PHISHING WEBSITE DETECTION USING MACHINE LEARNING

[1] **Mrs.G.PRIYANKA,[2] VANGALA BHANU , [3] BASU NAVEEN KUMAR , [4] DANDUGULA PRASHANTH ,[5] JUKANTI SATHVIKA**
[1]Assistant Professor, Department of IT, Sri Indu College of Engineering & Technology, Hyderabad.
[2,3,4,5] U.G. Scholor, Department of IT, Sri Indu College of Engineering & Technology, Hyderabad.

Abstract: *Phishing is a type of cyberattack in which attackers design fraudulent websites that closely imitate legitimate platforms to deceive users into revealing sensitive information such as login credentials, banking details, or personal data. Recent studies highlight the effectiveness of machine learning (ML) techniques in detecting phishing websites by analyzing features such as URL structures, webpage content, and domain-related information. Although these methods achieve high accuracy, they often lack real-time protection and seamless integration into user-friendly applications. To overcome these limitations, this project proposes a browser extension powered by a machine learning model that can detect phishing attempts during web browsing. The system extracts important features from web pages and processes them through a lightweight yet efficient ML classifier, which provides instant alerts to users when a potential threat is identified. The model is trained on a balanced dataset containing both legitimate and phishing websites to ensure reliable and consistent performance. By combining real-time detection with an easy-to-use interface, the proposed solution enhances user safety and provides an effective defense mechanism against phishing attacks in everyday internet usage.*

Keywords: Machine learning, Phishing, Website detection

I. INTRODUCTION

The phishing website detection system using machine learning is easy to use and doesn't require any technical knowledge. All you need to do is enter the website's URL into the tool, and it will quickly analyse it to tell you if it's safe or a phishing attempt. The system uses smart machine learning algorithms to identify even new phishing websites (called zero-hour attacks) that haven't been reported yet, making it more reliable than traditional methods like blacklists. It works automatically in the background, so users don't have to worry about complicated steps. The tool can also be added to browsers or antivirus software, offering real-time protection while you browse online. Some versions even explain why a website was flagged as phishing, helping users learn how to spot scams themselves. Whether on a phone, computer, or as a web app, this system makes detecting phishing websites fast, simple. Phishing is a big problem today because it's easy for attackers to create fake websites that look just like real ones. While experts can usually tell the difference, many regular users can't, and they end up being tricked. The main goal of phishing is to steal personal information, like bank account details. In the United States, businesses lose about $2 billion every year because of phishing. According to a report by Microsoft in 2014, phishing causes losses of up to $5 billion worldwide every year. The project's primary objective is to develop a system that integrates these key components, enabling real-time tracking of individuals within a crowd and counting the number of people in a crowd. The synergy between accurate person detection, continuous tracking, and counting represents a significant advancement in crowd analysis and offers a wide range of practical applications, including improving public safety, enhancing security, and aiding urban planning. This research project aims to provide a comprehensive and efficient solution to address the challenges encountered in real-world crowd analysis scenarios, ultimately contributing to safer and more efficiently managed public spaces.

II. METHODOLOGY

The process of detecting phishing websites using machine learning starts by gathering a dataset containing both genuine and malicious websites. Important characteristics like URL structure, domain registration details, and security features are extracted from each site. These attributes are then cleaned and prepared for training a machine learning algorithm. Once

trained, the model is embedded into a web browser extension for live monitoring. As users browse, the system checks each website and immediately notifies them if it detects a phishing threat.

STEP 1: IMPORT FLASK, CORS, AND PICKLE LIBRARIES. CREATE A FLASK APP INSTANCE AND ENABLE CORS FOR CROSSORIGIN REQUESTS.

STEP 2:
EXTRACT THE DOMAIN FROM THE URL (REMOVE PROTOCOL AND PATH).
-COUNT THE NUMBER OF HEXADECIMAL CHARACTERS IN THE DOMAIN (EXCLUDING PERIODS).
RETURN 1 IF THE NUMBER OF HEXADECIMAL CHARACTERS EQUALS OR EXCEEDS THE DOMAIN LENGTH (INDICATING AN IP ADDRESS), ELSE RETURN 0.

STEP 3: FUNCTION GETLENGTH(URL):
IF THE LENGTH OF THE URL IS LESS THAN 54, RETURN 0 (INDICATING LEGITIMATE). -
OTHERWISE, RETURN 1
(INDICATING SUSPICIOUS OR PHISHING).

STEP 4: FUNCTION HAVEATSIGN(URL): - DEFINE A LIST OF SPECIAL CHARACTERS.
CHECK IF ANY SPECIAL CHARACTER IS PRESENT IN THE URL. - RETURN 1 IF FOUND, OTHERWISE RETURN 0.

#STEP5: FUNCTION GETDEPTH(URL):
SPLIT THE URL PATH BY '/'. - COUNT NON-EMPTY COMPONENTS.
RETURN THE COUNT AS DEPTH.

STEP 6: FUNCTION FEATUREEXTRACTION(URL):
INITIALIZE AN EMPTY LIST FOR FEATURES.
Append relevant features: IP check, '@' sign, length, depth, redirection, HTTPS, shortening, prefix/suffix.
RETURN THE FEATURES LIST.

STEP 7: FUNCTION PREDICT():
GET URL FROM POST REQUEST.
CHECK IF URL EXISTS IN CSV (SET `DATAPHISH`).
IF `DATAPHISH` IS 0, RETURN "0".
OTHERWISE, EXTRACT FEATURES AND CHECK IF 14 OR 15 ZEROS - IF SO, RETURN "0". ELSE, PREDICT USING THE MODEL.

URL Length: Phishing websites often use long, complicated URLs to trick people into thinking they are legitimate. If a website's URL is unusually long or seems overly complicated, it could be a red flag for phishing.

Suspicious Keywords: Phishing sites frequently use words like “login,” “account,” “secure,” or “bank” in their URLs to lure people into entering personal information. If you see these words in a website’s URL, it might be trying to imitate a trusted website and steal your information.

Domain Name Analysis: Phishing sites sometimes try to fool users by using domain names that are very similar to the real, trusted websites. This is called "typo squatting." For example, they might use a name like "securepaypal.com" instead of "paypal.com." It’s also suspicious if the domain name is new and hasn't built up a reputation yet.

HTTPS Validation: Legitimate websites use HTTPS (a secure version of HTTP) to protect your data. If a website doesn't use HTTPS, or it shows an insecure "HTTP" connection, it could be a phishing site trying to steal your information without encryption.

Subdomains: Phishing sites sometimes use misleading subdomains to trick users. For example, a phishing site might look like "secure.bank.com," which appears to be a trusted bank website, but it's actually not. Checking the subdomains can help identify if the site is trying to disguise itself as something legitimate.

Webpage Content Analysis: Analyse meta tags, JavaScript functions, form elements, and hidden fields in the HTML of the webpage to detect suspicious activities.

Hyperlink Analysis: Extract and analyse the hyperlinks on the page to determine whether they lead to suspicious or known malicious websites.

Resource Requests: Monitor resources like images, scripts, and other external requests made by the page to detect abnormal behaviour.

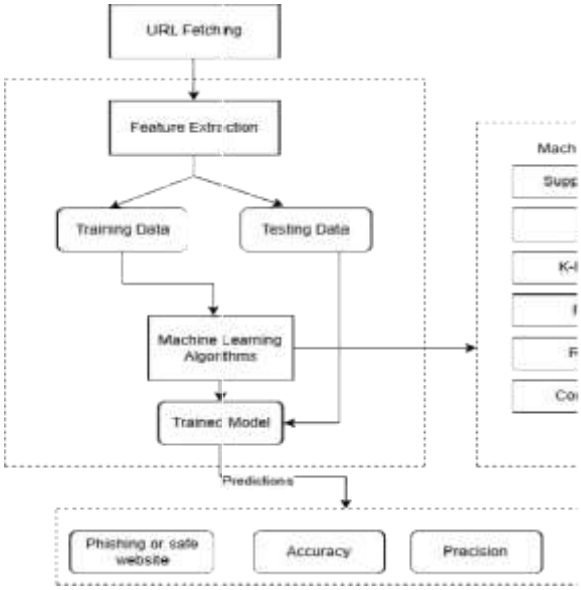

Figure 1: Flow Chart Accuracy of various model used for Phishing detection:

Sr. No	Algorithm	Accuracy	Precision	Accuracy_max_ft_3000
1	KN	0.905222 4	1.0	0.905223
2	NB	0.970986 4	1.0	0.971954
3	ETC	0.974854 9	0.991453	0.979691
4	RF	0.975822	0.982905	0.975822
5	SVC	0.975822 0	0.974789	0.974855
6	AdaBoost	0.924564	0.848837	0.961315
7	Xgb	0.967117 9	0.926229	0.968085
8	LR	0.958413	0.970297	0.956480

9	GBDT	0.946809	0.931373	0.959381
10	BgC	0.959381	0.861538	0.959381
11	DT	0.932302	0.838095	0.931335

Figure 2: Accuracy Table

III. LITERATURE SURVEY

Phishing detection has gained considerable attention over the past two decades due to the increasing sophistication and frequency of phishing attacks. Various methods have been explored in the literature, ranging from rule-based systems to modern machine learning and deep learning techniques.

Abdelhamid et al. (2014) proposed an intelligent phishing detection system using association rule mining and classification algorithms. Their system extracted lexical and host-based features from URLs and achieved moderate accuracy but struggled with zero-day attacks.

Meatal. (2009) introduced a real-time detection system using lexical and host-based features with supervised learning models such as Logistic Regression and Online Perceptron. Although effective in speed, it lacked deeper content analysis, limiting its detection scope.

Mohammad et al. (2015) created a phishing detection system based on rule-based learning and decision trees using the UCI phishing dataset. Their method showed good performance but suffered from poor adaptability to new phishing tactics.

Basnet et al. (2012) explored neural networks for phishing detection and reported improved accuracy over traditional classifiers. However, their model required extensive training time and was not optimized for real-time detection.

Marchal et al. (2016) proposed Phish Storm, which employed unsupervised learning to detect phishing in a streaming context. Their approach was effective for detecting previously unseen phishing attempts but required significant computational resources.

Verma and Das (2017) presented a comprehensive analysis comparing different machine learning techniques—SVM, Decision Trees, and Random Forests—for phishing detection. Their study concluded that Random Forests and ensemble methods tend to outperform single classifiers in both precision and recall.

Recent studies have started incorporating deep learning approaches, such as CNNs and LSTMs, to automatically learn features from URLs and webpage content (Zhang et al., 2020). While these models show high performance, they often require large datasets and high computational power, which limits their practical deployment in resource-constrained environments.

Overall, the literature indicates that while traditional machine learning models are effective and efficient, combining them with updated datasets, feature engineering, and hybrid models can significantly improve phishing detection performance. This paper builds on prior work by implementing and evaluating several classical ML algorithms with an emphasis on lightweight, scalable, and real-time capable solutions.

IV. EXISTING AND PROPOSED SYSTEM

4.1 Existing System

Traditional phishing detection systems primarily rely on

• Blacklist-based Detection: Maintains databases of known phishing URLs. While effective for previously reported threats, it fails to detect new or zero-day phishing attacks.

• Heuristic-based Methods: Use manually crafted rules to identify suspicious behavior. These are limited in adaptability and can generate high false positives.

• Rule-based Classifiers: Use fixed patterns (like suspicious domain names or abnormal JavaScript) to flag phishing attempts. These models lack flexibility and require frequent manual updates.

Some machine learning-based solutions exist, using basic URL features and simple classifiers. However, many of these

- Use outdated or limited datasets.
- Do not generalize well to real-world, unseen phishing attacks.
- Are not optimized for real-time detection and deployment.

Limitations of Existing Systems

- Inability to detect sophisticated, obfuscated phishing attempts.
- Poor performance on zero-day phishing websites.
- High false positive or false negative rates.
- Limited scalability and adaptability to changing phishing strategies.

4.2 Proposed System

The proposed system leverages machine learning algorithms for intelligent, adaptive phishing detection based on multiple features extracted from a given URL and associated metadata.

Key Features of the Proposed System:

- Feature Extraction: Uses a hybrid of URL-based, domain-based, and HTML-based features (if available) to create a comprehensive input vector.
- Machine Learning Models: Implements classifiers such as Decision Tree, Random Forest, SVM, and XGBoost to analyse the feature set and predict phishing probability.
- Pre-processing and Normalization: Ensures clean, normalized input data using techniques like label encoding, standardization, and balancing the dataset.
- Model Evaluation: Uses metrics such as accuracy, precision, recall, F1-score, and ROC-AUC for performance validation.

Advantages of the Proposed System:

- Ability to detect both known and zero-day phishing sites.
- Reduced false positives through robust feature selection and ensemble methods.
- High adaptability through retrainable models.
- Potential for real-time deployment using lightweight classifiers.

V. CHALLENGES AND FUTURE SCOPE

Challenges and Future Scope

5.1 Challenges

Despite promising results, several challenges remain in building a robust and real-time phishing detection system using machine learning:

1. Rapid Evolution of Phishing Techniques:

Phishers constantly change tactics, URLs, and page structures to evade detection. Static or outdated models may fail to detect new forms of attacks.

2. Data Quality and Diversity:

The availability of clean, labelled, and diverse datasets remains a major hurdle. Many public datasets are outdated or contain noise, affecting model generalizability.

3. Imbalanced Datasets:

In most datasets, legitimate websites significantly outnumber phishing websites, leading to biased learning. Handling class imbalance is crucial for model accuracy.

4. Feature Extraction Limitations:
Extracting certain features (e.g., HTML or JavaScript content) can be resource-intensive and potentially unsafe if the phishing site hosts malicious code.

5. Real-Time Detection Constraints:
Many ML models are computationally heavy and not optimized for real-time performance, limiting their integration into browsers or lightweight applications.

6. Adversarial Attacks on ML Models:
Attackers can craft URLs or webpage elements to intentionally deceive or mislead machine learning models, known as adversarial inputs.

5.2 Future Scope

To address the above challenges and improve phishing detection systems, future research can explore the following directions:

1. Deep Learning Models:
Use advanced models such as CNNs, LSTMs, or Transformer-based architectures to automatically learn deep patterns from raw URL strings and content data.

2. Real-Time Detection Systems:
Focus on developing lightweight and fast models suitable for deployment in web browsers, email clients, or proxy servers for instant detection.

3. Hybrid Approaches:
Combine multiple data sources (URL, domain reputation, HTML content, and user behaviour) and integrate rule based and ML-based systems for better accuracy.

4. Online Learning and Continuous Updates:
Implement online learning algorithms that continuously update the model with new phishing samples to stay effective against zero-day attacks.

5. Phishing Detection as a Service (PDaaS):
Develop cloud-based APIs or microservices that offer phishing detection capabilities to third-party developers and organizations.

6. Threat Intelligence Integration:
Enhance detection by integrating with global threat intelligence feeds to identify new phishing trends and block threats proactively.

VI. RESULT AND ANALYSIS

Figure 3: Home Page

Figure 4: Extension

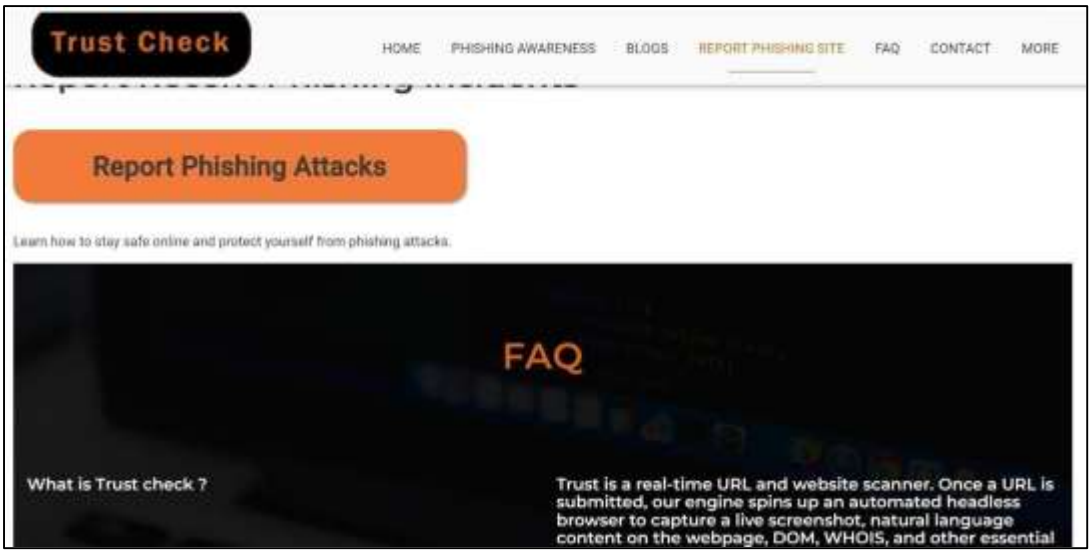

Figure 5: Report site

Figure 6: Phishing blogs

Figure 7: Accuracy of Algorithms

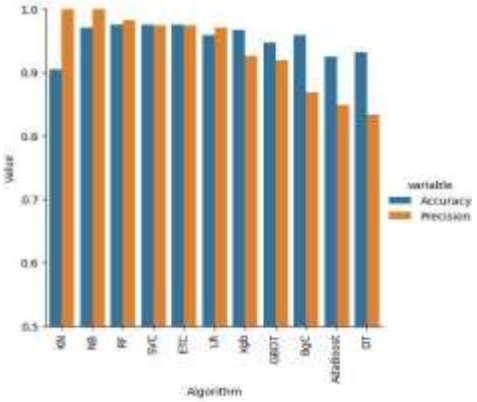

Figure 8: Precision Chart

Figure 9: Target Chart

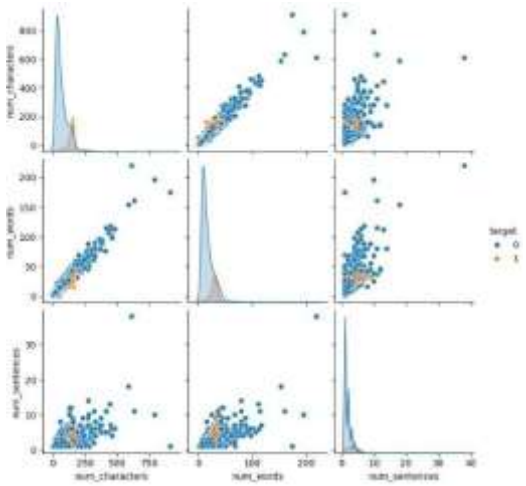

Figure 10: Regration chart

VII. CONCLUSION

The investigation into phishing website detection using machine learning techniques highlights the critical role that advanced algorithms play in enhancing cybersecurity. The study demonstrated that ensemble methods, particularly Random Forest and Gradient Boosting, provided superior accuracy and reliability in identifying phishing attempts when compared to traditional classification techniques. The integration of feature selection methods further optimized model performance, emphasizing the significance of utilizing relevant features derived from URLs and website content. Overall, the findings indicate that machine learning can significantly improve the effectiveness of phishing detection systems, ultimately protecting users from online threats

REFERENCES

[1] Aamir, M., Alhazmi, O. A., & Alzahrani, M. (2020). Phishing website detection using machine learning: A review. International Journal of Computer Applications, 975(1), 20- 26.

[2] Gupta, B., & Kaur, P. (2021). Phishing detection using machine learning algorithms. Journal of King Saud University - Computer and Information Sciences.

[3] Ghosh, A., & Saha, S. (2020). Detection of phishing websites using machine learning algorithms. International Journal of Advanced Research in Computer Science, 11(2), 23- 29.

[4] Rani, R., & Yadav, A. (2019). Phishing website detection using ensemble learning techniques. Proceedings of the 2019 IEEE International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), 186-190.

[5] Ramesh, P., & Srinivas, K. (2018). A survey of phishing detection techniques. International Journal of Computer Applications, 181(6), 29-34.

[6] Sharma, S., & Gupta, V. (2020). Machine learning techniques for phishing detection: A survey. Journal of Ambient Intelligence and Humanized Computing, 11(1), 249-267.

[7] Yadav, S. K., & Goyal, A. (2021). Phishing detection using artificial intelligence: A systematic review. Journal of Computer and Communications, 9(9), 1-18.