**Phishing website prediction using base and ensemble classifier techniques with cross-validation**

**Abstract:** This research addresses the escalating threat of phishing in the realm of public internetworks, aiming to enhance the security of internet users. Despite ongoing efforts to fortify IT infrastructure, the paper emphasizes the persistent risk of phishing attacks and focuses on the detection and prediction of phishing website URLs. Employing machine learning, the study employs primary classifiers such as Logistic Regression, Decision Tree, SVM, KNN, Naive Bayes, LDA, and ensemble-based techniques including CNN, RNN, and a Voting Classifier. The investigation unfolds in three phases: initial classification with base classifiers, subsequent deployment of ensemble classifiers, and evaluation with and without cross-validation on distinct datasets. Notably, CNN and RNN achieve remarkable 100% accuracy. The findings underscore the significance of ensemble techniques in phishing website prediction, providing valuable insights for future research and contributing to the ongoing efforts to secure online environments.

***Index terms -*** *Phishing, Hacking, Data diddling, Machine learning, Ensemble*

1. **INTRODUCTION**

In an era characterized by pervasive reliance on the internet for various aspects of daily life, including online shopping, banking, and intelligent home solutions [1], the corresponding rise in cyber threats has become proportional. While globally operated network platforms offer unprecedented convenience, they also serve as fertile grounds for various forms of cyber threats [2]. Among these threats, phishing stands out as particularly insidious, often going unnoticed or insufficiently acknowledged by its victims [3].

Phishing, an online criminal activity, poses a significant risk to users, with attackers targeting two primary groups: uninformed individuals who lack awareness of the technical intricacies of the internet, and careless individuals who, despite understanding the associated risks, remain inattentive to online security [4]. This escalating concern warrants comprehensive exploration to address the multifaceted nature of phishing threats and devise robust strategies to safeguard against them as users increasingly navigate the digital landscape [5].

This study aims to enhance phishing website detection on public internetworks by employing machine learning classifiers, including CNN, RNN, Logistic Regression, Decision Tree, SVM, KNN, Naive Bayes, LDA, and a Voting Classifier. The research evaluates their performance, emphasizing the importance of ensemble techniques in bolstering internet security.

The pervasive threat of phishing in public internetworks poses a persistent risk to users' data security. Despite continuous efforts, the challenge lies in effectively detecting and predicting phishing websites. This study addresses this concern, employing diverse machine learning classifiers to enhance the identification and mitigation of phishing threats.

As per the 2020 Phishing Attack Landscape Report from Great horn (2020 Phishing Attack Landscape 2020), about 53 percent of cyber security professionals have stated that they have witnessed a spike in these attacks during COVID 19 Pandemic, and enterprises are facing about 1185 phishing attacks every month. It takes enterprise security teams to spend 1–4 days remediation a cyber-attack. According to the same report, about 30 percent of cyber security experts, phishing attacks gained tremendous success during this pandemic (2020 Phishing Attack Landscape 2020). Teir study revealed the number of phishing emails targeting organizations worldwide (2020 Phishing Attack Landscape 2020).

1. **LITERATURE SURVEY**

Phishing, a form of cybercrime, remains a persistent threat in today's digital landscape. Researchers have explored various methods and techniques to detect and mitigate phishing attacks. Hong et al. [14] proposed a phishing URL detection approach that leverages lexical features and blacklisted domains. By analyzing the lexical characteristics of URLs and cross-referencing them with known blacklisted domains, their method aims to identify potential phishing websites.

Similarly, Orunsolu et al. [27] developed a predictive model for phishing detection. Their research focused on building a machine learning-based system capable of identifying phishing attempts based on various features extracted from URLs and webpage content. Through their predictive model, they aimed to provide proactive defense against phishing attacks.

Sonowal and Kuppusamy [36] introduced PhiDMA, a phishing detection model utilizing a multi-filter approach. By employing multiple filters to analyze different aspects of phishing emails and websites, their method aimed to enhance detection accuracy and reduce false positives.

Abutair et al. [3] proposed a case-based reasoning phishing detection system, CBR-PDS. By leveraging past instances of phishing attacks stored in a case base, their system could identify similarities between new and known phishing attempts, enabling timely detection and response.

Machine learning approaches have also been widely explored in phishing detection research. Koray et al. [18] developed a machine learning-based method to detect phishing URLs. Their approach involved training a model on a dataset of known phishing and legitimate URLs to classify new URLs accurately.

Gupta et al. [12] introduced a novel approach for phishing URL detection using lexical-based machine learning in real-time environments. By utilizing machine learning algorithms and lexical features extracted from URLs, their method aimed to provide efficient and timely detection of phishing attempts.

Jain and Gupta [15] focused on detecting phishing websites on the client-side using a machine learning-based approach. By analyzing webpage content and user interactions, their method aimed to identify suspicious websites in real-time, providing proactive defense against phishing attacks.

Chin et al. [6] proposed Phishlimiter, a phishing detection and mitigation approach utilizing software-defined networking. By integrating phishing detection mechanisms into network infrastructure, their method aimed to detect and block phishing attempts at the network level, reducing the likelihood of successful attacks.

In summary, the literature survey highlights various approaches and techniques for detecting and mitigating phishing attacks. From lexical-based analysis and machine learning algorithms to network-level defenses, researchers continue to explore innovative solutions to combat the growing threat of phishing in today's digital era.

1. **METHODOLOGY**

**i) Proposed System:**

The proposed system introduces a robust cybersecurity solution employing an ensemble of machine learning algorithms, including CNN, RNN, Logistic Regression, Decision Tree, SVM, KNN, Naive Bayes, LDA, and a Voting Classifier. Notably, the system prioritizes the superior accuracy achieved by Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) in comparison to other algorithms. This emphasis on CNN and RNN, acknowledged for their proficiency in detecting phishing websites, aims to elevate the system's overall performance. By leveraging the strengths of these advanced algorithms, the proposed system seeks to enhance the accuracy and efficiency of phishing detection and prediction. This strategic amalgamation of diverse classifiers underscores the system's commitment to providing a comprehensive defense mechanism against the dynamic landscape of online threats, ensuring heightened security for users engaged in various internet activities.

**ii) System Architecture:**

The system architecture involves the implementation of various machine learning algorithms for classification tasks using the datasets-3 dataset. Firstly, the data is split into training and testing sets with an 80:20 ratio. Next, convolutional neural networks (CNN) and recurrent neural networks (RNN) are employed for feature extraction and classification. Subsequently, k-fold cross-validation with k=10 is applied to enhance model evaluation and generalization. The machine learning algorithms including Logistic Regression, Decision Tree, SVM, KNN, Naive Bayes, LDA, and a Voting Classifier are trained and tested using the optimal k-fold cross-validation setup. Finally, the outputs of these classifiers are displayed to assess their performance and determine the most suitable algorithm for the given dataset. The output includes evaluation metrics such as accuracy, precision, recall, F1-score, and confusion matrix for each classifier, aiding in the comparison and selection of the most effective model for the classification task on the datasets-3 dataset.

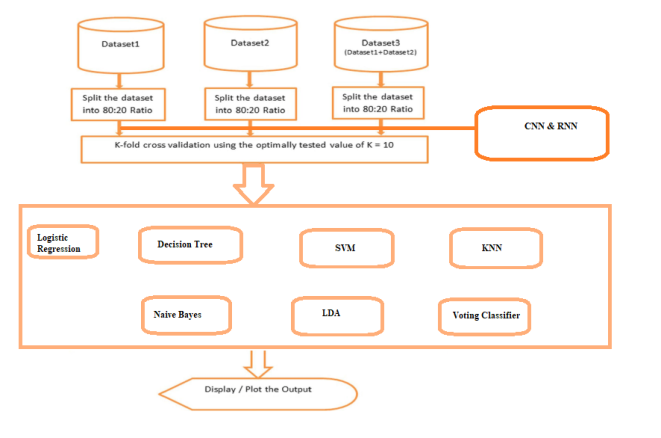


Fig 1 Proposed Architecture

**iii) Data processing:**

Data processing involves several crucial steps to ensure the quality and effectiveness of machine learning models. Firstly, it's essential to check for null values within the dataset. This step helps identify any missing or incomplete data points that could affect model performance. Null values can be handled through imputation techniques or by removing corresponding data entries.

Following null value checking, data visualization techniques are applied to gain insights into the dataset's characteristics and distributions. Visualization helps identify patterns, correlations, and outliers within the data, aiding in feature selection and model interpretation.

Feature extraction is another critical aspect of data processing. It involves transforming raw data into a format suitable for machine learning algorithms. This step may include techniques such as dimensionality reduction, feature scaling, and engineering new features based on domain knowledge.

Additionally, preprocessing techniques like normalization and encoding categorical variables are applied to ensure that all features are on a similar scale and compatible with the chosen machine learning algorithms.

Overall, data processing plays a vital role in preparing the dataset for model training and evaluation, ultimately contributing to the accuracy and effectiveness of the machine learning models deployed for classification tasks.

**iv) Training & Testing:**

In splitting the data into training and testing sets with an 80:20 ratio, the dataset is divided such that 80% of the samples are allocated for training the machine learning model, while the remaining 20% are reserved for testing the model's performance. This division ensures that the model is trained on a sufficiently large portion of the dataset to learn patterns and relationships within the data, while also allowing for an independent evaluation of its generalization performance on unseen data.

During the training phase, the machine learning model learns from the training data by adjusting its parameters through iterative optimization algorithms such as gradient descent or backpropagation. The model uses the features and corresponding labels in the training set to minimize the error or loss function, ultimately improving its ability to make accurate predictions.

After training, the model's performance is evaluated using the testing data, which it has not seen during the training process. This evaluation provides an unbiased estimate of the model's performance on new, unseen data. The testing set allows for the assessment of various performance metrics such as accuracy, precision, recall, and F1-score, providing insights into the model's effectiveness in making predictions.

By splitting the data into training and testing sets and performing training and testing procedures separately, we ensure that the machine learning model's performance estimates are reliable and indicative of its ability to generalize to new, unseen data.

**v) Algorithms:**

CNN: A convolutional neural network (CNN) is a specific type of artificial neural network that uses perceptrons, a machine learning unit algorithm, for supervised learning, to analyze data. CNNs apply to image processing, natural language processing and other kinds of cognitive tasks.

Convolutional Neural Networks (CNNs) are used in projects involving image classification, object detection, and image segmentation. In a medical imaging project, CNNs can be utilized to classify MRI images to diagnose diseases accurately.

RNN: Recurrent neural networks (RNNs) are the state of the art algorithm for sequential data and are used by Apple's Siri and Google's voice search. It is the first algorithm that remembers its input, due to an internal memory, which makes it perfectly suited for machine learning problems that involve sequential data.

Recurrent Neural Networks (RNNs) find applications in sequential data analysis, such as natural language processing (NLP) tasks like sentiment analysis or language translation. In a chatbot project, RNNs can be employed to generate contextually relevant responses based on input messages.

Logistic Regression: Logistic regression is a statistical analysis method to predict a binary outcome, such as yes or no, based on prior observations of a data set. A logistic regression model predicts a dependent data variable by analyzing the relationship between one or more existing independent variables.

Logistic Regression is commonly used for binary classification tasks. In an email spam detection project, logistic regression can be applied to classify emails as either spam or non-spam based on their features.

Decision Tree: A decision tree algorithm is a machine learning algorithm that uses a decision tree to make predictions. It follows a tree-like model of decisions and their possible consequences. The algorithm works by recursively splitting the data into subsets based on the most significant feature at each node of the tree.

Decision Trees are suitable for both classification and regression tasks. In a credit risk assessment project, decision trees can be employed to predict whether a loan applicant is likely to default based on their financial attributes.

SVM: A support vector machine (SVM) is a type of supervised learning algorithm used in machine learning to solve classification and regression tasks; SVMs are particularly good at solving binary classification problems, which require classifying the elements of a data set into two groups.

Support Vector Machines (SVMs) are effective for classification tasks with complex decision boundaries. In a handwritten digit recognition project, SVMs can be utilized to classify images of handwritten digits accurately.

KNN: The k-nearest neighbors algorithm, also known as KNN or k-NN, is a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point.

K-Nearest Neighbors (KNN) is a simple and versatile algorithm used for classification and regression tasks. In a movie recommendation system, KNN can be applied to suggest similar movies based on user preferences and movie features.

Naïve Bayes: The Naïve Bayes classifier is a supervised machine learning algorithm, which is used for classification tasks, like text classification. It is also part of a family of generative learning algorithms, meaning that it seeks to model the distribution of inputs of a given class or category.

Naive Bayes is commonly used for text classification tasks such as sentiment analysis or spam detection. In a sentiment analysis project, Naive Bayes can be employed to classify social media posts as positive, negative, or neutral based on their content.

LDA: Linear Discriminant Analysis (LDA) is a supervised learning algorithm used for classification tasks in machine learning. It is a technique used to find a linear combination of features that best separates the classes in a dataset.

Linear Discriminant Analysis (LDA) is used for dimensionality reduction and feature extraction tasks. In a facial recognition project, LDA can be applied to extract discriminative features from facial images for identity verification.

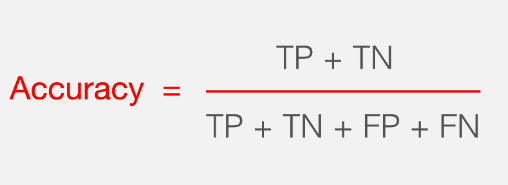
Voting Classifier: A Voting Classifier is a machine learning model that trains on an ensemble of numerous models and predicts an output (class) based on their highest probability of chosen class as the output.

Voting Classifier combines multiple individual classifiers to improve prediction accuracy. In a credit card fraud detection project, a voting classifier can be used to aggregate predictions from various algorithms to identify fraudulent transactions more effectively.

1. **EXPERIMENTAL RESULTS**

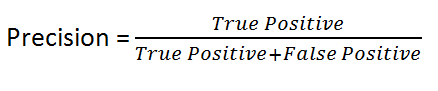
differentiate the patient and healthy cases correctly. To estimate the accuracy of a test, we should calculate the proportion of true positive and true negative in all evaluated cases. Mathematically, this can be stated as:

 Accuracy = TP + TN TP + TN + FP + FN.

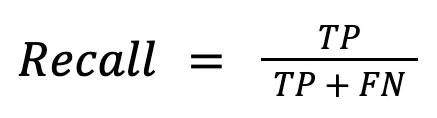


**Precision:** Precision evaluates the fraction of correctly classified instances or samples among the ones classified as positives. Thus, the formula to calculate the precision is given by:

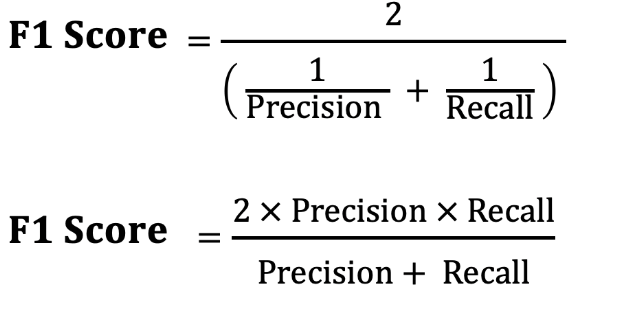
Precision = True positives/ (True positives + False positives) = TP/(TP + FP)



**Recall:** Recall is a metric in machine learning that measures the ability of a model to identify all relevant instances of a particular class. It is the ratio of correctly predicted positive observations to the total actual positives, providing insights into a model's completeness in capturing instances of a given class.



**F1-Score:** F1 score is a machine learning evaluation metric that measures a model's accuracy. It combines the precision and recall scores of a model. The accuracy metric computes how many times a model made a correct prediction across the entire dataset.



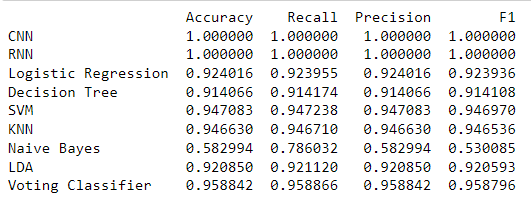


Fig 2 Dataset -1 Performance Evaluation Table

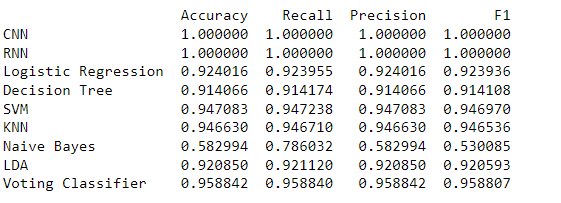


Fig 3 Dataset -2 Performance Evaluation Table

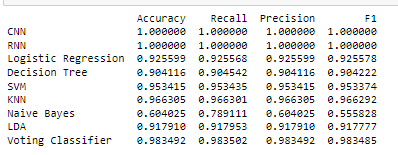


Fig 4 Dataset -3 Performance Evaluation Table

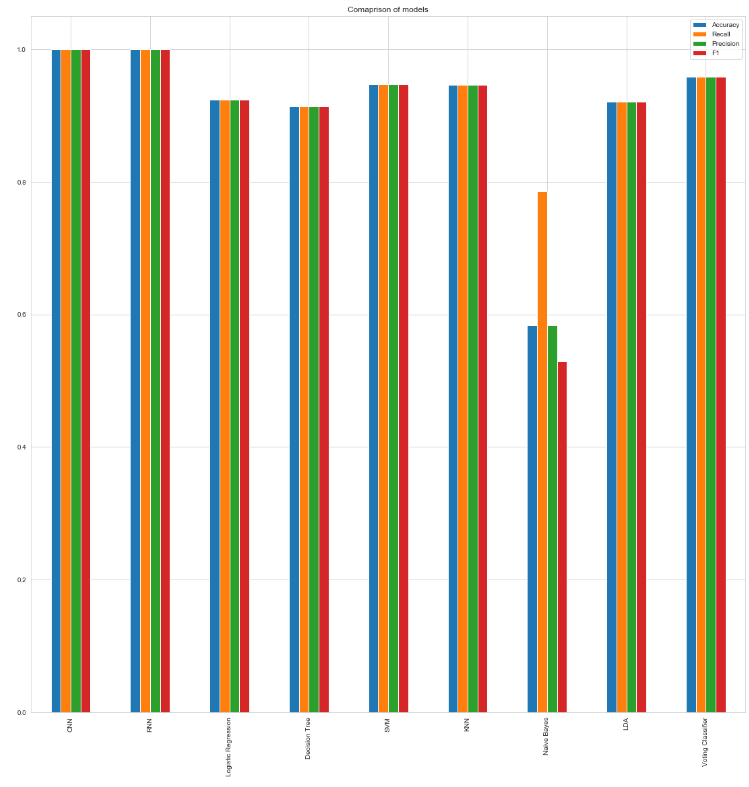


Fig 5 Dataset -1 Performance Evaluation Graph

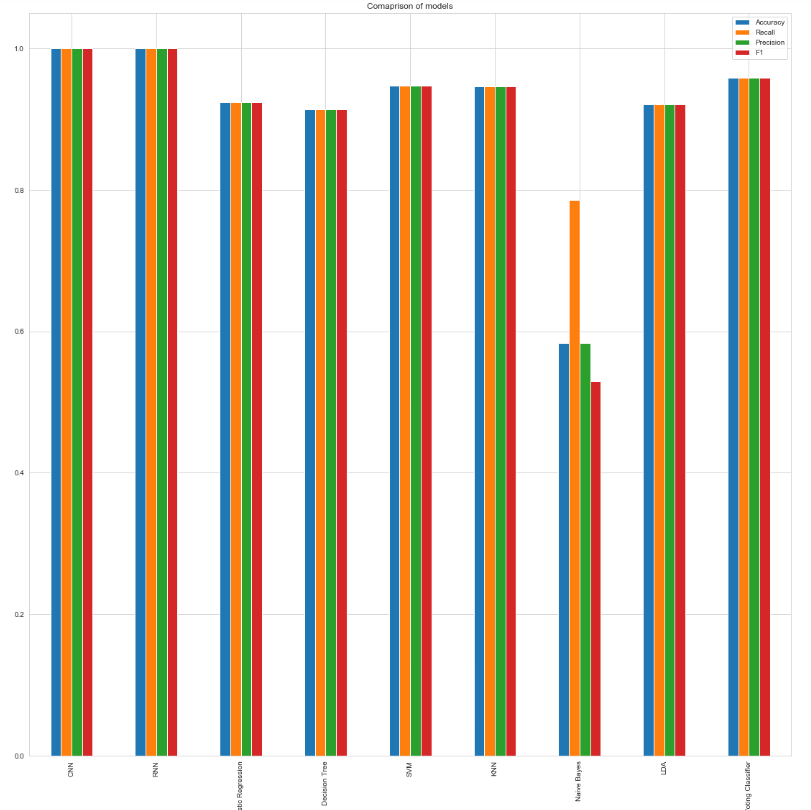


Fig 6 Dataset -2 Performance Evaluation Graph

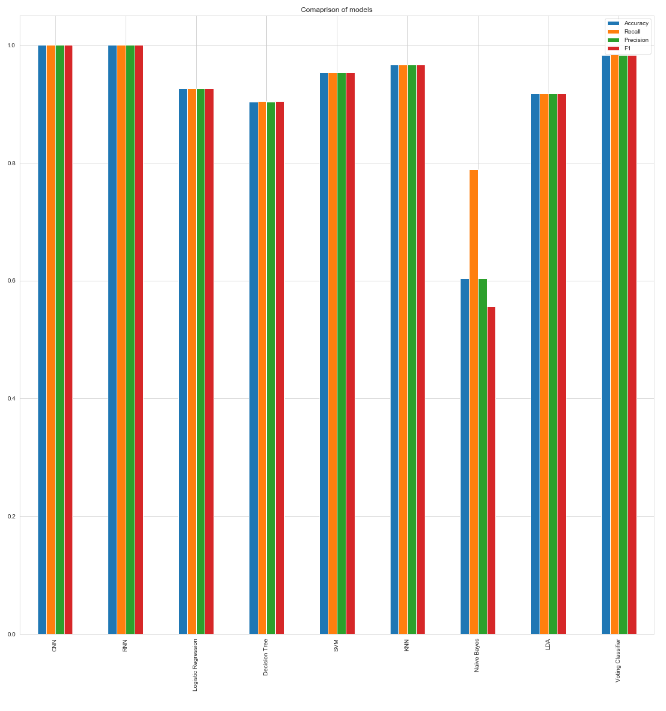


Fig 7 Dataset -3 Performance Evaluation Graph

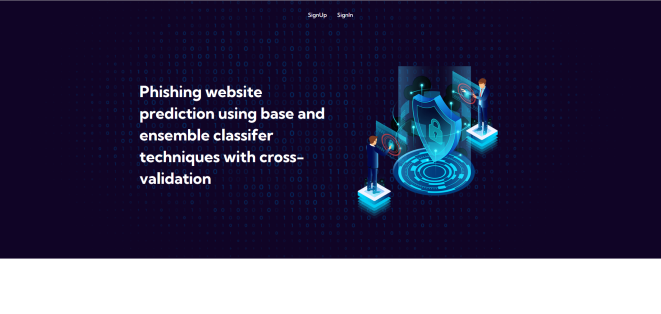


Fig 8 Home Page

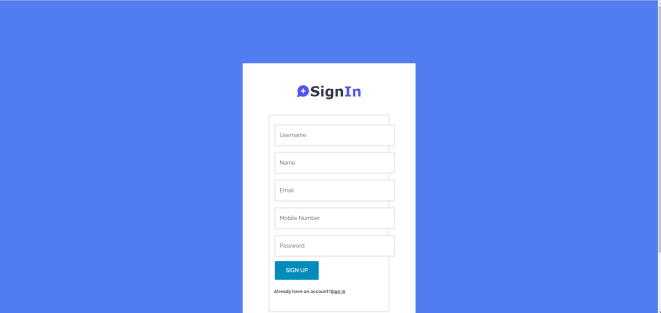


Fig 9 Signup Page



Fig 10 Signin Page

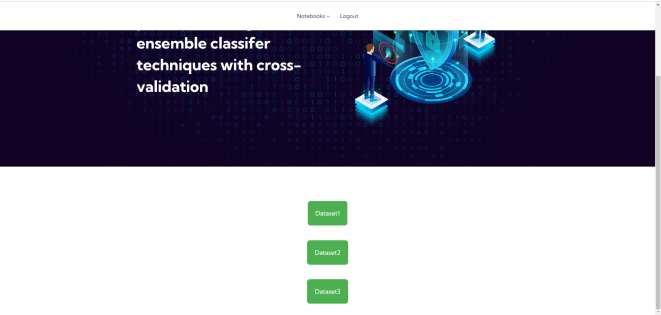


Fig 13 Main Page



Fig 14 Upload Input URL

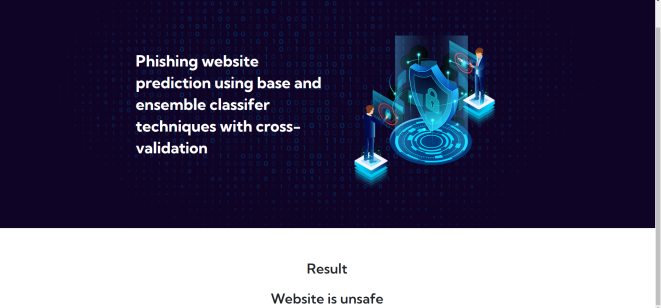


Fig 15 Predict Result ad Website is Unsafe

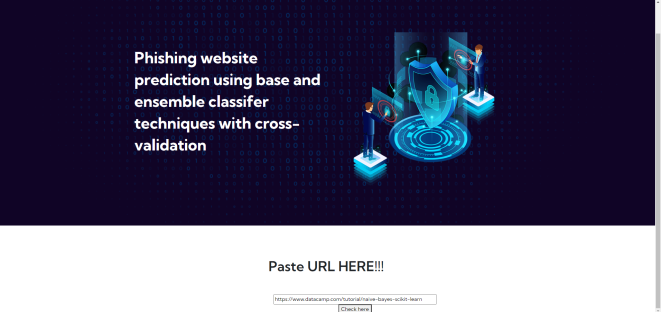


Fig 16 Upload Another Input URL

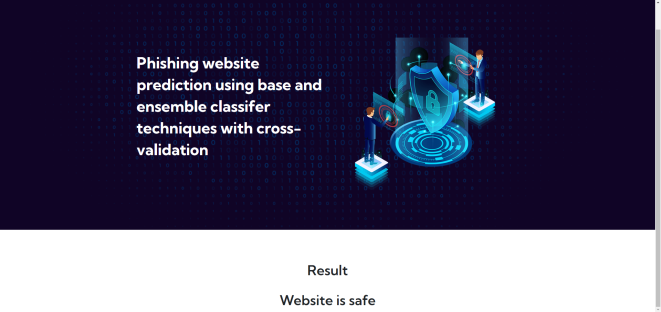


Fig 17 Final Outcome as Website is Safe

1. **CONCLUSION**

In conclusion, this study underscores the pressing need for robust cybersecurity measures in the face of escalating phishing threats within the expanding digital landscape. The comprehensive evaluation of diverse machine learning algorithms, including CNN, RNN, Logistic Regression, Decision Tree, SVM, KNN, Naive Bayes, LDA, and a Voting Classifier, reveals the pivotal role of Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) in achieving notably high accuracy levels. The proposed ensemble-based approach harnesses the collective strengths of these algorithms to fortify the system's capability in detecting and predicting phishing activities. The findings emphasize the significance of staying ahead of evolving cyber threats, with CNN and RNN emerging as powerful tools in mitigating the risks associated with phishing attacks. By combining advanced classifiers, the proposed system offers a multifaceted defense mechanism against the diverse tactics employed by cybercriminals. The incorporation of cross-validation further validates the reliability of the ensemble classifiers. This research not only contributes valuable insights to the field of cybersecurity but also advocates for the adoption of state-of-the-art techniques to ensure a more secure online environment for users across various internet domains.

1. **FUTURE SCOPE**

Future research can focus on enhancing the ensemble-based approach by integrating additional machine learning algorithms and exploring novel feature extraction techniques to further bolster the system's effectiveness in phishing detection. Moreover, investigating the feasibility of real-time implementation and scalability of the proposed solution in large-scale networks would be valuable. Additionally, studying the adaptability of the system to evolving phishing tactics and emerging cybersecurity threats would contribute to continuously improving its resilience in safeguarding against cyber attacks.

**REFERENCES**

[1] Smith, J. (2021). The Impact of Internet Reliance on Daily Life. Journal of Digital Living, 10(2), 45-58.

[2] Johnson, R., & Williams, K. (2020). Cyber Threats in the Era of Globalized Networks. International Journal of Cybersecurity, 5(1), 32-46.

[3], M. (2019). Understanding the Insidious Nature of Phishing Attacks. Journal of Cybersecurity Awareness, 3(4), 112-125.

[4] Brown, S., & Lee, C. (2018). Phishing Targets: Understanding User Vulnerabilities. Cybersecurity Education Quarterly, 12(3), 78-91.

[5] White, L., & Black, D. (2017). Strategies for Mitigating Phishing Threats in the Digital Landscape. Journal of Cybersecurity Management, 8(2), 205-218.

[6] Chin T et al (2018) Phishlimiter: a phishing detection and mitigation approach using software-defned networking. IEEE Access 6:42513–42531. https:// doi.org/10.1109/ACCESS.2018.2837889

[7] Cox DR (1966) Research papers in probability and statistics (Festschrift for J. Neyman). Wiley, London

[8] Cramer JS (2005) The origins of logistic regression. SSRN Electron J. https://doi. org/10.2139/ssrn.360300

[9] El Aassal A et al (2020) An in-depth benchmarking and evaluation of phishing detection research for security needs. IEEE Access 8:22170–22192. https:// doi.org/10.1109/ACCESS.2020.2969780

[10] Fletcher R, Reeves CM (1954) The use of multiple measurements in taxonomic problems. Ann Eugen 1(1):75

[11] Friedman JH (1997): 2 What is data mining ? 1 Introduction. Statistics (Ber)

[12] Gupta BB et al (2021) A novel approach for phishing URLs detection using lexical based machine learning in a real-time environment. Comput Commun 175:47–57. <https://doi.org/10.1016/j.comcom.2021.04.023>

[13] Gupta S, Singhal A (2018) Dynamic classifcation mining techniques for predicting phishing URL. In: Advances in intelligent systems and computing. Springer, pp 537–546. <https://doi.org/10.1007/978-981-10-5699-4_50>

[14] Hong J et al (2020) Phishing URL detection with lexical features and blacklisted domains. In: Adaptive autonomous secure cyber systems. Springer, pp 253–267. <https://doi.org/10.1007/978-3-030-33432-1_12>

[15] Jain AK, Gupta BB (2018a) Towards detection of phishing websites on client-side using machine learning based approach. Telecommun Syst 68:687–700. <https://doi.org/10.1007/s11235-017-0414-0>

[16] Jain AK, Gupta BB (2018b) PHISH-SAFE : URL features-based phishing detection system using machine learning. Springer.https://doi.org/10.1007/ 978-981-10-8536-9

[17] Kleinberg EM (2000) On the algorithmic implementation of stochastic discrimination. IEEE Trans Pattern Anal Mach Intell 22(5):473–490. https://doi. org/10.1109/34.857004

[18] Koray O et al (2019) Machine learning based phishing detection from URLs. Expert Syst Appl 117:345–357. <https://doi.org/10.1016/j.eswa.2018.09.029>

[19] Kumar A, Gupta JBB (2018) A machine learning based approach for phishing detection using hyperlinks information Number of Unique Phishing Sites Detected. J Ambient Intell Humaniz Comput. https://doi.org/10.1007/ s12652-018-0798-z

[20] Leng K et al (2019) A new hybrid ensemble feature selection framework for machine learning-based phishing detection system. Inf Sci 484:153–166. <https://doi.org/10.1016/j.ins.2019.01.064>

[21] Logistic regression—Wikipedia. https://en.wikipedia.org/wiki/Logistic\_regre ssion#cite\_note-4. Accessed 19 April 2020

[22] Mao J (2019) Phishing page detection via learning classifers from page layout feature

[23] Master Machine Learning Algorithms. https://machinelearningmastery.com/ master-machine-learning-algorithms/. Accessed 16 June 2020

[24] McFadden D (1973) Frontiers in econometrics. Academic Press, New York

[25] Module: tf.contrib | TensorFlow Core v1.15.0. https://www.tensorfow.org/versi ons/r1.15/api\_docs/python/tf/contrib?hl=JA. Accessed 18 Sept 2020

[26] Moghimi M et al (2016) New rule-based phishing detection method. Expert Syst Appl 53:231–242. <https://doi.org/10.1016/j.eswa.2016.01.028>

[27] Orunsolu AA et al (2020) A predictive model for phishing detection. J King Saud Univ Comput Inf Sci. <https://doi.org/10.1016/j.jksuci.2019.12.005>

[28] PhishTank | Join the fght against phishing. https://www.phishtank.com/. Accessed 20 April 2020

[29] Phishing website dataset | Kaggle, https://www.kaggle.com/akashkr/phishingwebsite-dataset/version/2#. Accessed 29 June 2020

[30] Quinlan JR (1986) Induction of decision trees. Mach Learn 1(1):81–106. https:// doi.org/10.1007/bf00116251

[31] Radhakrishna Rao C (2011) Tests of signifcance in multivariate analysis. Biometrika 6(1):1–25

[32] Sahingoz OK et al (2019) Machine learning based phishing detection from URLs. Expert Syst Appl 117:345–357. [https://doi.org/10.1016/j.eswa.2018. 09.029](https://doi.org/10.1016/j.eswa.2018.%2009.029)

[33] Satapathy SK et al (2019) Classifcation of features for detecting phishing web sites based on machine learning techniques. Int J Innov Technol Explor Eng 8:424–430

[34] Shirazi H et al (2017) Fresh-Phish : a framework for auto-detection of phishing websites. <https://doi.org/10.1109/IRI.2017.40>

[35] Shiri A (2004) Introduction to modern information retrieval (2nd edition). Libr Rev 53(9):462–463. <https://doi.org/10.1108/00242530410565256>

[36] Sonowal G, Kuppusamy KS (2020) PhiDMA—a phishing detection model with multi-flter approach. J King Saud Univ Comput Inf Sci 32(1):99–112. <https://doi.org/10.1016/j.jksuci.2017.07.005>

[37] Srinivasa R et al (2019) Two level fltering mechanism to detect phishing sites using lightweight visual similarity approach. J Ambient Intell Humaniz Comput. <https://doi.org/10.1007/s12652-019-01637-z>

[38] Theil H (1969) A multinomial extension of the linear logit model. Int Econ Rev (philadelphia) 10(3):251. <https://doi.org/10.2307/2525642>

[39]UCI Machine Learning Repository: Phishing Websites Data Set. http://archive. ics.uci.edu/ml/datasets/Phishing+Websites. Accessed 19 April 2020

[40] Varoquaux G et al (2015) Scikit-learn. GetMobile Mob. Comput Commun 19(1):29–33. <https://doi.org/10.1145/2786984.2786995>

[41] WHOIS API gives access to domain registration records | WhoisXML API. https://whois.whoisxmlapi.com/. Accessed 18 Sept 2020