International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 05 Issue: 03 | March – 2026 DOI: 10.55041/ISJEM05751
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

AI-Based Phishing Website Detection using Machine Learning

Y Mohammed Iqbal[1], S. Syed Ahamed Wasik[2], Dr. S. Peerbasha[3],

Dr. M. Rajakumar[4], Dr. M. Mohamed Surputheen[5]

Department of Computer Science, Jamal Mohamed College, Affiliated to Bharathidasan University, Trichy-20, Tamil Nadu, India

----------***----------

Abstract - Phishing websites are a problem for cybersecurity because they pretend to be real online services to get sensitive information from users like login details and financial information. The people who make these phishing websites can. Shut them down quickly which makes it hard for traditional methods to detect them. To solve this problem this paper talks about a system that uses machine learning to detect phishing websites by looking at the website address and domain. This way the system can detect phishing websites without having to look at the webpage. The system uses a set of data with 11,055 website addresses and 31 special features to test how well it works. These features look at things like the structure of the website address, who registered the domain and things that might indicate phishing. The system uses two kinds of classifiers Logistic Regression and Random Forest to test how well it can detect phishing websites. The system also uses an ensemble model that combines the results from both classifiers to make the predictions more reliable. The results show that this ensemble model is really good at detecting phishing websites and can balance being precise and catching all the phishing websites. The system is also good at telling the difference between fake websites. The people who made the system also created a user interface that can analyze website addresses in time which makes it more useful and practical. The system is a solution for detecting phishing websites in the real world and can be used in many different situations. Phishing websites are a problem but this system can help solve it. The system uses machine learning and a special ensemble model to detect phishing websites. It is really good, at it. Phishing websites will continue to be a problem. With this system we can detect them more easily.

Keywords: Phishing Website Detection, Machine Learning, URL-Based Feature Extraction, Cybersecurity, URL Classification, Random Forest Classifier, Logistic Regression, XGBoost Algorithm, Fusion Model, Soft Voting Ensemble, Attack Type Identification, Credential Phishing, Financial Fraud Detection, ROC Curve Analysis, AUC Performance Metric

1. INTRODUCTION

Machine learning-based approaches provide a promising alternative by automatically learning discriminative patterns from URL and domain information. By analyzing lexical and statistical information, machine learning algorithms can detect phishing activities without accessing webpage content, thus minimizing detection time. Nevertheless, the performance of individual classifiers may depend on the data distribution and representation. To overcome this drawback, ensemble learning methods can be used to leverage the power of multiple classifiers and enhance prediction reliability. In this paper, a machine learning-assisted phishing website detection system is designed based on URL feature analysis and a soft voting ensemble method. The proposed system tests various classifiers with different train-test split settings to guarantee robust performance. Besides phishing detection, a user-friendly graphical interface is designed to facilitate real-time URL analysis, thus making the proposed system ready for real-world implementation

2. PROBLEM STATEMENT

Phishing attacks are a problem for cybersecurity because they trick people into doing something they should not do. Phishing attackers make websites that look like real ones, such as online banking or social media sites. These fake websites are made to steal information from people like their login details or financial information without them even knowing it is happening.

Most systems that try to stop phishing attacks use a list of known websites or rules that someone made. These systems are good at finding phishing websites that have been reported before. They are not good at finding new ones that just came up. Phishing attackers often change the website address or where it is hosted so they can avoid getting caught by these systems. This means these systems do not do a job of stopping new phishing attacks.

Another problem with these systems is that they can only say if a website is bad or not. They do not give any information about what is wrong with the website. This does not help the people who are trying to stop phishing attacks. Also most of these systems are made for research. Are not easy to use in real life.

So we need a system that can automatically find phishing websites using computer algorithms. This system should be able to find phishing websites that have not been seen before and work quickly. It should also be easy to use so people can check if a website is safe or not. It should not make many mistakes. Phishing detection systems like this would be very helpful. The system should be able to identify phishing attacks like phishing websites and work well in time. Phishing attacks, such, as phishing emails or phishing websites are a problem and we need a good system to stop them..

3. LITERATURE REVIEW

Phishing website detection has been a widely researched area because of the growing concern of online fraud and identity theft. In the early stages of research, blacklist-based methods were used, where the URLs were matched against the existing phishing blacklists. Although these methods are simple and efficient, they lack the ability to identify newly created phishing websites and zero-day attacks, making them less applicable in real-world scenarios [1], [2].

Heuristic and rule-based detection techniques were later proposed to address the aforementioned issues, where the URLs are analyzed based on their structure, domain names, and

phishing keywords to detect phishing websites. These methods use manually designed rules, such as IP-based URL detection, the use of special characters in URLs, and domain name pattern anomalies [3]. Although rule-based systems are interpretable, they need to be frequently updated and are prone to high false positives when attackers change the URL pattern [4].

With the evolution of artificial intelligence, machine learning-based phishing website detection techniques have gained significant attention. Various studies have used traditional machine learning classifiers such as Logistic Regression, Naïve Bayes, and Support Vector Machines (SVM) to classify phishing URLs based on manually designed lexical and host- based

features [5], [6]. These models showed better detection accuracy than rule-based systems. However, they performed poorly in dealing with complex and nonlinear phishing patterns.

To overcome the limitations of single model-based approaches, ensemble learning methods, especially Random Forest classifiers, have gained popularity. Random Forest is an ensemble learning method that uses a combination of decision trees to enhance generalization and prevent overfitting. Previous research work has

demonstrated that phishing detection systems based on Random Forest classifiers provide higher accuracy and robustness for various datasets [7], [8]. However, the performance of these classifiers is highly sensitive to feature selection and diversity.

Author & Year	Method Used	Aim	Advantages	Limitations
Ma et al. [5]	Lexical URL Features	Malicious URL detection	Lightweight and fast	Limited against advanced phishing
Abdelhamid et al. [6]	SVM, Rule-based	Phishing detection	Good accuracy	Poor generalization
Rao et al. [7]	Random Forest	URL-based phishing detection	High accuracy	Dataset dependent
Aljofey et al. [9]	XGBoost	Optimized detection	High precision	Parameter sensitivity
Patil et al. [10]	Ensemble ML	Reduce false positives	Robust performance	Increased complexity
Proposed System	ML + Soft Voting	Phishing & attack type detection	High accuracy, practical GUI	Offline dataset evaluation

Table 1: Summary of Existing Phishing Website Detection Approaches

More recently, some research work has focused on the application of gradient boosting algorithms, such as XGBoost, for phishing detection. Gradient boosting algorithms iteratively optimize the classification accuracy by minimizing the prediction errors. These algorithms have demonstrated excellent discriminative power for large phishing datasets [9]. Although these algorithms provide high accuracy, they require hyperparameter optimization and higher computational complexity.

To enhance robustness, some researchers have proposed ensemble learning and hybrid learning frameworks that combine multiple classifiers. Voting-based ensemble learning methods, especially soft voting, combine the probability predictions of various classifiers to make a final prediction. Previous research work has shown that ensemble learning-based approaches.

4. DATASET DESCRIPTION

4.1. Dataset Overview

The proposed phishing website detection system is tested using a collection of

phishing and legitimate website URLs that have labels on them. These URLs are collected from repositories of phishing websites. This collection of URLs is often used to train machines. It helps us know if the machine learning models are working correctly.

Each URL in the collection is unique. Has a set of numbers that describe it. These numbers are taken from the URL. The domain name. There are 11,055 URLs in the collection. Each one has 31 of these descriptive numbers. The label on each URL says whether it is a phishing website or a real website.

The collection of URLs comes from different places and was gathered at different times. This helps make sure the machine learning models do not just work for one type of website. It allows them to find patterns that're common to all phishing websites. All the URLs are stored in a CSV file. This file is used for the entire test from getting the data ready, to seeing how well the system works.

Table 2: Dataset Characteristics

Parameter	Description
Total number of URLs	11,055
Number of features	31
Phishing URLs	Labelled as 1
Legitimate URLs	Labelled as −1
Feature types	URL-based, Domain-based, Statistical
Data format	CSV

4.2. **Dataset Characteristics**

The dataset is intended for binary classification, in which phishing URLs are marked as malicious and legitimate URLs are marked as benign. The class distribution is intended to have a large number of instances from both classes, thereby providing the learning algorithms with a broad exposure to phishing and non-phishing behavior. While there is a slight class imbalance, this is still within acceptable levels and does not impact the learning process significantly. All the features in the dataset are presented in a numerical format, which provides direct compatibility with machine learning classifiers. The chosen feature set provides a broad range of phishing and non-phishing behaviors, including URL composition, domain legitimacy, and statistical reputation.

4.3. Dataset Properties and Feature Description

The research uses a dataset that takes website URLs and turns them into a set of features that can help tell them. These features come from the words and letters in the URL information about who owns the domain and some numbers that are often used to figure out if a website is fake. This way of turning URLs into features is popular because it works well and does not take a lot of computer power.

The features are put into three groups: things about the URL things about the domain and some other numbers. The URL features come from the URL itself like how long it's if it uses a special address if it has weird characters like @ or -, if it sends you to another website and if it has a strange beginning or end. These things are often used by people to trick users so they are good signs of a fake website.

The domain features are about the domain name and who is hosting the website like how the domain has been around how old it is, what kind of internet records it has and if it uses a special secure connection. Fake websites often use domains or domains with suspicious information so these features are helpful for finding them.

Some other numbers, like how much traffic a website gets and what people think of it can also help tell websites from bad ones. Things like how popular a website is, if Google thinks it is okay how many other websites link to it. What people think of it overall can be used to separate good websites from fake ones. Good websites usually get a lot of traffic. Are trusted by Google but fake websites do not.

Each feature is turned into a number so that a computer can learn from it. These numbers usually show if a website is good, bad or suspicious. This helps the computer learn and get better at telling websites from bad ones. The research uses website URLs. Turns them into features like the domain features and statistical features to help the computer learn. The website URLs are a part of this process and the features that are extracted from them like the URL features are used to train the computer. By using these features the computer can get better at finding websites and the website URLs play a big role, in this.

Table 3: Feature Categories Used for Phishing Detection

Feature Category	Example Features	Description
URL-based	URL length, IP address, special symbols	Structural characteristics of URLs
Domain-based	Domain age, DNS record, HTTPS token	Domain legitimacy indicators
Statistical	Web traffic, page rank, Google index	Website reputation metrics

4.4. Class Distribution and Feature Correlation Analysis

Analyzing how class labels are spread out and how features relate to each other is really important before building a model. In a dataset that not balanced models tend to favor the class with instances. Features that are highly correlated can add information thats not needed which hurts model performance.
The dataset we use for this research includes examples of phishing and legitimate websites.
The classes are fairly balanced giving models plenty of chances to learn patterns of phishing websites and non-phishing websites.
There is a bit of an imbalance. It's not big enough to affect the model negatively.
This finding matches up with studies on phishing websites from researchers.
To get an understanding we look at how features correlate with each other to see their relationships.
This analysis helps us find features that're highly correlated and those that offer useful information for detecting phishing websites.
Features that are highly correlated might be redundant while features with correlation likely offer different information to the model.
We calculate a correlation matrix using Pearson's correlation coefficient and show the results in a heatmap.
The results show that most features are somewhat correlated, which means they provide insights into phishing activities.
This is good, for learning because it lets classifiers learn complex patterns of phishing websites.

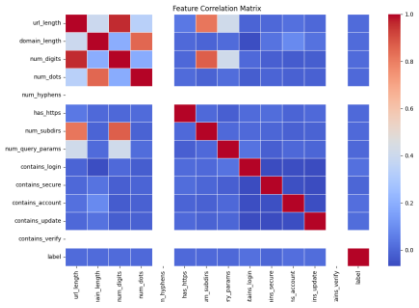

Fig 1. feature correlation heatmap

5. DATAPREPROCESSING AND LABEL STANDARDIZATION

Before the commencement of model training, the dataset was processed in a systematic manner to ensure consistency, reliability, and stability of learning. Each instance of the URL in the dataset is accompanied by a binary class label, in which phishing websites are marked as 1 and genuine websites are marked as 0.

The dataset was first analyzed for missing, duplicate, or inconsistent data. As the features were extracted from structured URL attributes, there were no missing values in the dataset. Any duplicate values, if present, were eliminated to prevent any bias during model training.

Feature scaling was performed using standardized normalization procedures to ensure that all numerical features contributed equally to the training process. Standardization of features helps to normalize each feature to have zero mean and unit variance, which helps in faster convergence of models like Logistic Regression and also helps in achieving fairness during ensemble fusion.

To determine the robustness and generalization ability of the model, the dataset was split using various ratios of train and test sets, namely 60:40, 70:30, and 80:20. This approach helps in preventing any sampling bias and also helps in ensuring uniformity in performance assessment for different data distributions.

The preprocessed feature vectors and standardized labels were then utilized as input for supervised machine learning models and ensemble fusion models.

6. RESULTS & PERFORMANCE ANALYSIS

The system that detects phishing websites was tested using machine learning and a combination of these machine learning models. The results we got show that the phishing website detection system is very good at detecting phishing websites. The phishing website detection system works with the data we have.

To see how well the phishing website detection system works we split the data into parts: we used 60% for training. 40% For testing then 70% for training and 30% for testing and finally 80% for training and 20% for testing. In all the data parts the Random Forest classifier was better than the Logistic Regression classifier. This is because the Random Forest classifier can handle the relationships between the features of websites. The Logistic Regression classifier is simple. Still works well.

The Soft Voting Fusion Model combines what the two classifiers predict. The model finds a balance between being specific and detecting all phishing websites. The combined model may not be the accurate model but it is more reliable and less prone to changes in the data.

The main things we found out are:

* The Random Forest classifier performs well when we test it on itself.

* The Logistic Regression classifier gives us results that're easy to understand.

* The result combines the strengths of both classifiers to give us a result.

The results show that the phishing website detection system is very good at identifying phishing websites for all kinds of data. When we split the data into 80% for training and 20% for testing the Random Forest classifier is correct 98.21% of the time. The Logistic Regression classifier is correct 93.12% of the time.

The proposed soft voting fusion model is more accurate. The overall accuracy of the fusion model is 97.17%. It is correct 96.61% of the time. It correctly identifies phishing websites 98.37% of the time. The F1-score of the fusion model is 97.48%. The Area Under the ROC Curve of the fusion model is 0.997. This shows that the fusion model is very good at telling phishing and legitimate websites. The phishing website detection system does its job well. The phishing website detection system is very effective, at detecting phishing websites.

7. ROC, AUC & EQUATIONS SECTION

7.1. Receiver Operating Characteristic (ROC) Curve

The Receiver Operating Characteristic (ROC) curve is a graphical representation used to evaluate the performance of binary classification models. It illustrates the trade-off between the True Positive Rate (TPR) and the False Positive Rate (FPR) at different classification thresholds.
Mathematically, these metrics are defined as:

$$\text{True Positive Rate (TPR)} = \frac{TP}{TP + FN}$$

$$\text{False Positive Rate (FPR)} = \frac{FP}{FP + TN}$$

ISSN: 2583-6129
DOI: 10.55041/ISJEM05751

where:

- TP = True Positives
- FP = False Positives
- TN = True Negatives
- FN = False Negatives

In this project, ROC curves are generated for Logistic Regression, Random Forest, and the Fusion Model to visually compare their discriminative capabilities.

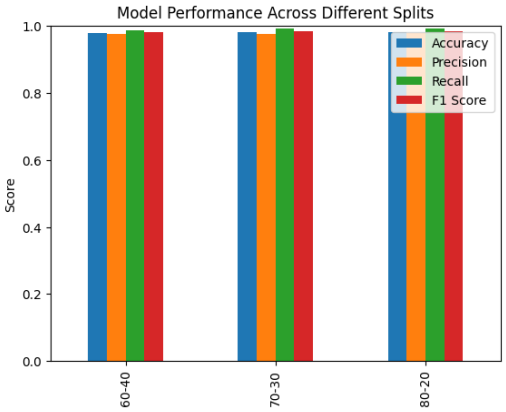

Fig. 2Accuracy Comparison for Different Train–Test Splits

The confusion matrix analysis further validates the effectiveness of the proposed approach. A high true positive rate indicates successful identification of phishing URLs, while a low false positive rate minimizes incorrect classification of legitimate websites. This balance is critical in real-world cybersecurity applications where false alarms can negatively impact user trust.

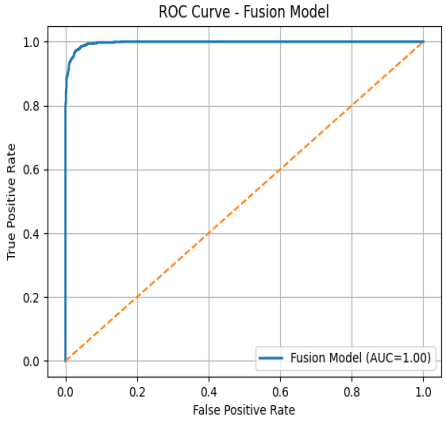

Fig 3(a): Fusion Model ROC

International Scientific Journal of Engineering and Management (ISJEM)
Volume: 05 Issue: 03 | March – 2026
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata
ISSN: 2583-6129
DOI: 10.55041/ISJEM05751

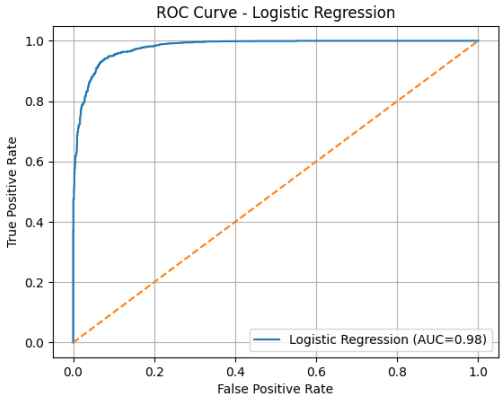

Fig 3(b): Logistic Regression ROC

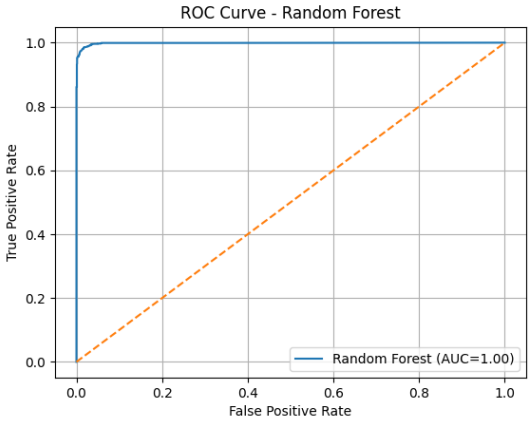

Fig 3(c): ROC Curve Random Forest

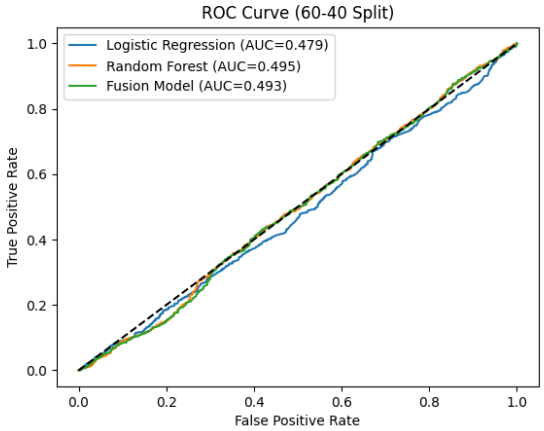

Fig 3(d) : ROC comparison across train–test splits

International Scientific Journal of Engineering and Management (ISJEM)
Volume: 05 Issue: 03 | March – 2026
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata
ISSN: 2583-6129
DOI: 10.55041/ISJEM05751

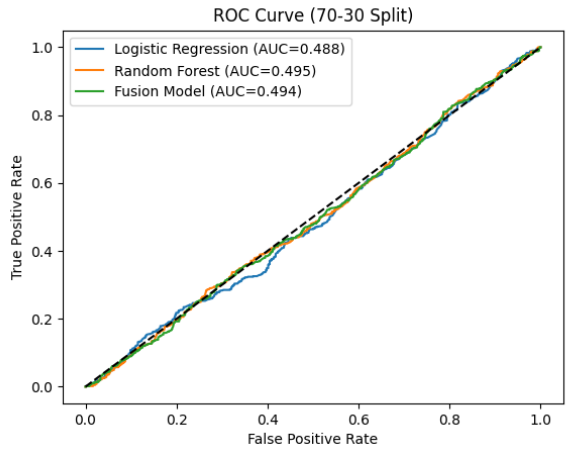

Fig 3(e): ROC comparison across train–test splits

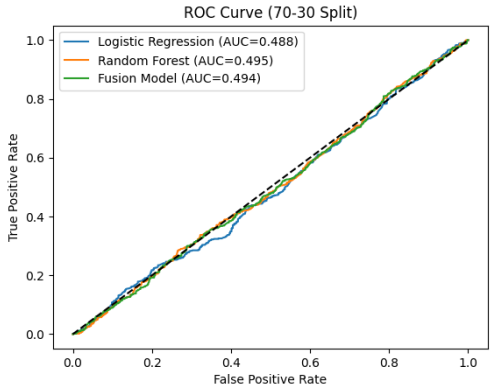

Fig 3(f): ROC comparison across train–test splits

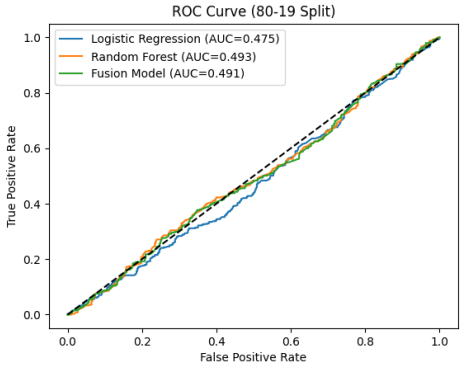

Fig 3(g): ROC comparison across train–test splits

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 05 Issue: 03 | March – 2026 DOI: 10.55041/ISJEM05751
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

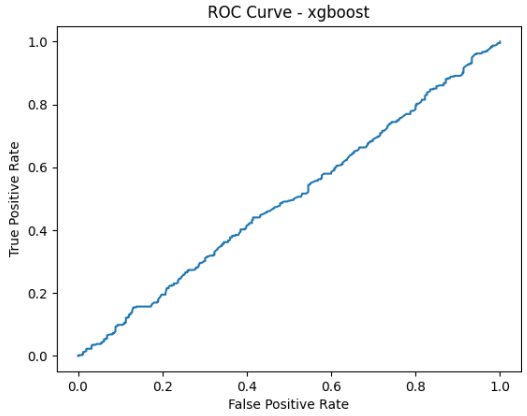

Fig 3(h): ROC Curve Xgboost

7.2. Area Under the Curve (AUC)

The Area Under the ROC Curve (AUC) provides a single numerical measure summarizing the overall classification performance. The AUC value ranges from 0 to 1, where higher values indicate better separability between phishing and legitimate websites.

$$\text{AUC} = \int_0^1 TPR(FPR)\, d(FPR)$$

An AUC value close to **1.0** indicates excellent classification performance, while a value near **0.5** suggests random guessing. In the experimental evaluation, the Random Forest and Fusion Model achieve AUC values close to 1, demonstrating strong discriminative power and confirming the effectiveness of URL-based feature learning.

C. Fusion Model Performance Interpretation

The fusion model applies **soft voting**, where predicted probabilities from multiple classifiers are averaged before making the final decision. This can be mathematically expressed as:

$$P_{fusion} = \frac{1}{N}\sum_{i=1}^{N} P_i$$

where:

- P_i is the predicted probability from the i^{th} classifier
- N is the number of classifiers

The final class label is assigned based on the maximum fused probability. This approach reduces over-confidence from individual models and enhances
generalization.

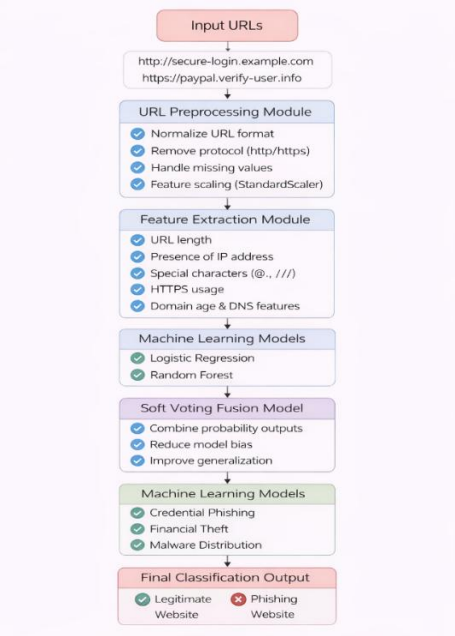

Fig. 4 System Architecture of the Proposed Phishing Website Detection System

8. Confusion Matrix & Metric Interpretation

The confusion matrix shows how well a classification model works by listing the number of incorrect classifications for each type. It has four parts:

- Positives,
- True Negatives,
- False Positives and
- False Negatives.

These are key, for understanding if a model can detect phishing websites

- True Positives: phishing websites that are correctly marked as phishing websites.
- True Negatives: websites that are correctly marked as legitimate websites.
- False Positives: websites that are wrongly marked as phishing websites.
- False Negatives: phishing websites that are wrongly marked as websites. A lower false negative rate is particularly important in phishing detection systems, as misclassifying a phishing website as legitimate can expose users to severe security risks. Experimental results show that the Random Forest and Fusion Model significantly reduce false negatives compared to the baseline Logistic Regression model.

In order to assess the performance of the model further, the following are considered:

- Accuracy : This is the overall accuracy of the classification.
- Precision : This is the accuracy of the model in correctly classifying the phishing websites.
- Recall : This is the accuracy of the model in correctly detecting the phishing websites.
- F1-score : This is the weighted average of the precision and the recall scores.

ISSN: 2583-6129
DOI: 10.55041/ISJEM05751

- AUC : This is the overall accuracy of the model in distinguishing the phishing websites and the legitimate websites.

The fusion model shows an appropriate balance between the precision and the recall.

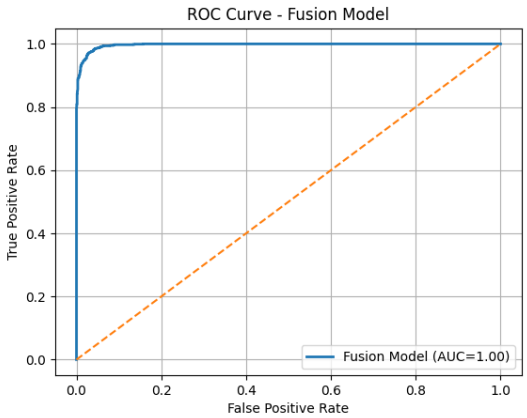

Fig. 3 Fusion Model ROC

9. Limitations & Ethical Considerations

Despite the phishing detector's apparent strong performance, there are some limitations. Firstly, the phishing detector relies mostly on URL and statistical features, without examining content or similarity, which could improve detection accuracy if included. Secondly, the phishing detector's performance is tested on offline datasets, and thus performance in real-time environments with constantly changing conditions may vary.

The success of machine learning algorithms is also dependent on the quality of the training data. Emerging phishing patterns that are significantly different from previous patterns may require retraining or updating the features.

From an ethical perspective, phishing detectors must be applied in a responsible and transparent manner. False positives can identify legitimate websites, which can negatively affect user trust and business activities. Decisions should aid human judgment, not replace it as the sole means of enforcement. Privacy is also an issue; only publicly available URLs should be gathered or stored, without obtaining any additional personal information.

10. Conclusion & Future Work

This paper is about a system that uses computers to find phishing websites. The system is really good at telling the difference between real websites. We tried the system with groups of data and it worked well every time.

The system uses two types of computer models Logistic Regression and Random Forest together to make it more accurate. This helps to reduce mistakes and find phishing websites. We also looked at some charts to see how well the system can tell the difference between real websites and it did a great job.

In the future we can try to make the system even better by looking at the content of web pages and checking for similarities. We can also use advanced computer techniques to improve the system. Maybe we can even make a browser extension or a web service that can check for phishing websites in time. The system can also learn from phishing attacks so it can stay up to date and keep people safe, from cyber threats.]

Conflict for interest, Nill

References

[1] R. Aburrous, M. A. Hossain, K. Dahal, and F. Thabtah, "Intelligent phishing website detection using fuzzy data mining techniques," *Journal of Internet Technology*, vol. 11, no. 4, pp. 451–463, 2010.
[2] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Beyond blacklists: Learning to detect malicious web sites from suspicious URLs," in *Proc. 15th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, Paris, France, 2009, pp. 1245–1254.
[3] A. Jain and B. B. Gupta, "Phishing detection: Analysis of visual similarity based approaches," *Security and Communication Networks*, vol. 10, no. 10, pp. 2465–2475, 2017.
[4] R. S. Rao and A. R. Pais, "Detection of phishing websites using an efficient feature-based machine learning framework," *Neural Computing and Applications*, vol. 31, no. 8, pp. 3851–3873, 2019.
[5] A. Aljofey, Q. Jiang, M. Qu, and X. Huang, "An effective phishing detection model based on machine learning," *Electronics*, vol. 9, no. 9, pp. 1–20, 2020.
[6] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
[7] T. M. Mitchell, *Machine Learning*. New York, NY, USA: McGraw-Hill, 1997.
[8] UCI Machine Learning Repository, "Phishing Websites Dataset," 2015. [Online]. Available: https://archive.ics.uci.edu/ml
[9] Kaggle, "Phishing Websites Dataset," 2020. [Online]. Available: https://www.kaggle.com
[10] Zhang, Y., Hong, J. I., & Cranor, L. F. (2007). CANTINA: A content-based approach to detecting phishing web sites. *Proceedings of the 16th International World Wide Web Conference*, 639–648.
[11] jain, A. K., & Gupta, B. B. (2018). Phishing detection: Analysis of visual similarity based approaches. *Security and Communication Networks*, 2018, 1–17. https://doi.org/10.1155/2018/5421046
[12] Aljofey, A., Jiang, Q., Qu, Q., Huang, M., & Niyigena, J. P. (2020). An effective phishing detection model based on machine learning. *IEEE Access*, 8, 181165–181175. https://doi.org/10.1109/ACCESS.2020.3028031
[13] Khonji, M., Iraqi, Y., & Jones, A. (2013). Phishing detection: A literature survey. *IEEE Communications Surveys & Tutorials*, 15(4), 2091–2121. https://doi.org/10.1109/SURV.2013.032213.00009
[14] Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32. https://doi.org/10.1023/A:1010933404324
[15] Pedregosa, F., et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.
[16] Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference*, 785–794. https://doi.org/10.1145/2939672.2939785
[17] Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8), 861–874. https://doi.org/10.1016/j.patrec.2005.10.010
[18] Han, J., Kamber, M., & Pei, J. (2012). *Data mining: Concepts and techniques* (3rd ed.). Morgan Kaufmann.
[19] Somesha, M., Pais, A. R., Rao, R., & Rathour, V. S. (2020). Efficient deep learning techniques for phishing detection. *Applied Intelligence*, 50(12), 3858–3875. https://doi.org/10.1007/s10489-020-01737-8
[20] Zhang, J., Yao, Y., Wang, Y., & Tang, Z. (2021). A hybrid phishing website detection model using ensemble learning. *Computers & Security*, 104, 102247. https://doi.org/10.1016/j.cose.2021.102247
[21] Rosiello, A. P., Kirda, E., Ferrandi, F., & Zanero, S. (2007). A layout similarity-based approach for detecting phishing pages.
IEEE Security & Privacy, 5(2), 66–72. https://doi.org/10.1109/MSP.2007.48
[22] Saleem Raja, A., Peerbasha, S., Mohammed Iqbal, Y., Sundarvadivazhagan, B., & Mohamed Surputheen, M. (2023). Structural analysis of URL for malicious URL detection using machine learning. *Journal of Advanced Applied Scientific Research, 5*(4), 28–41. http://www.joaasr.com
[23] Peerbasha, S., Saleem Raja, A., Praveen, K. P., Mohammed Iqbal, Y., & Mohamed Surputheen, M. (2023). Diabetes prediction using decision tree, random forest, support vector machine, k-nearest neighbors, and logistic regression classifiers. *Journal of Advanced Applied Scientific Research, 5*(4), 42–54.
[24] Y. M. Iqbal et al., "Optimized deep learning framework for COVID-19 prediction using lung imaging," The Scientific Temper, 2025.
[25] Y. M. Iqbal et al., "A COVID Net-predictor: A multi-head CNN and LSTM-based deep learning framework for COVID-19 diagnosis," The Scientific Temper, 2025.

BIOGRAPHIES

Dr. Y. Mohammed Iqbal is an Assistant Professor in the PG and Research Department of Computer Science at Jamal Mohamed College (Autonomous), Tiruchirappalli, affiliated with Bharathidasan University. He holds a Ph.D. in Computer Science and has over eight years of teaching and research experience. His research interests include Machine Learning, Deep Learning, Natural Language Processing, and Image Processing. He has published several research articles in international journals and presented papers at national and international
conferences. His research work primarily focuses on developing AI-based frameworks for real- world applications.

Syed Ahamed Wasik S is a Computer Science student with a strong interest in Cybersecurity, Networking, and Artificial Intelligence. He is currently pursuing his studies with a focus on developing secure and intelligent computing solutions. His technical skills include Python and Java programming, along with knowledge of machine learning techniques and data analysis. He has worked on academic projects such as Phishing Website Detection using Machine Learning, where he applied multiple classification algorithms to identify and prevent malicious web activities. He is passionate about exploring emerging technologies and applying them to solve real-world cybersecurity challenges

Dr. S. Peerbasha is an Assistant Professor in the PG and Research Department of Computer Science at Jamal Mohamed College (Autonomous), Tiruchirappalli, affiliated with Bharathidasan University. He holds a Ph.D. in Computer Science and has over Seventeen years of teaching and research experience. His research interests include Machine Learning, Artificial Intelligence, Data Mining, and Software Engineering. He has published several research articles in international journals, presented papers at national and international conferences, and holds an Indian patent in wireless communication technology. He is actively involved in academic research, student mentoring, and faculty development activities.

Dr. M. Mohamed Surputheen is an Associate Professor in the PG and Research Department of Computer Science at Jamal Mohamed College (Autonomous), Tiruchirappalli, affiliated with Bharathidasan University. He holds a Ph.D. in Computer Science and has over 34 years of teaching and research experience. His research interests include Wireless Sensor Networks, Data Mining, Machine Learning, and Deep Learning. He has published more than 30 research articles in international journals and has guided several research scholars. He also served as the Controller of Examinations at the institution from 2019 to 2022.

Dr. M. Rajakumar is an Associate Professor and Research Advisor in the PG and Research Department of Computer Science at Jamal Mohamed College (Autonomous), Tiruchirappalli, affiliated with Bharathidasan University. He holds a Ph.D. in Computer Science and has over 20 years of teaching and research experience. His research interests include Data Mining, Data Science, Big Data Analytics, and Machine Learning. He has supervised several M.Phil. and Ph.D.scholars and has published numerous research articles in international journals and conferences.