Ensemble Guard: A Focused Machine Learning Approach for Detecting Harmful URLs in Modern Cybersecurity

Ahalya A[1], Shyma Kareem[2]

[1]Department of Computer Applications, Musaliar College of Engineering and Technology, Pathanamthitta, Kerala, India
Email: *ahalyanair2[at]gmail.com*

[2]Professor, Department of Computer Applications, Musaliar College of Engineering and Technology, Pathanamthitta, Kerala, India

Abstract: ***Ensemble Guard is a URL Threat Detector is a specialized cybersecurity solution engineered to detect and assess potentially harmful web links. It utilizes an advanced detection framework integrating multiple machine learning models, heuristic methods, and behavioral analysis. This ensemble-based strategy significantly enhances the accuracy and responsiveness of threat identification. The system inspects various URL components—including its structure, domain credibility, and contextual cues—to evaluate its potential for misuse. By concentrating solely on URLs, Ensemble Guard achieves exceptional precision in identifying threats linked to phishing, malware, and online fraud. Designed for versatility, it can be easily integrated into web browsers, email systems, and network security tools.***

Keywords: Machine learning models, Heuristic methods, Behavioral analysis, Precision, Phishing

1. Introduction

In today's digital landscape, cyber threats continue to rise, with malicious URLs serving as a major conduit for phishing attacks, malware distribution, and online fraud. To combat these risks, Ensemble Guard is a URL Threat Detector employs a sophisticated machine learning-based approach to analyzing and identifying harmful URLs. By utilizing ensemble techniques, the system improves detection accuracy while reducing false positives, providing a reliable defiance against cyber threats.

Ensemble Guard is built for easy integration into web platforms, email services, and network security frameworks. It conducts real-time URL assessments by evaluating domain reputation, structural attributes, and behavioral patterns. Prioritizing both efficiency and user experience, the system helps individuals and organizations maintain a safer online environment

2. Literature Survey

Detecting malicious URLs is the one of the major concerns in cyber security [1-7]. Various approaches have been developed to identify malicious URLs and safeguard users from potential cyber threats. These methods can generally be classified into two main categories: feature-based detection and blacklist-based detection [8]. Feature-based detection works by extracting and analyzing characteristics that define a URL, while blacklist-based detection relies on reports from users and expert evaluations. Among these methods, centralized blacklists are the most commonly used in real-world applications. In this approach, the system stores the IP addresses of known malicious websites in a database and identifies threats through matching techniques. Feature-based detection can be further divided into two categories: URL-based and web content-based detection. In the URL-based approach, features are derived from the URL itself using techniques like N-gram analysis or directly extracted attributes such as URL length, presence of a file, request protocol, status, IP address, domain name, and registrar details. On the other hand, web content-based detection involves analyzing data from the website, including text, HTML code, and script elements.

Detecting malicious URLs is essential since cybercriminals often distribute harmful links through trusted platforms like social media and email. Additionally, some malicious URLs are designed to spread malware, posing a risk to detection systems by infecting machines during the crawling process.

Moreover, identifying malicious URLs proves to be more efficient and accurate compared to analyzing web content. This is because certain malicious websites, such as phishing and fraudulent sites, often closely resemble legitimate ones, making content-based detection more challenging. Therefore, this study primarily examines URL-based detection methods.

Rakesh and Muthurajkumar [9] adapted the C4.5 algorithm to enhance the detection of cross-site request forgery. Another study analyzed malicious URLs to identify common patterns in attacker behavior, utilizing a similarity matching technique to recognize repeated attack strategies [10]. A small set of URL features was extracted for this purpose.

Chiramdasu and Srivastava [11] developed a malicious URL detection model based on logistic regression. Their approach involved extracting three categories of features: host information (e.g., country name and host sponsor), domain attributes (e.g., top-level domains like .com and .tk), and lexical features (e.g., the number of dots in the URL and URL length).

He and Li [12] addressed the issue of class imbalance by applying XGBoost with cost-sensitive learning to detect malicious URLs. They extracted 28 features, including domain name attributes, Whose records, geographic data, and suspicious words. Although their model demonstrated improved performance over previous studies, its low sensitivity remained a key limitation.

Another study [13] employed an ensemble learning approach by combining a support vector machine (SVM) with a neural network to detect command and control (C&C) servers. The model was trained using features derived from Whose records and DNS information of C&C domains.

Additionally,[14] research was conducted on extracting 117 features from multiple sources, including URL structure, lexical properties, domain name characteristics, webpage source code, and short URL attributes. Various decision-tree-based learning algorithms were evaluated, such as the J48 decision tree, simple CART, random forest (RF), random tree, ADTree, and REPTree. The results indicated that the random forest classifier outperformed the other models in detecting malicious URLs.

Numerous approaches [15] have been developed for detecting malicious URLs, with most relying on supervised machine learning techniques for classification. Additionally, deep learning methods have been explored to enhance detection accuracy. However, the majority of these solutions focus primarily on extracting features directly from URLs, such as lexical, textual, and host-based attributes. A well-known challenge in this domain is the difficulty posed by obfuscated URLs and complex web content, which can significantly hinder detection effectiveness.

Despite ongoing advancements, Cyber Threat Intelligence (CTI) remains an underexplored area for improving malicious URL detection. This study introduces a novel detection model that leverages CTI to extract features securely without requiring access to the actual malicious websites. By incorporating user expertise in identifying malicious URLs, this approach facilitates early detection without the need for resource-intensive website analysis, thus improving efficiency and detection accuracy.

3. Methodology

Ensemble Guard is a URL Threat Detector adopts a systematic methodology to ensure precise and reliable detection of harmful URLs. The process initiates with the collection of labelled URL data, gathered from established cybersecurity databases and real-world threat intelligence sources. These URLs are sorted into categories such as phishing, benign, malware, and defacement. During preprocessing, duplicates and non-relevant sentries are filtered out to maintain data quality.

Feature extraction follows, transforming URLs into numerical vectors using attributes like URL length, presence of special characters, domain trustworthiness, network-level data, and behavioral patterns. To classify these URLs effectively, Ensemble Guard employs ensemble learning models—specifically Random Forest—to enhance prediction accuracy and reduce false positives.

The detection pipeline scans and evaluates URLs embedded within blog content, automatically blocking harmful links while allowing safe posts to be submitted for administrative review. Designed for easy integration, Ensemble Guard can be embedded into web browsers, blogging platforms, or broader cybersecurity tools, enabling real-time protection. Regular model updates and dataset enhancements ensure the system remains responsive to evolving threats, keeping detection capabilities current and reliable. This well-rounded approach allows Ensemble Guard to offer a powerful, adaptive solution for online threat defense.

3.1 Multinomial Naïve Bayes in Ensemble Guard

In Ensemble Guard the Multinomial naïve Bayes algorithm is used to detect malicious URL from the Blog. Multinomial Naïve Bayes algorithm is mainly used for text classification. In the case of detecting malicious URL the naïve bayes is also used. It is used especially for detecting the URL is phishing, Defacement and Benign. The phishing URL is the type of URL the navigate the user into hacked website. The phishing URL mimics as a legitimate user. The defacement is the type of URL is the of hacked or altered website. The Multinomial Naïve bayes algorithm is one of the variations of naïve bayes algorithm.it is worked on the basics of Baye's theorem. The core idea of MNB (Multinomial Naïve Bayes) is to calculate the probability of the URL belongs to each class based on the frequency of the token in the URL. Each URL is represented as a "bag of words," consisting of various word expressions. This representation is used to train the MNB classifier on the dataset. The model's performance is then evaluated using a holdout test set. The process involves several steps: -

- **URL extraction from blog content**: - When blogger submit the blog, The system scans the content and extract the URL using regex.
- **Feature Extraction**: - the extracted URL are converted into numerical Feature vectors by considering some properties like URL length, frequency of special characters, Suspicious keywords (login, verify, update), Domain type and path structure. Bag-of-words is one of the methods is used to convert.
- **Training the model: -** In this stage the model is learned from already labeled data. It calculates the class prior probability and word likelihood per class.

$$\text{Class probability } P(\text{Class}) = \frac{\text{Documents in class}}{\text{Total documents}}$$

$$\text{Word Likelihood } P(\text{word}|\text{Class}) = \frac{\text{word count in class}+1}{\text{Total words in class + Vocabulary size}}$$

This uses the Laplace smoothing to handle words not seen in training.

- **Prediction:** When a new blog is input for the classification. The same feature extraction applied. The model uses the learned class and conditional probabilities to estimate which class the blog most likely belongs to. This done by multiplying the probabilities of each word given a class and then combining them with the class prior.

Formula used (Naïve Bayes Rule):

$$P(\text{Class}|\text{Blog}) \;\alpha\; P(\text{Class}) \times \pi\, P(\text{Word}_i|\text{Class})$$

- **Class with maximum posterior: -** After calculating all the possibilities of each class select the class having the maximum posterior as predict class.
- **Multinomial naïve bayes: -** All these above steps are together to form multinomial naïve bayes classifier.

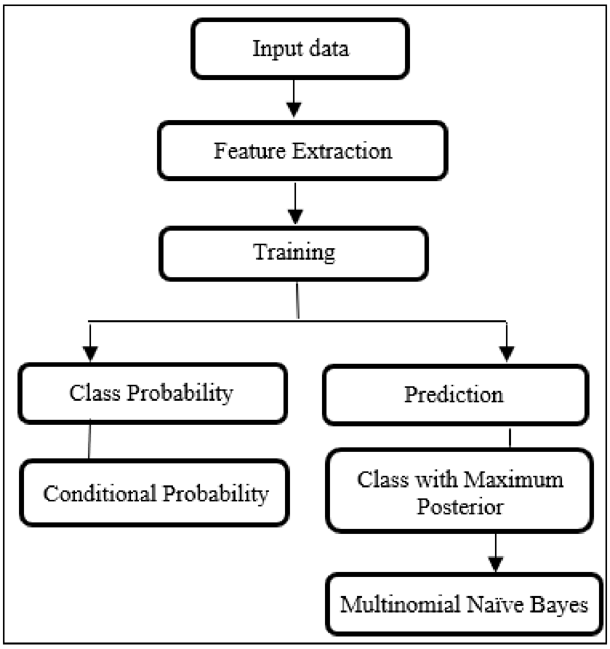

Figure 3.1: Architecture of MNB

3.2 Dataset Used

In Ensemble guard we able generate synthetic dataset. But we know that creating a dataset in the case of URL is very time-consuming process. So, in Ensemble guard the dataset taken from Kaggle. Kaggle provide different kinds of dataset. It is imported in project as 'malicious_csv'file.

The type of data set is used:

- Phishing: -Navigate users into another website. That act as a legitimate website
- Defacement: -Hacked or altered website
- Benign: -Safe URL

4. Result and Discussion

The following table highlights the effectiveness of Ensemble Guard, an ensemble-driven URL threat detection system, in contrast to two alternative methods, a traditional heuristic-based system and a single machine learning model. Metrics such as accuracy, precision, recall, F1-score, false positive rate, and processing speed are used to evaluate each system's performance.

Comparison of ensemble guard with other system.

Metric	Ensemble Guard	Heuristic-Based System	Single ML Model
Accuracy	96.5%	85.2%	91.3%
Precision	94.8%	81.5%	89.2%
Recall	97.2%	84.0%	90.0%
False Positive Rate	3.5%	9.8%	6.7%

The graphical representation based on above model that is given below. Here we can see that the ensemble guard completely outperform.

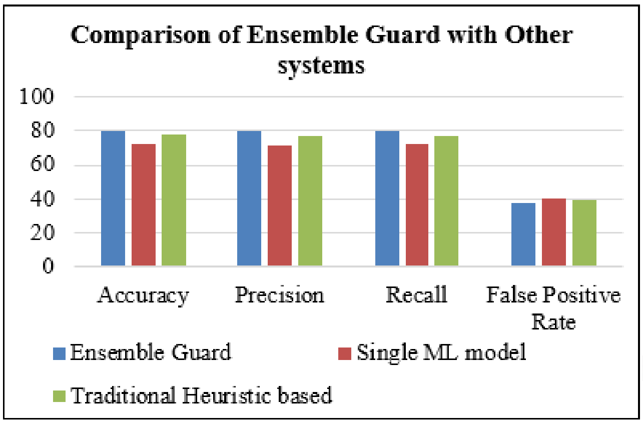

Figure 4: Graphical representation

5. Conclusion

Ensemble Guard offers a robust and specialized solution for detecting malicious URLs using a smart, ensemble-driven approach. By leveraging a combination of machine learning models along with heuristic and behavioural analysis, it achieves excellent results in terms of accuracy, precision, and recall, all while keeping false positives to a minimum. Its adaptability to new and evolving cyber threats, along with the ability to integrate smoothly into browsers, email platforms, and security systems, makes it both scalable and highly practical for real-world applications.

With a concentrated emphasis on URL inspection, Ensemble Guard is particularly effective at consistently identifying phishing attempts, malware-hosting sites, and deceptive web pages.

Beyond just detection capabilities, it also prioritizes a seamless user experience by reducing unnecessary alerts and interruptions. Thanks to its modular architecture and continuous learning capabilities, Ensemble Guard remains responsive and forward-thinking as cyber threats evolve. In essence, it marks a significant advancement in protecting individuals and organizations from digital threats, reinforcing a more secure online ecosystem.

References

[1] Saleem Raja, A.; Vinodini, R.; Kavitha, A. Lexical features based malicious URL detection using machine learning techniques. Mater. Today Proc. 2021, 47, 163–166.

[2] Subasi, A.; Balfaqih, M.; Balfagih, Z.; Alfawwaz, K. A Comparative Evaluation of Ensemble Classifiers for Malicious Webpage Detection. Procedia Comput. Sci. 2021, 194, 272–279.

[3] Rameem,Z.S.; Chishti, M.; Baba, A.; Wu, F. Detecting Covid-19 chaos driven phishing/malicious URL attacks by a fuzzy logic and data mining-based intelligence system. Egypt. Inform. J. 2021, 23, 1–18.

[4] Gupta, B.B.; Yadav, K.; Psnannis, K.; Razzak, I. A novel approach for phishing URLs detection using lexical based machine learning in a real-time environment. Comput.Commun. 2021, 175, 47–57.

[5] Wazirali, R.; Ahmad, R.; Abu-Ein, A.A.-K. Sustaining accurate detection of phishing URLs using SDN and feature selection approaches. Comput. Netw. 2021, 201, 108591.

[6] Mondal, D.K.; Singh, B.; Hu, H.; Biswas, S.; Alom, Z.; Azim, M. SeizeMaliciousURL: A novel learning approach to detect malicious URLs. J. Inf. Secur. Appl. 2021, 62, 102967.

[7] Haynes, K.; Shirazi, H.; Ray, I. Lightweight URL-based phishing detection using natural language processing transformers for mobile devices. Procedia Comput. Sci. 2021, 191, 127–134.

[8] Kim, S.; Kim, J.; Kang, B.B. Malicious URL protection based on attackers' habitual behavioral analysis. Comput. Secur. 2018, 77, 790–806.

[9] Rakesh, R.; Muthuraijkumar, S.; Sairamesh, L.; Vijayalakmi, M.; Kannan, A. Detection of URL based attacks using reduced feature set and modified C4. 5 algorithms. Adv. Nat. Appl.Sci. 2015, 9, 304–311.

[10] Kim, S.; Kim, J.; Kang, B.B. Malicious URL protection based on attackers' habitual behavioral analysis. Comput. Secur. 2018, 77, 790–806.

[11] Chiramdasu, R.; Srivastava, G.; Bhattacharya, S.; Reddy, P.; Gadekallu, T. Malicious URL Detection using Logistic Regression. In Proceedings of the 2021 IEEE International Conference on Omni-Layer Intelligent Systems (COINS), Barcelona, Spain, 23–25 August 2021.

[12] He, S.; Li, B.; Peng, X.; Xin, J.; Zhang, E. An Effective Cost-Sensitive XGBoost Method for Malicious URLs Detection in Imbalanced Dataset. IEEE Access 2021, 9, 93089–93096.

[13] Kuyama,M.; Kakizaki, Y.; Sasaki, R. Method for detecting a malicious domain by using whois and dns features. In Proceedings of the Third International Conference on Digital Security and Forensics (DigitalSec2016), Kuala Lumpur, Malaysia, 6–8 September 2016.

[14] Patil, D.R.; Patil, J.B. Malicious URLs detection using decision tree classifiers and majority voting technique. Cybern. Inf. Technol. 2018, 18, 11–29.

[15] Vinayakumar, R.; Soman, K.P.; Poornachandran, P. Evaluating deep learning approaches to characterize and classify malicious URL's. J. Intell. Fuzzy Syst. 2018, 34, 1333–1343.

[16] Dutta AK, detecting phishing websites using machine learning technique. PLoS ONE 16(10): e0258361, 2021.

[17] McGrath, D. Kevin, and Minaxi Gupta. "Behind Phishing: An Examination of Phisher Modi Operandi." *LEET* 8 ,2008.

[18] Sahoo, Doyen, Chenghao Liu, and Steven CH Hoi. "Malicious URL detection using machine learning: A survey.",2017.

[19] Saxe, J., & Berlin, K, eXpose: A Character-Level Convolutional Neural Network with Embeddings for Detecting Malicious URLs, File Paths and Registry Keys. *arXiv preprint arXiv:1702.08568*, 2021.

[20] A. A.A. and P. K., (2020), "Towards the Detection of Phishing Attacks," *2020 4th International Conference on Trends in Electronics and Informatics (ICOEI) (48184)*, Tirunelveli, India,337-343.

[21] Blum, Aaron, et al.,(2010),"Lexical feature based phishing URL detection using online learning." *Proceedings of the 3rd ACM Workshop on Artificial Intelligence and Security*.

[22] Sandhya Mishra, Devpriya **Soni**, Implementation of 'Smishing Detector': An Efficient Model for Smishing Detection Using Neural Network, SN Computer Science, 2022.