ISSN: 0976-9102 (ONLINE)
DOI: 10.21917/ijivp.2026.0563

REAL TIME PHISHING DETECTION USING FEATURE EXTRACTION AND HYBRID ENSEMBLE MODELS

B. Kalaimathi
Department of Electronics and Communication Engineering, Sri Ramakrishna Engineering College, India

Abstract

Cyber security attacks and threats are prone among the websites used by people now a days. The most vicious threat for people surfing on the websites is phishing. Phishing is recognized as a malicious act of cybercrime to deceive users and steal important data such as Banking Credentials, ATM PINs, cards, and CVV numbers. Our proposed method is based on detecting these websites using a Hybrid Ensembler model. The URL is given as input and the feature extraction is done by thirty-one parameters which mainly include URL length, HTTPS request/response, IP address, sub-domain, etc The modal is trained using various machine learning algorithms like decision tree, K-Nearest, logistic regression, naïve bayes and hybrid ensembler and the output is obtained numerically. The model is trained in such a way that it gives more accuracy and efficiency than the existing method. The Hybrid Ensembler is used to pick a particular algorithm that gives better results for a certain dataset. Python functions are written for each feature extraction and the model is trained in google colab. The instructed modal is transfigured to a pickle file using python pickling for deployment in the flask framework. Flask framework and other UI development tools are used to create web applications where the user can enter the URL to check whether it is a Phishing website or a legitimate website.

Keywords:

Cybersecurity, Machine Learning, Web Development, Phishing Detection, Ensemble

1. INTRODUCTION

In Modern Days we come across different websites for various purposes which include Online Shopping, Film Booking, Study Portals, Journal Pages, Streaming Websites, social media, Tutorial platforms, etc. Some of these websites involve the process of online money transactions and exposure of bank credentials like ATM Pin, UPI Pin, Account Numbers, and OTPs like Online Travel Ticket Booking Websites, Film Booking Websites, etc., and some don't. Also, some websites replicate as same as the websites that involve the process of online transactions, these websites are called Phishing Websites. The concern here is the user might end up on a Phishing Website thinking that it is a legitimate website for their specific purpose. To detect those phishing websites our proposed method is Web Application for Phishing Detection Using Hybrid Ensembler with other machine learning algorithms.

Machine learning is an unique approach which was initially employed for constructing analytical models in supervised learning. By utilizing machine learning techniques, models for identifying phishing activities based on classifying old web pages can be developed and then integrated into web browsers. For instance, when a user accesses a website, the machine learning models can rapidly distinguish between authorized and fraudulent websites and provide the output to the user. It is a form of cyber-threat in which an individual is tricked into visiting illegitimate websites and disclosing sensitive data such as their personal identity, banking information, credit card details, passwords, and more. Given the paramount importance of online security, phishing has piqued the interest of numerous professionals and researchers.

2. PRINCIPLE

This system aims to detect whether a given website is legitimate or a phishing website that aims to steal sensitive information from users. The proposed method is a web application where the user needs to enter the complete URL i.e., with the internet protocol extension (HTTP/HTTPS) as the input, and the feature is extracted from the given URL. The model is trained with Hybrid Ensembler as its main algorithm and other various algorithms. The Algorithm which gives the highest Accuracy is used for training dataset modals. The Feature Extraction contains 29 parameters with 29 python functions for each parameter the feature for a given input is extracted and the result is predicted most efficiently and accurately possible. The most popular Python Framework for Web Development called the Flask Framework is used for UI Development along with HTML and CSS are used for styling. Also, Bootstrap is used for animation while the output is being processed in the backend which makes the UI more sophisticated. The result is displayed in the UI whether it is Phishing Website or not.

2.1 LITERATURE SURVEY

Sundara Pandiyan et al. [1] as our base paper in which they used 17 parameters for feature extraction while we used 29 parameters which enables the model to get to know about the uniqueness of the given URL and predict the results with more accuracy. Also, we have used Decision Tree, Random Forest, and SVM as common algorithms compared to this paper and the decision tree and random forest are trained far more better and consistent and rigorous datasets are given. Our proposed method is one step ahead compared to as we have used a greater number of parameters for feature extraction, produced higher accuracy in common algorithms, and developed a web application that makes this more interactive for the user. The Proposed Method's Average Train Data Accuracy percent increased by around 10.47% and Average Test Accuracy increased by 9.62% compared to. It is Furthermore explained in the comparison part.

Aydin et al. [2], we got an idea and exposure to the process of feature Extraction from the URL. Although they have used 133 parameters for feature extraction, our 29 parameters for feature extraction are more than enough for detecting modern-day Phishing Websites. Since, the paper [2] explains only feature extraction and nothing about phishing Detection with Feature Extraction. We have used the papers [3] and [11] to study the python flask framework for web development. It is the most Popular Web Development Framework across the world. The

software which uses python as its backend language and needs to be developed as a Web Application, Flask is the most suited and reliable framework to do so. In our base paper they didn't develop the model as a web application and it works more like a kind of console application. To overcome this Setback, we came up with the idea to develop our model as a Web app. The papers [13], [12] and [5] have been used to learn about the ensemble techniques and models. As mentioned earlier in the principle, Hybrid Ensembler is an algorithm that chooses the best algorithm to train the model based on accuracy.

These papers were of great assistance with the implementation of the Hybrid Ensembler Algorithm. The papers [8] and [4] were referred for implementation of the process of feature extraction to detect phishing websites. However, there are limitations to these papers like predicting the Phishing websites of modern days as legitimate websites, since they have used only limited parameters for feature extraction. i.e., [4] used 9 parameters and [8] used 12 parameters for feature Extraction. The studies on web development have been carried out by referring to [14] which gave us a clear objective of how to develop a sophisticated and responsive web page. CSS (Cascading Style Sheets) were used for styling, and HTML acts as a building block of the web page. The paper [15] has been used to study the Bootstrap Framework used for animation in web pages. The Table.1 furnishes the details of the number of parameters used in previous methods along with the proposed method.

Table.1. Comparison of the Number of Parameters

S. No	Related Works	Number of Parameters
1	Proposed Method	29
2	Sundara Pandiyan [1]	17
3	Patil et al. [4]	12
4	Ravindra et al. [8]	9

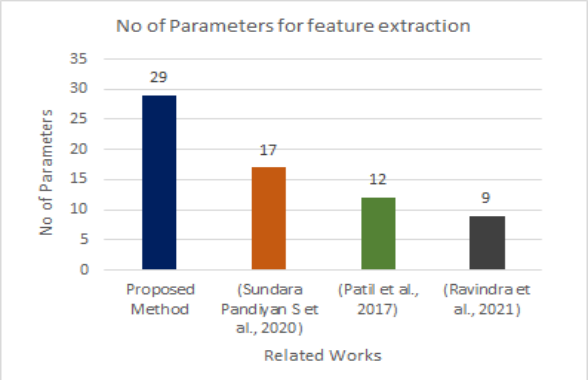

Fig.1. Comparison in number of parameters for feature extraction

The Fig.1 shows the graphical representation for increased number of parameters in the proposed method

3. PROPOSED SYSTEM ARCHITECTURE

The Fig.2 shows the workflow and proposed architecture. The top half of the diagram is about importing datasets, splitting train data and test data, and training the model using ML algorithms. The bottom half explains feature extraction. First step in this process is acquiring raw data. This set of data is derived from University of California ,Irvine ML Repository and Kaggle. Phish Tank, an open-source service is used to collect the phishing URLs in various formats like CSV, JSON, etc which is updated hourly at https://www.phishtank.com. The data is then compiled in advance to improve its authenticity and definiteness for ML algorithms to learn from. The preprocessing step usually involves data cleansing, where junk or missing values are removed, and invalid values are fixed to ensure that the dataset is of good quality. This step is critical since scrutinize the data, the efficient it is for ML algorithms to provide accurate outcomes. The next step in the process is building machine learning models using versatile algorithms such as logistic regression, naive Bayes, K-Nearest Neighbours, Decision Tree, SVM, Random Forest, AdaBoost and Hybrid Ensembler Each algorithm has its advantages and is used to produce the outcome. They have their own advantages to check whether the data passed has features that could be either a phished or legitimate one.

After the models have been built, their results are compared, and the best one is chosen by its performance. The hybrid ensembler method is used to improve the performance and the reliability of the chosen model. It combines the predictions of multiple models to give a more precise outcome. To further enhance the performance and reliability of the system, The dataset of newly entered URLs by the user is obtained by performing dynamic feature extraction. The algorithm can then produce accurate results for newly entered data. This step is critical since phishing websites are continually evolving, and new features need to be added to the system to keep up with the changes.

Finally, the software that uses all the above process must be capable of checking whether the URL is phished or legitimate one. This prediction is based on the features extracted from the URL . The following are the steps involved in detecting phishing websites using ensemble learning:

- **Data Collection**: The first step is collecting the dataset of existing known phishing and legitimate websites on the world wide web. The Algorithms are trained using this Dataset.
- **Feature Extraction**: Secondly, our aim is to extract features from the websites. The features can be based on the URL, HTML content domain age, URL length, number of subdomains, and presence of certain keywords and other certain features.
- **Model Training**: The next step is to instruct the ML algorithms on the collection of data. different algorithms like decision tree, support vector machines logistic regression and neural networks can be used
- **Ensemble Learning**: In the next step the overall accuracy is improved by combining the outputs of the different models to the accurate result. Ensemble techniques such as bagging, boosting, and stacking can be used.
- **Evaluation**: The final step is to test the performance of the model on a separate dataset. Common metrics used for evaluation include precision, recall, F1 score and accuracy.

Fig.2. Diagram of Proposed Architecture

4. PROPOSED WORK

4.1 TYPES OF FEATURES

In the existing model is the usage of limited features, whereas in our project we will be using features from the URL like

1. **URL-Based Features**: The URL plays a pivotal role in determining whether the website is phishing or not . When the URL is deduced, features linked to these points acquire, the following are some URL-based features:

- Digit count in the URL
- Total length of URL
- Having @ Symbol in the URL.
- Prefix/Suffix in the URL
- Number of subdomains in the URL
- Is Top Level Domain (TND)

2. **Domain-Based Features**: Phishing Domain Detection involves identifying fraudulent domain names through the use of passive queries to gather relevant data. Certain domain-based features have proven useful in distinguishing between legitimate and phishing domains, including:

- Presence of domain name or its IP address in blacklists of reputed services
- Count of days passed since the domain was registered.

3. **Page-Based Features**: It makes use of data from reputation. Ranking system to calculate information about pages, some of these characteristics show that the website is trustworthy. The following are some of the page-based features: Page rank, Google Index and Web Traffic.
4. **Content-Based Feature**: To achieve these features, an consistent scan of the target domain is required, involving the analysis of page content to determine if it is being utilized for phishing. The following details pertain to pages that have undergone processing: Page Titles, Meta Tags, Hidden Text, Text in the Body and Images etc.

4.2 FEATURE EXTRACTION

Feature extraction is an important step in predicting a phishing site because it allows machine learning algorithms to process and analyze the characteristics of the website in question. By extracting relevant features from the website, such as the URL, the HTML code, the visual design, and the content, the algorithm can identify patterns and correlations that are indicative of a phishing attack. For example, some common features that are extracted from phishing websites include the use of misleading or deceptive URLs, the presence of fake login forms or pop-ups, and the use of engineering techniques to influence users into reveling the sensitive information. By analyzing these features, machine learning models can learn to distinguish between legitimate websites and phishing sites with high level of accuracy. Overall, feature extraction is a critical step in the process of predicting phishing sites because it enables machine learning algorithms to identify the subtle cues and patterns that are indicative of a phishing attack. Without feature extraction, it would be much more difficult to accurately identify and classify phishing websites, which could lead to serious security risks for individuals and organizations alike.

In the existing method, they are extracting only 17 features. But in the proposed method we are extracting 29 features from the website. The Unique features of the stated methodology compared to the existing method are listed in the Table.2.

Table.2. List of Unique Features

1. Having @ Symbol	12. Abnormal URL
2. Having_Sub_Domain	13. Redirect
3. SSL_Final_State	14. On-mouseover
4. Domain_Registration _Length	15. Right-Click
5. Favicon	16. Pop-up Button
6. Port	17. Web Traffic
7. HTTPS token	18. Page Rank
8. URL of Anchor	19. Google Index
9. Links in Tags	20. SFH
10. Links Pointing to page	21. Statistical Report
11. Submitting to E-mail	

The 29 features extracted from the website by URL are listed and described below,

- **having_IP_Address**: The possession of IP Address can raise suspicion if it is utilized as the domain of a URL (e.g.,

http://172.253.123.99/abc.html), indicating a potential attempt to steal information.

- **URL Length**: URL Length is another factor that can indicate phishing. If the length of the URL is less than 75 characters, it can be classified as a legitimate website. However, if it is equal to or exceeds 75 characters, it might probably be a phishing website.
- **Shortining_Service**: Shorter URLs can be created using the technique called URL Shortening. it is the service that redirects them to longer web page URLs (e.g. http://productbase.ac.nic.ig/ can be shortened to "bit.ly/56FLW6"). However, URL shortening services may also be used by phishing sites.
- **Having @ Symbol**: The use of the @ symbol in a URL causes the web browser to everything that follows.
- **Double Slash Redirecting**: The user will be redirected to another website, if the characters "//" appear in a URL. Basically, this occurs in the sixth position of the URL, as exemplified by the following link: http://randomsite.nac.in.
- **Prefix or Suffix**: Although it is uncommon for authentic URLs to include hyphens, phishers might include a hyphen to create an image of a legitimate website through domain name. To deceive users into believing it is genuine, they might establish a site such as http://www.randomsite-browse.com.
- **Having Subdomain**: Domain name can contain a code that represents to a particular country (ccTLD) or educational institution (such as "id" or "ac," respectively). This combination of codes is called a two-level domain (SLD). A URL can be determined suspicious if it contains multiple number of dots after removing "www", "ccTLD", and SLD. A legitimate URL generally do not have dots or subdomains.
- **SSL_final__State**: HTTPS is a critical component of any website that prioritizes security and authenticity. An URL with HTTPS as its transfer protocol is considered to be a secured one but URL with HTTP might not be as much secure as HTTPS. Phishing often appears to be with HTTP transfer protocol.
- **Domain Registration Length**: They are only available for a short period of time and registered for a year, which can be a sign that they're illegitimate.
- **Favicon** : It is an image appears in the browser tab that represents a website. If it is displayed separately from the website address, it could be a sign that it is a phishing website
- **Port**: Ports are used for verifying certain services like HTTP, and can be automatically blocked or opened with firewalls, proxies, and Network Address Translation (NAT). However, if all ports are open, phishers can take advantage. They can exploit loopholes and gain access to desired services, there by stealing information. If the ports of the website are open, then the probability of the URL being phishing website id high.
- **HTTPS Token**: Phishers can add a fake HTTPS token to a domain URL, like in the example http://amazon-secure-login.com/secure/login.php?token=https://www.amazon.com/. This type of phishing website is designed to look like a legitimate login page for Amazon, but the URL contains a fake token that redirects the user to a phishing site. The aim is to deceive the user into entering their login information on the fake site, which the phisher can then use to steal their personal information.
- **Abnormal URL**: If a website is genuine, its URL should contain the website's identity.
- **Links in Tags**: During our investigation of webpage source code, we found that legitimate one commonly utilize the tags to provide Metadata about the HTML document. But, phishing websites might contain irregular Metadata about the HTML document.
- **SFH**: SFHs that include an empty string which are meant to be taken into account suspicious as action need to be taken on the submitted information.

Additionally, if the domain name in the webpage mismatches domain name of the SFH, it indicates that the webpage is questionable since external domains rarely handle submitted information.

- **Submitting to email**: When personal information is submitted on an official website, it is sent to the server for processing. The server-side scripts like "mail()" and client-side scripts like "mailto()" can be used to determine whether a phisher is involved during the exchange of information or not.
- **Abnormal URL**: The WHOIS database can reveal whether a website has a legitimate URL, which typically includes its identity.
- **Redirect**: distinguishing phishing websites from legitimate ones is an easy task. In our set of data, legitimate websites have been redirected at most once, while phishing websites have been redirected to the least of four times.
- **On Mouse Over**: Phishers use JavaScript to display a illegitimate URL in the status bar, which is identified by checking the source code of the webpage.
- **Right Click**: Phishers turn off the right-click function using JavaScript to prevent users from accessing and saving the webpage source code. It is similar to "Depending on Mouse over to hide the Link," and can be detected by searching for the event "event. Button==2" in the webpage source code to see if the right-click function is turned off .
- **Pop-up Butto**n: The website which asks about private information of the user is considered to be a phishing website. It is uncommon for legitimate websites to ask for personal information through pop-up windows. However, some legitimate websites use this feature to warn users about illegitimate activities or share a welcome message, but they do not request private information through pop up windows.
- **IFrame**: It allows for displaying another webpage within the currently displayed page. Phishers may exploit this tag by making it invisible using the "frame border" attribute.
- **Age of Domain**: The nature of the website can be identified using the age of domain. The age of a domain can be acquired from the WHOIS database. Typically, phishing websites have a short lifespan. Our dataset shows that legitimate domains have span of 6 months.

- **DNS Record**: Illegitimate website will either have an unrecognized claimed identity in the WHOIS database or no records associated with the hostname. If it is empty or not found, the website is considered illegitimate. On the other hand, if it is present, the website is legitimate.
- **Web Traffic**: Web traffic is a measure of a website's popularity based on the number of viewers and pages viewed. Since, phishing websites have short lifespans, they may not be recognized by the Alexa database. It is said that legitimate websites typically rank among the top 100,000, even in the worst cases. It is considered as phishing if it is not recognized by Alexa, it is classified as phishing, while recognized domains with low traffic are classified as suspicious.
- **Page Rank**: It is a value between zero and one that measures a webpage's importance on the internet. In our dataset, 95% of the phishing webpage have no page rank while 5% have value upto 0.2. It has high probability of being a phishing website.
- **Google Index**: Google Index examine whether it is included in Google's index or not. A site that is indexed by Google appears in search results. Phishing webpages are typically available for a short time, so many may not be found in the Google index.
- **Links Pointing to Page**: The Legitimacy of a website can be figured out by the count of links directing to that particular website even though they are from the same domain. we found that 98% of them in our dataset had no links pointing to them. On the other hand, Legitimate websites have at least two external links pointing to them.
- **Statistical Report**: Statistical reports on phishing websites are regularly published by various sources, such as Phish Tank. These reports may be monthly or quarterly.

4.3 MACHINE LEARNING ALGORITHMS

In this proposed method, the Hybrid Ensembler algorithm acts as the main algorithm which chooses the best algorithm to train the model for specific data based on accuracy. We will distinguish the error rate and execution time of different algorithms and propose the best one for our task. Additionally, we will consider incorporating additional features that we believe could improve the accuracy of our predictions.

To begin, we imported the phishing dataset from UCI and Kaggle into the Colab environment and pre-processed it by checking for null values and verifying that the target class was evenly distributed. We then identified correlations between features and removed multicollinear and weakly correlated columns to simplify the machine learning model and improve its performance. The dataset was then split into training and validation data at a 75:25 ratio using sklearn libraries. We used the same libraries to train and create the machine learning models, feeding input and output data to allow the models to learn. We predicted values using the validation data and used sklearn to generate a score, accuracy, and classification report.

In the testing phase, we will obtain real-time data such as a URL, and parse it. We will apply the necessary conditions to classify the features (-1, 0, or 1) and then use the best-trained model to predict whether the given data represents a phishing website or not. The obtained Train accuracy and Test accuracy for various Machine Learning Algorithms are listed in the Table.3.

Table.3. Algorithms and their Data Accuracy

Algorithm	Train Accuracy	Test Accuracy
Decision Tree	0.9879	0.9591
Random Forest	0.9879	0.9645
SVM	0.7096	0.7015
K-nearest neighbour	0.9828	0.9489
Logistic Regression	0.9272	0.9218
Bernoulli's Naïve Bayes	0.9115	0.9059
Adaboost	0.9342	0.9290
Hybrid Ensembler	0.9863	0.9621

The average Train data accuracy is 0.9284 and the average Test data accuracy is 0.9116.

5. CONFUSION MATRIX AND CLASSIFICATION REPORT OF PROPOSED WORK

A table-shaped arrangement known as a confusion matrix is utilized to appraise the effectiveness of a classifier on a dataset with already known true values.

Table.4. Results for decision tree

	Precision	Recall	F1- Score	Support
-1	0.95	0.96	0.96	1236
1	0.97	0.96	0.96	1528
Accuracy			0.96	2764
Macro average	096	0.96	0.96	2764
Weighted average	0.96	0.96	0.96	2764

Table.5. Results for Random Forest

	Precision	Recall	F1 Score	Support
-1	0.97	0.95	0.96	1236
1	0.96	0.97	0.97	1528
Accuracy			0.96	2764
Macro average	0.96	0.96	0.96	2764
Weighted average	0.96	0.96	0.96	2764

Table.6 Results for SVM

	Precision	Recall	F1-score	Support
-1	0.67	0.66	0.66	1236
1	0.73	0.74	0.73	1528
Accuracy			0.70	2764
Macro Average	0.70	0.70	0.70	2764
Weighted Average	0.70	0.70	0.70	2764

Despite the possibly bewildering terms used in the matrix, comprehending it is typically straightforward and uncomplicated.

The confusion matrix has several terms, including True Positive(TP) where the classifier predicts yes (Phishing URLs), and True Negatives(TN) where the classifier predicts no (non-Phishing URLs). When a classifier predicts "yes" incorrectly, it's termed False Positive(FP), which is a Type I error. Conversely, when a classifier predicts "no" inaccurately, it's called False Negative (FN), which is a Type II error. The confusion matrix derives True Positive and True Negative values from the classifier's expectations, also referred to as predictions. In contrast, True and False terms are obtained from external observations. The Table.4-Table.6 provides the results for the Train accuracy and Test accuracy by the algorithms Decision Tree, Random Forest and SVM.

5.1 WEB AND UI DEVELOPMENT

We have created a web application using the python flask framework. Machine learning models can be trained, deployed, and integrated into web applications in a streamlined and efficient manner by Using the flask Framework. Building, training, and deploying machine learning models is made simple by Flask's support for a number of machine learning libraries, including TensorFlow, Scikit-learn, and Keras. Thus, when a user inputs a URL, the extension uses the GET method to retrieve the URL and then sends it to the Python code via the extension's JavaScript to the Python backend. After that, the Python function creates an array by extracting every feature from the URL. Next, we evaluate this using the Hybrid Ensembler Model that has been trained.

6. COMPARISON OF RESULTS

The Table.7 shows the performance comparison of common algorithms used in existing method and our proposed method.

Table.7. Results for common algorithms

Algorithm	Existing Method		Proposed Method	
	Train Accuracy	Test accuracy	Train Accuracy	Test Accuracy
Decision Tree	0.815	0.808	0.9879	0.9591
Random Forest	0.807	0.808	0.9879	0.9645
SVM	0.803	0.796	0.7096	0.7015

The Fig.3 shows the comparison of Train accuracy in common algorithms and "Fig.4" shows the comparison of Test accuracy in common algorithms.

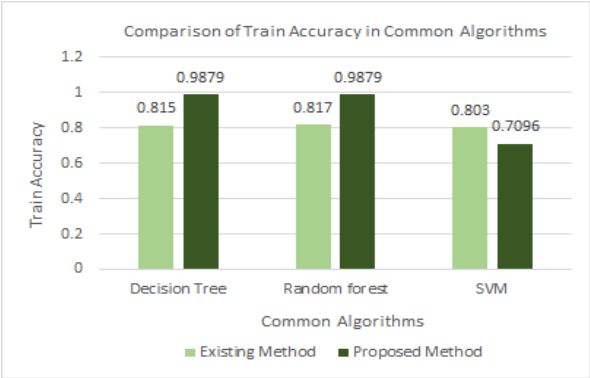

Fig.3. Comparison of Train Accuracy in Common Algorithms

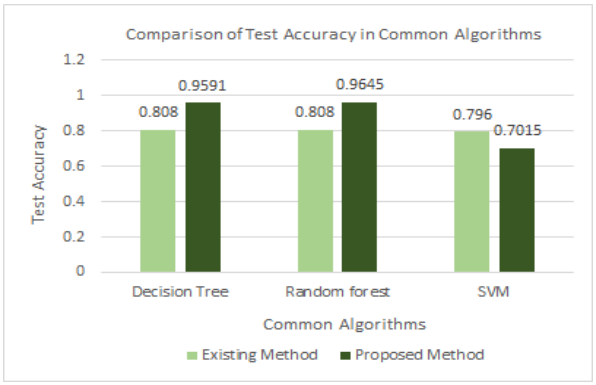

Fig.4. Comparison of Test Accuracy in Common Algorithms

The Table.7 and "Fig.5" shows the performance comparison of Average Train Accuracy and Average Test Accuracy of the existing method and our proposed method.

Table.7. Comparison of Average Train Accuracy and Average Test Accuracy

Data	Existing Method	Proposed Method
Average Train Data Accuracy	0.840	0.928
Average Test Data Accuracy	0.831	0.910

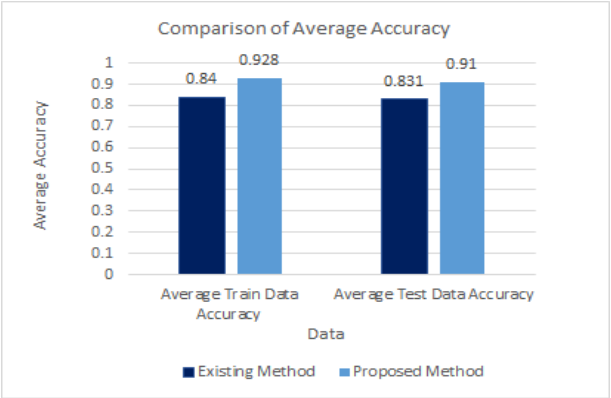

Fig.5. Comparison of Average Accuracy

7. CONCLUSION

In this world of rapid technological development phishing is a great threat to website users. Through our project we have developed the number of algorithm as it is tough to detect them. Through our project wemum accuracy of "0.9863" and test accuracy of "0.9620" is given by Hybrid Ensembler as shown in the proposed table. The accuracy percentage is around 98% while the accuracy percentage of our base paper is around 86%. Hence, our accuracy percentage is increased. about 12% more than the current approach. In conclusion, the best classifier for identifying fraudulent websites is Hybrid Ensembler. It generates a final prediction by integrating the forecasts from these separate models. It employs a voting system in which the final prediction is determined by the majority vote, with each model's prediction serving as a vote.

REFERENCES

[1] S. Sundara Pandiyan, Prabha Selvaraj, Vijay Kumar Burugari, P. Julian Benadit and P. Kanmani, "Phishing Attack Detection using Machine Learning", Measurement Sensors, Vol. 25, pp. 100476-100489, 2022.

[2] Aydin Mustafa and B. Nazife, "Feature Extraction and Classification Phishing Websites Based on URL", *Proceedings of IEEE International Conference on Communications and Network Security*, pp. 769-770, 2015.

[3] Mohammad Robihul Mufid, Arif Basofi, M. Udin Harun Al Rasyid, Indhi Farhandika Rochimansyah and Abdul Rokhim, "Design an MVC Model using Python for Flask Framework Development", *Proceedings of IEEE International Conference on Electronics and Communications*, pp. 1-5, 2019.

[4] Prajakta Patil, Rashmi Rane and Madhuri Bhalekar, "Detecting Spam and Phishing Mails using SVM and Obfuscation URL Detection Algorithm", *Proceedings of International Conference on Inventive Systems and Control*, pp. 1-7, 2017.

[5] Anjaneya Awasthi and Noopur Goel, "Phishing Website Prediction using Base and Ensemble Classifier Techniques with Cross-Validation", *Cybersecurity*, Vol. 5, No. 22, pp. 1-17, 2022.

[6] Mahan Mayuri Vilas, Kakade Prachi Ghansham, Sawant Purva Jaypralash and Pawar Shila, "Detection of Phishing Websites using Machine Learning Approach", *Proceedings of International Conference on Electrical, Electronics, Communication, Computer Technologies and Optimization Techniques*, pp. 1-6, 2019.

[7] Anand Desai, Janvi Jatakia, Rohit Naik and Nataasha Raul, "Malicious Web Content Detection using Machine Learning", *Proceedings of IEEE International Conference on Recent Trends in Electronics Information and Communication Technology*, pp. 34-41, 2017.

[8] Salvi Siddhi Ravindra, Shah Juhi Sanjay, Shaikh Nausheenbanu Ahmed Gulzar and Khodke Pallavi, "Phishing Website Detection Based on URL", *International Journal of Scientific Research in Computer Science Engineering and Information Technology*, Vol. 56, No. 2, pp. 1-13, 2021.

[9] F. Xu, V.S. Sheng and M. Wang, "A Unified Perspective for Disinformation Detection and Truth Discovery in Social Sensing: A Survey", *ACM Computing Surveys*, Vol. 55, No. 1, pp. 1-33, 2021.

[10] Ashit Kumar Dutta, "Detecting Phishing Websites using Machine Learning Techniques", *PLOS ONE*, Vol. 34, No. 2, pp. 1-15, 2021.

[11] Vangala Rama Vyshnavi and Amit Malik, "Efficient Way of Web Development using Python and Flask", *International Journal of Recent Research Aspects*, Vol. 6, No. 2, pp. 16-19, 2019.

[12] Przemysław Kazienko, Edwin Lughofer and Bogdan Trawinski, "Hybrid and Ensemble Methods in Machine Learning", *Journal of Universal Computer Science*, Vol. 19, No. 4, pp. 457-461, 2013.

[13] Elham Nikookar and Ebrahim Naderi, "Hybrid Ensemble Framework for Heart Disease Detection and Prediction", *International Journal of Advanced Computer Science and Applications*, Vol. 9, No. 5, pp. 1-17, 2018.

[14] Nian Li and Bo Zhang, "The Design and Implementation of Responsive Web Page Based on HTML5 and CSS3", *Proceedings of IEEE International Conference on Machine Learning, Big Data and Business Intelligence*, pp. 1-7, 2019.

[15] Suraj Shahu Gaikwad and Pratibha Adkar, "A Review Paper on Bootstrap Framework", *IRE Journals*, Vol. 2, No. 10, pp. 1-3, 2019.