URL PHISHING DETECTION USING MACHINE LEARNING

AUTHORS:

Darga Ganesh[*1] Emmadi Sai Saketh [*2] Badam Gokul[*3]

dargaganesh@gmail.com , sakethemmadi4@gmail.com , badamgokul80@gmail.com

Department of Data Science(CSE) Sphoorthy EngineeringCollege, Hyderabad, India.

Nilesh D Mhaiskar

ndmhaiskar@gmail.com

ABSTRACT

Malicious Web In recent years, with the increasing use of mobile devices, there is a growing trend to move almost al l real-world operations to the cyberworld. Although this makes easy our daily lives, it also brings many security breaches due to the anonymous structure of the Internet. Used antivirus programs and firewall systems can prevent most of the attacks. However, experienced attackers target on the weakness of the computer users by trying to phish them with bogus webpages. These pages imitate some popular banking, social media, e-commerce, etc. sites to steal some sensitive information such as, user-ids, passwords, bank account, credit card numbers, etc. Phishing detection is a challenging problem, and many different solutions are proposed in the market as a blacklist, rule-based detection, anomaly-based detection, etc. In the literature, it is seen that current works tend on the use of machine learning-based anomaly detection due to its dynamic structure, especially for catching the "zero-day" attacks. In this paper, we proposed a machine learning-based phishing detection system by using eight different algorithms to analyze the URLs, and three different datasets to compare the results with other works. The experimental results depict that the proposed models have an outstanding performance with a success rate. Phishing is one of the most widely practised Internet frauds. It focuses on the theft of sensitive personal information such as passwords and credit card details.

INTRODUCTION:

In an era where the internet serves as a lifeline for both personal and professional activities, the prevalence of cyber threats looms larger than ever. Among these threats, phishing stands out as one of the most insidious and widespread. Phishing attacks, particularly those utilizing deceptive URLs, continue to pose significant risks to individuals, businesses, and organizations worldwide.

PROJECT AIM AND OBJECTIVE

The aim of this project is to develop an efficient and robust system for detecting URL phishing attacks, thereby enhancing cybersecurity measures to safeguard individuals and organizations from potential cyber threats.

The objectives of this project are to:

- To study various automatic phishing detection methods

To identify the appropriate machine learning techniques and define a solution using the selected method

- To select an appropriate dataset for the problem statement
- Evaluate the performance metrics of the detection system, including accuracy, precision, recall, false positive rate, and computational overhead, to assess its suitability for practical deployment in various cybersecurity environments.

PROPOSED SYSTEM

- The proposed system involves an integrated approach where the blacklist serves as an initial filter for known phishing URLs, and the Random model provides additional analysis for potentially new or unidentified threats. Regular updates to the blacklist ensure that the system remains effective in detecting and preventing phishing attacks as they evolve over time.

ADVANTAGES:

- In depth preprocessing
- High Accuracy
- High Efficiency
- Fast Processing

SCOPE OF THE STUDY

The scope of URL phishing detection encompasses the development and implementation of advanced algorithms and techniques to identify and mitigate malicious activities conducted through deceptive URLs. It involves real-time detection, leveraging machine learning and AI, behavioral analysis, and integration with web browsers and security tools. Additionally, it includes user education, cross-platform compatibility, reducing false positives, compliance with regulations, and continuous monitoring and updates. Overall, the scope aims to provide comprehensive protection against phishing attacks, ensuring the security of users' personal information and online transactions.

SYSTEM STUDY FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed

on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

LITERATURE SURVEY

The goal of the study is to carry out ELM employing 30 different primary components that are characterized using ML [1]. To prevent being discovered, most phishing URLs use HTTPS. Website phishing can be identified in three different ways. The first method evaluates several URL components; the second method assesses a website's authority, determines if it has been introduced or not, and determines who is in charge of it; the third method verifies a website's veracity.

In the study, the highest correlated features from two distinct datasets were chosen. These features combined content-based, URL and domain- based features. Then, a comparison of the performance of a number of ML models was carried out[2]. The results also sought to pinpoint the top characteristics that aid the algorithm in spotting phishing websites. The Random Forest (RF) method produced the best classification results for both datasets.

In their study ,the user-received URLs will be entered to the machine learning model, which will then process the input and report the results, indicating whether the URLs are phishing or not. SVM, Neural Networks, Random Forest, Decision Tree, XG boost, and other machine learning algorithms can all be used to categorize these URLs[2]. The suggested method uses the Random Forest and Decision Tree classifiers. With an accuracy of 87.0% and 82.4% for Random Forest and decision tree classifiers, respectively, the suggested technique successfully distinguished between Phishing and Legitimate URLs.

In [4] the study of system for Detection of Phishing Websites using Machine learning. Their proposed method uses both Classification and Association algorithms to optimise the system, making it faster and more effective than the current approach. The proposed system's inaccuracy rate is reduced by 30% by combining these two algorithms with the WHOIS protocol, making it an effective technique to identify phishing websites.

.

FUNDAMENTALS

In ML and statistics, classification method is an approach involving supervised learning where computer program gains information from input and afterward utilizes this figuring out how to characterize new observations. Here are few classification techniques used in the detection of phishing URLs.

K-NEAREST NEIGHBORS

The K-Nearest Neighbors (KNN) algorithm is a simple yet powerful supervised machine learning algorithm used for classification and regression tasks. In KNN, an object is classified based on the majority class of its nearest neighbors in the feature space. The algorithm works by calculating the distance between the new data point and all other data points in the training dataset, typically using Euclidean distance. The "K" in KNN represents the number of nearest neighbors to consider for classification. Once the distances are computed, the KNN algorithm selects the K nearest neighbors and assigns the class label of the majority of these neighbors to the new data point. KNN is non-parametric, meaning it does not make assumptions about the underlying distribution of the data, making it suitable for a wide range of applications.

Kernal Support Vector Machine

The Kernel Support Vector Machine (SVM) is a powerful supervised machine learning algorithm used for classification and regression tasks. SVMs are particularly effective in handling high-dimensional data and are widely used in various applications such as image classification, text categorization, and bioinformatics.

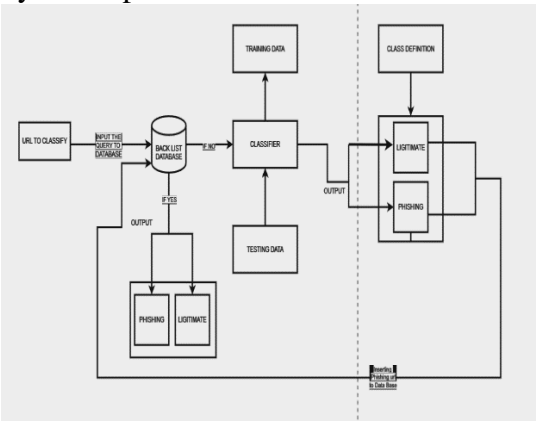
gradient boosting

A Gradient Boosting is a popular boosting algorithm in machine learning used for classification and regression tasks. Boosting is one kind of ensemble Learning method which trains the model sequentially and each new model tries to correct the previous model. It combines several weak learners into strong learners.

SYSTEM DESIGN:

SYSTEM ARCHITECTURE

The architecture of the system is as shown in fig 4.1; the URLs to be classified as legitimate or phishing is fed as input to the appropriate classifier. Then classifier that is being trained to classify URLs as phishing or legitimate from the training dataset uses the pattern it recognized to classify the newly fed input.



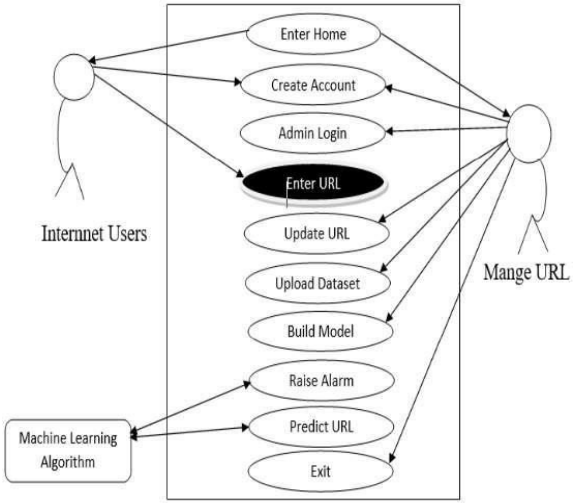
UML DIAGRAMS

The Unified Modeling Language (UML) is used to specify, visualize, modify, construct and document the artifacts of an object-oriented software intensive system under development. Complex applications need collaboration and planning from multiple teams and hence require a clear and concise way to communicate amongst them. The UML represents a collection of best engineering practices that have proven

successful in the modeling of large and complex systems. The UML is a very important part of developing object orieanted software and the software development process. The UML uses mostly graphical notations to express the design of software projects. non-programmers do not understand code. So UML becomes essential to communicate with non-programmers about essential requirements, functionalities, and processes of the system.

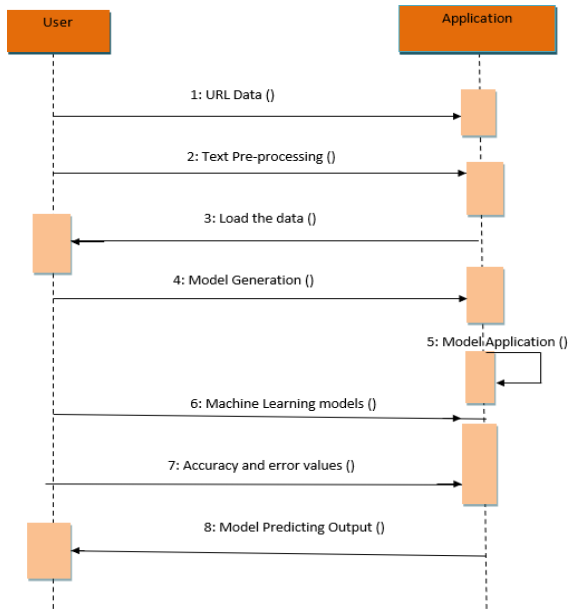
USE CASEDIAGRAM

Use Case Diagram is a vital tool in system design, it provides a visual representation of how users interact with a system. It serves as a blueprint for understanding the functional requirements of a system from a user's perspective, aiding in the communication between stakeholders and guiding the development process.



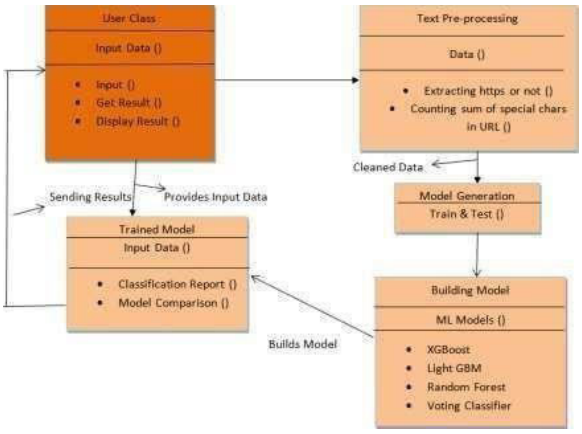
SEQUENCE DIAGRAM

A sequence diagram is a type of Unified Modeling Language (UML) diagram that illustrates the interactions and communication sequences between objects or components within a system over a specific period of time.It is a modeling language in the field of software engineering that aims to set standard ways to visualize the design of a system. Sequence diagrams are often used to model the dynamic behavior of a system, showing how various objects or components collaborate and exchange messages to achieve a specific functionality or scenario.



CLASS DIAGRAM

The class diagram is the main building block of object-oriented modelling. It is used for general conceptual modelling of the structure of the application, and for detailed modelling translating the models into programming code.



IMPLEMENTATION

This chapter of the report illustrates the approach employed to classify the URLs as

either phishing or legitimate. The methodology involves building a training set. The training set is used for training a machine learning model, i.e., the classifier. Fig 5.1 shows the diagrammatic representation of the implementation.

Implementation

Process Involved in implementation

The first step of the research work was determining the right data set. The dataset selected was collected from Kaggle for this task. The reasons behind selecting this dataset are several. It includes:

- The data set is large, so working with it is intriguing
- The number of features in the data set is 30 giving a wide range of features making the predictions a little more accurate. The fig 5.2 shows the features being considered.
- The number of URLs is quite evenly distributed among the 2 categories.

1	having_IP_Address	16	SFH
2	URL_Length	17	Submitting_to_email
3	Shortining_Service	18	Abnormal_URL
4	having_At_Symbol	19	Redirect
5	double_slash_redirecting	20	on_mouseover
6	Prefix_Suffix	21	RightClick
7	having_Sub_Domain	22	popUpWidnow
8	SSLfinal_State	23	Iframe
9	Domain_registeration_length	24	age_of_domain
10	Favicon	25	DNSRecord
11	port	26	web_traffic
12	HTTPS_token	27	Page_Rank
13	Request_URL	28	Google_Index
14	URL_of_Anchor	29	Links_pointing_to_page
15	Links_in_tags	30	Statistical_report

- **Splitting**: the dataset into training part of dataset and testing part of dataset. The dataset was split into training and testing dataset with 75% for training and25% for testing using the "train test split" method. The splitting was done after assigning the dependent variables and independent variables.
- **Preprocessing**: Preprocessing involves filling the missing data or removing the missing data and getting a clean dataset . But the dataset chosen was already preprocessed and did not require any further preprocessing from my end. The only step to be performed in preprocessing was feature scaling.
- **Feature scaling**: Feature Scaling is a procedure to normalize the independent variable present in the information in a fixed range. It is performed during the data pre-processing to deal with varying magnitudes. There are two ways of feature scaling – Normalization and Standardization. The project uses standardization feature scaling methods.
- The variables should be put in the same scale, else one variable might dominate others hence might affect the result.

Standardization: Standardization is another scaling procedure where the values are based on the mean with a unit standard deviation. This implies the mean of that attribute gets zero and the resultant distribution has a unit standard deviation.

Normalization: Normalization is a scaling method where values are moved and rescaled so they wind up going somewhere in the range of 0 and 1. It is otherwise called Min-Max scaling.

CLASSIFIERS

sklearn.neighbors.KNeighborsClassifier:

Parameters used:

- N neighbors: It is the number of neighbors to be considered while categorizing and was considered 5 in the algorithm
- Metric: It depicts the distance metric to be used. The one used in the algorithmis 'minkowski'

- p: It is the power parameter for the metric. The algorithm uses p = 2 which is equivalent to Euclidean distance.

sklearn.tree.DecisionTreeClassifier

Parameters used:

- criterion: the function that is used to measure the quality of a split. The one that is used in the algorithm is "entropy"
- m is "entropy"

sklearn.ensemble.GradientBoostingClassifier

Gradient boosting is an ensemble technique where multiple weak learners (usually decision trees) are built sequentially, with each subsequent model focusing on the errors made by the previous ones. The final model is a weighted combination of these weak learners.

Parameters used:

- N estimators: The number of boosting stages to perform. The number used in the algorithm is 10.

TESTING AND VALIDATION

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

In this chapter, we check for the working of the proposed system by testing and comparing the result of the algorithm and the actual result. It is basically validating the system. The testing is done for each algorithm with a legitimate and phishing URL and the results are as follows.

UNIT TESTING:

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

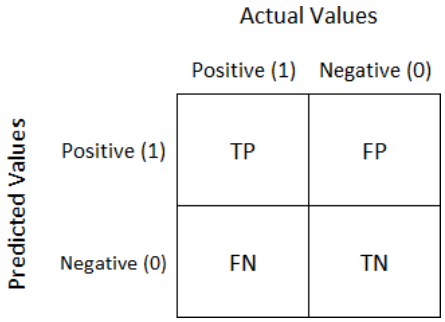
The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects

encountered.

FUNCTIONAL TEST

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system



documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input: identified classes of valid input must be accepted.

Invalid Input: identified classes of invalid input must be rejected.

Functions: identified functions must be exercised.

Output: identified classes of application outputs must be exercised. Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

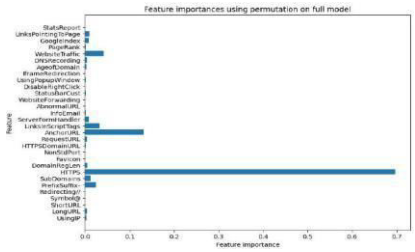
EXPERIMENTAL ANALYSIS AND RESULT

EXPERIMENTAL ANALYSIS

Confusion matrix(CM) is a graphical summary of the correct predictions and incorrect predictions that is made by a classifier that can be used to determine the performance. In abstract terms, the CM is as shown in fig

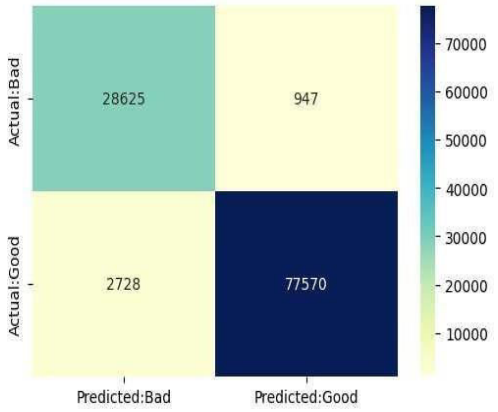
Confusion matrix

In the above figure TP is True positive, TN is True negative, FP is False Positive



and FN is False Negative. The confusion matrix of the algorithms used are as shown:

GRADIENT BOOSTING



Output Design Input Interface:

The input interface allows users to enter a URL for phishing detection. It consists of:

Input Box: A text field where users can input the URL to be checked.

Submit Button: A button to submit the URL for analysis.

OUTPUT DISPLAY:

The output display presents the results of the URL phishing detection. It includes:

Phishing Status: Indicates whether the URL is classified as phishing or legitimate.

Confidence Score: Provides a confidence score or probability associated with the classification result.

Additional Information: Optionally, display additional information such as features extracted from the URL, metadata, or WHOIS data.

USER FEEDBACK:

Provide feedback to the user based on the classification result, such as:

Warning Message: If the URL is classified as phishing, display a warning message advising the user not to proceed.

Confirmation Message: If the URL is classified as legitimate, display a confirmation message indicating that the site is safe to visit.

INTEGRATION WITH BLACKLIST DATABASE:

If the URL is not found in the database and needs to be checked with the machine learning model, display:

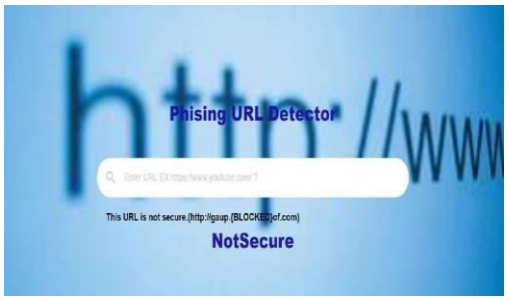
Database Check Result: Indicates whether the URL was found in the blacklist database or not.

Machine Learning Prediction: Displays the phishing status predicted by the machine learning model.

Database Update Status: Indicates whether the URL was inserted into the blacklist database.

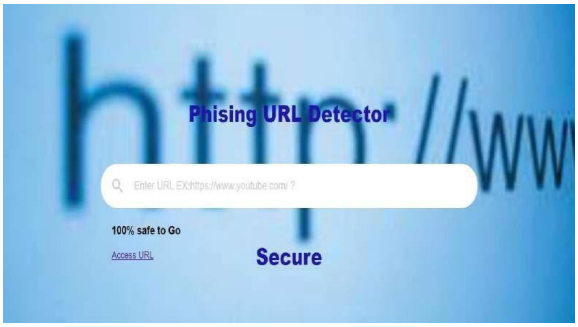
RESULTS:

- Input URL - http://gaup.{BLOCKED}of.com
- Algorithm – Gradient Boosting Algorithm
- Expected outcome – Phishing
- Obtained – Phishing



Gradient Boosting

- Input URL - https://www.youtube.com/
- Algorithm – Gradient Boosting Algorithm
- Expected outcome – Legitimate
- Obtained – Legitimate



CONCLUSION:

In summary, developing a URL phishing detection system in learning techniques to identify and classify URLs as either phishing or legitimate. The system typically integrates with a blacklist database containing known phishing URLs and utilizes machine learning models to classify URLs not found in the database. However, several limitations must

be considered when designing and evaluating such a system.

These limitations include challenges related to data availability, feature engineering, class imbalance, concept drift, adversarial attacks, model interpretability, false positives/negatives, regulatory and privacy concerns, resource constraints, and user awareness and education. Addressing these limitations requires a comprehensive approach that combines technical solutions, user education, and ongoing monitoring and adaptation.

By understanding and acknowledging these limitations, organizations can develop more realistic expectations, implement appropriate

machine learning based approach for phishing detection using hybrid features.

REFRENCES:

[1] S. Alrefaai, G. Özdemir, A. Mohamed, Detecting Phishing Websites Using Machine Learning, in Proceedings of the 2022 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), Ankara, Turkey (2022)

[2] M. D. Bhagwat, P. H. Patil and T. S. Vishawanath, A MethodicalOverview on Detection, Identification and Proactive Prevention of PhishingWebsites, in Proceedings of the 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), Tirunelveli, India (2021)

[3] P. Bhavani, Amba, Chalamala, Madhumitha, Likhitha, Sree Sai, C. P. Sai, Intl. J. Appl. Res. Tech 8, 2511 (2022)

[4] B. B. Gupta, K. Yadav, I. Razzak, K. Psannis, A. Castiglione, X. Chang, Comp. Communi. 175 (2021)

[5] S. Parekh, D. Parikh, S. Kotak, and S. Sankhe. A new method for detection of phishing websites: Url detection. In 2018 Second International Conference on In

[6] M. M. Yadollahi, F. Shoeleh, E. Serkani, A. Madani, and H. Gharaee. An adaptive

[7] P. Yang, G. Zhao, and P. Zeng. Phishing website detection based on multidimen sional features driven by deep learning. IEEE Access, 7:1519615209, 2019

[8] Muhammet Baykara and Zahit Gurel. Detection of phishing attacks. pages 15, 03 2018

[9] E. Zhu, Y. Chen, C. Ye, X. Li, and F. Liu. Ofs-nn: An e ective phishing websites detection model based on optimal feature selection and neural network. IEEE Access, 7:7327173284, 2019

[10] E.Poornima, N.Kasiviswanath, &C.ShobaBinduSecured Data Sharing in Groups using Attribute-Based Broadcast Encryption in Hybrid Cloud,in Proceedings of the Emerging Trends in Expert Applications and Security. Advances in Intelligent Systems and Computing, Springer, Singapore, 841 (2019)

[11] D. K. Mondal, B. C. Singh, H. Hu, S. Biswas, Z. Alom, M. A. Azim, J. Inform. Secu. Appl 62 (2021)

[12] H. Shirazi, K. Haefner, and I. Ray. Fresh-phish: A framework for auto-detection of phishing websites. In 2017 IEEE International Conference on Information Reuse and Integration (IRI), pages 137143, 2017