DOI: http://dx.doi.org/10.26483/ijarcs.v15i4.7095

Volume 15, No. 4, July-August 2024

ISSN No. 0976-5697

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

HYBRIDIZED MACHINE LEARNING PREDICTION FOR THE EXPOSURE OF PHISHING WEBSITES

Ankit Prajapati
Research Scholar
Department of CSE
RITS, Bhopal, India

Chetan Agrawal & Pawan Meena
Assistant Professor
Department of CSE
RITS, Bhopal, India

Abstract: One of the most popular ways that people utilise the internet for illegal activities is phishing. Phishing websites are those that impersonate trustworthy websites while still appearing and sounding authentic. The purpose of their fabrication is to trick the receiver into believing that the item is authentic. These days, phishing schemes are riskier and more intricate than ever. Artificial intelligence-based ML & deep learning techniques can be used to anticipate phishing websites. A classification system based on ML may be employed to detect possible phishing websites. We provide a mixed machine learning approach for phishing site forecast in this study. The outcomes of the studies demonstrate that the recommended process performs more effectively in categorizing malevolent URLs than more contemporary methods. A graphical illustration of the accuracy comparison of several approaches is shown in Figure 4. The simulated results demonstrate that a compared to current method, the suggested hybrid classification strategy achieves higher accuracy. While the current KNN achieves 96% accuracy, the hybrid approach achieves 98.14% accuracy.

Keywords: AI, Machine Learning, Hybrid, and Phishing Websites, Accuracy.

I. INTRODUCTION

Right nowadays Folks are starting to be sure of more and more on innovative efforts identical mobile infrastructure and the cloud as the significance of digital operations has grown over the past few years. Phishing. According to PhishLabs, assault volumes are also rising as actors' techniques shift to reflect changes in the digital environment. Additionally, during the previous four years, phishing attempts have increased as a result of using free hosting services, rising from 3.0% in 2020 to 13.8% in 2021. Furthermore, creating phishing websites is really easy using phishing toolkits. A single actor may simply develop a huge number of phishing websites that imitate real websites since these toolkits are so readily available. For sample, PhishLabs observed a notable rise in the numeral of attacks in month August 2022. This attempt made use of at minimum 2,000 publicly held phishing websites; all formed using the matching kit. There will most likely be an increase in phishing attempts in the future due to the widespread accessibility of these kits provided by organised criminal groups. Actors' persistent efforts to capitalise on novel opportunities are demonstrated by their procedure of free hosts, phishing kits, & SSL certificates. As a result, the number of phishing assaults has been steadily increasing from year to year. Due to the dynamic nature of the assaults, it is difficult to design a trustworthy phishing detection tool. [1]

To access the website and "update" his personal statistics, the victim is required to provide his online banking login credentials, which consist of his user name and password. If the victim gives the phisher his real login credentials on the fake website, the phisher can then take the victim's identity. This might provide the attacker the power to drain the victim's account of money or do them other harm. Because victims are establishing a direct connection with a website they believe to be reliable and trustworthy, such attacks tend to be quite successful. It should be highlighted that phishers have been able to adapt despite widespread media coverage of phishing, which has increased the number of Internet users aware of the method.

Phishers usually utilise target page spoofing to create phishing websites. Opened phishing sites have the potential to trick recipients into thinking they are real and supplying the desired data. If they are successful, phishers may either keep a sizable portion of the money for themselves or sell the consumers' personal data to other criminals.

For example, many phishing emails that target people these days ask them to validate their personal information for "security reasons," presumably so that the organisation they are targeting can protect themselves from the risk of phishing. Bestowing to the Anti-Phishing Working Clutch, there has been a discernible rise in the phishing issue over the last several years.

Artificial Intelligence, Machine Learning, and Phishing are email scams where the sender assumes the identity of a reliable company or organization in an effort to obtain private data like login passwords or credit card numbers. Phishing email criminals may pose as IT administrators, banks, auction sites, or well-known social networks in order to trick the civic. It's a pernicious form of societal engineering hybrid websites and phishing sites.[2]

Third-party monitoring enables the collection and correlation of browsing patterns from web users. The development of new technique known as As ad-blocking software and other anti-tracking techniques proliferate, tracking service benefactors resort to "CNAME cloaking" to circumvent them. The problematic subdomain uses a CNAME to resolve to a third-party domain associated with tracking, but it fools browsers into discerning that a entreaty for the subdomain comes from the website that was visited. The secrecy protections against such third-party aiming are evaded by employing this strategy. We flinch by defining CNAME cloaking-based chasing by evaluating websites and tracking organisations that utilise CNAME cloaking to monitor users' behaviour using a CNAME blocklist.[3]

The paper is divided into the following five sections. An overview of earlier research that is relevant to our work on phishing detection systems is given in Section II. The planned technique is provided in section III. The imitation and outcome creation are shown in Section IV, and the supposition is shown in Section V.

CONCEPT AND DEFINITIONS

The use of machine learning techniques to recognise and categorise websites that participate in phishing operations is known as "phishing website recognition using machine learning approaches." Phishing is the term for dishonest attempts to trick individuals into divulging cloistered statistics—alike passwords, credit card numbers, or own information—by pretending to be a trustworthy organisation.[4]

Phishing website detection refers to the process of teaching machine learning models to differentiate between websites that are authentic and those that are part of phishing schemes. By analyzing various features and patterns, machine learning algorithms can learn to differentiate between trustworthy websites and malicious ones, thereby providing an automated and scalable solution to detect and mitigate phishing threats.

1. ***Data Collection:*** Gather a diverse and representative dataset containing both legitimate and phishing website samples. Include features that capture relevant characteristics of the websites, such as URL attributes, domain information, HTML content, etc. Ensure that the dataset is properly labeled, distinguishing between legitimate and phishing websites.[5]
2. ***Feature Extraction:*** Extract meaningful features from the website data. This can include: URL-based features: URL length, presence of special characters, domain age, etc. Domain-based features: WHOIS information, domain reputation, etc. HTML-based features: Presence of suspicious HTML tags, JavaScript-based redirection, form submission URLs, etc. Content-based features: Keywords, similarity to known phishing websites, etc. Consider using feature engineering techniques to derive new topographies that may enrich the predictive power of the exemplary.[6][7]
3. ***Feature Pre-processing:*** Normalize and pre-process the extracted features to ensure compatibility and effectiveness in the subsequent steps. Perform tasks such as feature scaling, one-hot encoding for categorical features, handling missing values, and removing outliers. Riven the dataset into training & testing sets, ensuring that the class distribution is maintained in both sets.
4. ***Hybrid Model Construction*****:** Build an ensemble of machine learning models that leverages different algorithms and techniques. Choose a mix of classifiers, including neural networks, SVM, logistic regression, decision trees, etc. Utilising the training dataset, train each model separately and adjust the hyper parameters for best results. Consider using different subsets of features or feature representations for different models to introduce diversity.[8]
5. ***Model Fusion:*** Syndicate the extrapolations of distinct models to style absolute prophecy. Explore ensemble techniques such as voting (majority voting, weighted voting), stacking, or boosting. Experiment with different fusion strategies and select the one that yields the best results based on validation performance.
6. ***Evaluation and Validation:*** Use relevant measures to assess the hybrid model, such as area under the ROC curve, recall, accuracy, and precision. To make sure the model can generalise, use holdout validation or cross-validation. Evaluate the model's performance using the testing dataset to confirm that it works as intended in practical situations.
7. ***Model Optimization and Refinement:*** Conduct thorough analysis of the model's strengths and weaknesses. Fine-tune the model by adjusting hyper parameters, exploring feature selection techniques, or incorporating additional features. Regularly apprise the model with innovative data and acclimatize it to evolving phishing techniques to maintain its predictive power.
8. ***Deployment and Monitoring:*** Integrate the hybrid model into a production environment where it can be utilized for real-time or batch phishing website prediction. Monitor the model's performance over time and periodically retrain or update it with fresh data to ensure its accuracy and reliability.

PHISHING SYSTEM AND OVERVIEW

In order to create an automated system for detecting phishing websites, researchers have been examining several methods in recent years, including examining the website's content, appearance, and URL. These strategies may generally be divided into two groups. The first method looks for inherent properties of phishing internet site and tries to identify these assaults based on their unique qualities. Recent studies using deep learning and machine learning approaches have provided support for this kind of approach.

These techniques have been publicized to be effective at discovering phishing, but because they rely so heavily on characteristics of phishing websites that may change or stop being relevant in the future (such as particular kinds of web forms or unusual URL structures), they are less resistant to concept drift. The second method, meanwhile, looks for signs of phishing by comparing phishing websites to the authentic website that is being targeted. This approach is a smaller amount of resilient to zero-day phishing attacks than the prior one. Similarity-based approaches are helpful for rapidly weeding out a hefty number of phishing websites earlier putting them into machine learning-based algorithms, which frequently yield longer to categoriese.[1]

Figure 1. Phishing detection URL frameworks shown. This an important part of the defence against phishing attacks. They can aid in preventing users from being victims of these assaults and having their personal data stolen.

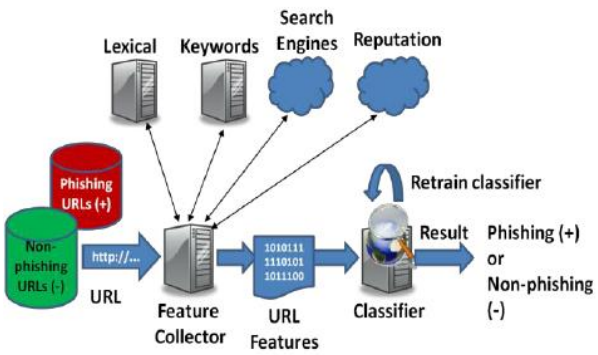

Fig 1. Phishing URL Detection Framework

II. RELATED WORK

The detection of phishing is a topic of extensive investigation. Some research concentration on the usage of whitelists and blacklists in anti-phishing organisms. Blacklist-based techniques maintain a gradient of known phishing website sphere names or links and warn users when they attempt to access those websites. However, phishing websites are extremely vibrant, and they often only exist for a short period of time. Zero-hour phishing assaults frequently avoid detection by blacklist-based techniques with ease. Whitelist-based systems, however, typically make it difficult for users to explore outside the secure websites.

"Phishing Detection: A Machine Learning Approach" by Kamalam, G.K. Suresh et al. (2022): This paper examines the effectiveness of many classifiers and suggests a phishing detection system based on machine learning. It makes advantage of attributes including page content, SSL certificate details, domain age, and URL length. The following classifiers are assessed: Support Vector Machines (SVM), k-Nearest Neighbours, Naïve Bayes, and decision trees. [9]

A comparison of several ML algorithms for phishing website identification is carried out by the authors of "Phishing Website Detection Via Machine Learning Techniques" by Dhanya. They assess algorithms like Support Vector Machines (SVM), Random Forest, Decision Trees, and Naive Bayes[10] on a list of websites that are phishing. The study examines how well these algorithms perform in terms of recall, accuracy, precision, and F1-score, highlighting both their advantages and disadvantages.

"Detecting Phishing Websites Using Machine Learning Techniques" by Mishra is another pertinent piece of literature. The effectiveness of machine learning algorithms, such as Decision Tree, Random Forest, and k-Nearest Neighbours (k-NN), is examined by the writers.[11] and SVM, in detecting phishing websites. They evaluate the algorithms using real-world datasets and equate their concert in terms of exactitude and computational efficiency. The study contributes to the understanding of different machine learning techniques and their applicability to phishing detection.

“Adremover: an enhanced machine learning technique for ad blocking” by M. Abdulaziz Saad Bubukayr and M. Frikha (2022): In recent decades, there has been a significant surge in online ads. Like most other websites, Facebook, Google, and Twitter all have advertisements on them. When you encounter ads that are tailored to your present internet browsing behaviour, you may have the constant feeling that someone is monitoring you. These occurrences can be the result of web monitoring. The original purpose of commercials was to assist companies in marketing to customers and closing deals on goods.[12]

"Phishing Websites Classification uses Random Forest Algorithm" by Dr. G Ramesh, R.B. Lokitha (2023): This work focuses on using the Random Forest algorithm for phishing website classification. Features like URL-based, HTML-based, and domainbased features are used for training the classifier. The study demonstrates that the Random Forest algorithm achieves good performance in detecting phishing websites. [13]

"A New Method for Identifying Phishing Websites through Machine Learning Techniques" authored by Ashit Kumar Dutta (2021): This study suggests a distinctive method for ascertaining phishing websites that makes use of machine

learning techniques. Features that are extracted and utilised as input include those that are domain-based, HTML-based, and URL-based. The writers test out many machine learning techniques, such as Random Forest, Naive Bayes, and Decision Trees. The study achieves high accuracy in predicting phishing websites and compares the performance of different classifiers.[14]

S. M. Istiaque et al.'s "Artificial Intelligence Based Cybersecurity: Two-Step Suitability Test"[15] Big data, IoT devices, global digitalization, and social media all present noteworthy threats to organisational and individual sanctuary in today's networked environment. Traditional security systems often fall short when it comes to cybersecurity for individuals and enterprises. Artificial intelligence (AI) is smart enough and adaptable enough to handle the constantly shifting world of cyber security. The application of AI is very beneficial for the following tasks: managing access, detecting spam, malware, and botnets; authenticating users; studying user behavior; and managing access. Models for machine learning (ML) are essential to artificial intelligence (AI). This study shows AI's value in the field of cyber security via a novel, hands-on method. It is displayed through a range of machine learning techniques.

H. Dao et al.'s paper "CNAME Cloaking-Based Tracking on the Web: Uncovering, and Protection" Online users' glancing habits can be gathered and allied obligations to third-party surveillance. To get around ad-blocking software and other anti-tracking procedures, the chasing service benefactors employed a novel tactic known as "CNAME cloaking". When a subdomain of the webpage a user is presently seeing uses a CNAME to resolve to a third-party dominion linked to tracking, it tricks browsers into thinking that the request is coming from the page they are currently reading.[3]

According to Kasif Saleem, Zainab Alshigiti, and colleagues (2023), "DeepPhish: A Phishing Detection Method Based on Deep Learning using CNN, LSTM, and LSTM-CNN": An endwise deep neural network called DeepPhish is proposed by the authors for phishing detection. Convolutional neural networks (CNNs) and long short-term memory (LSTM) networks are the archetypal syndicates. According to the study, DeepPhish works better than conventional machine learning techniques and achieves high accuracy.[16]

R. Pandey and colleagues' "Phishing Website Detection using Machine Learning Techniques: A Efficient Literature Review" (2020): An overview of machine learning methods for phishing website identification is given in this study of the literature. The study investigates a number of features that are often employed in phishing detection, such as domain-based, HTML-based, and URL-based features. A variety of machine learning methods are examined, including Random Forest, SVM, Decision Trees, and Neural Networks.[4]

III. PROPOSED METHOD

To increase the precision and resilience of the prediction system, hybrid machine learning systems for phishing website prediction combine many models or algorithms. Figure 2.The entire approach is explained in the flowchart below:

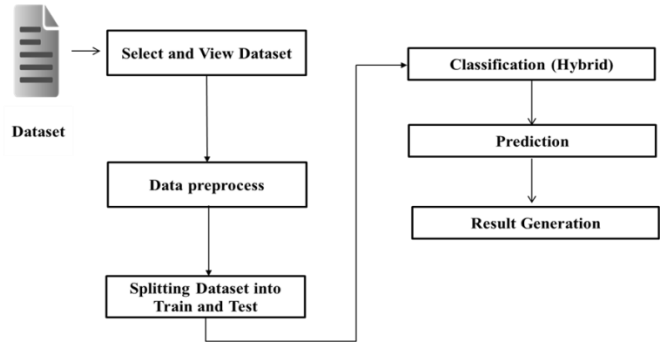

Fig 2. Flow Chart

Algorithm Steps:

Inputs:

DS (training dataset)
LA (Learning algorithm)
MC (model mixture procedure)
M (numeral of model to generate)
N (numeral of novel examples to generate)

Technique Hybrid (DS, LA, MC, m, n):

For i =1 to m
  DS_i (Disparity of DS)
  Let M_i = produced model by smearing LA to DS_i
For j = 1 to n
  a (arbitrarily generated model)
  c (class assigned to a by $MC_{M1.....Mm}$)
DS = DS U {(a , MC)}
Model = formed model by applying LA to DS
Return Model.

Statistics Collection and Loading: The progression of choosing a dataset and loading it into the Python environs is notorious as data selection.

Statistics Pre-processing: Selecting a dataset and importing it into a Python environs is notorious as data selection. The First Data Processing Steps. Pre-processing involves filtering away unsolicited data from a dataset.

Excruciating Dataset into Train & Test Data: The practice of separating a dataset in half, usually for use in a cross-validator, is known as statistics splitting. Semi of the data is used to physique a prediction model, while the other half is used to assess the mock-up's enactment.

Feature Extraction: Feature extraction is one way to standardise data independence. Commonly carried out at the

pre-processing juncture of data analysis, normalisation is another name for it.[17]

Classification: A combination of a gradient boosting classifier and random forest was employed to make the distinctions.

A supervised learning strategy used in machine learning may be thought of as a means of classifying or organising certain unknown data into a unique set of groups. Finding the relationship between a collection of highlight factors and an objective variable of interest is the aim of categorization. The objective attribute in classification is a straight out factor with discrete values. Classification creates the class grade for an unlabeled test case using the objective labels and a collection of predefined information points.

For classification, there are many different algorithms and machine learning approaches that may be used, such as naïve bayes, logistic regression, support vector machines, neural networks, logistic regression, k-nearest neighbour, decision trees, and linear discriminant analysis.

a) Random Forest: Making an N-by-1 random forest of decision trees is the first stage of Random Forest, and using those trees to make predictions is the second.[2][18]

The steps and a diagram illustrating the process are described below:

- Select at random K training data points.
- Second , build the decision trees that are linked to the contested points (Subsets).
- Choose N as the whole numeral of decision trees you intend to build in the third step.
- Repeat steps 1 and 2 in Step 4 to review.

Locate each decision tree's forecast for the newly added data points, then assign them to the category with the highest number of votes.

The random forest prediction can be expressed mathematically as follows:

$$\mathbf{H(X)} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{h_i}\,(\boldsymbol{X})$$

Where:

- $H(X)$ is the prediction made by the Random Forest for input data X.
- N is the numeral of decision trees in the forest.
- $h_i(X)$ is the prophecy of the i^{th} decision tree.

Each decision tree $hi(X)$ is proficient on a random subcategory of the training data & makes its peculiar prediction. The final prediction of the Random Forest is determined by a majority vote or averaging, depending on whether it's a classification or regression problem.

For phishing website prediction, the input features X would include various characteristics of a website, and the output $H(X)$ would indicate whether the website is classified as phishing or not.

b) Gradient Boosting: Gradient Boosting-Gradient Descent, one of the utmost well-liked optimisation strategies, is frequently castoff to train machine learning replicas by minimising the discrepancy amid expected and actual results. Gradient descent is also used in neural network training. Using gradient descent techniques on previously gathered data, moulds are formed using beginning constraints, and the rate purpose is computed by iteratively changing the constraints in the hopes of decreasing the rate utility.

The number of phases needed to spread the bottom is the definition of this phrase. This is a very small value that is tracked and modified based on the behaviour of the cost function. The possibility of overshooting the minimum is traded off for a quicker learning rate, which leads to larger advances. On the other hand, low learning rates expose the small step sizes, which sacrifice overall effectiveness in favour of higher precision.[2][19]

Here's a simplified representation of the formula:

$$\boldsymbol{F(X)} = \sum_{i=1}^{M} \boldsymbol{\gamma_i}\, \boldsymbol{h_i}\,(\mathbf{X})$$

Where:

- $F(X)$ is the final prediction through by the Gradient Boosting model for input data X.
- M is the numeral of trees in the communal.
- γ_i is the weight or learning rate allotted to the i^{th} tree.
- $h_i(X)$ is the forecast of the i^{th} decision tree.

c) K-Nearest Neighbors Algorithm (KNN): A supervised learning classification system called K-Nearest Neighbours uses a big number of named points to figure out how to identify new ones. This calculation classifies cases based on how similar they are to each other. In K-Nearest Neighbours, information points that are near to one another are called neighbours. This viewpoint is the basis of K-Nearest Neighbours. As a result, the distance between two circumstances expresses how different they are from one other. Finding the similarity or difference between two information points may be done in a number of ways. This should be possible, for instance, using Euclidean distance.[1]

$$\mathbf{H(X) = MajorityClass(N_{nearest}(X))}$$

Where:

- $h(X)$ is the prediction made by the KNN algorithm for input data X.
- N_{nearest} (X) represents the set of K nearest neighbors to the input data X.
- Majority Class $(\cdot)$ is a function that determines the majority class among the K nearest neighbors.

The algorithm estimates the detachment between the input data point X & all other data points in the training set, then selects the K nearest neighbors based on this distance. The prediction for X is determined by the majority class among these neighbors.

d) Decision Tree Algorithm: Recursive partitioning is a method that divides the training set into discrete nodes, each of which includes all or most of a single category of the data, in order to produce decision trees that classify data. By considering each feature separately, a decision tree can be made[12]:

- Select an trait from our dataset first. Determine the attribute's prominence in the data splitting process.
- After that, divide the data bestowing to the best attribute's value.
- After that, reiteration the process for the remaining properties in each branch. Once this tree is constructed, you may use it to forecast the category of cases that are unknown.

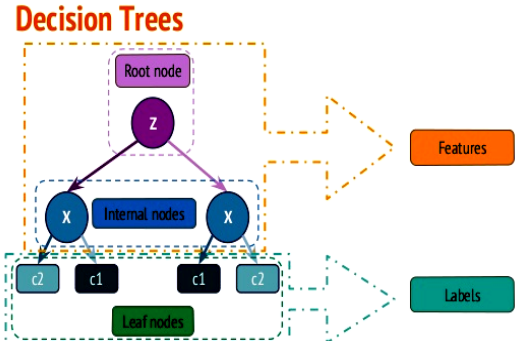

Fig 3. Decision Tree

A characteristic is checked in decision trees, and the cases are branched based on the test's result:

1. Every internal node has a test counterpart.
2. Every branch relates to a test result.
3. A patient is assigned to a class by each leaf node.

$$\mathbf{h(X) = Tree(X, \theta)}$$

Where:

- $h(X)$ is the prediction made by the decision tree for input data X.
- Tree(X,Θ) represents the decision tree function, which takes input data X and a set of parameters ΘΘ to make predictions.

The decision tree function recursively splits the data based on features in X according to the parameters Θ until a stopping criterion is met. The concluding prediction is made based on the widely held class in the leaf node where the input data falls.

Prediction

- It is a technique for identifying fraudulent Android apps within a database.
- Because of this study, the prediction findings of the research have performed better overall, which has led to more accurate forecasting of the dataset's data.

Experimental Setups:

This part provides further information about the phishing and genuine website dataset that we utilised in our learning, after which we briefly outline the investigate and technique to assess our suggested method.

A. Evaluation Methodology

Two well-liked ensemble learning methods in machine learning for both regression and classification applications are Random Forest and Gradient Boosting. These algorithms, which are part of the tree-based technique family, combine many weak learners—typically decision trees—to create a strong, more accurate model with the goal of enhancing prediction performance. While both ensemble techniques employ decision trees, Random Forest and Gradient Boosting take different approaches.

A confusion matrix is a bench that is commonly used to designate the recital of a classification prototypical (also notorious as a "classifier") on a set of trial data for which the true values are recognized. There can be some misinterpretation with the accompanying terminology, even though the confusion matrix itself is not too hard to comprehend.

Performance Evaluation Parameters:

True Positive (TP): Predicted tenets correctly predicted as authentic positive.

False Positive (FP): Predicted tenets incorrectly prophesied an actual positive. i.e., Negative values prophesied as positive

False Negative (FN): Positive values prophesied as negative

True Negative (TN): Predicted tenets correctly predicted as an authentic negative

Accuracy: The ability of the classifier is raised to as accurateness. It truthfully predicts the class label, and predictor accuracy trials how meritoriously a particular predictor can estimate the expected attribute value for a fresh set of data.

$$AC = \frac{TP+TN}{TP+TN+FP+FN}$$

Precision: The amount of true positives alienated by the total number of true positives + false positives is the definition of precision.

$$Precision = \frac{TP}{TP+FP}$$

Recall: The amount of right consequences alienated by the total number of results that ought to have been reverted is known as recall..

$$Recall = \frac{TP}{TP+FN}$$

F-Measure: The biased harmonic mean of the test's precision and recall is notorious as the F measure, often notorious as the F1 score or F score. It serves as a gauge of a test's accuracy..

$$\text{F-measure} = \frac{2TP}{2TP+FP+FN}$$

Error Rate: The term "error of the method" refers to the expected output values' inaccuracy. The error is given as an error rate if the goal values are categorical. This is the percentage of instances in which the forecast is incorrect.

$$\text{Error Rate} = 100 - \text{Accuracy}$$

Sensitivity: The capability of a machine learning prototypical to identify positive specimens is restrained by its sensitivity. It is sometimes referred to as recall or the true positive rate (TPR). Since it shows us how many positive cases the model was able to correctly detect, sensitivity is used to assess model performance.

$$\text{Sensitivity} = \frac{TP}{TP+FN}$$

Specificity: The ability of the algorithm or prototypical to forecast a true negative for every accessible sort is known as specificity.

$$\text{Specificity} = \frac{TN}{TN+FP}$$

B. Datasets

We used latest dataset entirely dissimilar from the ones used for the resemblance scrutiny and choosing the best distance threshold to assess the performances. The experimental dataset will be made available for use in upcoming research projects. The phishing URLs that PhishTank[20] offers are frequently phoney login or sign-in pages, or folios with contribution forms for account facts theft.

There are over 13,300 phishing websites in total that have been collected. As an alternative of retrieving the Hyper Text Markup language cause code using the built-in we get or ringlet tackles, which gave us the Hyper Text Markup language as solidified in a system application, we utilised Selenium & Chromium to get the website content. We partake 9,034 phishing websites[5] that were testified by users to PhishTank amid 28/05/2020 and February 22, 2021 after cleaning and eliminating error/empty pages. By looking at the site URL of the mooring folio and removing folios whose domain is among the topmost 500 spheres, we also deleted cloaked phishing websites, indicating that they were leading to legitimate websites.

In the meantime, we assembled authentic pages in the Conjoint Crawl database to generate a sample dataset of legitimate websites. We chose the top 4,000 domains from the Tranco list2, which was produced on March 14, 2021. We pulled 100 pages from the February/March 2021 crawl archive of Common Crawl for each site.3 180,302 webpages remain after empty and error pages have been eliminated.[21]

IV. RESULT GENERATION

All in all, classification and forecasting will be applied to build the ultimate result. Various measures, including accuracy and error rate, are employed to evaluate the effectiveness of a certain strategy.

Based on the overall categorization and forecast, the outcome is produced. This recommended method's effectiveness is evaluated using a numeral of metrics, including sensitivity, specificity, recall, accuracy, precision, and F1-measure. Based on the total categorization and projection, the ultimate outcome will be determined. Bench 1, displays the hybrid method's suggested simulation results.

Bench 1. Assessment b/w preceding work & planned work

Sr. No.	Parameters	Preceding Work [1]	Planned Work
1	Method	KNN	Hybrid (Proposed)
2	Precision (%)	95	96
3	Recall (%)	94	97
4	F Measure (%)	95.93	97.60
5	Sensitivity (%)	90.14	97.34
6	Specificity (%)	95.34	99.42
7	Accuracy (%)	87.14	98.14
8	Classification Error (%)	4	2.04

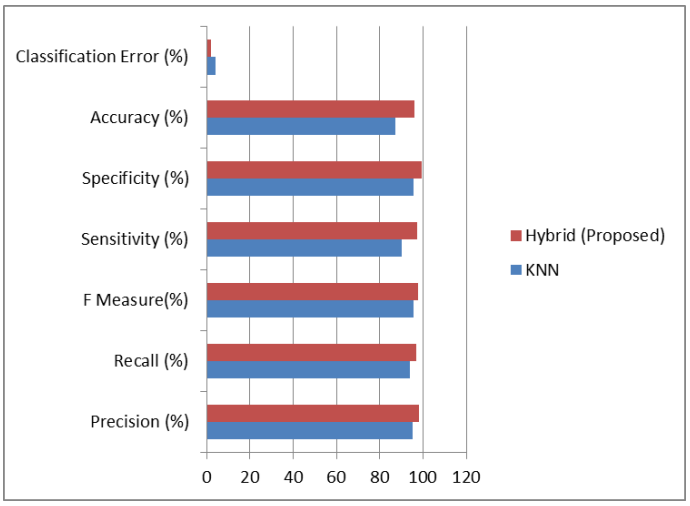

Fig 4. Comparison of preceding work & planned work

In Figure 4, is screening the result comparison of the preceding and planned work. Compared to previous art forms, this new one was more accurate.

Bench 2. Outcome Assessment

Sr. No.	Procedures	Exactitude (%)
1.	Decision Tree	91.51
2.	K-Nearest Neighbour	97.69
3.	Random Forest	94.44
4.	Hybrid (Proposed)	98.14

Bench 2, displays the accuracy of the random forest at 94.44%, the decision tree at 91.51%, the KNN at 97.69%, and the suggested hybrid at 98.14%. Thus, it is evident from the simulation results that the suggested work much outperformed the findings of the previous study.

V. CONCLUSION

Phishing is currently one of the largest threats to computer security. Furthermore, fresh phishing scams that pose a greater risk to security are always emerging. Phishing is the act of sending out electronic messages professing to be an honourable source in an endeavour to get sensitive information, including passwords, credit card details, and, on rare occasions, money. Phishing, when done maliciously, aims to gain credit card numbers, usernames, and other sensitive information (and, on rare occasions, money indirectly).

This presents an efficient technique for phishing prediction with performance augmentation. In this work, the machine learning classifiers are used to predict the phishing website. The pre-processing approach uses the phishing data as input data. Use the label encoding in the pre-processing procedure after cleaning the dataset. In the feature selection procedure that follows, the dataset is riven into training & testing datasets. The simulated domino effect show that the planned hybrid classification approach performs more accurately than the state-of-the-art techniques. The hybrid network achieved 97.96% accuracy, while the current KNN only managed 96%. Because of this, the simulation results clearly show that the recommended study has outperformed earlier research in terms of results. Because of this, the simulation results clearly show that the recommended study has outperformed earlier research in terms of results.

To reach even higher performance, the suggested clustering and classification methods may be extended or modified in the future. We plan to enhance the stacking ensemble structural design and apply our methodology to other real-world scenarios in the future.

VI. REFERENCES

[1] F. Yahva et al., "Detection of Phising Websites using Machine Learning Approaches," 2021 Int. Conf. Data Sci. Its Appl. ICoDSA 2021, pp. 40–47, 2021, doi: 10.1109/ICODSA53588.2021.9617482.

[2] M. K. Pandey, M. K. Singh, S. Pal, and B. B. Tiwari, "Prediction of phishing websites using machine learning," Spat. Inf. Res., vol. 12, no. 02, pp. 18–21, 2022, doi: 10.1007/s41324-022-00489-8.

[3] H. Dao, J. Mazel, and K. Fukuda, "CNAME Cloaking-Based Tracking on the Web: Characterization, Detection, and Protection," IEEE Trans. Netw. Serv. Manag., vol. 18, no. 3, pp. 3873–3888, Sep. 2021, doi: 10.1109/TNSM.2021.3072874.

[4] R. Mahajan and I. Siddavatam, "Phishing Website Detection using Machine Learning Algorithms," Int. J. Comput. Appl., vol. 181, no. 23, pp. 45–47, 2018, doi: 10.5120/ijca2018918026.

[5] "Kaggle: Your Home for Data Science." https://www.kaggle.com/datasets/isatish/phishing-dataset-uci-ml- csv?select=uci-ml-phishing-dataset.csv (accessed Jul. 27, 2023).

[6] I. Tyagi, J. Shad, S. Sharma, S. Gaur, and G. Kaur, "A Novel Machine Learning Approach to Detect Phishing Websites," 2018 5th Int. Conf. Signal Process. Integr. Networks, SPIN 2018, pp. 425–430, Sep. 2018, doi: 10.1109/SPIN.2018.8474040.

[7] A. Joshi and P. T. R. Pattanshetti, "Phishing Attack Detection using Feature Selection Techniques," SSRN Electron. J., Jul. 2019, doi: 10.2139/SSRN.3418542.

[8] M. H. Alkawaz, S. J. Steven, A. I. Hajamydeen, and R. Ramli, "A comprehensive survey on identification and analysis of phishing website based on machine learning methods," ISCAIE 2021 - IEEE 11th Symp. Comput. Appl. Ind. Electron., pp. 82–87, Apr. 2021, doi: 10.1109/ISCAIE51753.2021.9431794.

[9] G. K. Kamalam, P. Suresh, R. Nivash, A. Ramya, and G. Raviprasath, "Detection of Phishing Websites Using Machine Learning," 2022 Int. Conf. Comput. Commun. Informatics, ICCCI 2022, no. June, 2022, doi: 10.1109/ICCCI54379.2022.9740763.

[10] S. Jain and C. Gupta, "A Support Vector Machine Learning Technique for Detection of Phishing Websites," 2023 6th Int. Conf. Inf. Syst. Comput. Networks, ISCON 2023, 2023, doi: 10.1109/ISCON57294.2023.10111968.

[11] V. Muppavarapu, A. Rajendran, and S. K. Vasudevan, "Phishing detection using RDF and random forests," Int. Arab J. Inf. Technol., vol. 15, no. 5, pp. 817–824, 2018.

[12] M. Abdulaziz Saad Bubukayr and M. Frikha, "Web Tracking Domain and Possible Privacy Defending Tools: A literature review," J. Cyber Secur., vol. 4, no. 2, pp. 79–94, 2022, doi: 10.32604/jcs.2022.029020.

[13] R. B. Lokitha, R. R. Monisha, N. S. Neha, T. Nadu, and T. Nadu, "Phishing Detection System using Random Forest Algorithm," vol. 8, no. 4, pp. 510–514, 2023.

[14] A. K. Dutta, "Detecting phishing websites using machine learning technique," PLoS One, vol. 16, no. 10 October, pp. 1–17, 2021, doi: 10.1371/journal.pone.0258361.

[15] S. M. Istiaque, M. T. Tahmid, A. I. Khan, Z. Al Hassan, and S. Waheed, "Artificial Intelligence Based Cybersecurity: Two-Step Suitability Test," 2021 IEEE Int. Conf. Serv. Oper. Logist. Informatics, SOLI 2021, 2021, doi: 10.1109/SOLI54607.2021.9672437.

[16] Z. Alshingiti, R. Alaqel, J. Al-Muhtadi, Q. E. U. Haq, K. Saleem, and M. H. Faheem, "A Deep Learning-Based Phishing Detection System Using CNN, LSTM, and LSTM-CNN," Electron., vol. 12, no. 1, pp. 1–18, 2023, doi: 10.3390/electronics12010232.

[17] N. Megha, K. R. Remesh Babu, and E. Sherly, "An Intelligent System for Phishing Attack Detection and Prevention," Proc. 4th Int. Conf. Commun. Electron. Syst. ICCES 2019, pp. 1577–1582, Jul. 2019, doi: 10.1109/ICCES45898.2019.9002204.

[18] S. P. Ripa, F. Islam, and M. Arifuzzaman, "The emergence threat of phishing attack and the detection techniques using machine learning models," 2021 Int. Conf. Autom. Control Mechatronics Ind. 4.0, ACMI 2021, Jul. 2021, doi: 10.1109/ACMI53878.2021.9528204.

[19] K. E. Aydin and S. Baday, "Machine Learning for Web Content Classification," Proc. - 2020 Innov. Intell. Syst. Appl. Conf. ASYU 2020, Oct. 2020, doi: 10.1109/ASYU50717.2020.9259833.

[20] "PhishTank | Join the fight against phishing." https://phishtank.org/ (accessed Jul. 27, 2023).
[21] K. S. Swarnalatha, K. C. Ramchandra, K. Ansari, L. Ojha, and S. S. Sharma, "Real-Time Threat Intelligence-Block Phising Attacks," CSITSS 2021 - 2021 5th Int. Conf. Comput. Syst. Inf. Technol. Sustain. Solut. Proc., 2021, doi: 10.1109/CSITSS54238.2021.9683237.