International Journal of Scientific Research in Computer Science, Engineering and Information Technology

ISSN : 2456-3307 OPEN ACCESS

Available Online at : www.ijsrcseit.com

Volume 10, Issue 7, May-June-2024 | Published Online : 20th June 2024

Dynamic Deep Learning : Advancements in Anomaly Detection through Deep Learning

D Roja Ramani[1], B Rajalakshmi[2], Abhinav D[3], Ashish Kumar Jha[3], Antony Marvic Murera[3]

[1]Associate Professor, Department of Computer Science and Engineering New Horizon College of Engineering Bangalore, India

[2]Professor, Department of Computer Science and Engineering New Horizon College of Engineering Bangalore, India

[3]Department of Computer Science and Engineering New Horizon College of Engineering Bangalore, India

***Abstract*—** Globally, the Internet spans across nations, but it faces looming threats of network attacks. The proliferation of information and its global accessibility has escalated risks to data integrity and confidentiality. Consequently, breaching security measures has become increasingly effortless. Consequently, there is a pressing need to bolster network security. Leveraging advancements in machine learning, numerous solutions have been proposed for phishing detection. In this study, we introduce a deep learning-driven framework tailored for identifying phishing websites. Our framework encompasses various machine learning methodologies, including logistic regression, Naive Bayes, K-Nearest Neighbors, and Decision Trees, to bolster intrusion detection. We rigorously evaluate the effectiveness of our approach through unit testing, system testing, white box testing, and black box testing across different algorithms to yield comprehensive experimental outcomes.

***Keywords* —** Deep learning, K-Nearest Neighbors and Decision Trees.

I. INTRODUCTION

In today's business landscape, information systems serve as the backbone of enterprises, regardless of their size or industry. However, the data stored and services offered by these systems make them prime targets for various forms of attacks. Many online services operate through user membership systems, requiring individuals to register and log in to access personalized features. Consequently, users must provide personal information to enjoy these convenient services. While network security technologies safeguard the transmission and storage of information in secure environments, cybercriminals employ diverse methods to attack and pilfer personal data. Given the breadth and specificity of these attacks, their consequences can be devastating, making computer security a paramount challenge. Consequently, research in this field is burgeoning, leading to the development of numerous tools and mechanisms to meet the safety demands of modern life.

A phishing website is a deceptive online platform designed to dupe users into disclosing sensitive information, such as login credentials, credit card details, or personal data. These fraudulent websites often mimic legitimate ones, including popular services like online banking and social media platforms. Their primary objective is to deceive unsuspecting users into believing they are interacting with authentic sites and then coax them into revealing confidential information. Phishing attacks typically occur via emails, social media messages, or other forms of communication that direct users to these fraudulent websites. Common dissemination methods include emails, text messages, and social media platforms. By employing social engineering tactics, attackers manipulate users into clicking on phishing links. Hence, detecting such risks through network security technology and alerting users represent effective countermeasures to prevent data leakage.

Phishing websites employ various strategies to appear genuine and reliable. They may replicate logos, layouts, and design elements from legitimate sites they impersonate. Additionally, they might create URLs closely resembling those of legitimate sites or use similar domain names. Upon landing on a phishing website, users may encounter prompts to enter login credentials, personal

information, or financial details. These platforms often utilize fake login forms, online forms soliciting personal data, or false security alerts to coerce users into providing their information promptly.

It's imperative to recognize that phishing websites are both illegal and malicious. Their sole purpose is to exploit unsuspecting individuals and compromise their sensitive information. Falling victim to a phishing attack can result in identity theft, financial loss, unauthorized account access, and other severe consequences.

II. RELATED WORK

Nearest Neighbour: It's a supervised machine learning method used for solving classification and regression problems. In this approach, certain hyperparameters are utilized, including 'leaf_size' ranging from 25 to 40, 'n_neighbors' ranging from 5 to 20, and 'p' taking values of 1 or 2. For the K-nearest neighbors (KNN) algorithm, the confusion matrix is represented as:

[[3186 284]

[247 4021]]

Naive Bayes: Naive Bayes classifiers encompass a set of classification algorithms rooted in Bayes' Theorem. Although not a singular algorithm, they share the common principle of assuming independence among features being classified. Gaussian Naive Bayes, a simple technique, is employed with default hyperparameters. The confusion matrix for Naive Bayes is given by:

[[3470 0]

[3097 1171]]

Decision Tree: Recognized as a potent tool for classification and prediction, the decision tree employs a flowchart-like structure. Each internal node signifies a test on an attribute, branches indicate outcomes of tests, and leaf nodes hold class labels. Hyperparameters such as 'max_leaf_nodes' ranging from 2 to 100 and 'min_samples_split' taking values of 2, 3, or 4 are utilized. The confusion matrix for the Decision Tree method is:

[[3193 277]

[210 4058]]

Gradient Boosting: This technique, applicable to regression and classification problems, forms a prediction model through an ensemble of weak prediction models, typically decision trees. Hyperparameters including 'min_samples_leaf' ranging from 0.1 to 0.5, 'max_depth' with values of 3, 5, or 8, 'max_features' taking either "log2" or "sqrt", and 'n_estimators' set to 10 are employed. The confusion matrix for Gradient Boosting is:

[[2880 590]

[107 4161]]

$$\sum(M + L) = x_n$$

$$\text{Input} = \sum_{n=0}^{m} x_n$$

$$\text{Malicious} = \text{Output}\_\text{RNN}(\text{Input}(\text{Pm}))$$

$$\text{Legitimate} = \text{Output}\_\text{RNN}(\text{Input}(\text{Pl}))$$

Fig. 1. Equation for neural network with LSTM for malicious Url

Let M and L ∈ xn signify the malicious and legitimate URLs, respectively, each containing the properties Pm and Pl, respectively. The proposed framework utilizes Recurrent Neural Network (RNN)—Long Short-Term Memory (LSTM) to identify the properties Pm and Pl in order to classify URLs as malicious or legitimate. Equations 1 to 4 outline the method for identifying malicious URLs.

The term "Recurrent Neural Network" encompasses two broad groups of networks with similar general structures: one with finite input and the other with infinite input. Both groups exhibit time dynamic behavior. A recurrent network with finite input forms a directed acyclic graph that can be replaced by a purely feedforward neural network, while a recurrent network with infinite input forms a directed cyclical graph that cannot be altered. LSTM is a modified version of RNN, serving as a deep learning technique that addresses the gradient problem encountered by RNNs. LSTM employs multiple gates to enhance performance and prevents backpropagation. Each input to LSTM generates an output, which then serves as input for the subsequent layer or module of LSTM. Equations 1 to 4 elucidate the core concept of the proposed study.

Figure 2 illustrates the data collection processes employed in this study. Data repositories such as Phishtank and a custom crawler are utilized to gather both malicious and benign URLs. Specifically, the crawler is designed to extract URLs from the

AlexaRank website, which publishes a set of URLs along with their rankings to support the research community. Between June 2020 and November 2020, the crawler collected a total of 7658 URLs from AlexaRank. Additionally, 6042 URLs were obtained through the Phishtank datasets. Throughout the data collection process, the extracted data are stored in a repository denoted as W and returned as W1, along with the number of URLs (N) collected.

Input: Data Repositories
Output: Raw Data

1: **procedure** DATA COLLECTION
2: $W \leftarrow$ ExtractData(Repositories)
3: $W1 \leftarrow$ FilterInvalidURL(W)
4: $N \leftarrow$ Count(W1)
5: **return** $W1$,N
6: **end procedure**

Fig. 2. Algorithm for Data Collection Process for Malicious and Benign URLs

III. A REVIEW OF THE LITERATURE

This pioneering research delves into the realm of phishing website detection, leveraging cutting-edge deep learning methodologies. It utilizes the variational autoencoder (VAE) to extract key features from URLs and employs a deep neural network (DNN) for effective classification. While achieving an impressive accuracy of 97.85%, the study emphasizes addressing a slightly higher false positive rate (2.19%), with plans to integrate generative modeling for further improvements [1]. Additionally, the introduction of the SI-BBA algorithm demonstrates a commendable 94.8% classification accuracy in distinguishing phishing websites, with future efforts focusing on fine-tuning parameters like epochs and learning rate [2].

This research introduces an advanced system for detecting phishing websites but faces challenges in verifying URL active status, impacting overall accuracy. Addressing this involves refining feature engineering and accelerating training to validate website states effectively [5]. Furthermore, in the domain of intrusion detection, the study presents an enhanced approach based on recurrent neural networks (RNNs) and conducts a comparative analysis of intrusion detection methods using a reduced subset of the KDD-99 dataset, comprising 1,000 cases [6].

In the realm of intrusion detection, the study unveils RNN-IDS, a pioneering system grounded in recurrent neural networks. Demonstrating superior performance in both binary and multiclass classifications, RNN-IDS emerges as a robust model that outperforms traditional techniques [8]. This study marks a significant milestone by integrating various machine learning methodologies comprehensively for phishing website detection. Utilizing a categorical framework and the XG Boost algorithm, it achieves an exceptional accuracy of 99.05% with the lowest false positive rate, highlighting the importance of augmented training data in enhancing classifier performance [9].

IV. PROPOSED METHODOLOGY

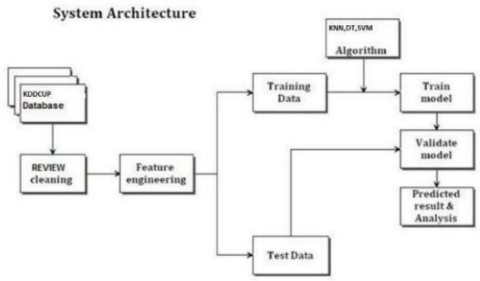

Fig. 3. Proposed System

The methodology outlined integrates various aspects of system architecture design, diagrammatic representation, and testing, emphasizing the significance of comprehending the goals, content, users, and navigation philosophy of a web application for designing an appropriate system architecture. This involves crafting the overall hypermedia structure, organizing content for presentation and navigation, and defining the structure of the web application to manage user interaction, internal processing, navigation, and content presentation.

The system follows the following procedure:

i. Data Collection: This initial step is critical in intrusion detection. The type of data source and where the data is collected significantly influence the design and effectiveness of an Intrusion Detection System (IDS). The study aims to enhance the security level of targeted hosts or networks by recommending a network-based IDS to implement suggested strategies. The recommended IDS analyzes incoming network traffic on the router nearest to the victim(s). Data samples obtained are categorized based on transport/Internet layer

protocols and labeled during the training stage according to domain knowledge. However, during the testing phase, data are only categorized based on protocol types.

ii. Data Preprocessing: Data collected in the previous phase are processed to extract fundamental features, akin to those in the KDD Cup 99 dataset. This phase comprises three main stages.

iii. Data Transferring: The trained classifier needs to represent each record in the input data as a vector of real numbers. Therefore, each symbolic feature in the dataset is assigned a numerical value. For example, in the KDD CUP 99 dataset, symbolic and numerical properties coexist. Symbolic properties like protocol type (TCP, UDP, ICMP), service type (HTTP, FTP, Telnet), and TCP status flag (SF, REJ) are converted into numeric values.

iv. Data Normalization: After converting symbolic attributes into numerical values, normalization becomes crucial. It involves scaling each attribute's value into a proportional range, thereby removing bias towards features with higher values. Each feature in a record is normalized within the [0-1] range. This ensures consistency in the test data for comparison with other systems.

v. Feature Selection: Although datasets contain numerous attributes, only certain features are essential for building an IDS. Identifying the most useful aspects of traffic data is crucial for optimal performance. The study employs a method to determine the optimal number of required features for training a classifier, ultimately chosen based on achieving the highest classification accuracy in the training dataset. Additionally, the suggested feature selection algorithm is applied to the various classes in the KDD Cup 99 dataset.

V. WORKFLOW OF PROPOSED SYSTEM

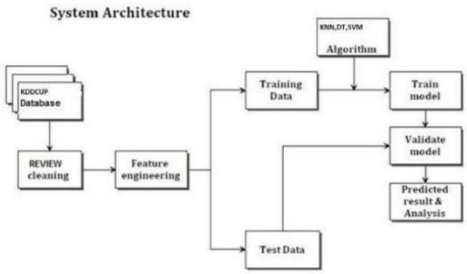

Fig. 4. Proposed System

A data flow diagram (DFD) provides a visual representation of how information flows through a data system, depicting its processing components. It is often used as an initial step to provide a high-level overview of the system before delving into details, which can be addressed later. Additionally, DFDs serve as a graphical tool for presenting data processing. They illustrate the types of input and output data in the system, the direction of data flow through the system, and where the data will ultimately be stored.

VI. RESULT AND DISCUSSIONS

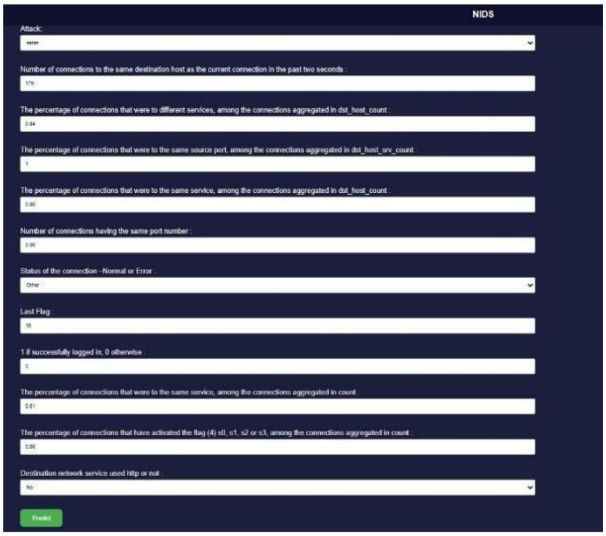

Fig. 5. Performance Result and Evaluation metrics. reference [5]

Fig. 6. Performance Result and Evaluation metrics. reference [5]

Above are the snapshots of the implementation outputs. There are more than 40 attributes for detecting the intrusion in the network. By applying Feature extraction, we use only the important attributes. The values for the attributes are given as provided in the KDD dataset. Here are a few of the attributes used in detecting the intrusion:

Attack: Satan: The "Attack" feature indicates the specific type of attack being

performed. In this case, the attack is labeled as "Satan."

Number of connections to the same destination host as the current connection in the past two seconds: 175: This feature represents the count of connections made to the same destination host within the past two seconds.

Percentage of connections that were to different services, among the connections aggregated in dst_host_count: 0.84: This feature calculates the percentage of connections made to different services among the aggregated connections to the destination host. If most connections are targeting different services, it could suggest a scanning or reconnaissance activity, which is common in probe attacks.

Number of connections having the same port number: 1: This feature indicates the count of connections that have the same port number.

Status of the connection - Normal or Error: Other: This feature represents the status of the connection, which is labeled as "Other" in this case. It indicates that the connection status does not fall into the predefined categories of "Normal" or "Error."

Last Flag: 18: The "Last Flag" feature captures the last observed flag in the network connection, potentially aiding in attack classification.

1 if successfully logged in; 0 otherwise: 0: This feature indicates whether the connection was successfully logged in or not. A value of 1 typically means successful login, while 0 means unsuccessful or no login attempt was made. It can provide information about the authentication status of the connection.

Percentage of connections that were similar to the service, among the connections aggregated in count: 0.01: This feature calculates the percentage of connections made similar in service among the aggregated connections. A low percentage suggests that the connections are spread across different services, which might be an indication of probing or scanning behavior.

Percentage of connections that have activated the flag (4) s0, s1, s2, or s3, among the connections aggregated in count: 0.10: This feature calculates the percentage of connections that have activated specific flags (s0, s1, s2, or s3) among aggregated connections. These flags often represent abnormal or suspicious activities in the network.

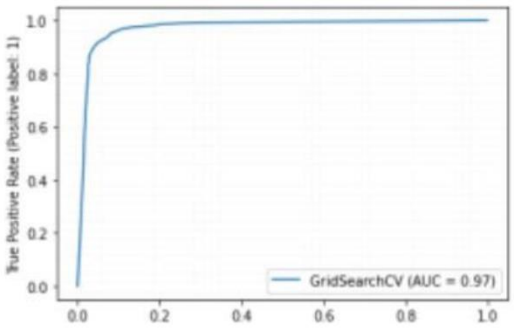

Fig. 7. ROC curve of KNN

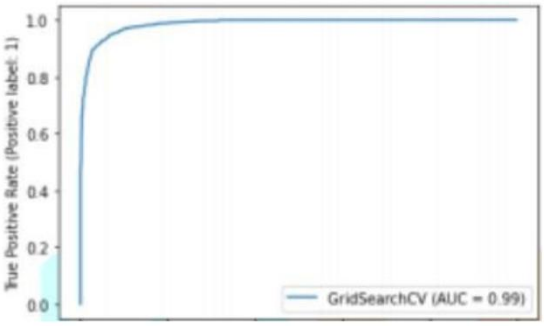

Fig. 8. ROC curve for Naive Bayes

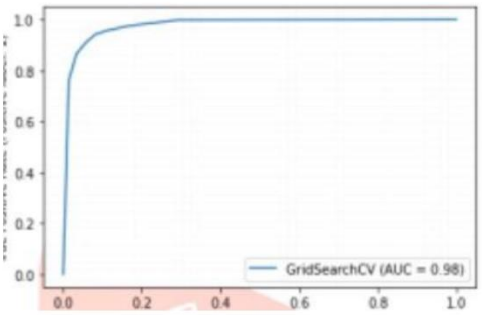

Fig. 9. ROC curve for Decition Tree

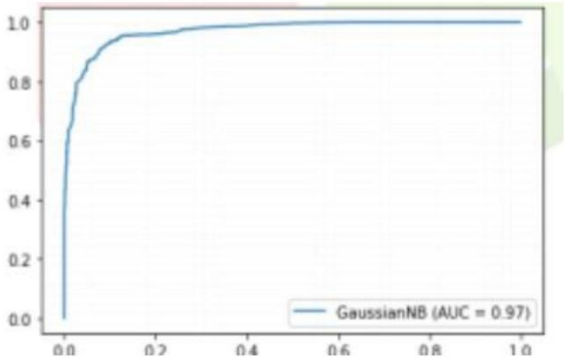

Fig. 10. ROC curve of Gradient Boosting

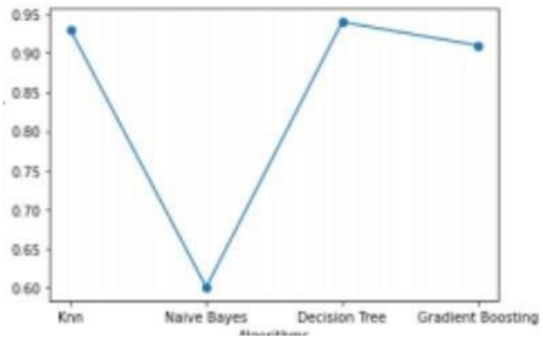

Fig. 11. Comparison of f1-Score accuracy of all algorithms

VII. CONCLUSION

In recent years, numerous machine learning-based solutions have been proposed to address phishing attacks. However, these solutions often lack verification in live browsing environments, and there is a notable absence of analysis and research on products for phishing detection. In this paper, we introduce a framework for phishing detection in a real-time browsing environment. We present a practical and efficient Network Intrusion Detection System (NIDS) utilizing classification and deep learning techniques, which can also be applied to other machine learning algorithms and existing systems. NIDS offers a realistic and functional solution by providing a specific defense definition in our large and contemporary networks, addressing potential uncertainties in computer and networking programs.

Key features of our framework include:

Real-time operation without delays, displaying prediction results when web pages are opened.

Trackable experimental data, with an automated model training process and storage of execution results in a real-time database.

Development of a browser extension as a client product accessible to ordinary users.

Extendable implementation of predictive services, allowing for the combination of individual detection services such as blacklist filtering or computer vision.

Independent feature extraction process in the deep learning model, avoiding reliance on third-party services.

Utilization of closed-loop data to enhance machine learning model performance, leveraging high-quality feedback data from users for improved accuracy and sensitivity.

In the future, we plan to deploy the entire system on a cloud platform, configuring machines with NVIDIA GPUs for model training to enhance efficiency through GPU parallel computing power. Users will be able to download the extension via the Chrome Web Store, and we aim to implement our framework as a plug-in for other browsers as well.

REFERENCES

1. Cabaj, K.; Domingos, D.; Kotulski, Z.; Respício, A. Cybersecurity Education: Evolution of the Discipline and Analysis of Master Programs. Comput. Secur. 2018, 75, 24–35. [Google Scholar] [CrossRef]
2. Iwendi, C.; Jalil, Z.; Javed, A.R.; Reddy, G.T.; Kaluri, R.; Srivastava, G.; Jo, O. KeySplitWatermark: Zero Watermarking Algorithm for Software Protection Against Cyber-Attacks. IEEE Access 2020, 8, 72650–72660. [Google Scholar] [CrossRef]
3. Rehman Javed, A.; Jalil, Z.; Atif Moqurrab, S.; Abbas, S.; Liu, X. Ensemble Adaboost Classifier for Accurate and Fast Detection of Botnet Attacks in Connected Vehicles. Trans. Emerg. Telecommun. Technol. 2020, 33, e4088. [Google Scholar] [CrossRef]
4. Conklin, W.A.; Cline, R.E.; Roosa, T. Re-Engineering Cybersecurity Education in the US: An Analysis of the Critical Factors. In Proceedings of the 2014 47th Hawaii International Conference on System Sciences, IEEE, Waikoloa, HI, USA, 6–9 January 2014; pp. 2006–2014. [Google Scholar]
5. Javed, A.R.; Usman, M.; Rehman, S.U.; Khan, M.U.; Haghighi, M.S. Anomaly Detection in Automated Vehicles Using Multistage Attention-Based Convolutional Neural Network. IEEE Trans. Intell. Transp. Syst. 2021, 22, 4291–4300. [Google Scholar] [CrossRef]
6. Mittal, M.; Iwendi, C.; Khan, S.; Rehman Javed, A. Analysis of Security and Energy Efficiency for Shortest Route Discovery in Low-energy Adaptive Clustering Hierarchy Protocol Using Levenberg-Marquardt Neural Network and Gated Recurrent Unit for Intrusion Detection System. Trans. Emerg. Telecommun. Technol. 2020, 32, e3997. [Google Scholar] [CrossRef]

7. Bleau, H.; Global Fraud and Cybercrime Forecast. Retrieved RSA 2017. Available online: https://www.rsa.com/en-us/resources/2017-global-fraud (accessed on 19 November 2021).
8. Computer Fraud & Security. APWG: Phishing Activity Trends Report Q4 2018. Comput. Fraud Secur. 2019, 2019, 4. [Google Scholar] [CrossRef]
9. Hulten, G.J.; Rehfuss, P.S.; Rounthwaite, R.; Goodman, J.T.; Seshadrinathan, G.; Penta, A.P.; Mishra, M.; Deyo, R.C.; Haber, E.J.; Snelling, D.A.W. Finding Phishing Sites; Google Patents: Microsoft Corporation, Redmond, WA, USA, 2014. [Google Scholar]
10. What Is Phishing and How to Spot a Potential Phishing Attack. PsycEXTRA Dataset. Available online: https://www.imperva.com/learn/application-security/phishing-attack-scam/ (accessed on 20 November 2021).
11. Gupta, B.B.; Tewari, A.; Jain, A.K.; Agrawal, D.P. Fighting against Phishing Attacks: State of the Art and Future Challenges. Neural Comput. Appl. 2016, 28, 3629–3654. [Google Scholar] [CrossRef]
12. Zhu, E.; Ju, Y.; Chen, Z.; Liu, F.; Fang, X. DTOF-ANN: An Artificial Neural Network Phishing Detection Model Based on Decision Tree and Optimal Features. Appl. Soft Comput. 2020, 95, 106505. [Google Scholar] [CrossRef]
13. Machine Learning Decision Tree Classification Algorithm—Javatpoint. Available online: https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm (accessed on 25 November 2021).
14. Breiman, L. Random Forests. Mach. Learn. 2001, 45, 5–32. [Google Scholar] [CrossRef] [Green Version]
15. Friedman, J.H. The Elements of Statistical Learning: Data Mining, Inference, and Prediction; Springer Open: Berlin/Heidelberg, Germany, 2017. [Google Scholar]
16. Mr.Gunjal Somnath P, Prof. Aher S M proposed "Network Intrusion Detection using Recurrent Neural Network Algorithm, Vol 7, July 2020. DOI: E-ISSN 2277 – 4106, P-ISSN 2347–5161, https://www.irjet.net/archives/V7/i7/IRJET-V7I7381.pdf.
17. Phishing Website Detection Based on Multidimensional Features Driven by Deep Learning, pp. 15196 - 15209, 11 January 2019, DOI: 10.1109/ACCESS.2019.2892066, https://ieeexplore.ieee.org/document/8610190.
18. Phishing Website Detection using Deep Learning, pp. 83-88, 11 August 2022, DOI: 10.1145/3542954.3542967, https://dl.acm.org/doi/abs/10.1145/3542954.3542967.
19. Phishing Website Detection Using Machine Learning, pp. 45-47, Jan 2023, DOI: 10.5120/ijca2018918026, https://www.researchgate.net/publication/328541785_Phishing_Website_Detection_using_Machine_Learning_Algorithms.