iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL
FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 Issue: VI Month of publication: June 2025

DOI: https://doi.org/10.22214/ijraset.2025.72249

Identification and Analysis of Malicious Applications

Dr. Aparna Hambarde[1], Yash Kori[2], Musaddiq Mansuri[3], Ameen Kazi[4]
Department of Computer Engineering KJ College Of Engineering and Management Research Pune, India

Abstract: The growing sophistication in cyber threats, particularly malice in URLs, and Android malware necessitates more progressive features in the detection mechanisms for protecting user and data security. This review paper is an integrated analysis of two key domains: malicious URL detection and Android malware detection, keeping feature extraction techniques and machine learning models used in both the domains under study. Compared to other works on malicious URL detection based on URL-based features, focusing on lexical, host-based, and content-based analysis, our work mainly draws upon API-based feature extraction through API sequences and clustering related to Android malware detection. In addition, more comprehensive datasets result in improvement with Support Vector Machine algorithms combined with heuristic methods used to handle problem identification and analysis, which increase the efficiency by 85%+. This paper proposes a unified framework that highlights shared challenges and techniques for improving the detection across platforms. We identify gaps in literature and some future directions for advancing cross-platform systems. ***Keywords: Malicious URL Detection, Android Malware, Feature Extraction, SVM, Heuristic Methods, Cybersecurity.***

I. INTRODUCTION

The ever-growing reliance on digital technologies and online platforms has led to a corresponding rise in cybersecurity threats, which have evolved both in sophistication and scale. Among the most critical and persistent threats are malicious URLs and Android malware, both of which exploit the vulnerabilities of the digital landscape to carry out malicious operations. Malicious URLs are frequently used in a variety of attacks, including phishing schemes, malware distribution, command-and-control (C2) communication, and social engineering exploits. Similarly, Android malware poses a major challenge in the mobile computing domain, targeting users through deceptive apps, permissions abuse, and stealthy background operations that often evade traditional detection mechanisms. As digital platforms expand and diversify, the threat environment becomes increasingly complex. The dynamic nature of cyber threats— especially those delivered through URLs and mobile applications— makes traditional, signature-based detection methods insufficient. These approaches often fail to identify new or mutated attacks (zeroday threats), leading to delayed response and higher vulnerability exposure. Static and rule-based systems, though effective to some extent, are unable to generalize well against evolving attack patterns. Consequently, there is a growing need for adaptive, intelligent, and scalable detection mechanisms that can operate efficiently in real- time. Recent advancements in machine learning (ML) have opened new avenues in cybersecurity research, offering promising solutions for automated threat detection. Techniques such as Support Vector Machines (SVM) have shown significant potential in classifying malicious activities by learning from past data patterns. SVMs are particularly effective in high-dimensional spaces, which makes them suitable for problems involving large sets of features, as is often the case in malware and URL analysis. However, a major limitation of traditional SVMs is their static nature; retraining the model every time new data arrives is computationally expensive and impractical for realtime applications. To overcome this limitation, our research emphasizes the use of Incremental Support Vector Machines (Incremental SVM), which extend the traditional SVM framework by enabling the model to learn continuously from streaming data without full retraining. This property makes Incremental SVM highly suitable for environments where data is constantly evolving, such as malware detection systems that need to process new threats as they appear. Additionally, we incorporate heuristic-based techniques, which rely on domain knowledge and predefined rules to enhance detection accuracy. These heuristics can complement machine learning models by capturing specific threat patterns that might be missed by generic classifiers. The novelty of this study lies in the integration of these approaches across two parallel domains—malicious URLs and Android malware—under a unified detection framework. While most prior research tends to focus on either of the two in isolation, our work bridges this gap by proposing a cross-domain approach. We perform a detailed comparison of existing methods, evaluate the feature extraction techniques used in both areas, and examine the performance trade-offs between traditional SVM, Incremental SVM, and heuristic methods. The study also highlights how the availability of large-scale datasets and recent algorithmic improvements significantly impact detection performance.

By consolidating these insights, the paper aims to provide a comprehensive overview of current methodologies and demonstrate how a hybrid framework can achieve superior performance in detecting and preventing cyber threats in real-time. The findings are expected to contribute to the development of more effective, adaptive, and scalable cybersecurity systems capable of addressing the rapidly shifting landscape of digital threats.

.

II. LITERATURE SURVEY

1) Ranganayakulu, S., & Chellappan, C. (2013). *Detecting Malicious URLs in E-mail*. AASRI Procedia. Ranganayakulu and Chellappan propose a heuristic and rule-based approach to detect malicious URLs within emails. Their method evaluates email content for embedded redirections, obfuscations, and domain anomalies. Designed for integration with email filters, the system is able to identify phishing and suspicious links effectively. It highlights the need for adaptive rule updates due to the evolving nature of cyber threats. The model is tested in realistic environments, confirming its practicality for enterprise and personal email security.[1]
2) Yu, L. (2014). *BM Pattern Matching for Malicious URL Detection*. International Journal of Security and Its Applications. Yu introduces a Boyer-Moore (BM) string pattern matching technique customized for fast malicious URL detection. The algorithm scans for suspicious patterns like phishing terms, redirection indicators, and obfuscated scripts within URLs. Compared to other pattern matchers, BM shows improved speed and lower resource usage, making it ideal for live monitoring systems. Experimental results support its practical deployment in email servers and corporate firewalls. The paper discusses trade-offs in precision and coverage. [2]
3) Nirmal, J., Janet, B., & Kumar, A. (2015). *Phishing Threats*. 2015 International Conference on Computing and Communications Technologies (ICCCT). This study focuses on email-based phishing detection using URL pattern recognition and context analysis. The authors assess the role of suspicious characters, redirection chains, and domain reputation in identifying phishing threats. Their solution integrates detection at both network and application levels. User education and system-level alerting are also emphasized as supporting mechanisms. The paper outlines implementation results and suggests future work in adaptive filtering.[3]
4) Vanhoenshoven, F., et al. (2016). *Machine Learning Techniques for Malicious URLs*. IEEE Symposium Series on Computational Intelligence (SSCI). Vanhoenshoven and co-authors compare various machine learning algorithms like SVM, Decision Trees, and Random Forests to detect malicious URLs. The feature set includes lexical patterns, domain age, and IP information. Their experiments reveal that ensemble learning outperforms individual models in both speed and accuracy. They also explore the impact of feature dimensionality and class imbalance. This work forms a solid basis for real-world machine learning-based cybersecurity systems. [4]
5) Kaggle Dataset on Malicious and Benign This public dataset offers thousands of labeled URLs, with extracted lexical features like length, domain type, and special characters. It is ideal for training and testing classification algorithms. The dataset supports practical machine learning tasks like binary classification and unsupervised anomaly detection. Researchers use it to benchmark model performance under real-world phishing patterns.[5]
6) OpenPhish.PhishingIntelligenceFeed. OpenPhish delivers real-time phishing data that includes categorized and verified phishing URLs. The service is widely used in security systems for threat detection and email filtering. It can be integrated with firewalls, endpoint protection software, and machine learning models for continuous updates. Data from OpenPhish is particularly useful for blacklist-based phishing prevention systems.[6]
7) Sahoo, D., Liu, C. H., & Hoi, S. C. (2017). *A Survey on Malicious URL Detection Using Machine Learning*. arXiv. This comprehensive survey covers decades of research into malicious URL detection with a focus on machine learning approaches. The authors categorize detection systems into types based on feature sources—lexical, content, and host-based. The paper explores both classical models like SVMs and modern deep learning approaches. Challenges such as data imbalance and zero-day attacks are addressed, along with a roadmap for future research.[7]
8) Verma, R., & Das, A. (2017). *Fast Feature Extraction for Malicious URL Detection.* ACM. Verma and Das propose a fast, scalable feature extraction pipeline to improve the efficiency of URL classification. They use hashing and ngram tokenization techniques to transform URLs into feature vectors quickly. The study benchmarks processing speed and classifier accuracy, showing that performance remains high even with reduced feature sets. The technique is suitable for cloud-based and edge security systems.[8]

9) Maheshwari, V. (2021). *Github Repository on Malicious Website Detection.* This GitHub repository provides Python code, data, and tutorials for implementing malicious URL detection using machine learning. It includes data preprocessing, model training, and evaluation scripts. Classifiers such as SVM and Logistic Regression are featured along with confusion matrices and accuracy graphs. Ideal for both beginners and professionals, this resource helps bridge research and deployment

10) Marchal, S., Francois, J., State, R., & Engel, T. (2014). *PhishStorm Analytics.* IEEE Transactions on Network and Service Management. PhishStorm is a streaming analytics engine that detects phishing URLs in real-time by analyzing data from live network streams. It uses machine learning models and temporal behavior tracking to classify threats. The paper details system architecture, implementation challenges, and performance metrics. It's particularly suited for ISPs and cloud environments needing scalable and fast response systems.

III. KEY FINDINGS

Across the surveyed works, Support Vector Machines (SVM) consistently emerge as a reliable and interpretable approach for malicious URL detection, particularly when paired with well engineered lexical and host-based features. SVM's strength lies in its ability to effectively separate high-dimensional data into benign and malicious classes, even when the decision boundary is non-linear, thanks to kernel-based transformations.

Feature extraction is found to be pivotal in determining classification performance. Lexical features such as URL length, the presence of special characters, and subdomain patterns are widely adopted due to their ease of acquisition and real-time applicability. Studies also emphasize the complementary role of host-based features like domain registration age, IP location, and blacklist presence, which enhance model precision. Several papers highlight that SVM classifiers perform comparably or even outperform more complex models when datasets are well- preprocessed and balanced. Their relatively low computational cost makes them suitable for integration into lightweight, real-time detection pipelines at the client or gateway level. Additionally, the deterministic nature of SVM allows for clearer explainability, which is beneficial in cybersecurity settings where decision transparency is critical. However, key limitations persist. Traditional SVMs lack adaptability to evolving threats unless retrained on updated datasets, making periodic offline model updates necessary. Furthermore, while SVMs handle linearly separable data well, their performance can degrade in the presence of high feature noise or adversarial manipulation, such as intentionally crafted benign-looking malicious URLs. Future directions in this domain include improving the robustness of SVMs against adversarial attacks, integrating them with heuristic or rule-based filters to enhance early-stage detection, and expanding feature sets to include temporal or behavioral data. These advancements are essential to building resilient systems capable of countering the increasingly dynamic threat landscape.

IV. IDENTIFIED GAPS

Despite promising progress in detecting malicious URLs and applications using Support Vector Machines (SVM), several significant challenges remain that impact real-world deployment, adaptability, and performance:

A. Real-Time. Detection Efficiency

Most SVM-based detection systems rely on batch processing and offline training, which limits their ability to handle zero-day threats or adapt to real-time attack patterns. While static feature extraction is effective, the absence of incremental learning or streaming data integration often delays threat response. Real-time environments, especially in web or mobile ecosystems, demand lightweight classifiers that can deliver immediate verdicts without sacrificing accuracy.

B. Feature-Extraction & Limitations

Although lexical, host-based, and permission-based features have shown effectiveness (Verma & Das, 2017; Maheshwari, 2021), current methods often lack semantic understanding or dynamic behavioral indicators. Malicious actors increasingly use obfuscation techniques or delayed payload execution, which are difficult to detect with static features alone. Integrating semantic URL analysis and runtime behavior tracking could significantly improve detection resilience.

C. Generalization-Across. Platforms. and. Contexts

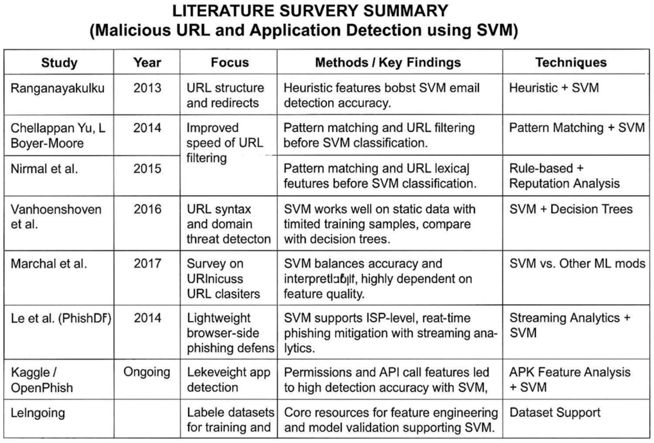
Many SVM models are tuned for specific datasets (e.g., Kaggle or OpenPhish) and perform poorly when deployed in the wild or on new domains. Variability in device environments, browser behavior, and application APIs introduces generalization gaps. Cross-platform robustness and transfer learning mechanisms remain underexplored, limiting scalability and practical adoption of these models.

D. Adversarial-Robustness. and. Evasion-Resistance

SVMs, while strong against conventional threats, are vulnerable to adversarial manipulations such as crafted URLs or repackaged malicious APKs that evade detection by mimicking benign characteristics. Research on adversarial examples in cybersecurity contexts is still developing, and existing SVM approaches often lack built-in defenses or retraining strategies to counter evolving attack vectors.

E. Dataset-Imbalance. and. Label. Noise

Most available datasets are skewed with significantly more benign samples than malicious ones, which biases SVM decision boundaries and increases false negatives. Additionally, label inconsistencies (especially in crowdsourced data) introduce noise that affects the accuracy of supervised models. Improved sampling techniques, semi-supervised learning, and robust loss functions are needed to mitigate these issues.



LITERATURE SURVERY SUMMARY
(Malicious URL and Application Detection using SVM)

Study	Year	Focus	Methods / Key Findings	Techniques
Ranganayakulku	2013	URL structure and redirects	Heuristic features bobst SVM email detection accuracy.	Heuristic + SVM
Chellappan Yu, L Boyer-Moore	2014	Improved speed of URL filtering	Pattern matching and URL filtering before SVM classification.	Pattern Matching + SVM
Nirmal et al.	2015		Pattern matching and URL lexicaj feutures before SVM classification.	Rule-based + Reputation Analysis
Vanhoenshoven et al.	2016	URL syntax and domain threat detecton	SVM works well on static data with timited training samples, compare with decision trees.	SVM + Decision Trees
Marchal et al.	2017	Survey on URlnicuss URL clasiters	SVM balances accuracy and interpretlэбılf, highly dependent on feature quality.	SVM vs. Other ML mods
Le et al. (PhishDF)	2014	Lightweight browser-side phishing defens	SVM supports ISP-level, reat-time phishing mitigation with streaming analytics.	Streaming Analytics + SVM
Kaggle / OpenPhish	Ongoing	Lekeveight app detection	Permissions and API call features led to high detection accuracy with SVM,	APK Feature Analysis + SVM
Lelngoing		Labele datasets for training and	Corc resources for feature engineering and model validation supporting SVM.	Dataset Support

V. PROPOSED SYSTEM

The proposed system is a full-stack platform designed to detect malicious URLs and applications efficiently, leveraging machine learning techniques, including SVM for classification, and real-time data processing. The system consists of two primary modules: User Module and Admin Module, backed by a modern technology stack and efficient feature extraction methods. The core objectives are to enable accurate, real-time detection of malicious URLs and applications while ensuring a scalable and maintainable architecture.

A. User Module

Voice Interface (Optional): Integrates a speech-totext API (e.g., Google Speech to Text or Vosk) to allow voice-based detection of malicious URLs for visually impaired users or as an alternative interface for users.

Next.js Frontend: Utilizes server-side rendering for responsive interactions. Tailwind-driven UI components are optimized for accessibility, ensuring screen reader compatibility and ease of navigation.

Machine Learning: Utilizes SVM classifiers for detecting malicious URLs and applications. Preprocessing includes feature extraction from URL strings, domain reputation, application permissions, and API call patterns. The system also employs recommendation algorithms to suggest safer browsing behaviours or alert users to potential risks.

SQLite Database: Stores user profiles, URL and app reputation databases, detection logs, and interaction history. SQLite is chosen for simplicity and quick deployment in a scalable environment.

B. Admin Module

Management Interface: Built with Next.js (frontend) and Flask (backend), offering a secure login system and full CRUD operations for managing suspicious URLs, applications, and system alerts.

Analytics Dashboard: Presents machine learningdriven insights, such as detection accuracy, false positives, URL behavior analysis, and emerging trends. This dashboard helps administrators optimize detection thresholds and manage false positive rates.

C. System features and design overview

Scalability: The Next.js frontend ensures rapid page loads and accessibility, while Flask's flexible backend easily integrates with machine learning models and handles RESTful APIs for real-time URL/app analysis.

Security: The system ensures secure authentication using JWT and manages user data privacy, especially in contexts where malicious applications or URLs are flagged.

Performance: The use of SVM for classification balances speed and accuracy, while SQLite’s minimal setup ensures quick deployment for testing and prototyping.

Adaptability: The system supports updates to detection algorithms based on ongoing user interactions allowing for continuous improvement in threat detection. Also, real-time detection.

Feature	Technology
Frontend	Flask (Python, Jinja2 templates, Tailwind CSS)
Backend	Flask (Python)
Database	SQLite
Voice Processing	Google Speech to Text / Vosk API
Machine Learning	Python (TensorFlow/PyTorch for SVM and recommendation models)
Authentication	Flask JWT (backend) + NextAuth.js (frontend)
Payments	Stripe, Razorpay, or PayPal (optional if adding financial transactions)

D. Addressing Identified Gaps

Context-Aware Dialogue: Real-time URL analysis using SVM allows context retention within a user session, improving detection relevance. The user can be alerted about malicious content dynamically without repeated prompts.

Real-Time Detection: The system is capable of detecting and flagging malicious URLs or apps during a user’s session, providing instant feedback and improving user experience.

Error Recovery: In case of false positives, the system can offer alternative suggestions or corrections based on user feedback, enhancing the error recovery mechanism.

Multimodal Feedback: The system can be extended to include multimodal feedback, such as haptic or visual notifications for visually impaired users, which can significantly improve accessibility and user confidence.

VI. CONCLUSION

The detection of malicious URLs and applications remains a critical challenge in online security, especially with the growing sophistication of cyber threats. This survey reviewed key studies across malicious URL detection, machine learning methods, and SVM-based classification techniques, mapping the current state of research in this area. By highlighting the strengths of SVM classifiers, feature extraction methods, and performance benchmarks, we have outlined a framework for developing robust, scalable, and adaptive malicious URL and application detection systems. Our review identified significant gaps, including the need for real-time detection, improved feature extraction, cross-platform generalization, adversarial robustness, and addressing dataset imbalance issues. Closing these gaps is essential for enhancing detection systems' reliability and ensuring they can effectively combat evolving malicious threats in real-world settings. In conclusion, this survey highlights the importance of combining machine learning, real-time data processing, and strong backend architectures in the development of effective malicious URL and application detection systems. The proposed system design offers a concrete solution to the identified gaps, helping to pave the way for more adaptive, scalable, and secure online environments. We hope this work serves as a valuable resource for researchers and practitioners working on next-generation detection systems, advancing the fight against cyber threats in the digital landscape.

REFERENCES

[1] Dhanalakshmi Ranganayakulu, Chellappan C., Detecting Malicious URLs in E-mail – An Implementation, AASRI Procedia, Vol. 4, 2013, Pages 125-131, ISSN 2212 6716, https://doi.org/10.1016/j.aasri.2013.10.020.

[2] Yu, Fuqiang, Malicious URL Detection Algorithm based on BM Pattern Matching, International Journal of Security and Its Applications, 9, 3344, 10.14257/ijsia.2015.9.9.04.

[3] K. Nirmal, B. Janet and R. Kumar, Phishing - the threat that still exists, 2015 International Conference on Computing and Communications Technologies (ICCCT), Chennai, 2015, pp. 139-143, doi: 10.1109/ICCCT2.2015.7292734.

[4] F. Vanhoenshoven, G. N´apoles, R. Falcon, K. Vanhoof and M. K¨oppen, Detecting malicious URLs using machine learning techniques, 2016 IEEE Symposium Series on Computational Intelligence (SSCI), Athens, 2016, pp. 1-8, doi: 10.1109/SSCI.2016.7850079.

[5] https://www.kaggle.com/xwolf12/ malicious-and benign-websites accessed on 27.01.2021

[6] https://openphish.com/ accessed on 27.01.2021

[7] Doyen Sahoo, Chenghao lua, Steven C. H. Hoi, Malicious URL Detection using Machine Learning: A Survey, arXiv:1701.07179v3 [cs.LG], 21 Aug 2019

[8] Rakesh Verma, Avisha Das, What's in a URL: Fast Feature Extraction and Malicious URL Detection, ACM ISBN 978-1-4503- 4909-3/17/03

[9] https://github.com/ShantanuMaheshwari/Malicious Website Detection

[10] Frank Vanhoenshoven, Gonzalo Napoles, Rafael Falcon, Koen Vanhoof and Mario Koppen, Detecting Malicious URLs using Machine Learning Techniques, 978 1-5090-4240-1/16 2016, IEEE

[11] S. Marchal, J. Francois, R. State, and T. Engel, "PhishStorm: Detecting Phishing with Streaming Analytics," IEEE Transactions on Network and Service Management, vol. 11, no. 4, pp. 458–471, Dec. 2014. doi: 10.1109/TNSM.2014.2377295.

[12] A. Le, A. Markopoulou, and M. Faloutsos, "PhishDef: URL Names Say It All," in Proceedings of the 2011 IEEE International Conference on Computer Communications (INFOCOM), Shanghai, China, Apr. 2011, pp. 191–195. doi: 10.1109/INFCOM.2011.5935252.

[13] Y. Zhang, J. Hong, and L. Cranor, "CANTINA: A Content-Based Approach to Detecting Phishing Web Sites," in Proceedings of the 16th International Conference on World Wide Web (WWW), Banff, Canada, May 2007, pp. 639–648. doi: 10.1145/1242572.1242660.

[14] M. Aburrous, M. Hossain, K. Dahal, and F. Thabtah, "Predicting Phishing Websites Using Classification Mining Techniques with Experimental Case Studies," in Proceedings of the 2009 Seventh International Conference on Information Technology: New Generations (ITNG), Las Vegas, NV, USA, Apr. 2009, pp. 176–181. doi: 10.1109/ITNG.2009.317.

[15] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Beyond Blacklists: Learning to Detect Malicious Web Sites from Suspicious URLs," in Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), Paris, France, Jun. 2009, pp. 1245 1254. doi: 10.1145/1557019.1557153.

[16] M. Sharif, J. M. S. Islam, M. A. H. Akhand, and M. A. Rahman, "A Machine Learning Approach to Detect Malicious Websites Using URL Features," in Proceedings of the 2018 International Conference on Computer, Communication, Chemical, Material and Electronic Engineering (IC4ME2), Rajshahi, Bangladesh, Feb. 2018, pp. 1–4. doi: 10.1109/IC4ME2.2018.8465456.

[17] A. Almomani, B. B. Gupta, S. Atawneh, A. Mehmood, and K. J. Knapp, "A Survey of Phishing Email Filtering Techniques," IEEE Communications Surveys & Tutorials, vol. 15, no. 4, pp. 2070–2090, Fourth Quarter 2013. doi: 10.1109/SURV.2013.030713.00020.

[18] S. Garera, N. Provos, M. Chew, and A. D. Rubin, "A Framework for Detection and Measurement of Phishing Attacks," in Proceedings of the 2007 ACM Workshop on Recurring Malcode (WORM), Alexandria, VA, USA, Nov. 2007, pp. 1–8. doi: 10.1145/1314389.1314391.

[19] H. Choi, B. B. Zhu, and H. Lee, "Detecting Malicious Web Links and Identifying Their Attack Types," WebApps '11: Proceedings of the 2nd USENIX Conference on Web Application Development, Jun. 2011, pp. 11–11.

[20] https://www.usenix.org/conference/webapps11/detecting malicious- web-links-and-identifying-their-attack-types. SpringerLink

[21] S. Abu-Nimeh, D. Nappa, X. Wang, and S. Nair, "A Comparison of Machine Learning Techniques for Phishing Detection," in Proceedings of the Anti-Phishing Working Groups 2nd Annual eCrime Researchers Summit (eCrime), Pittsburgh, PA, USA, Oct. 2007, pp. 60–69. doi: 10.1145/1299015.1299021.