, detected URLs can be logged and analysed to identify emerging patterns. A feedback mechanism can also be implemented to periodically retrain the model using updated datasets, allowing the system to adapt to evolving cyber threats and maintain high detection accuracy over time.

4.9 System Architecture

The architecture of the proposed CyberSentinel system consists of multiple interconnected modules including authentication, URL analysis, machine learning detection, feedback processing, and visualization components

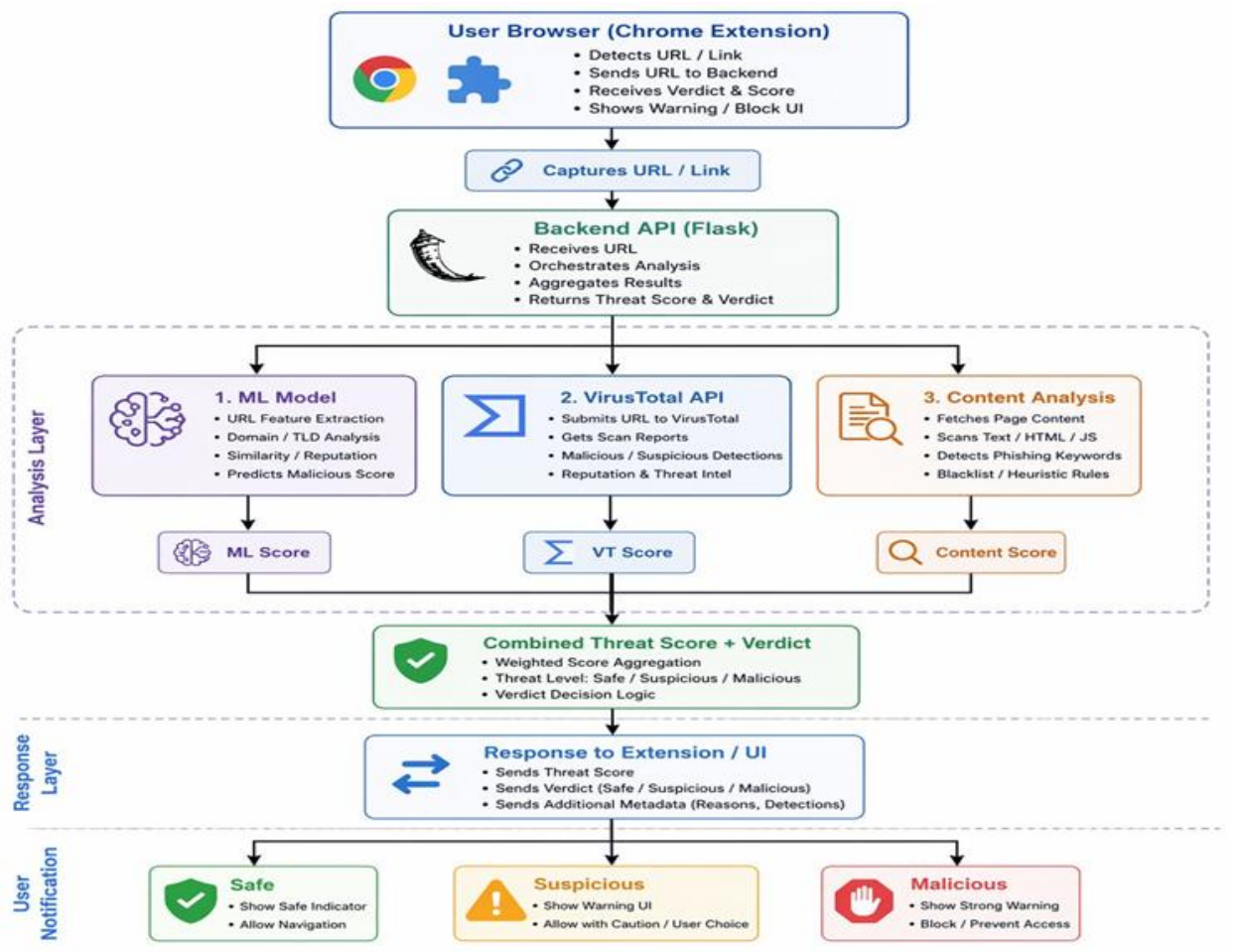


Fig1: System Architecture

V. PERFORMANCE EVALUATION METRICS FOR MODEL ACCURACY

A. Accuracy

Accuracy represents the overall correctness of the model by measuring the proportion of correctly classified instances (both safe and malicious) out of the total number of predictions. It provides a general indication of system performance.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

Where:
TP = True Positives
TN = True Negatives
FP = False Positives
FN = False Negatives

B. B. Precision

Precision measures the proportion of correctly identified malicious URLs out of all URLs predicted as malicious. It reflects how reliable the system is when it flags a URL as a threat.

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

Where:
TP = Malicious URLs correctly detected
FP = Legitimate URLs incorrectly flagged as malicious

C. Recall (Sensitivity)

Recall measures the ability of the model to correctly identify all actual malicious URLs. It indicates how effectively the system detects threats without missing them.

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

Where:
TP = Malicious URLs correctly detected
FN = Malicious URLs incorrectly classified as safe

D. F1 Score

The F1-score provides a balanced measure of performance by combining precision and recall. It is particularly useful when both false positives and false negatives need to be minimized.

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision+Recall} \quad (4)$$

Where:
Precision = Ratio of correctly predicted malicious URLs
Recall = Ratio of correctly detected malicious URLs

E. Classification Function

At the core of your system, the model predicts whether a URL is safe or malicious based on extracted features:

$$y = f(x_1, x_2, x_3, \ldots, x_n) \quad (5)$$

Where:
$x_1, x_2, \ldots, x_n$represent extracted URL features
$y \in \{0,1\}$, where 0 = Safe, 1 = Malicious
fis the trained machine learning model.

F. F. Logistic Regression

$$P(y = 1 \mid x) = \frac{1}{1+e^{-(w^T x+b)}} \quad (6)$$

Where:
w= weight vector
x= feature vector
b= bias term

VI. SYSTEM FEATURES

6.1 User Authentication Module

The system includes a secure login mechanism that ensures only authorized users can access the platform. This enhances system security and allows personalized tracking of user activities.

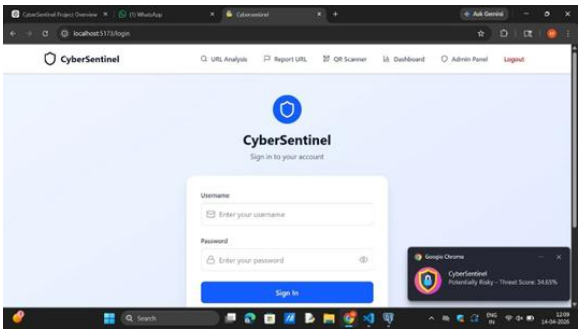


Fig 2: User Authentication

6.2 URL Detection Module

Users can manually input URLs to verify whether they are safe or potentially malicious. The system analyses

the input using the trained machine learning model and provides immediate classification results.

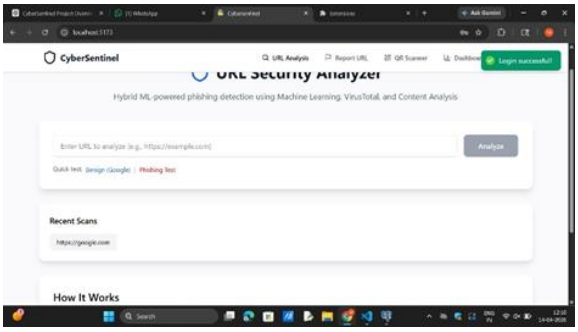


Fig 3: URL Detection Interface

6.3 Feedback based Learning Module

To improve detection accuracy, the system includes a feedback module where users can submit URLs that were incorrectly classified. These inputs are stored and used for future model retraining, enabling a reinforcement-like learning process.

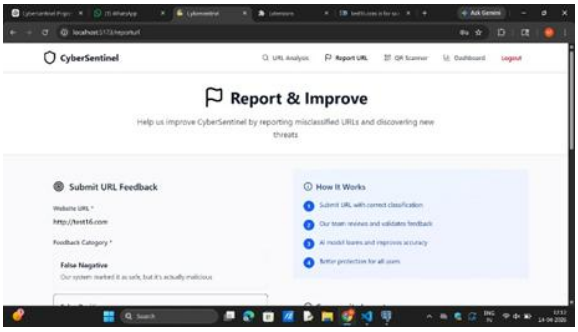
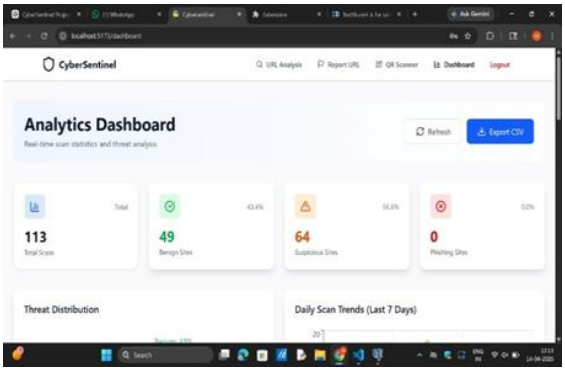


Fig 4: Feedback Page

6.4 Analytics Dashboard

A comprehensive dashboard is provided to visualize system performance and threat insights. It displays information such as threat distribution, daily scan trends, and detection statistics, helping users understand cybersecurity patterns



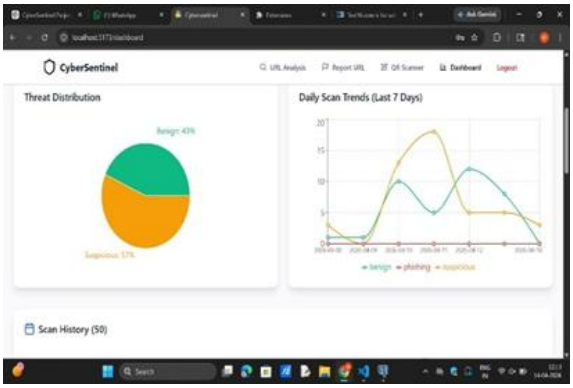


Fig 5&6: Analytics Dashboard

6.5 QR Code Security Scanner

The system includes a QR code analysis feature that extracts embedded URLs and evaluates them for potential threats. This helps prevent attacks delivered through malicious QR codes.

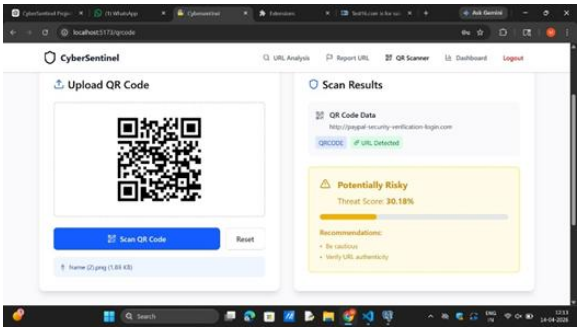


Fig 7: QR Code Analysis

6.6 Feedback Review Panel

An administrative panel is implemented to manage user-submitted feedback. It allows reviewing, approving, or rejecting submitted URLs before incorporating them into the training dataset, ensuring data quality and system reliability.

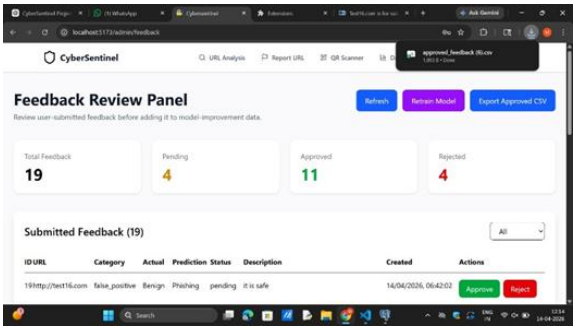


Fig 8: Feedback Review Panel

6.7 Browser Extension Integration

The CyberSentinel system includes a browser extension module designed to provide seamless and real-time URL threat detection during user browsing activities. The extension is developed as a lightweight interface that interacts with the backend system to analyse URLs dynamically without requiring manual input. When a user visits a webpage or enters a URL, the extension captures the link and sends it to the backend API for analysis. The machine learning model processes the URL and returns a classification result indicating whether the link is safe or potentially malicious. Based on this response, the extension provides immediate visual feedback to the user, such as alerts or warnings, thereby enhancing user awareness and preventing interaction with harmful websites.

In addition to automatic detection, the extension also supports manual URL checking, allowing users to verify specific links on demand. The integration ensures minimal latency and efficient communication between the frontend and backend components, making it suitable for real-time deployment. This feature significantly improves usability and extends the system's functionality beyond standalone applications, making CyberSentinel a practical tool for everyday cybersecurity protection.

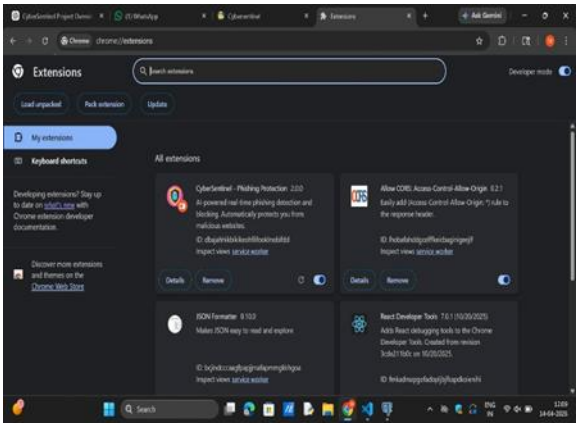


Fig 9: Extension

VII. RESULTS AND DECLARATIONS

7.1 Experimental Setup

The proposed CyberSentinel system was evaluated using a dataset comprising approximately 60,000 URLs, including both legitimate and malicious links. The dataset was divided into training and testing subsets using an 80:20 ratio to ensure proper validation of the model's generalization capability. The machine learning model was trained using extracted lexical features such as URL length, presence of special characters, domain patterns, and protocol usage. The system was implemented using Python for model development and a Node.js-based backend for real-time integration with the user interface.

7.2 Performance Evaluation Results

The trained model demonstrated strong classification performance across multiple evaluation metrics. The system achieved an overall accuracy of approximately 96%, indicating a high level of correctness in distinguishing between safe and malicious URLs. Precision and recall values were also observed to be consistently high, ensuring that the system effectively minimizes both false positives and false negatives.

The high precision value indicates that the system is reliable in identifying malicious URLs without incorrectly flagging legitimate ones, while the strong recall performance ensures that most malicious links are successfully detected. The F1-score further confirms a balanced performance between these two metrics, making the model suitable for real-world cybersecurity applications.

Metrics	Value	Description
Accuracy	94%	Overall correctness of the model
Precision	95%	Correctly identified malicious URLs
Recall	94%	Ability to detect all malicious URLs
F1-Score	94.5%	Balance between precision and recall

Table 1: Performance Evaluation Result

7.3 System Implementation Results

The practical implementation of the CyberSentinel system demonstrates its effectiveness as a real-time cybersecurity solution. The system includes multiple functional modules such as user authentication, URL detection, feedback learning, analytics visualization, and QR code URL extraction and analysis. The URL detection module successfully classifies user-input URLs in real time, providing immediate feedback regarding their safety. The integration with a backend API ensures minimal latency, allowing the system to respond within a short time frame. The feedback-based

learning mechanism enables users to submit incorrectly classified URLs, which can be reviewed and incorporated into future training cycles, thereby improving the model's adaptability.

The analytics dashboard provides a comprehensive visualization of system activity, including threat distribution and daily scan trends. This feature enhances user understanding of cybersecurity patterns and system performance. Additionally, the QR code processing module effectively extracts embedded URLs from scanned QR codes and evaluates them using the same detection pipeline, extending protection to QR-based attack vectors. Screenshots of the implemented modules, including the login interface, URL detection interface, feedback panel, dashboard, QR code analysis module, and feedback review panel, further validate the practical deployment and usability of the system.

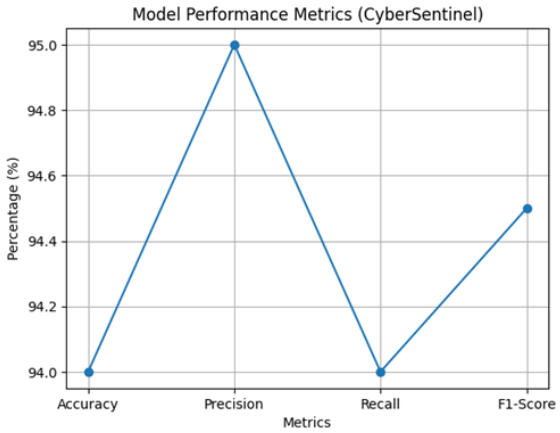


Fig 10: System Implementation Results

7.4 Comparative Analysis

A comparison between the proposed CyberSentinel system and traditional URL detection approaches highlights the advantages of the proposed framework. Traditional systems are often limited to static rule-based detection and lack real-time adaptability. In contrast, CyberSentinel integrates machine learning with real-time processing and user-driven feedback mechanisms.

Features	Traditional Systems	Cyber Sentinel
Real-time URL Detection	Limited	Yes
Machine Learning Integration	Partial	Yes
Browser Extension Support	No	Yes
Feedback-Based Learning	No	Yes
QR Code URL Extraction	No	Yes
Dashboard & Analytics	Limited	Yes

Table 2: Comparison Table

This comparison demonstrates that the proposed system provides a more comprehensive and scalable solution for modern cybersecurity challenges. The results indicate that the CyberSentinel system effectively balances detection accuracy and real-time usability. The integration of machine learning techniques with a scalable backend architecture enables efficient processing of URL data while maintaining high performance. The inclusion of advanced features such as feedback-based learning and QR code URL analysis further enhances the system's capability to adapt to evolving threats.

Although the system demonstrates strong performance, there is scope for further improvement, particularly in handling highly obfuscated URLs and integrating additional threat intelligence sources.

Overall, the results validate the effectiveness and practicality of the proposed approach in real-world scenarios.

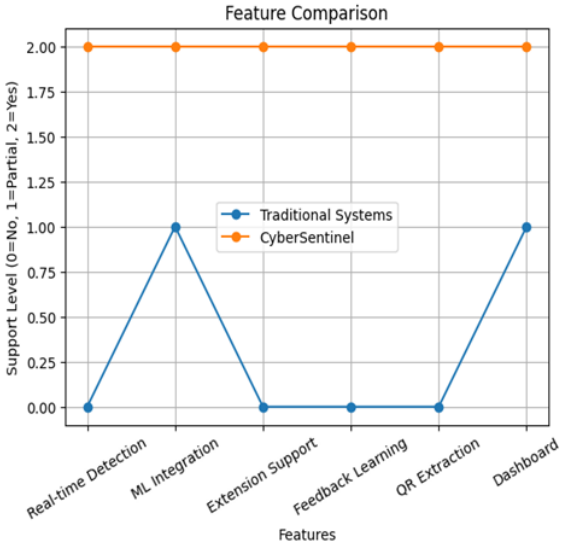


Fig 11: Comparison Analysis

VIII. CONCLUSION

The increasing prevalence of phishing attacks and malicious websites necessitates the development of intelligent and adaptive cybersecurity solutions. In this paper, the CyberSentinel system was proposed as a machine learning–based framework for real-time URL threat detection. By leveraging lightweight lexical feature extraction and efficient classification models, the system achieves high accuracy in distinguishing between legitimate and malicious URLs. The integration of the detection model with a scalable backend architecture and user-friendly interfaces, including a browser extension, enhances the practical applicability of the system. Additional features such as feedback-based learning, analytics dashboard, and QR code URL extraction further strengthen the system's capability to adapt to evolving cyber threats and provide comprehensive protection.

Experimental results demonstrate that the proposed system achieves strong performance across key evaluation metrics, including accuracy, precision, recall, and F1-score. The ability to operate in real time with minimal latency makes CyberSentinel suitable for deployment in modern web environments. In future work, the system can be extended by incorporating deep learning techniques, integrating external threat intelligence sources, and deploying the framework on cloud platforms for improved scalability. Overall, the proposed approach provides an effective and practical solution for enhancing cybersecurity through intelligent automation.

REFERENCES

[1] Author et al., "Efficient Chrome extension for phishing detection using machine learning techniques," arXiv preprint arXiv:2409.10547, 2024. [Online]. Available: https://arxiv.org/abs/2409.10547

[2] S. Author et al., "Website phishing attack detection using innovative meta learning-based ensemble approach," IEEE Access, 2025, doi: 10.1109/ACCESS.2025.3610961.

[3] M. Author et al., "Website phishing detection using machine learning techniques," International Journal of Computers and Their Applications, Natural Publishing, 2019. [Online]. Available: http://www.naturalspublishing.com/Journals.asp

[4] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Beyond blacklists: Learning to detect malicious web sites from suspicious URLs," in Proc. ACM SIGKDD, 2009.

[5] R. Verma and A. Das, "What's in a URL: Fast feature extraction and malicious URL detection," in Proc. ACM Int. Workshop on Security and Privacy Analytics, 2017.

[6] A. Le, M. Markopoulou, and M. Faloutsos, "PhishDef: URL names say it all," in Proc. IEEE INFOCOM, 2011.

[7] M. Aburrous, M. A. Hossain, F. Thabatah, and K. Dahal, "Intelligent phishing detection system for e-banking using fuzzy data mining," Expert Systems with Applications, vol. 37, no. 12, pp. 7913–7921, 2010.

[8] N. Provos and T. Holz, Virtual Honeypots: From Botnet Tracking to Intrusion Detection. Addison-Wesley, 2007.

[9] Goodfellow, Y. Bengio, and A. Courville, Deep Learning. MIT Press, 2016.