Phishing Website Detection Using Machine Learning

Krithika Kumar[1], Hamsa Gowlikar[2], Anusha Kommarajula[3]
[1,2,3]*Department of Computer Science and Engineering, Stanley College of Engineering and Technology for Women, Hyderabad, India*

Abstract—**Phishing attacks represent a major danger for international cybersecurity efforts. Hackers have developed better social engineering methods which they use to trick people into sharing their confidential financial details through fake websites. The existing detection methods which include rule-based systems and manual blacklisting fail to identify new zero-day attacks and emerging URL threats. This research presents an innovative detection system which uses machine learning to analyze URLs in real time based on their structural characteristics. The project collects multiple numerical features for training purposes which will be used to develop five classification algorithms including Logistic Regression Random Forest Decision Tree K-Nearest Neighbours and Linear Support Vector Machine. The data set provides character ratios and entropy measurements and URL length information from more than 566000 unique URLs. The results show high effectiveness across multiple performance indicators, including F1-score and accuracy and precision and recall. The K-Nearest Neighbours (KNN) algorithm achieved an accuracy of 96.39% while it produced an F1 score of 0.9137 which made it superior to all other algorithms. The most precise model operates through a browser extension which notifies users about dangerous websites. The demonstration shows practical applications of the system. The research demonstrates that automated threat detection through machine learning protection systems provides better security for users in digital environments.**

Index Terms—**Classification Algorithms, Machine Learning, Phishing Website Detection, Cybersecurity, K-Nearest Neighbours, URL Feature Analysis.**

I. INTRODUCTION

Phishing attacks remain a major online threat because cybercriminals create fake websites which imitate real sites to obtain user passwords and sensitive data. The constant emergence of new website domains renders conventional security systems which rely on blacklists ineffective because these systems cannot defend against new security threats. Researches are using machine learning and deep learning techniques to build precise fraud detection systems which they will evaluate through their current investigations.

The findings from various research studies have developed improved detection methods. The PhiDMA model established a five-layer system which combined whitelisting with lexical patterns to achieve an accuracy of 92.72% [1]. Researchers focus on developing new predictive machine learning models which require less computation resources [2] while Gradient Boosting Classifiers enable users to recognize minute differences which standard methods cannot identify [3]. SVM and Random Forest algorithm comparison shows how ML technology benefits this specific domain [4]. Current research on "Evasion Space" demonstrates how attackers use URL manipulation to bypass detection systems which results in significant drops in detection efficiency [5]. Deep learning-based real-time frameworks deliver effective solutions because their systems can generate alerts within six seconds [6]. Today architectural systems use advanced technology to create accurate results through their enhanced design abilities.

The URL classification algorithm achieves high performance through its combination of ResMLP methods and inverted residual block design [7]. At the character level, CNN models enable end-users running web content identifiers without necessarily having to look at the source code of web pages [8]. The research examined two methods to handle unbalanced data sets through SMOTE and fine-tuning techniques [9]. The combination of WordNet-based lemmatization with LDA topic modeling yields almost 98.85 per cent accuracy [10].Recent evaluations continue to prove that ML is essential for protecting online transactions [11]. PhishHaven uses ensemble learning to identify phishing URLs which are created by both humans and artificial intelligence [12].

Feature selection remains critical factor research indicates that Random Forest models achieve 98.23% accuracy through optimized selection techniques [13]. The complete detection methods overview exists through comprehensive surveys which divide detection methods into two groups: list-based approaches and similarity-based approaches [14].The ongoing research examines the structural components of URLs together with its methods of hybrid optimization. In 2025, research indicated that the detection to be reliable, and the works were primarily related to URLbased characteristics [15].Advanced deep learning models serve as the main defence system against emerging security threats [16].while researchers focus on developing optimization methods that improve system performance [17]. The development of hybrid deep learning techniques enables real-time applications to deliver user protection without any time delay [18]. The combination of Deep CNNs with Random Forests in ensemble systems achieves exceptional performance because these systems successfully detect both local patterns and categorical features which leads to improved classification results [19].

II. LITERATURE SURVEY

Multiple researchers have examined different methods to detect phishing websites and malicious URLs using machine learning and artificial intelligence technologies. Their primary objective involved developing robust systems which would identify fraudulent websites to protect users from potential cyber threats.

The study shows that analyzing website domain URL attributes is a great way to explore websites. The technology uses machine learning to identify hazardous websites through its ability to learn special website attributes which conventional detection methods cannot detect. The group developed a method which uses feature selection together with K-Means clustering to identify important website features that they use to detect unusual patterns. When compared to conventional techniques, the researchers discovered that feature selection technology increased detection speed and accuracy [24].

The field of architecture research continuously introduces new architectural designs which include Graph Convolutional Networks which enable phishing detection. The system protects users from online threats by monitoring internet traffic to detect dangerous behaviour patterns which it identifies as hazardous [25]. The research team evaluated how various language models functioned by testing 21 different model versions. The researchers demonstrated that "few-shot" prompting which requires the use of specific examples to create questions produced best results when used with certain models which achieved 91% accuracy [26].

Alhaidari and his team developed a specific machine learning approach to detect phishing websites containing Arabic content. The research team examined a dataset which included 4048 websites to determine which websites were either malicious or legitimate through their analysis. Through their implementation of Random Forest and XGBoost algorithms, they discovered that Random Forest achieved the highest performance with an accuracy rate of 92.96%. The researchers developed a Chrome extension which enables users to detect unsafe websites as part their efforts to create public access to their technology [20]. Remya and his team developed a system which they named BGL-PhishNet. The system uses BERT to assess website content while its Graph Neural Networks study the website's design and LightGBM assesses the website's duration of existence. The system detected elements through its multiple assessments which resulted in a detection success rate of 97% [21].

Ghafoor and his team studied attacker behavior by investigating how attackers create and improve their phishing websites to bypass detection systems. The researchers observed that attackers have developed more advanced techniques to hide their activities. The researchers created BYPHISH to evaluate evasion techniques which succeeded in hiding phishing websites from detection for 80 to 85 of the time. The research demonstrates that detection systems need ongoing development to achieve effective performance [22]. Another research group investigated the use of intelligence-based models for URL detection. The researchers built a browser tool which protects users through real-time monitoring of malicious URLs after they collected data and analyzed features and tested different machine learning models [23]. The research shows that artificial intelligence acts as a vital protection mechanism which defends against phishing attacks.

The second approach uses a system which operates on website content by studying paragraph structures and HTML tags to detect fraudulent activities. The content-based machine learning model achieved outstanding performance results when evaluated with benchmark datasets [27]. Many developers now use deep learning in conjunction with machine learning to detect trends and anomalies in the data. The security team uses clustering techniques together with decision tree algorithms to identify websites that show suspicious activities. Organizations experience difficulties because their systems lack the agility needed to combat new phishing techniques despite technological advancements. Research studies show that ongoing artificial intelligence technology integration creates secure online systems which benefit all users [28].

III. PROPOSED METHODOLOGY

DATASET DESCRIPTION

The research developed a phishing URL detection system by using three different datasets to create the system. The use of multiple datasets. The model can learn to identify patterns that exist in both phishing sites and trustworthy sites. The first dataset contains URLs along with other features that describe their structure. The characteristics of the URL include its length and the number of digits and letters and special characters and subdomains and various ratio- based elements which show how the URL is structured. The dataset contains a label which shows whether the URL is a phishing site or an authentic site.

The structure of the second dataset was marginally different. The dataset divided the URL into protocol, domain, and path columns in addition to a number of other website-related attributes rather than storing the entire URL in a single column. These columns were combined to reconstruct the entire URL so that the URLs could be used for analysis. To ensure that the dataset matched the format of the others, only the URL and label columns were kept after the complete URL was constructed.

The third dataset contains a big set of URLs which researchers divided into two groups of legitimate URLs and phishing URLs. The dataset contains labels which researchers stored in a column called "type" using text values. The research team converted the labels into numerical values for machine learning algorithms which use 0 to represent legitimate URLs and 1 to represent phishing URLs.

The panda’s concatenation method was used to combine the three datasets. The resulting dataset contained more than 567000 URLs. About 566000 unique URLs remained after duplicate URLs were eliminated to prevent duplicate entries. The dataset size improves both the performance and reliability of the phishing detection model.

DATA PREPROCESSING

The process of machine learning model training begins with data preparation, which ensures that all analysis-ready data will be stored in its initial form while its complete and understandable form remains accessible. The researchers used the Panda library to load three different datasets that they obtained from their research; they looked at each dataset's column structure because the datasets existed in different structural formats; they selected essential columns to maintain consistency across all datasets; and they removed all feature columns from the initial dataset except the URL and label columns.

The second dataset's URL required the extraction of its three components which included protocol and domain and path elements. The entire URL restoration process required the combination of these two specific columns. The dataset maintained its initial structure because the URL reconstruction method left only the URL and label columns. The third dataset maintained its labels in text format which included the terms "phishing" and "legitimate." The data were converted into numerical labels which assigned 0 to legal URLs and 1 to phishing URLs.

The pandas concat function was used to merge the standardized structures of the three datasets. The team removed duplicates from the dataset to ensure that each URL appeared only once in the final dataset. The approach produced a unified dataset which enabled professionals to modify feature data and build models correctly.

FEATURE ENGINEERING

Feature engineering enables machine learning models to discover data patterns when unprocessed data transforms into useful features. The system requires transformed text strings because machine learning algorithms lack the ability to process URLs in their original form. The extraction process created multiple numerical attributes which defined the distinct characteristics and structural elements of each URL. The project needed the Python libraries re math and urllib.parse to develop an original feature extraction system. The URL parse module enabled the team to deconstruct each URL into its basic elements which included the domain and path and query parameters. The elements created multiple features which resulted in multiple output results. The URL structure analysis requires three main elements which include the URL length and domain length and subdomain count.

The system needs to check both HTTPS usage and domain-to-IP address mapping because both scenarios indicate potential suspicious behavior. The system obtained character-based URL data by measuring all letters and numbers and special symbols present in the URL. The research determined how often specific symbols occurred in the study by counting their times they appeared, including dots (.), dashes (-), underscores (_), question marks (?), and ampersands (&). The feature extraction process analyzed URL structure through its calculation of multiple ratio-based features which included letter ratios and digit ratios and special character ratios. The system used entropy as a vital element to assess URL complexity because it helps identify suspicious automated links.

The research team extracted additional structural features which included both query length and path length measurements. The team created a new dataset which contained both the characteristics of the study and their corresponding labels. The researchers used the structured dataset to train machine learning models which could detect phishing URLs.

FEATURE SCALING

Feature scaling guarantees that all features will exhibit equivalent measurement ranges. Machine learning datasets contain multiple features which display significant value differences. The learning process will be controlled by high-value features when scaling is not applied which results in negative effects on model accuracy. StandardScaler from the scikit- learn library was utilized to solve this problem. StandardScaler transforms data by converting each feature to a distribution which has zero mean and one standard deviation. This normalization procedure during model training ensures that all features maintain equal impact on the model. Machine learning algorithms achieve better performance through feature scaling which enables faster training progress.

MODEL TRAINING

The dataset became ready for model training after the completion of preprocessing and feature extraction and feature scaling processes. The dataset was first divided into two parts through the train-test split method which created a training set and a testing set. The model training process typically used 80 percent of the data while testing used 20 percent of the data. The model is trained on a given set of training data in order to learn the pattern of the features obtained during extraction; eventually, the model is tested on some testing data set to draw conclusions about the knowledge transferability while reaching out class assignments over unseen data. The project evaluated various machine learning algorithms which included K- Nearest Neighbours (KNN) Random Forest Decision Tree Logistic Regression and Support Vector Machine. The data is developed for the development of the system so that models could determine when a URL was considered as a phishing attack and when it was considered to represent a valid website. The model performance evaluation requires performance metrics to be compared against actual data results. The evaluation process used accuracy as the main metric to assess how many URLs the model successfully detected. The evaluation process identified the final model for the phishing detection system as the best option which achieved both highest accuracy and superior detection capabilities. The system established enhanced reliability through its testing on multiple datasets while the model achieved better phishing URL detection through its complete feature extraction and data cleaning efforts.

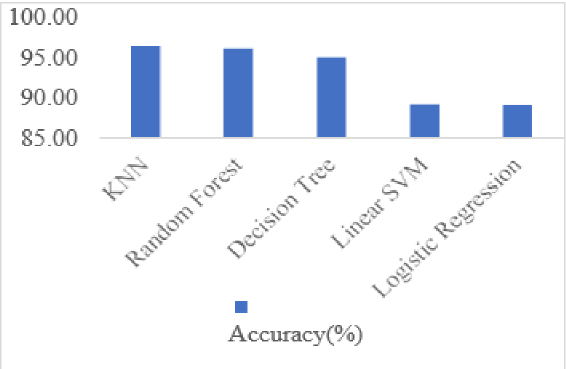

Figure I. Accuracy Comparison of Machine Learning Models

TABLE I shows the precision values of various machine learning approaches. It can be observed that the KNN classifier achieves the highest precision value of 0.935924 when compared with other evaluated models such as Random Forest, Decision Tree, Linear SVM, and Logistic Regression, it is represented in Figure II.

Precision: Precision shows how many of the URLs predicted as phishing were actually phishing. It measures how "trustworthy" the model is when it raises a phishing alert.

$$\text{Precision} = \frac{TP}{TP+FP} \qquad ---- (2)$$

Table I. Precision Comparison of Different Machine Learning Models

S.No	Model	Precision (%)
1	KNN	93.59%
2	Random Forest	91.50%
3	Decision Tree	88.29%
4	Linear SVM	70.48%
5	Logistic Regression	69.99%

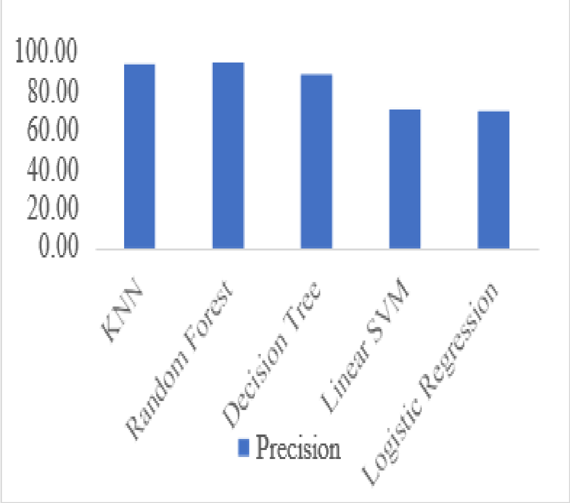

Figure II. Precision Comparison of Machine Learning Models

IV. RESULT ANALYSIS

EXPERIMENTAL SETUP:

The project was operated on a Windows 11 laptop which had an Intel i7 13th generation processor and 16GB of RAM. The system could handle our machine learning experiments because it contained sufficient processing power to handle our entire dataset. And selected Jupyter Notebook as our data processing tool because it provides excellent features for creating and testing our work with visual demonstrations. Used Visual Studio Code for our browser extension development because it offered all the necessary coding tools. The system maintenance system allowed us to build the entire system without any difficulties. Our research experiments reached completion because our hardware and software components delivered essential findings.

Accuracy: Accuracy tells us how many predictions the model got right out of all predictions it made. It gives an overall idea of the model's performance.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \qquad ---- (1)$$

Table II. Accuracy Comparison of Different Machine Learning Models

S.No	Model	Accuracy (%)
1	KNN	96.40%
2	Random Forest	96.10%
3	Decision Tree	94.97%
4	Linear SVM	89.09%
5	Logistic Regression	89.01%

TABLE II compares the accuracy values of various machine learning models. From the table, we can see that the KNN classifier achieves the best performance with an accuracy of 0.9639, showing better overall performance than the other evaluated models such as Random Forest, Decision Tree, Linear SVM, and Logistic Regression. The results are represented graphically in Figure I.

Recall: Recall measures the proportion of actual positives which a model successfully identified. The metric requires complete accuracy because it determines success through its capacity to identify all actual positive cases.

$$\text{Recall} = \frac{TP}{TP+FN} \qquad -----(3)$$

Table III. Recall Comparison of Different Machine Learning Models

S.No	Model	Recall (%)
1	KNN	90.13%
2	Random Forest	89.25%
3	Decision Tree	88.17%
4	Linear SVM	84.23%

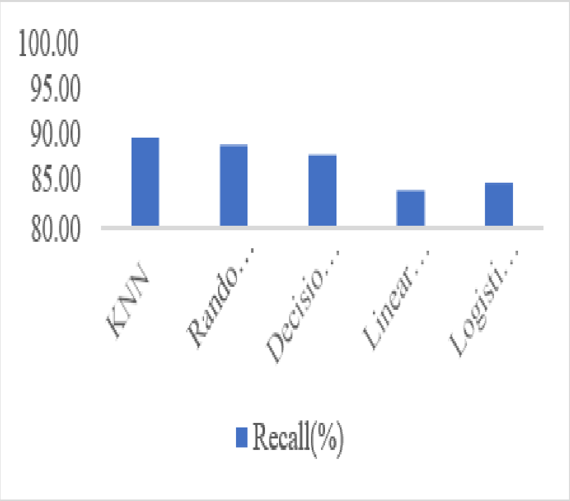

Figure III. Recall Comparison of Machine Learning Models

TABLE III presents a comparison between different machine learning models through their respective recall measurement results. The KNN classifier shows a recall value of 0.9013 which demonstrates superior performance when compared to the other evaluated methods that include methods that include Random Forest Decision Tree Linear SVM and Logistic Regression as shown in Figure III.

F1-Score: The F1 score serves as a machine learning evaluation metric which assesses classification models by calculating the harmonic mean of precision and recall. The score ranges from 0 to 1 which means that higher scores indicate better model quality through their assessment of both false positive and false negative errors.

$$\text{F1 Score} = 2 \times \frac{(Precision \times Recall)}{(Precision + Recall)} \quad \text{----} (4)$$

Table IV. F1-Score Comparison of Different Machine Learning Models

S.No	Model	F1-Score (%)
1	KNN	91.36%
2	Random Forest	90.80%
3	Decision Tree	88.23%
4	Linear SVM	76.74%
5	Logistic Regression	76.77%

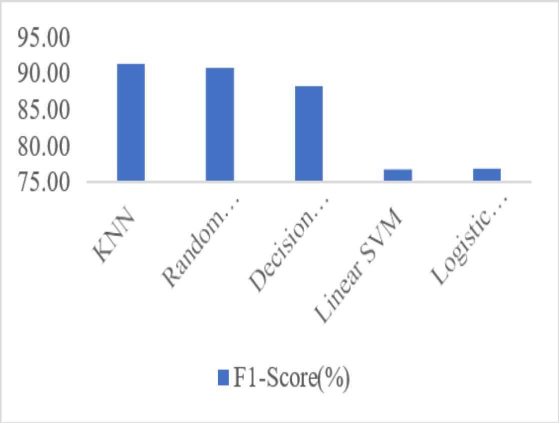

Figure IV. F1-Score Comparison of Machine Learning Models

The F1-score performance of different machine learning models appears in TABLE IV. The KNN classifier achieves its best F1-score performance of 0.9137 because it exhibits better overall precision and recall balance when compared to all other tested models which include Random Forest, Decision Tree, Linear SVM, and Logistic Regression. The results are represented graphically in Figure IV.

V. CONCLUSION

It is a well-spread cyber-threat of phishing because the fakers are spoofing the original site and making a copy of the fake site to log in to their account to fetch and steal confidential information. Blacklists and other traditional defences are no longer sufficient because new phishing sites create constant online threats. The project develops a machine-learning-based system which detects phishing URLs to address this research gap. The combined three datasets which contained both authentic links and malicious links generated over 566000 unique records. The team extracted URL length domain structure character patterns and four additional features which included ratios and entropy from the data after they processed it through duplicate removal. The team used StandardScaler to transform the data because this method improved model training efficiency. The researchers evaluated multiple machine learning algorithms which included KNN Random Forest Decision Tree Logistic Regression and Linear SVM. The KNN model attained superior performance compared to all other models through its measurement of precision and recall together with F1-score calculations and its achievement of 96.39 percent accuracy. The developed system functions as a browser extension which protects users from phishing attacks while alerting them to potentially dangerous websites. The research shows that machine learning systems can successfully detect phishing websites when researchers use advanced preprocessing methods together with their selected feature extraction techniques.

REFERENCES

[1] G. Sonowal and K. S. Kuppusamy, "PhiDMA – A phishing detection model with multi-filter approach," J. King Saud Univ. - Comput. Inf. Sci., vol. 32, no. 1, pp. 99–112, 2020, doi: 10.1016/j.jksuci.2017.07.005.

[2] A. A. Orunsolu, A. S. Sodiya, and A. T. Akinwale, "A predictive model for phishing detection," J. King Saud Univ. - Comput. Inf. Sci., vol. 34, no. 2, pp. 232–247, 2022, doi: 10.1016/j.jksuci.2019.12.005.

[3] K. Omari, "Phishing Detection using Gradient Boosting Classifier," Procedia Comput. Sci., vol. 230, pp. 120–127, 2023, doi: 10.1016/j.procs.2023.12.067.

[4] S. Abu-Nimeh, D. Nappa, X. Wang, and S. Nair, "A comparison of machine learning techniques for phishing detection," in Proc. Anti-Phishing Working Groups 2nd Annu. eCrime Researchers Summit (eCrime '07), 2007, pp. 60–69, doi: 10.1145/1299015.1299021.

[5] Y. Yuan, G. Apruzzese, and M. Conti, "Multi-SpacePhish: Extending the Evasion-space of Adversarial Attacks against Phishing Website Detectors Using Machine Learning," Digital Threats, vol. 5, no. 2, Art. no. 16, Jun. 2024, doi: 10.1145/3638253.

[6] L. Tang and Q. H. Mahmoud, "A Deep Learning-Based Framework for Phishing Website Detection," IEEE Access, vol. 10, pp. 1509–1521, 2022, doi: 10.1109/ACCESS.2021.3137636.

[7] S. Remya, M. J. Pillai, K. K. Nair, S. Rama Subbareddy, and Y. Y. Cho, "An Effective Detection Approach for Phishing URL Using ResMLP," IEEE Access, vol. 12, pp. 79367–79382, 2024, doi: 10.1109/ACCESS.2024.3409049.

[8] A. Aljofey, Q. Jiang, Q. Qu, M. Huang, and J.-P. Niyigena, "An Effective Phishing Detection Model Based on Character Level Convolutional Neural Network from URL," Electronics, vol. 9, no. 9, Art. no. 1514, 2020, doi: 10.3390/electronics9091514.

[9] S. R. Abdul Samad et al., "Analysis of the Performance Impact of Fine-Tuned Machine Learning Model for Phishing URL Detection," Electronics, vol. 12, no. 7, Art. no. 1642, 2023, doi: 10.3390/electronics12071642.

[10] E. S. Gualberto, R. T. De Sousa, T. P. De B. Vieira, J. P. C. L. Da Costa, and C. G. Duque, "From Feature Engineering and Topics Models to Enhanced Prediction Rates in Phishing Detection," IEEE Access, vol. 8, pp. 76368–76385, 2020, doi: 10.1109/ACCESS.2020.2989126.

[11] N. F. Almujahid, M. A. Haq, and M. Alshehri, "Comparative evaluation of machine learning algorithms for phishing site detection," PeerJ Comput. Sci., vol. 10, Art. no. e2131, 2024, doi: 10.7717/peerj-cs.2131.

[12] M. Sameen, K. Han, and S. O. Hwang, "PhishHaven—An Efficient Real-Time AI Phishing URLs Detection System," IEEE Access, vol. 8, pp. 83425–83443, 2020, doi: 10.1109/ACCESS.2020.2991403.

[13] M. S. Karim, K. Mustofa, S. B. Belhaouari, and S. R. K. Joga, "Phishing Detection System Through Hybrid Machine Learning Based on URL," IEEE Access, vol. 11, pp. 36805–36822, 2023, doi: 10.1109/ACCESS.2023.3252366.

[14] R. Zieni, L. Massari, and M. C. Calzarossa, "Phishing or Not Phishing? A Survey on the Detection of Phishing Websites," IEEE Access, vol. 11, pp. 18499–18519, 2023, doi: 10.1109/ACCESS.2023.3247135.

[15] V. Kumar, A. Prathiba, A. Ashritha, N. H.

Reddy, and X. S. A. Shiny, "Phishing Website Detection Based on URL Features," Int. J. Sci. Res. Eng. Technol., vol. 5, no. 2, pp. 73–78, 2025.

[16] U. Zara et al., "Phishing Website Detection Using Deep Learning Models," IEEE Access, vol. 12, pp. 167072–167087, 2024, doi: 10.1109/ACCESS.2024.3486462.

[17] K. Barik, S. Misra, and R. Mohan, "Web-based phishing URL detection model using deep learning optimization techniques," Int. J. Data Sci. Anal., vol. 20, pp. 4449–4471, 2025, doi: 10.1007/s41060-025-00728-9.

[18] B. C. Ujah-Ogbuagu, O. N. Akande, and E. Ogbuju, "A hybriddeep learning technique for spoofing website URL detection in real-time applications," J. Electr. Syst. Inf. Technol., vol. 11, Art. no. 7, 2024, doi: 10.1186/s43067-023-00128-8.

[19] R. Yang, K. Zheng, B. Wu, C. Wu, and X. Wang, "Phishing Website Detection Based on Deep Convolutional Neural Network and Random Forest Ensemble Learning," Sensors, vol. 21, no. 24, Art. no. 8281, 2021, doi: 10.3390/s21248281.

[20] M. Aljabri et al., "Kashif: A Chrome extension for classifying Arabic content on web pages using machine learning," Appl. Sci., vol. 14, no. 20, Art. no. 9222, 2024, doi: 10.3390/app14209222.

[21] S. Remya, M. J. Pillai, B. S. Aparna, S. R. S. Reddy, and Y. Y. Cho, "BGL-PhishNet: Phishing website detection using hybrid model BERT, GNN, and LightGBM," IEEE Access, vol. 13, pp. 47552–47569, 2025, doi: 10.1109/ACCESS.2025.3551542.

[22] M. Ghafoor, A. Shah, M. A. Al-Naeem, and C. Maple, "Decoding phishing evasion: Analyzing attacker strategies to circumvent detection systems," IEEE Access, vol. 13, pp. 78513–78526, 2025, doi: 10.1109/ACCESS.2025.3565619.

[23] F. Malik et al., "Optimizing Malicious Website Detection with the XGBoost Machine Learning Approach," J. Comput. Biomed. Inf., vol. 7, no. 2, 2024. [Online]. Available: jcbi.org

[24] M. Zouina and B. Outtaj, "A novel lightweight URL phishing detection system using SVM and similarity index," Hum.-Centric Comput. Inf. Sci., vol. 7, Art. no. 17, 2017, doi: 10.1186/s13673-017-0098-1.

[25] A.-S. Alabdulkarim, M. Elhoseny, and S. Kamel, "Enhanced K-Means clustering with CfsSubsetEval for phishing website detection," Electronics, vol. 13, no. 18, Art. no. 3677, 2024, doi: 10.3390/electronics13183677.

[26] H. Huang, L. Qian, and Y. Wang, "A SVM-based technique to detect phishing URLs," Inf. Technol. J., vol. 11, no. 7, pp. 921–925, 2012, doi: 10.3923/itj.2012.921.925.

[27] M. A. K. Raaian, W. Monika, A. Onan, and Y. Murakami, "Benchmarking open-source large language models for phishing URL detection under multiple prompting strategies," Information, vol. 16, no. 5, Art. no. 366, 2025, doi: 10.3390/info16050366.

[28] S. Saleem, A. Rehman, M. A. Khan, and M. Asif, "Web content analysis-based malicious webpage classification using machine learning techniques," J. Adv. Res. Appl. Sci. Eng. Technol., vol. 47, no. 1, pp. 105122, 2025, doi: 10.37934/araset.47.1.105122.