IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

DETECTION OF PHISHING WEBISTES

[1]K. Angel, [2]Prof. B. Prajna, [3]K. Chandana Sai Sri , [4] k.sathwika , [5]K. Amogha , [6]K.Susmitha

[1]Student, [2]Head of the Department, [3]Student, [4]Student, [5]Student, [6]Student
[1]Department of Computer Science and Systems Engineering,
[1]Andhra University College of Engineering for Women, Visakhapatnam, Andhra Pradesh, India

ABSTRACT

Phishing is one of the familiar attacks that detects users to access malicious content and gain their information. Gradient Boosting Classifier is the model we utilised in our suggested strategy to identify phishing websites based on aspects of URL significance. By extracting and comparing different characteristics between legitimate and phishing URLs, the suggested method uses gradient boosting classifier to identify phishing URLs.

Keywords: Phishing, Attack, Malicious , Gradient Boosting Classifier , Strategy , URL , Legitimate , Characteristics , Detection

I. INTRODUCTION

Phishing has emerged as a significant concern for cyber security researches due to the ease with Which malicious actors create deceptive websites that closely resemble legitimate ones. While cyber security experts may be able to identify these fake websites, many users lack the knowledge to distinguish them from genuine sites, making them vulnerable to phishing attacks. These attacks typically aim to obtain sensitive information such as bank account credentials, resulting in substantial financial losses for businesses, with estimates suggesting annual losses of around $2 billion in the United States alone. Globally, the impact of phishing is estimated to be as high as $5 billion annually, according to the Microsoft Computing Safer Index Report released in February 2014.

Phishing attacks exploit the vulnerabilities of users, making them difficult to mitigate effectively. Traditional methods of detecting phishing websites involve updating blacklists with known malicious URLs and Internet Protocol (IP) addresses. However, attackers employ various techniques, including URL obfuscation, fast-flux, and algorithmic generation of new URLs, to evade detection. One major drawback of this blacklist method is its inability to detect zero-hour phishing attacks, where the malicious URL is newly created and not yet identified.

II. LITERATURE REVIEW

[1] Sahin Goz, O. K., Buber, E., Demir, O., &Diri, B. "Machine Learning-Based Phishing Detection from URL's, "Expert Systems with Applications. The dataset used is self-constructed. Where phishing websites belong to Phish Tank and legitimate URLS are from Yandex Search API. The main purpose was to detect the word, which is like, brand names, to detect keywords, the words, which ae formed from random characters. Various classification algorithms such as Naive Bayes, Random Forest, KNN, Ada boost and Decision Tree including some feature extraction types such as NLP-based features, word vectors and Hybrid are used.

[2] J. James, Sandhya L. and C. Thomas, "Detection of phishing URLs using machine learning techniques," International Conference on Control Communication and Computing (ICCC): The system proposes used a method based on lexical features, host properties and properties related to the page for the detection of phishing websites. For getting a proper understanding of the pattern of URLs, various data mining algorithms are used. So, Tree-based classifiers are best suited for phishing URL classification.

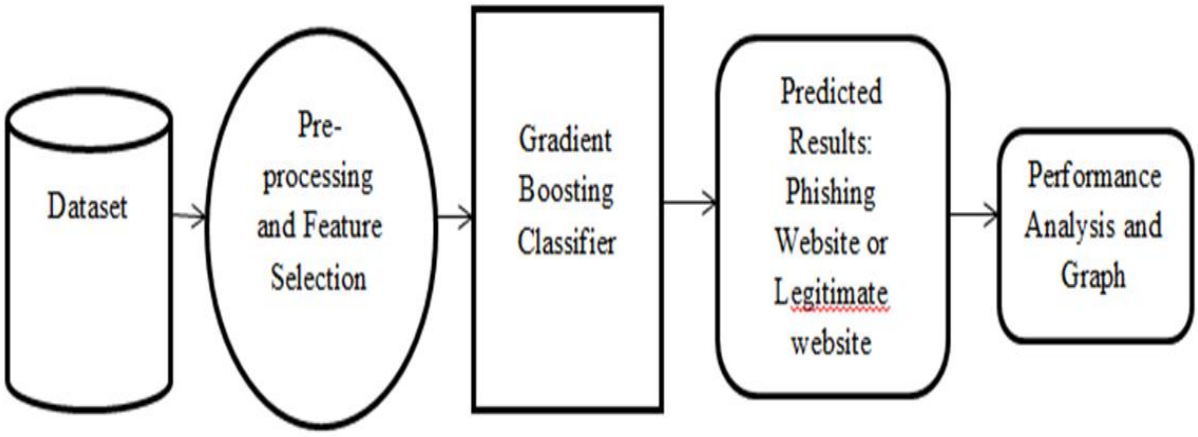
[3] Dipayan Sinha, Dr. Minal Moharir, Prof. Anitha Sandeep, "Phishing Website URL Detection using Machine Learning," International Journal of Advanced Science and Technology: Detection of phishing websites is performed by using machine learning techniques like Logistic Regression, Decision Tree, Gradient Boosting. Data collection involves phishing and legitimate websites. Extracting useful features has two steps: URL-based involves IP Address,'@' symbol in URL, dashes in URL, long URL, presence of unusual number, dot count, sub-domains in URL, etc. Domain and based includes Page Rank of the website, age of the Domain, and Validity of the website. The dataset is split into training and testing set in the ratio 80:20.

According to Erzhou Zhu (2018), phishers typically put up a false website where victims were tricked into providing passwords and perceptive information. As a result, it's critical to detect rogue websites before they cause any harm to their victims. This study proposes a new method based on deep reinforcement to model and detects malicious URLs, fuelled by the dynamic nature of criminal websites to steal sensitive information [4]. The suggested model may learn the properties related to phishing website identification by accommodating the dynamic behaviour of phreaking websites [5]. Using Deep Forest, as well as a range of contemporary machine learning models, such as GBDT and XG Boost, are used to represent URLs in vector form may be applied to detect sensitive identity theft [6].

III. SYSTEM ARCHITECTURE

A system architecture is the conceptual model that declinate the structure, behavior, and various perspectives of a system.

Typically, within the realm of architecture, the term "system" usually pertains to the architecture design of the software rather than the physical arrangement of the machinery. The system architecture in essence, mirrors the system utilization and thus adapts overtime in response to its usage patterns as the system is used.



IV. METHODOLOGY

In the proposed system, our model undergoes preprocessing where we tokenize words and apply stemming. Data processing is integral, transforming data for seamless machine transfer and aiding the algorithm in defining features effortlessly. As we move forward, vectorizing our URLs becomes imperative, given the varying importance of words within them. For instance, terms like "virus" or ".exe" carry more weight. To achieve this, our model employs Count Vectorizer alongside a tokenizer to amalgamate words effectively, creating a vector representation of URLs. We utilize a regular expression tokenizer, a tool adept at separating strings based on specified patterns, such as 's+', which synchronizes one or more gaps when '+' is appended. Stemming emerges as a crucial step globally, pivotal for queries and Internet search engines alike.

For deployment, we leverage the Fast API framework, renowned for its efficiency and compatibility with Python 3.6+ and standard Python type hints. Fast API stands out for its rapidity, rivalling the performance of NodeJS and Go, thanks to Starlette and Pedantic. It furnishes the user interface by seamlessly integrating the machine learning model. The system's architectural flow is visually represented in Fig. 1.

Advantages of our system include:

- Provision of a user interface for ease of interaction.
- Training of the model utilizing an extensive array of features.
- Delivering a high level of accuracy, crucial for reliable performance.

Incorporating Extreme Gradient Boosting (XG Boost) enriches our system. XG Boost represents the pinnacle of engineering aspirations, pushing computational resources to their limits for boosted tree algorithms. It serves as a software library, downloadable and installable on various platforms, accessible through a diverse array of interfaces.

utilization of Fast API framework for deployment underscores the system's commitment to efficiency and user-friendly interaction. Fast Api's compatibility with Python 3.6+ and standard Python type hints streamlines development and deployment processes. Its integration with the machine learning model facilitates the creation of a seamless user interface, enhancing user experience while ensuring robust functionality. By leveraging Fast API, the system achieves rapidity comparable to other high-performance frameworks like NodeJS and Go, without compromising on reliability or scalability.

Furthermore, the emphasis on preprocessing techniques such as tokenization, stemming, and vectorization underscores the system's dedication to optimizing data representation for effective model learning. By transforming raw data into structured, feature-rich inputs, the model gains insights that enhance its ability to discern phishing websites accurately.

This meticulous approach to data processing ensures that the model is well-equipped to handle the complexities and nuances inherent in URL analysis, ultimately leading to higher detection accuracy and reliability in real-world scenarios.

Graphical workflow of proposed model for detection phishing of website:

SYSTEM REQUIREMENTS

Software Requirements:

- Operating System: Windows 10
- Coding Language: Python
- Web Framework: Flask

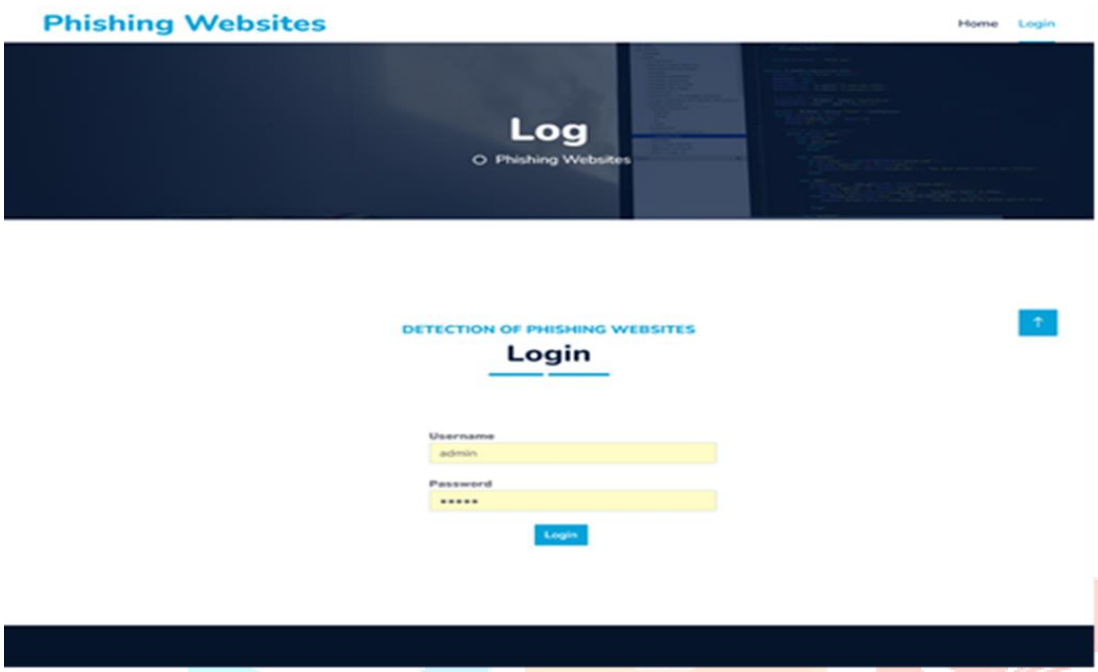
Hardware Requirements:

- System Processor: Pentium i3
- Hard Disk:500GB
- Monitor:15” LED
- Input Devices: Keyboard, Mouse
- Ram: 4GB

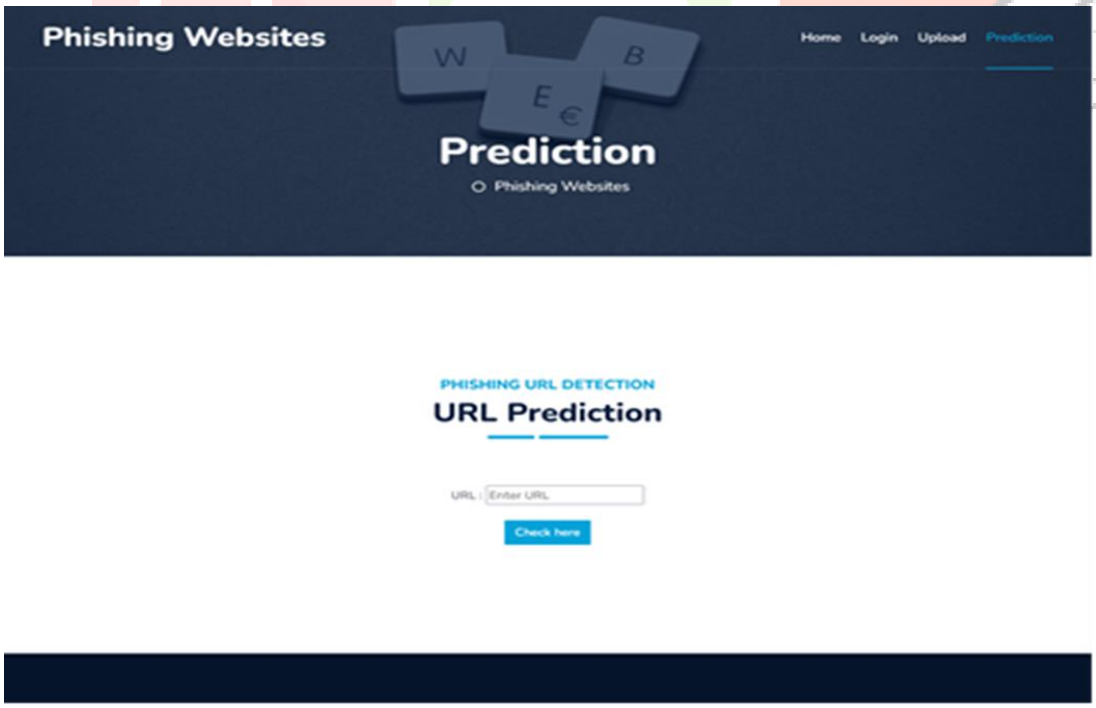
V. RESULTS AND DISCUSSION

LANDING PAGE

LOGIN PAGE

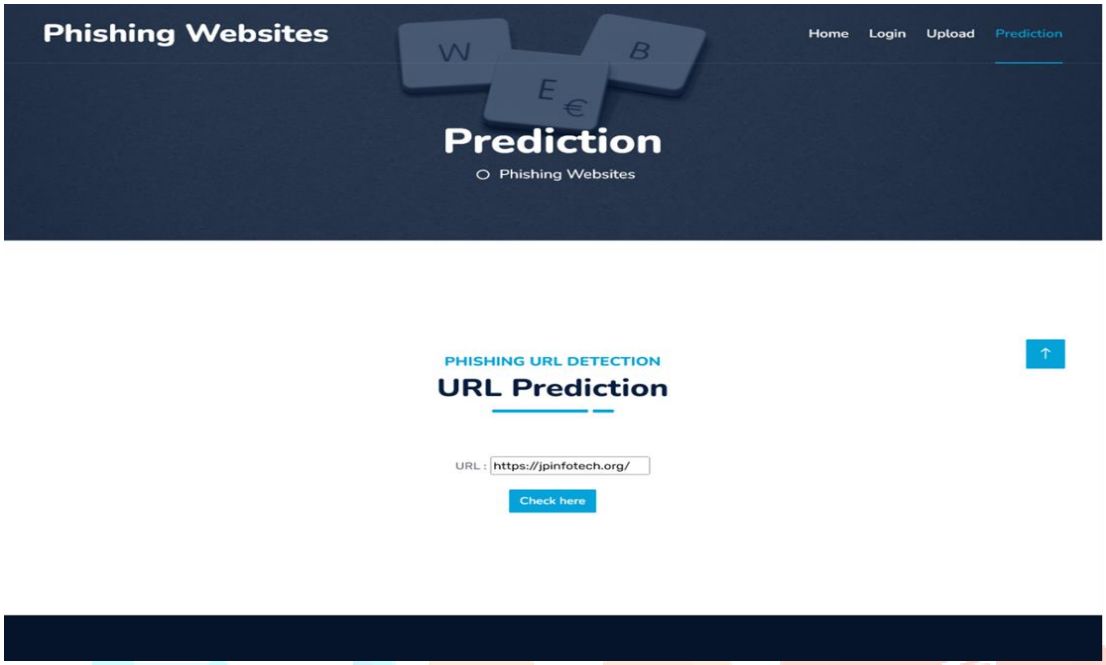


PREDICTION PAGE

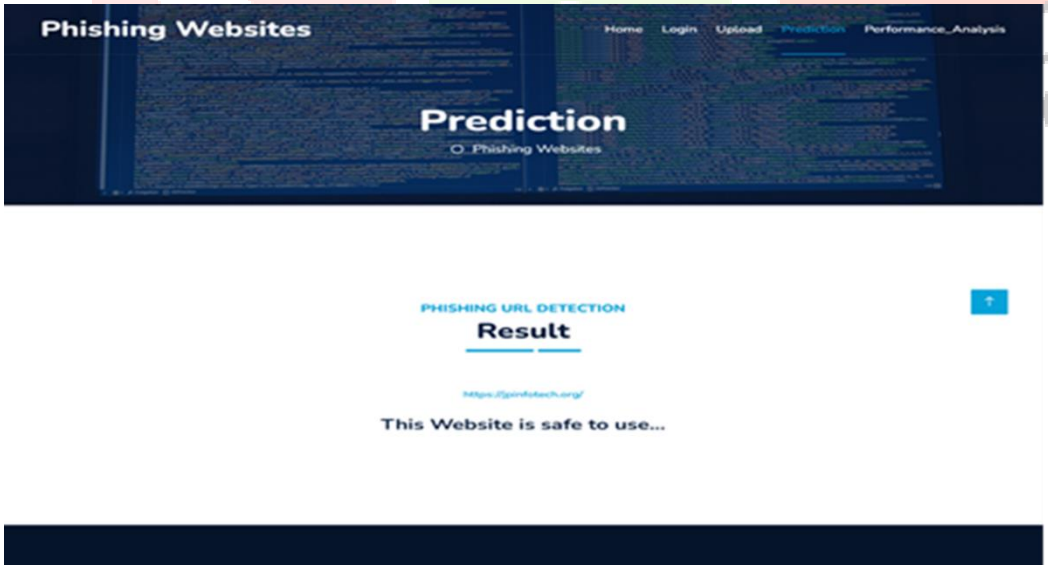


OUTPUT

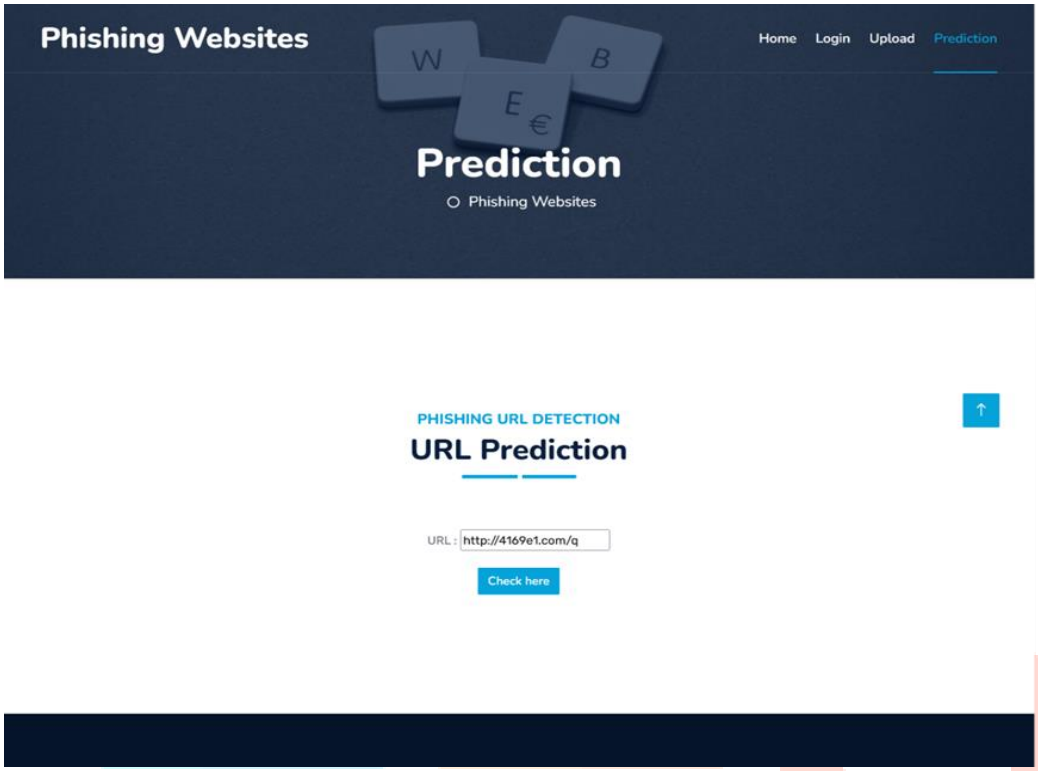
LEGITIMATE WEBSITE URL :



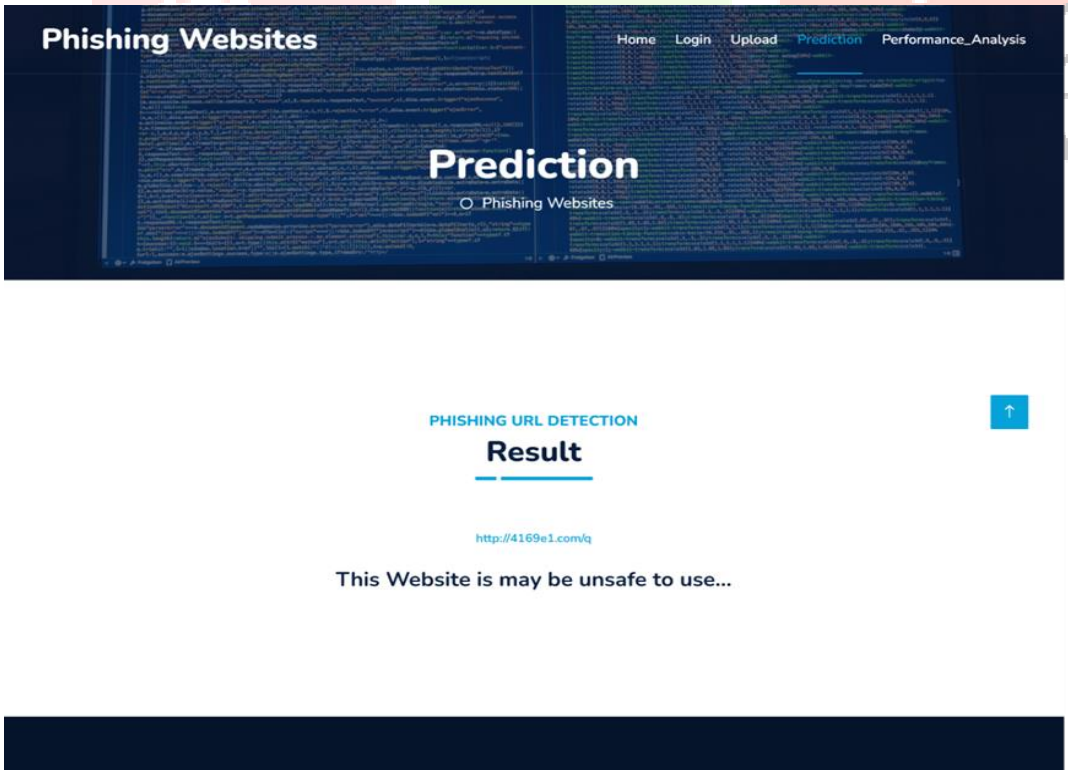
RESULT:



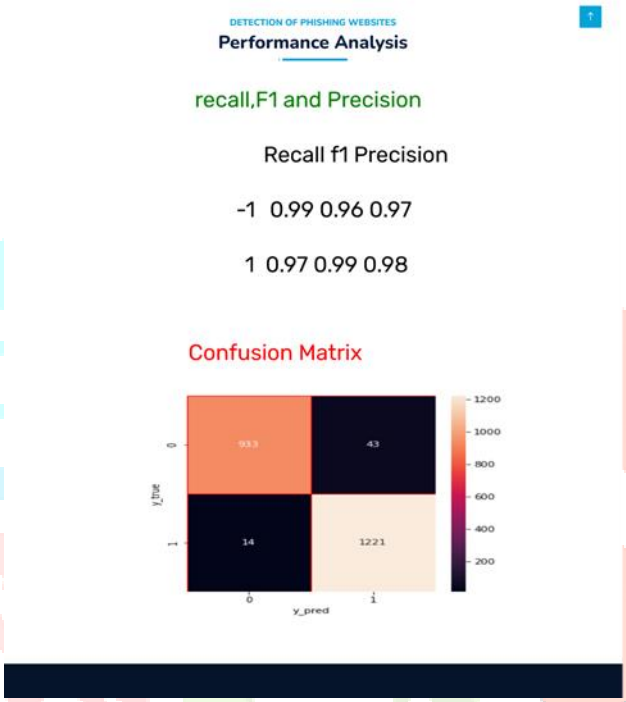
PHISHING WEBSITE URL :

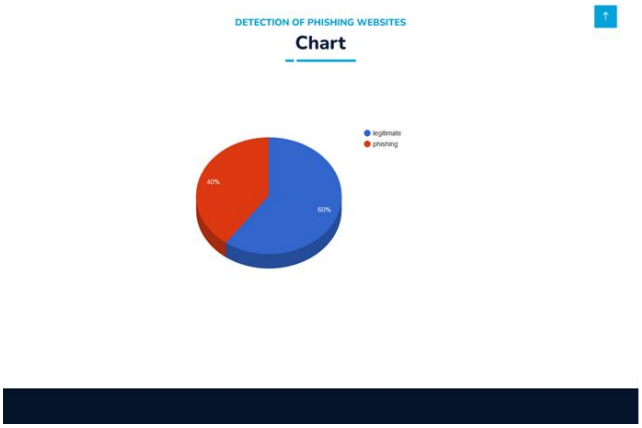


RESULT :



PERFORMANCE ANALYSIS





VI. CONCLUSION & FUTURE WORK

An effective anti-phishing system requires timely detection using tools like Gradient Boosting Classifier, achieving 97% accuracy. Careful Preprocessing, feature engineering, and rigorous evaluation are crucial, with metrics like accuracy, precision, recall, and F1 score being vital. Combining URL lexical features with others, such as the host, enhances effectiveness. Future enhancements aim to create a scalable web service with online learning capabilities for improved accuracy. The future of phishing detection involves technological advancements and interdisciplinary collaboration to stay ahead of evolving threats. To fortify detection

capabilities against evolving phishing strategies, combining URL lexical features with host information proves essential. Our forthcoming strategy involves constructing a scalable web service endowed with online learning functionalities to swiftly adapt to novel attack patterns. Sustained progress in technology and cross disciplinary cooperation remains paramount in outpacing emerging cyber threats

VII. REFERENCES

- [1] Sahin Goz, O. K., Buber, E., Demir, O., & Diri, B. "Machine Learning-Based Phishing Detection from URLs, "Expert Systems with Applications, vol.117, pp. 345-357, January 2019.

- [2] J. James, Sandhya L. and C. Thomas, "Detection of phishing URLs using machine learning techniques," International Conference on Control Communication and Computing (ICCC), December 2013.

- [3] Dipayan Sinha, Dr. Minal Moharir, Prof. Anitha Sandeep, "Phishing Website URL Detection using Machine Learning," International Journal of Advanced Science and Technology, vol.29, no. 3, pp. 2495-2504,2020.

- [4] Andrew Jones, Mahmoud Kh Andrew Jones, Mahmoud Khonji, Youssef Iraqi, Senior Member A Literature Review on Phishing Detection 2091-2121 in IEEE Communications Surveys and Tutorials, vol. 15, no. 4, 2013.

- [5] Alessandro Acquisto, Idris Ad jerid, Rebecca Bale Bako, Laura Brandimarte, Lorrie Faith Cranor, Saranga Komanduri, Pedro Giovanni Leon, Norman Sadeh, Florian Schaub, Many Understanding and Assisting Users' Online Choices with Nudges for Privacy and Security 50(3), Article No. 44, ACM Computing Surveys, 2017.

- [6] "Discovering phishing target based on semantic link network," Wenyin, Liu, and colleagues wrote. Computer Systems, vol. 26.3, no. 3, pp. 381-388, 2010.

- https://www.interviewbit.com/
- https://www.javatpoint.com/