# An Explainable Ensemble Learning Framework for Phishing website Detection using Random Forest and XGBoost with SHAP Explainer

Kanimalar Rajaram[1], R.K.Pongiannan
[1]*Artificial Intelligence, SRM Institute of Science and Technology Chennai, India*
[2]*Dept. of Computing Technologies, SRM Institute of Science and Technology Chennai, India*

***Abstract*** **- Phishing contributes to the most common cyberattacks, because it manipulates user behavior rather than system weak points. Even with improvements in security mechanisms, many existing solutions still depend on constant methods such as blacklisting, which are not useful for handling newly created threats like Zero Day phishing websites. These websites normally exist only for a short duration, making them difficult to detect using traditional approaches. In this work, an ensemble based phishing detection model is proposed by combining Random Forest and XGBoost. The model uses around lexical and domain based features extracted from URLs to identify whether a website is legitimate or malicious. Instead of depending on a single model, the approach helps improves the consistency of prediction and reduces the chances of overfitting.**

**Many machine learning models function like black boxes, making it difficult to understand how the concept arrive to the prediction. To overcome this issue SHAP(SHapley Additive exPlanations) is used to understand how each feature contributes to the prediction. This improves the transparency of the system and easier to trust, especially in realtime security applications.**

**The model was evaluated on the UCI phishing dataset and achieved the Accuracy of 97. 1% and a recall of 98. 5%, which is performing better than individual models like Decision Tree and SVM. Besides the performance, the paper focus on the transparency gap in the cybersecurity field by adding the SHAP explainability and implementing a diagnostic dashboard, which is live and Streamlit-based. This way, security analysts can understand the results by viewing Reason Codes generated on the spot and making the system a useful tool for professional security audits.**

***Keywords*****— Phishing Detection, Ensemble Learning, Random Forest, XGBoost, Explainable AI (XAI), SHAP.**

## I. INTRODUCTION

The advancements in digital applications and electronic transactions have simplified our daily lives, but at the same time, they have attracted cybercriminals to infiltrate systems and compromise personal and confidential information. Among various cyberattacks, phishing is the most common. Phishing can be described as an act of deceiving individuals to disclose specific information like user ID, passwords, credit card numbers and any other sensitive data. Phishing is unique among other forms of attack in that most attacks exploit system vulnerabilities, but these exploits humans.

Although many security technologies have evolved and been optimized over the years, many existing systems still use outdated techniques such as blacklisting and signature matching. Such techniques are directly useful for recognizing known depository sites, but they are not effective against recently launched phishing sites. Such sites usually exist for a short time and are not registered on any blacklist. Hence, reactive techniques leave a gap in protection.

Systems for phishing detection have become more effective because machine learning was introduced to the field. Experts claim that machine learning permits the system to gain knowledge from information instead of following only the original instructions. It has been observed that various successful algorithms are difficult to understand clearly. Within a practical digital safety setting, the simple labelling of a web location as harmful is not enough. The reasons for such a categorisation must be understood by security specialists because the machine learning model loses value without clear explanations.

Enhancing the precision and the clarity of phishing detection is the primary goal of this academic paper. The combined approach of Random Forest & XGBoost is suggested and this technique is trained using various characteristics taken from URLs. When multiple characteristics are joined, the machine

learning model produces more stable predictions. Many people believe that SHAP clarifies the way every characteristic affects the final result. Through SHAP, the machine learning model is made more transparent for the users.

These academic efforts focus on constructing a detection system with strong capabilities and providing detailed justifications. A tool like this is often proposed to be more useful in practical situations. In these real situations, the understanding of findings is prioritized as well as the exactness. The logic is visible and the detection system becomes a more valuable asset for the professional environment.

## II. RELATED WORK

Phishing detection methods have evolved over time from blacklists for identifying malicious websites to advanced machine learning models. Most of these methods revolved around blacklists, lists of URLs identified as used in phishing attacks. But these blacklists work as a good method for blocking websites, Mohammad and Thabtah [1] argue that this method is reactive in dealing with the issue where there is a time delay between site creation and user reporting. Because of these reasons, the authors propose using supervised learning algorithms for finding patterns in URLs themselves.

From the referenced paper [2], where this research studied SVMs and Decision Trees, accuracies of 89% and 92%, respectively, were achieved. These classifiers work much better than the basic filter but still have problems with complex page data, or even with simple modifications designed to fool the filter. The issue of overfitting holds back these new models and failed with accuracy.

To improve detection reliability, there are several techniques that have been involved, including ensemble methods, which have been proposed, but Phishing detection has been improved by methods like Random Forests and Gradient Boosting, which have reduced false positives and enabled the detection of even the most sophisticated forms of phishing. The main issue with these highly accurate solutions is that they are hardly transparent. As per the referenced paper, XGBoost models are as black boxes model [3], but in reality, many of these methods are highly accurate, security professionals are still hesitate to trust the result because they do not know which feature such as URL length or domain age, led to the alert.

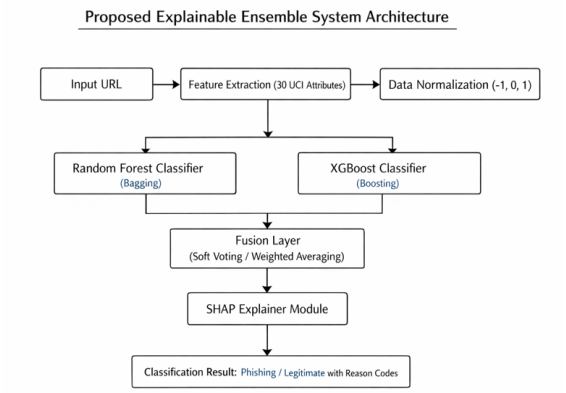
As the referenced paper [4] states, Brand Spoofing might cause the loss of public trust and, very importantly, deeply affect user trust as it is not only a technical problem. It has led us to a point where we demand that the system be both precise and transparent.

## III. PROPOSED METHODOLOGY

Our framework is based on an explicit pipeline that we designed to evaluate real-time phishing URL detection. Rather than simply applying single classifiers on data, we proposed a system that integrates a data driven environment, an ensemble of classifiers and the explainability layer to have a result understandable. The system is proactive because it uses features derived from the URL text as soon as it appears. We assemble an ensemble of classifiers to reduce false negatives, which is important for protecting sites such as online banking and e-commerce.

### *A. Ensemble Learning Architecture*

To obtain better results than a single model, we used a heterogeneous ensemble. Single models, such as Decision Trees, tend to have high variance, whereas linear models tend to be more biased. Combining different algorithms can help "balance" the bias-variance trade-off.




Random Forest is the ensembling classifier, and the individual classifiers are decision trees and multiple one of many decision trees are generated, and a voting system is used to combine their outputs. This is a way to reduce variance and improve the classifier's generalization. Also, we used XGBoost, a gradient boosting implementation with great performance on identifying complex links between features. By using

both, the system becomes more robust against different types of phishing URLs.

*B. Data Acquisition and Feature Analysis*

In our case, for the experiment we used the UCI phishing dataset which has 11,055 records. It is a very famous dataset in the field, so it would be easy to compare our results with those of other researchers. We extracted 30 relevant features from URLs. It is divided into a very small number of main categories. Initially, features of the address bar were examined, such as the use of an IP address instead of a domain name or if shortening of a URL had been conducted.

TABLE I. URL FEATURE CATEGORIES

| Category | Feature Count | Sample Attributes | Why it Matters |
|---|---|---|---|
| Address Bar | 12 | IP Address, URL Length, TinyURL | Attackers often hide the real website using IP addresses and links. |
| Abnormality | 6 | Redirects, Right-click disabled | These behaviors are used to confuse users or stop them from checking. |
| HTML/JS | 5 | Anchor URL, Request URL | Suspicious external links may redirect users to unsafe or unrelated sites. |
| Domain-Based | 7 | Domain Age, DNS Record | Many phishing sites are newly created and don't have stable domain history. |

These are typical techniques hackers use to mask their identity. Features of the anomaly were considered, such as variations in the request URL or anchor tags. Also included were features based on the domain itself, such as the creation date of the domain or its DNS records. Many phishing websites are less than a few days old. The domain's age was also a useful indicator of the malicious site.

*C. Explainability via SHAP (SHapley Additive exPlanations) and Feature Importance*

In the cybersecurity adoption, the main issue is that high precision models are tend to operate as black boxes. Though this ensemble technique provides high accuracy, the internal decision making is still not visible.

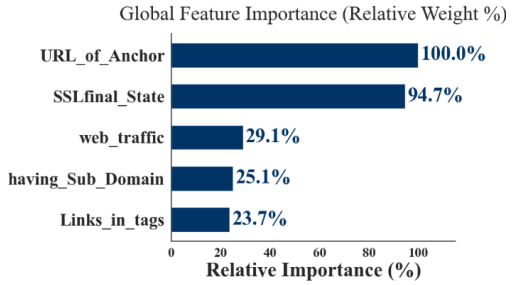



Fig. 2. SHAP Feature Importance (Top 5).

By integrating the SHAP game theoretic approach, which helps to provide transparency to every prediction, indicating how much each feature contributes to a prediction. SHAP helps to approximate the effect of each feature using the attribution logic below for the model to make a decision.

The image above explains the Global SHAP analysis and the different URL attributes that help to identify phishing.

SHAP is based on coalitional game theory, where every feature is treated as a player in a game. The below attribution logic calculates the contribution of each feature to the final prediction score with the reason code why the site was flagged for what reasons.

.

$$\phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(n - |S| - 1)!}{n!} [v(S \cup \{i\}) - v(S)]$$

Instead of assigning the label by saying the website is a malicious or Legitimate label but this model provides the basis and reason for this decision, such as short domain age or IP address. This transparency is important for the analysts who needs to take care of the security alerts accurately why the sites was flagged for the instance.

## IV. EXPERIMENTAL SETUP

*A. Experimental Setup*

The model was built using Python 3.10 version and tested. The libraries used in the implementation were standard libraries such as scikit-learn, which was used for Random Forest, whereas XGBoost was used for the gradient boosting library. The SHAP library was used for analyzing the impact of various features.

- Prototype and Analysis: Jupyter Notebook environment used for model training and data analysis.
- Deployment and Visualization: Streamlit framework for web application hosting and interactive user interface rendering.
- Machine learning Libraries: Installing scikit-learn for dataset splitting, XGboost for Gradient boosting, Shap for game theory, Pandas and Numpy for Matrix operations.

*B. Data Description and Pre processing:*

In the tests, we used the UCI phishing set that has 11,055 samples. Before training, the data set was validated for missing values and data consistency. Our set was already well-structured and has 30 distinct features which is extracted from web traffic but preprocessing was required for algorithmic requirements. The UCI dataset has the target variable as -1 for Phishing and 1 for Legitimate. But the bagging component where this Random Forest processes these labels and the boosting component which is XGBoost enforces the binary constraints and those expecting a domain of as 0,1. Hence ensure compatibility within the Fusion Layer the normalization mapping function applied prior to the training split as follows.

- Legitimate Sites: Maintained as 1.
- Phishing Sites: Re-mapped from -1 to 0.

After the normalisation, the dataset was partitioned using an 80/20 train test split, which is 8,844 instances for training and 2,211 instances for testing.

*C. Ensemble Model Configuration:*

To balance the variance and bias reduction, the heterogeneous ensemble method was implemented with the following hyperparameter configurations:

- Bagging Component is Random Forest Classifier where n_estimators = 100, which is configured to generate 100 independent decision trees to construct the forest for Optimisation and stability.
- Boosting Component is for XGBoost Classifier, which is configured to utilize the logarithmic loss for evaluating the performance and penalising false classifications.
- The Fusion Layer is Soft Voting where Instead of binary voting, this framework works on the final prediction logic

## V. RESULTS AND DISCUSSION

*A. Performance Analysis via Confusion Matrix Metrics*

To evaluate the model, we considered common metrics such as accuracy, precision, recall, and F1-score. In the case In phishing detection, recall is especially important because missing a malicious website can have serious consequences.

So, more attention was given to reducing false negatives and false positives. Blocking a legitimate site is called as a false positive, which disrupts the business operations and user experience. False negatives allow phishing sites to go undetected, which causes severe security breaches.

Performance analysis of Random forest:

The Random Forest works as strong baseline for this study because it tackles the problems individual classifiers have, like high variance and overfitting, by creating many decision trees and integrating their results through the voting method, resulting in better stability and capability than a single tree.

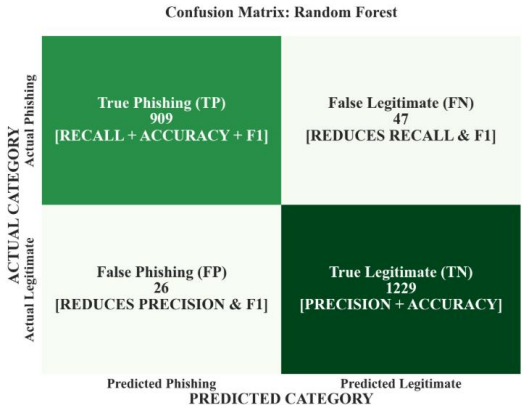



Fig. 3. Confusion Matrix- Random Forest

The Random Forest achieved an accuracy of 96.7%, outperforming the accuracy of the traditional benchmarks for the Decision Trees (95.7%) and SVM (94.7%).

- Recall: The Random Forest achieved 97.9% Recall. This means that all these actual phishing sites were identified in the TP(True Phishing)

box and a small number fell into the FN (False Legitimate) box.

- Precision: Random Forest model has a precision of 96.3%. This precision indicates that when the model flags a site as Phishing, it is correct 96.3% of the time, with the FP (False Positive) box indicate the small error rate.
- Accuracy: The overall percentage of correct predictions (TP + TN) and the value for the Random Forest is 96.70%.
- F1 Score: This is the balance of the whole Phishing detection which is TP, FN, FP. The F1 Score for Random forest is 97.12% though it shows that very strong baseline, but still has room for improvement through the ensemble.

Performance analysis of XGBoost:

XGBoost is unlike Random Forest, which builds trees independently and focuses on boosting, where each new tree is designed to correct the errors made by the previous tree. In this research framework, XGBoost is used to identify complex and non linear patterns within the 30 URL features extracted from the UCI dataset.

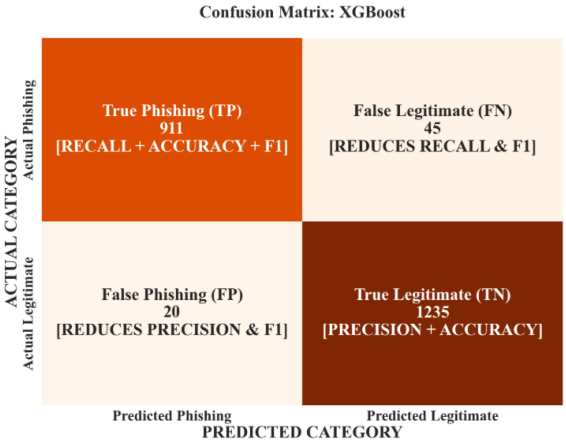



Fig. 4. Confusion Matrix- XGBoost

- Recall (98.4%): XGBoost is designed to reduce bias, because it is better at catching phishing attempts than the other models, which have high recall value, which is its primary strength.
- Precision (96.4%) is slightly better than Random Forest, but it still produces a small number of false alarms.
- Accuracy (97.0%) is the combined sum of the diagonal boxes (TP + TN), showing that XGBoost is more reliable than the Random Forest.
- F1 Score (97.44%) is very strong because XGBoost has such high recall and highly effective model for security audits.
- Despite its high accuracy, XGBoost is a black box model because the internal decision making is still not visible. By integrating the SHAP approach, which helps to provide transparency to every prediction by the reason codes of the model decision.

Performance Analysis of Proposed Ensemble Model:

The final ensemble model achieved the highest performance by combining bagging and boosting methods: Random Forest and XGBoost with SHAP.

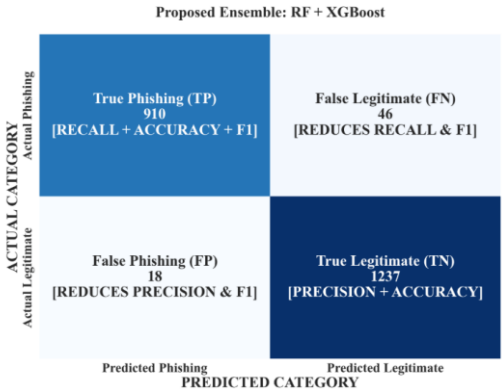



Fig. 5. Confusion Matrix- Proposed Ensemble Model(RF+ XGBoost with SHAP Explainer)

- The testing performed to evaluate the reliability of the partition of 2,211 instances, which is the testing sample. True Phishing Detections are 910 cases were identified out of 956 malicious sites, achieving a 98.5% Recall rate. This high sensitivity is required for preventing security breaches.
- True Legitimate Verifications are 1,237 cases in which the model allowed 1,237 safe sites, ensuring that legitimate business traffic is not disrupted.
- False Positive Mitigation is 18 cases, which is 0.8% of legitimate sites that were incorrectly flagged. This low error rate is required for maintaining user trust in a real world security applications
- False Negative Management is 46 cases where the model failed to find 46 phishing attempts which cause minimal user disruption.

Final Accuracy is 97.11% by implementing both models where the framework resulted in a high peak predictive accuracy on the UCI dataset. The F1-score

of 97.48% further confirms that the model is highly consistent in handling the safe and malicious categories.

*B. Model Comparison*

The accuracy of the proposed ensemble algorithm was compared to individual algorithms such as Decision Tree, Support Vector Model(SVM), Random Forest(RF) and XGBoost. The ensemble got an accuracy of 97.1%, which is higher than other models. Decision Tree and SVM had lower accuracy, while Random Forest & XGBoost had higher accuracy, with a slight difference between the two. It provides evidence that the ensemble can improve the stability and predictive accuracy.

TABLE II. COMPARATIVE PERFORMANCE ANALYSIS

| Algorithm | Accuracy (%) | Precision (%) | Recall (%) | F1-Score (%) |
|---|---|---|---|---|
| Decision Tree | 95.7 | 96.2 | 96.2 | 96.2 |
| SVM | 94.7 | 94.2 | 96.5 | 95.4 |
| Random Forest | 96.7 | 96.3 | 97.9 | 97.1 |
| XGBoost | 97 | 96.4 | 98.4 | 97.4 |
| Proposed Ensemble (RF+XGBoost) | 97.1 | 96.4 | 98.5 | 97.4 |

The image below which explained the superiority of Proposed Ensemble (RF + XGBoost) with other individual models.

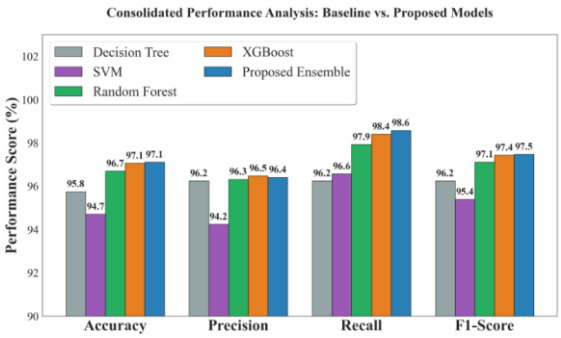



Fig. 6. Comparative Performance analysis of Models.

The analysis showed that the two most important features were URL anchor and domain age. It makes sense because most phishing sites are created for a very short period, and do not have a long history, therefore a new domain is a very serious warning signal.

We also saw that many sites redirect through strange anchor links to fake login pages. Another thing we notice is that the models are not dependent on a single feature to make a call. There are many instances where a long URL alone is not sufficient to identify a site as phishing. Because when the same URL lacks an HTTPS token and has multiple redirects, the model is more certain. This is where the ensemble outperformed decision trees.

Using SHAP, we generated reason codes for each case. It's not only a decision or a simple yes or no when a security analyst can see precisely which features made the model lean toward a phishing outcome. Such transparency is much more usable for a real security audit framework. It is a valuable process, because a model achieving high accuracy is not all that useful if it gives you no insight into why a given website was classified as malicious. We examined feature importance to identify which sections of a URL placed the greatest stress on the classifier.

*C. System Implementation and Deployment:*

Instead of only running a local script in Python, the system was created with a real-time monitoring display. This enabled the tool not only security experts to use it, but also non-programmers who wish to develop their skills.

Development setup: The Graphical user interface was created with Streamlit. Streamlit is a Python package which is commonly used by developers when building machine learning web applications. Our main reason for choosing Streamlit was its feature to embed sophisticated visualisation components like Matplotlib and SHAP in a web browser.

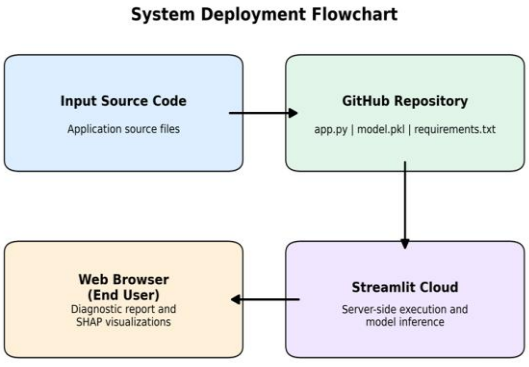



Fig.7 System Deployment Flowchat(GitHub-Streamlit)

Version control and hosting: We have all our code, the pretrained ensemble model and feature extraction

techniques in a GitHub repository. Apart from version and change tracking, this also allows other developers to collaborate on the detection logic.

Live Deployment: To make it available to the public, the project repository has been connected to Streamlit Cloud which works as a hosting platform for the application to be executed live. Because of this, anyone can enter a URL and get a security report right away without having to locally install the Python environment.

User Interface Design: The Control Panel is located on the left side of the streamlit dashboard, and makes it easy to enter the URL, and the main display area shows the Result Output. Here, the model outputs a binary class, saying as Phishing or Legitimate and the confidence percentage, and then the SHAP based reason codes are shown to justify the prediction

*F.Real-time System Validation and Output*

The Streamlit web real time dashboard has been created based on the Streamlit library which is not only a show of the setup but it also provide the doorway to real-world testing for this prediction.

Validation Instance for Analysis of a Phishing Threat and Malicious URL Detection, where the Input URL is www. update. com. The system raised a red flag labeled as PHISHING DETECTED with a full 100% threat confidence score and the embedded SHAP explainer module unlocked the decision making contents which were two structural red flags as the URL_of_Anchor and the SSLfinal_State.

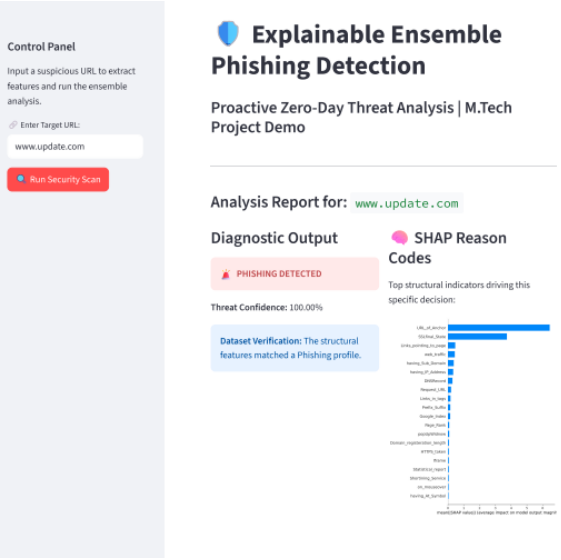



Fig.8 Validation of Phishing threat input(www.update.com) showing 100% confidence and SHAP reason codes

The site was fingerprinted as a phisher because the links on its page targeted external domains and it did not have a secure certificate and the Reason Codes, the structure arms the security expert with the proof necessary for flagging the threat and deciding the way forward as Phishing.

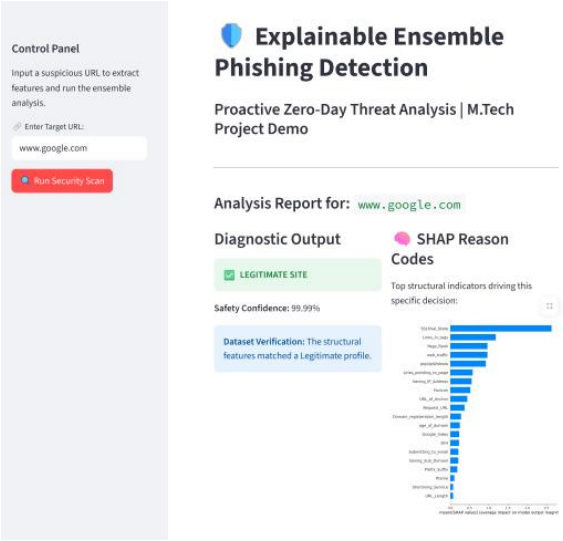



Fig.9 Validation of legitimate domain (www.google.com) showing 99.99% safety confidence

Validation Instance for analysing a Legitimate Domain with Positive Feature Recognition where the setup with a safe domain and the Input URL is www. google. com. The model did not misclassify it as a LEGITIMATE SITE with a safety confidence score of 99. 99%. On this occasion, the SHAP reason codes were changed to highlight positive safety features. The model concluded that the presence of a valid and the high level SSL certificate and also having a robust global PageRank were the primary indicators of a site authenticity.

This Validation for the ensemble method is not only searching for the negative patterns but also recognizing the structural differences between the trusted entities and malicious security threats. This validation method demonstrates that the system can be deployed in professional settings where trust matters as much as accuracy.

## V. FUTURE WORK

The current framework works well, but there are a few directions we want to explore in the future. We want to ensure our system cannot be fooled by attackers who may attempt to manipulate URL features to avoid

being explained by our system, and also intend to explore the use of visual feature integration.

As of now, the system focuses only on URL text, but we can improve its ability to detect more types of brand spoofing by incorporating screenshot analysis or webpage content. A future goal is building a real time browser plugin. This way, users can quickly be briefed by Reason Codes on the safety of suspicious links.

## VI. CONCLUSION

In this project, we learned how ensemble learning enhances the ability of classifying phishing websites. By ensembling various models, for example, Random Forest and XGBoost, to extract various characteristics from the input features of the dataset. The overall prediction accuracy of the system improved to 97.1% on the UCI phishing dataset. The ensemble model achieved more consistent results with less misclassifications

Furthermore, the interpretability of the models was another essential source of this study. In general, machine learning applications, especially in cybersecurity, merely provide predictions instead of explanations. Therefore, SHAP was called upon to identify the contribution of each feature. The results showed that features such as domain age and URL related features contributed significantly to phishing detection. Such explanations would be more helpful to analysts in interpreting the security implications.

However, during this work, we saw that for real world threat detection, performance alone is sometimes not enough, because a model that works perfectly in theory is of limited use if it cannot be trusted or explained. The system proposes to fill this gap by balancing the performance and the interpretability.

This paper demonstrates that a combination of an ensemble method and an explainability approach can give a more robust and the automated phishing detection system. But still, there is room for improvement and further work in this area of attack detection, which helps system development towards an image that is dependable and more trusted to operate in real world applications.

## REFERENCES

[1] R. Mohammad and L. Thabtah, "Phishing Websites Features," UCI Machine Learning Repository, 2015. [Online]. Available: https://archive.ics.uci.edu/dataset/327/phishing+websites.

[2] T. Nguyen, P. Chen, and K. Le, "ML Techniques in Phishing Detection: A Comparative Study," *Journal of Cybersecurity Research*, vol. 9, no. 2, pp. 45-58, 2023.

[3] S. Khalid et al., "Shaping the Future of Cybersecurity: The Convergence of AI, Quantum Computing, and Ethical Frameworks," *Research Square*, Preprint, 2025.

[4] R. M. Mohammad, F. Thabtah, and L. McCluskey, "Tutorial and Predictor for Phishing Websites using Neural Networks," *UCI Machine Learning Repository*, 2015.

[5] A. B. Carter, W. J. Perry, and J. D. Steinbruner, *A New Concept of Cooperative Security*. Washington, D.C.: Brookings Institution Press, 2010.

[6] S. S. Shafin, "An explainable feature selection framework for web phishing detection with machine learning," *Data Science and Management*, vol. 8, pp. 127–136, 2025.

[7] K. Thakur et al., "A systematic review on deep-learning-based phishing email detection," *Electronics*, vol. 12, no. 21, p. 4545, 2023.

[8] M. Chauhan and S. Shiaeles, "An analysis of cloud security frameworks, problems and proposed solutions," *Network*, vol. 3, no. 3, pp. 422-450, 2023.

[9] K. Thakur et al., "Cloud computing and its security issues," *Application and Theory of Computer Technology*, vol. 2, no. 1, pp. 1-10, 2017.

[10] T. Kehkashan et al., "Explainable phishing website detection for secure and sustainable cyber infrastructure," *Scientific Reports*, vol. 15, no. 41751, 2025. doi: 10.1038/s41598-025-27984-w

[11] K. Thakur, J. Shan, and A. S. K. Pathan, "Innovations of phishing defense: The mechanism, measurement and defense strategies," *Int. J. Commun. Netw. Inf. Secur.*, vol. 10, no. 1, pp. 19-27, 2018.

[12] M. H. Alkawaz et al., "Detecting phishing website using machine learning," in *Proc. 16th IEEE Int. Colloquium on Signal Processing & Its Applications (CSPA)*, 2020, pp. 111–114.

[13] R. Alazaidah et al., "Website phishing detection using machine learning techniques," *J. Stat. Appl. Probab.*, vol. 13, pp. 119–129, 2024.

[14] T. O. Ojewumi et al., "Performance evaluation of machine learning tools for detection of phishing attacks on web pages," *Scientific African*, vol. 16, p. e01165, 2022.

[15] N. Q. Do et al., "Deep learning for phishing detection: Taxonomy, current challenges and future directions," *IEEE Access*, vol. 10, pp. 36429–36463, 2022.

[16] [A. Ramana, K. L. Rao, and R. S. Rao, "Stop-phish: An intelligent phishing detection method using feature selection ensemble," *Soc. Netw. Anal. Min.*, vol. 11, no. 110, 2021.

[17] [Z. Alkhalil, C. Hewage, L. Nawaf, and I. Khan, "Phishing attacks: A recent comprehensive study and a new anatomy," *Front. Comput. Sci.*, vol. 3, p. 563060, 2021.

[18] A. Aljofey, Q. Jiang, Q. Qu, M. Huang, and J.-P. Niyigena, "An effective phishing detection model based on character level convolutional neural network from URL," *Electronics*, vol. 9, p. 1514, 2020.

[19] M. Bahaghighat, M. Ghasemi, and F. Ozen, "A high-accuracy phishing website detection method based on machine learning," *J. Inf. Secur. Appl.*, vol. 77, p. 103553, 2023.

[20] R. Mahajan and I. Siddavatam, "Phishing website detection using machine learning algorithms," *Int. J. Comput. Appl.*, vol. 181, pp. 45–47, 2018.

[21] A. Altaher, "Phishing websites classification using hybrid SVM and KNN approach," *Int. J. Adv. Comput. Sci. Appl.*, vol. 8, 2017.

[22] M. A. A. Siddiq, M. Arifuzzaman, and M. Islam, "Phishing website detection using deep learning," in *Proc. 2nd Int. Conf. on Computing Advancements*, 2022, pp. 83–88.

[23] W. Ali and A. A. Ahmed, "Hybrid intelligent phishing website prediction using deep neural networks with genetic algorithm-based feature selection and weighting," *IET Inf. Secur.*, vol. 13, pp. 659–669, 2019.

[24] Y. Wei and Y. Sekiya, "Feature selection approach for phishing detection based on machine learning," in *Int. Conf. on Applied CyberSecurity*, Springer, 2021, pp. 61–70.

[25] R. Alenezi and S. A. Ludwig, "Explainability of cybersecurity threats data using SHAP," in *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2021, pp. 1–10.

[26] S. K. Bajaj and S. Hansen, "Social Effects of Phishing on E-Commerce," in *IADIS Int. Conf. e-Commerce*, 2008.

[27] E. Humphreys, "Information security management standards: Compliance, governance and risk management," *Information Security Technical Report*, vol. 13, no. 4, pp. 247-255, 2008.

[28] S. Banik and P. R. Kothamali, "Developing an End-to-End QA Strategy for Secure Software: Insights from SQA Management," *Int. J. Mach. Learn. Res. Cybersecur. AI*, vol. 10, no. 1, pp. 125-155, 2019.

[29] J. M. Stewart, E. Tittel, and M. Chapple, *CISSP: Certified Information Systems Security Professional Study Guide*. John Wiley & Sons, 2011.

[30] D. Anand and V. Khemchandani, "Data security and privacy functions in fog computing for healthcare 4.0," in *Fog Data Analytics for IoT Applications*, Springer, 2020, pp. 387-420.