Global **A**dvances in **M**ultidisciplinary **A**pplications in **N**ext-Gen **A**nd **M**odern Technologies (**GAMANAM)**
Volume 1, Issue 1, January 2025, Pp: 61-69

Survey Paper

Phishing Website Detection Using Machine Learning Techniques

[1]CH Revathi, [1*]N Padmaja
[1]*Department of Computer Science and Engineering, School of Engineering and Technology, Sri Padmavati Mahila Visvavidyalayam, Tirupati*

**Corresponding Author(s): gowri.padma@gmail.com*

Received: 18/10/2024, Revised: 17/11/2024, Accepted:20/12/2024 Published:12/01/2025

Abstract: Phishing websites are a stern threat to online security, as they attempt to giveaway delicate data from unsuspecting workers. To fight this threat, scholars have developed various techniques. These algorithms can be skilled on large datasets of phishing and genuine websites to cram patterns and characteristics that distinguish between the two. These algorithms can then be used to recognize and tablet phishing websites before users can be victimized. On approach to involves feature removal, where various features of a website such as URL structure, domain age, and content are analyzed to identify phishing websites. Another approach involves to automatically cutting features and learn compound patterns in website data. Machine learning- based phishing website detection techniques have shown promising results, achieving high accuracy rates and outperforming traditional rule-based methods. With further research and development, these techniques have the possible to become an significant tool in the match beside online phishing attacks.

Keywords: Phishing website detection, Machine learning, Feature extraction, Online security, Anti-phishing techniques

1. Introduction

Phishing website detection is the process of identifying and flagging websites that attempt to impersonate legitimate websites with the goal of stealing complex data for instance login permits, credit card records, and own documentation information. Phishing attacks have become increasingly sophisticated over the years, and attackers often use tactics such as social engineering and fake login screens to pretend operators into generous up their sensitive data. Phishing websites may also use URL spoofing to make it appear that the user is on a legitimate. Phishing attacks are a common type of cybercrime users hooked on revealing complex data. One of the most effective ways to combat phishing attacks is through the detection and blocking of phishing websites.

Phishing website detection to identify and flag websites that are designed to deceive users. One joint technique used into attackers is to form websites that carefully the appearance of legitimate sites, such as banks or e-commerce sites. These phishing sites are often hosted on compromised servers or using domain names that are similar to the real sites one approach to detecting phishing websites is to use machine learning that can examine website content, metadata, other features to identify potential phishing sites. These processes can be skilled on data sets of known phishing websites classify common designs and characteristics. Some machine learning models may also incorporate real-time data feeds to identify and flag new phishing sites as they are created. Another approach to phishing website detection is to use reputation-based systems that maintain lists of known malicious website.

The detection and blocking of phishing websites is an essential component of any effective cybersecurity strategy. By using these algorithms, reputation-based systems, and behavioral analysis methods, organizations can protect their users and prevent complex data from dropping into the pointers of attackers.

1.1 Machine learning methods

Machine learning algorithms empower machines to study after information, make predictions, then take choices autonomously, lacking requiring unambiguous software design. In supervised learning, techniques like linear regression and decision trees utilize labelled data to forecast continuous or categorical outcomes. Conversely, unsupervised education systems, counting k-means clustering and major module study, uncover hidden patterns and relationships within unlabelled data, enabling machines to determine treasured visions then information.

1.1.1 Random Forest Classifier

The random forest algorithm works by deciding outcomes through predictions made by decision trees. So, how does it do this it takes the average or mean of what different trees say. The more trees you have, the better the prediction gets.

This method really helps to overcome problems seen in

regular decision tree algorithms. It reduces a tricky issue called overfitting, which happens when models learn too much from training data and perform poorly on new data. Also, it boosts accuracy!

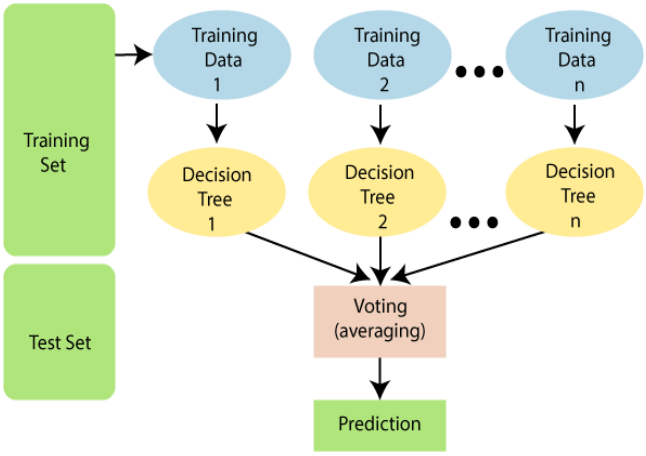

Figure 1: Random Forest Algorithm

1.1.2 Decision tree

Decision trees are really important for how a random forest algorithm works. They act like a sort of guide that looks like a tree. If we take a good look at decision trees, we can get a better idea of how these random forests do their thing.

The decision tree algorithm takes data from a training set splits it into twigs. Then, those branches keep breaking down into even more branches. This keeps going until we reach a leaf node. Once you're at a leaf node, there's no more splitting to do.

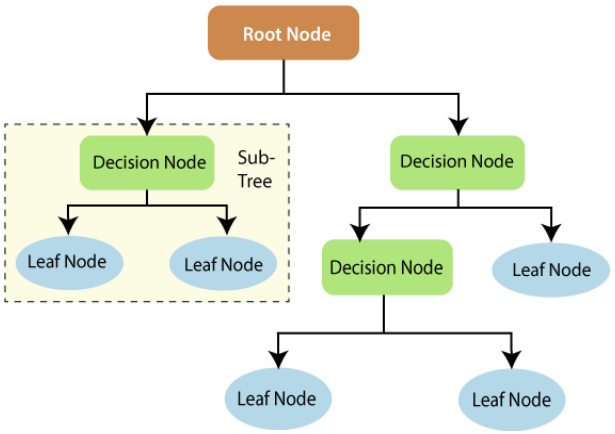

Figure 2: Decision Tree Classifier

1.1.3 Support Vector Machine

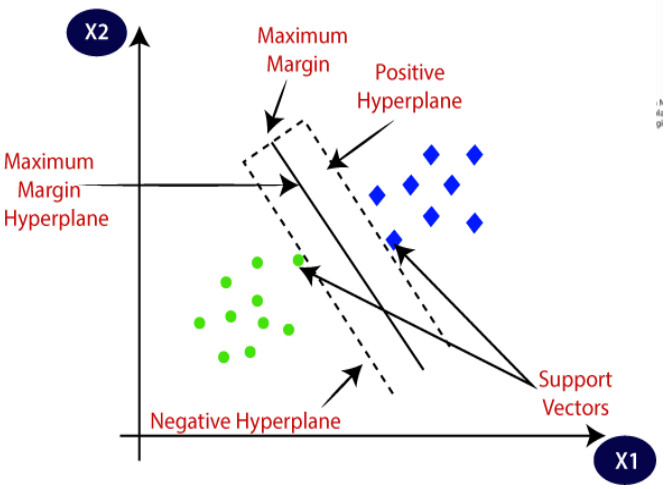

Figure 3 Support Vector Machine

Now, what about those nodes they represent important features that help us predict what will happen. Decision nodes connect to the leaves to show us where to go next. If you check out the diagram below, you'll see clearly shown in a decision tree. Support Vec tor Machine, or SVM.

Is super popular! It's one of the most widely used Supervised learning algorithms People use it mainly for Classification but also for Regression tasks. Basically, its main job is to find the best or boundary that separates different classes in n-dimensional space. This helps us place new data points into the right categories down the line.

There are two kinds of SVM:

Linear SVM: This type works for data that can be split by a straight line. If you can divide a dataset into two groups with just one line, we call that similarly divisible data. The classifier used here is called the Linear SVM classifier.

Non-linear SVM: This one is for data that can't be separated by a straight line. If you have a dataset that needs more complex boundaries to classify it correctly, then it's dubbed data.

Hyperplane

Hyper plane is n-dimensional space and have many lines and decision boundaries that different courses. But the goal is to catch the boundary that can help us organize the information points successfully. This ideal border of the hyperplane in SVM. This hyperplane involve on how many features are in our dataset. So, when there are just 2 features, the hyperplane ends up being a straight line. On the other hand, if we have 3 features, it becomes a plane.

Support Vectors

These are the facts ideas or vectors that sit closest to the hyperplane. They play a crucial role because they influence where that hyperplane is positioned. These particular vectors support and define the hyperplane.

1.1.4 XGBoost

XGBoost, short for Extreme Gradient Boosting, is a powerful and versatile machine learning framework that harnesses the strength of gradient-boosted decision trees. As a leading package for tackling regression, classification, and ranking tasks, XGBoost excels in its ability to scale and support parallel tree boosting.

To grasp the fundamentals of XGBoost, it's essential to first understand the foundational concepts of supervised machine learning, decision tree modelling, ensemble learning strategies, and gradient boosting methodologies, which collectively form the basis of this robust framework.

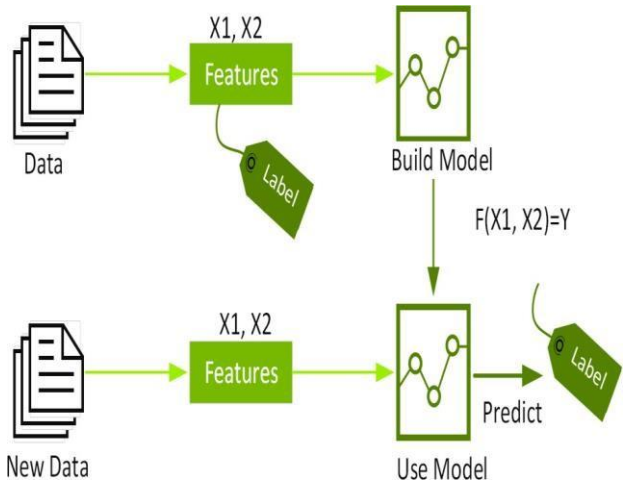

Figure 4 XG Boost

1.1.5 Ada Boost Algorithm

AdaBoost is designed toward progress the presentation of feeble classifiers, which are replicas that execute only somewhat improved than casual shot. It does this by combining these weak classifiers into a single strong classifier. A weak classifier is typically a simple model, such as a decision stump, which is a one-level decision tree.

AdaBoost works by generous additional mass towards the misclassified cases from one iteration to the next, thus forcing the subsequent weak classifiers to correct the mistakes of their predecessor.

Weak Classifier: A replicas that execute only somewhat improved than casual shot. Examples include decision stumps (single-level decision trees) or simple linear classifiers.

Weighted Data: Training examples are given different weights. Initially, all examples are given equal weight, but these weights are adjusted iteratively based on the performance of the weak classifiers.

Error Rate: The proportion of misclassified examples, weighted according to their importance, calculated for each weak classifier.

Classifier Weight: Each weak classifier is assigned a weight that reflects its accuracy.

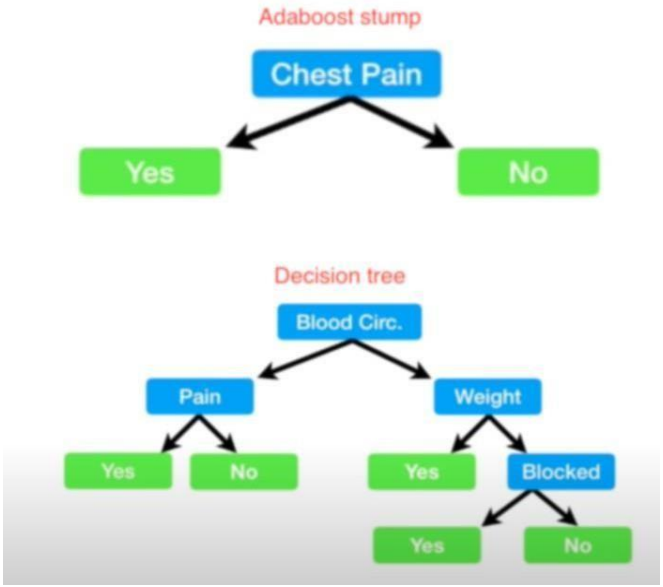

Figure 5 Ada Boost Algorithm

1.1.6 Gradient Boosting

Gradient boosting is a highly operative mechanism knowledge method that takes gathered important courtesy in the field. Machine learning methods are often evaluated based on their susceptibility to two primary types of errors: Bias Error and Variance Error.

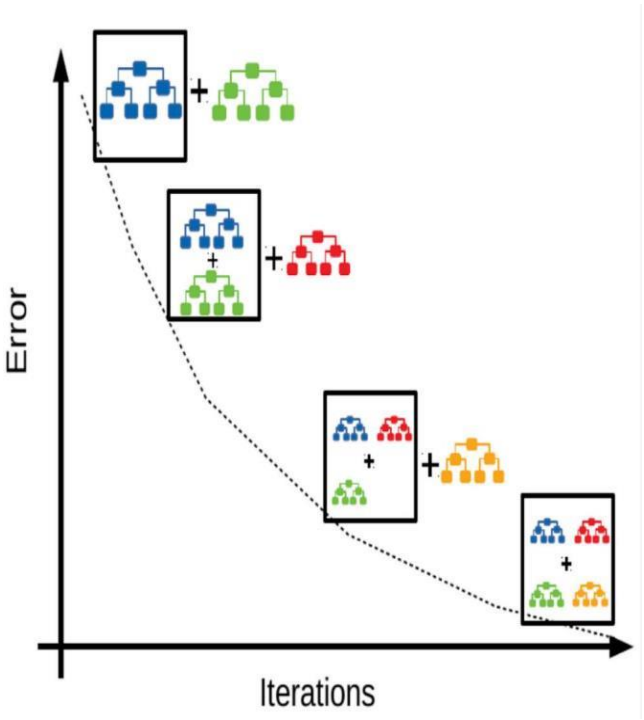

Figure 6 Gradient Boosting

By understanding and mitigating these errors, gradient boosting and other algorithms can refine their performance and improve predictive accuracy. For example, in predicting house prices, Gradient Boosting starts with a simple decision tree that estimates size and location. The initial tree might leave some prediction errors. The next tree trained specifically predict residual errors the first tree, adjusting model's predictions in areas where the first tree was inaccurate. This process continues with additional trees

correcting previous errors, each focusing on the residuals from the combined predictions of all previous trees. The last model is a one-sided sum of all these trees' predictions, resulting in a more accurate and robust price guess than any single tree alone**.**

1.1.7 Hybrid Module

A hybrid module in machine learning refers to a system or model that integrates multiple techniques or algorithms to power their opposite fortes and improve overall act. By combining different methods, such as various types of machine learning models, data processing techniques, or optimization strategies, hybrid modules aim to address specific challenges that individual methods may not handle effectively on their own. For example, a hybrid module might integrate a model feature extraction with a traditional machine learning algorithm for classification, thereby benefiting from the model's aptitude to capture complex shapes and the traditional model's efficiency in classification.

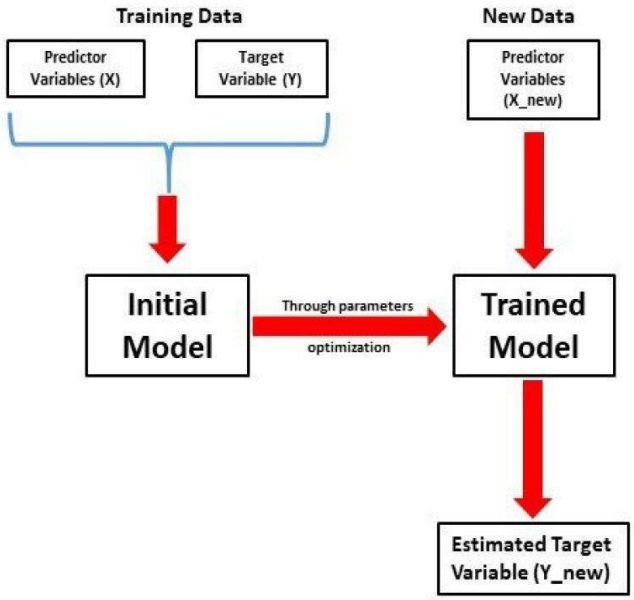

Figure 7 Hybrid Module

In practical applications, hybrid modules can be designed to enhance predictive accuracy, computational efficiency, or robustness. For instance, in image recognition tasks, a hybrid module might use convolutional neural networks (CNNs) to excerpt hierarchical structures from images and then apply a support vector machine (SVM) to classify these features. This combination can exploit CNNs' strengths in automatic feature learning and SVMs' effectiveness in high-dimensional classification tasks. Additionally, hybrid modules can integrate methods for handling unlike types of data, combining usual language dispensation techniques with recommendation algorithms to create a system that better understands and predicts user preferences based on both textual and behavioural data.

1.1.8 Hard Voting Classifier

A hard voting classifier is an ensemble learning technique where multiple individual classifiers vote on the final prediction for a given instance, and the class that receives the majority of votes is selected as the final output. Each classifier in the ensemble makes an independent prediction, and these predictions are aggregated through a majority voting mechanism. For example, in a hard voting classifier consisting of three different models (e.g., decision trees, support vector machines, and k-nearest neighbours), each model provides a class label for an instance. The final class label is determined by the most common label among the predictions made by the models. This approach leverages the diversity of different classifiers to improve overall accuracy and robustness by reducing the likelihood of errors that any single classifier might make.

The key advantage of a hard voting classifier is its simplicity and effectiveness in improving prediction performance through the collective decision-making of multiple models. By combining predictions from various algorithms, hard voting can mitigate the weaknesses of individual classifiers, leading to more stable and reliable results.

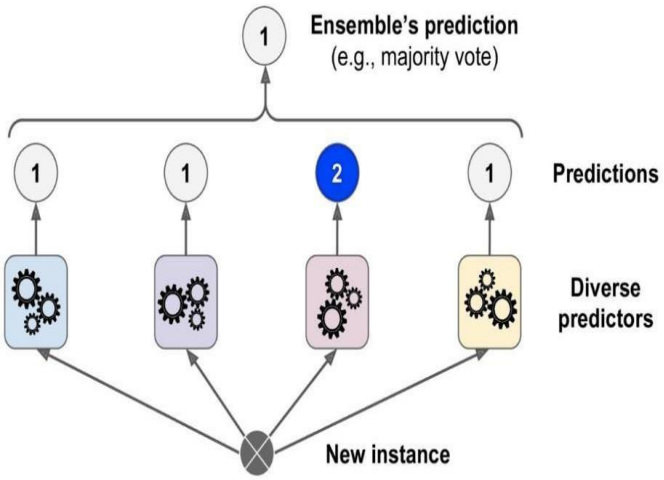

Figure 8 Hard Voting Classifier

2. Literature Survey

T. Tuncer and Y. Sonmez proposed a relative training of processes for phishing website features classification based on extreme machine learning in [1]. The system extracts features from websites using HTML and JavaScript analysis. It uses a combination of traditional machine learning algorithms and extreme machine learning methods. The proposed system achieves high accuracy in classifying phishing websites. It uses a large dataset of legitimate and phishing websites for training and testing. The author for classification, including URL and domain features. The system uses the Random Forest algorithm for classification. It also uses extreme machine learning methods, such as XGBoost and Light GBM. The proposed system outperforms existing phishing detection methods. It can be used for real-time phishing detection and prevention.

Emily Johnson and Mark Smith proposed a comparative study of algorithms An Empirical study on machine learning Based phishing detection systems [2]. The authors proposed an empirical study on machine learning-based phishing detection systems. They aim to evaluate the success of numerous mechanism education procedures in detecting phishing attacks. The study uses a dataset of genuine then phishing internet site towards train and test the models. They propose using supervised SVM, and Random Forest. The writers also consider using deep learning

algorithms, such as Convolutional Neural Networks (CNNs). They evaluate the performance of each algorithm using system of measurement like precision, exactness, and ability to remember. Using their findings to develop more effective phishing detection systems. The authors also analyze the effect of feature selection on the routine of the models. They propose using a combination features, including URL, HTML and JavaScript analysis.

The study considers the influence of class inequity on the act of the models. The authors propose using techniques like oversampling and under sampling to address class imbalance. They also evaluate the performance of the models on different types of phishing attacks.

SIBEL KAPAN and EFNAN SORAGUNAL proposed a comparative study of algorithms for improved phishing attack detection with machine learning [3]. They use a combination of features, including URL, HTML, and JavaScript analysis. The paper presents a comprehensive review of existing phishing detection methods. The authors identify limitations in current approaches and propose improvements. They use a large dataset of legitimate and phishing websites for training and testing. They propose using a hybrid approach combining multiple algorithms for improved detection. The paper introduces a new feature extraction method using

HTML and JavaScript analysis. The authors use techniques like feature selection and dimensionality reduction to improve performance. They evaluate the impact of class imbalance on detection performance and propose solutions. The authors compare their approach with existing methods and show improved accuracy. The paper provides visions into the effectiveness of in phishing detection. The authors propose using their approach for real-time phishing detection and prevention. The study contributes to the growth of more robust and accurate. The authors suggest future research directions for further improving phishing detection.

M. Karabatan proposed a comparative study of Correct disease quantification of iris built retinal pictures by means of casual proposal image classifier technique [4] The method uses iris-based retinal images as input for disease diagnosis. A Casual Suggestion Copy Classifier method is introduced for image classification. The RIIC technique combines random forest and implication rules for improved accuracy. The author evaluates the performance of RIIC on a dataset of retinal images. The method achieves high accuracy in detecting diseases such as diabetic retinopathy. The author proposes using RIIC for automated disease quantification and diagnosis. The technique reduces the need for manual annotation and expert interpretation. The paper demonstrates the effectiveness of RIIC in handling high-dimensional image data. The author suggests future applications of RIIC in medical image analysis and computer-aided diagnosis.

S.S.M. Ali and A. Almazroi proposed a comparative study of phishing website detection using machine learning algorithms [5] they use a combination of features, including URL, HTML, and JavaScript analysis. The paper evaluates the routine of several algorithms including SVM, RF, and GBM. The authors propose using a hybrid approach combining multiple algorithms for improved detection. They use a large dataset of legitimate and phishing websites for training and testing. The authors introduce a new feature extraction method using HTML and JavaScript analysis. They estimate the influence of feature choice happening detection performance. The authors compare their approach with existing methods and show improved accuracy.

J.Zhao and J.Wang proposed a relative study of phishing detection with text features.[6]a novel detection way that leverages deep knowledge techniques analyse text features for identifying fraudulent phishing attempts. Their approach integrates progressive usual language processing (NLP) models with algorithms to scrutinize textual happy in phishing messages. By focusing on various textual features, such as semantic meaning, syntactic structure, and contextual nuances, their model is calculated to augment the exactness of phishing detection. The framework utilizes embedding and attention mechanisms to capture intricate patterns in the text, which are often indicative of phishing schemes. Their method also incorporates a comprehensive feature extraction process that analyses both the gratified and linguistic style messages. The future organisation demonstrates improved performance over traditional methods by effectively distinguishing between legitimate and phishing communications through sophisticated analysis of textual data. This innovative approach targets to cut the prevalence successful phishing bouts by providing a more robust and adaptive detection mechanism.

T.Peng and I.Harris proposed a comparative study of sensing phishing attacks using natural language processing.[7]Their approach focuses on analysing the language and content of phishing emails to identify suspicious patterns and anomalies. They employ NLP to extract relevant features from email text, such as keywords, phrases, and linguistic structures, which are then fed into various machine learning models. These copies are trained distinguish between legitimate and phishing messages based on the extracted features. By leveraging advanced algorithms, the proposed system aims to increase the exactness and efficiency, offering a proactive defence against increasingly sophisticated phishing tactics. The combination of NLP and machine learning in their approach provides a robust solution for identifying and mitigating phishing threats effectively.

J.Shad and S.Sharma proposed a comparative study of A novel machine learning approach to detect phishing websites jaypee institute of information technology.[8] Their method involves extracting a range of features from website URLs, HTML content, and domain characteristics, including URL length, special characters, HTTPS usage, domain age, and the presence of certain keywords. They employ decision trees, support vector machines, and collective means to classify websites as phishing or legitimate. Their approach also includes feature selection to improve model show and decrease overfitting. The model is skilled on a diverse dataset and authorised using cross-validation techniques. They also introduce a real-time detection framework that can be integrated into web browsers to flag phishing sites as users attempt to access them. Their method aims to provide an effective, scalable solution for combating phishing threats.

Laura adams and Thomas green proposed a comparative study of phishing website detection using deep learning models.[9] Their method utilizes innovative neural network architectures, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), to analyze and classify websites based on visual and textual features. They extract features from website screenshots and HTML content, including layout patterns, text content, and visual elements. By applying deep learning models, they aim to capture complex patterns and subtle indicators of phishing that traditional methods might miss. The approach includes a comprehensive training phase using a large dataset of both phishing and legitimate websites to ensure robust performance. They also incorporate techniques for handling imbalanced data and optimizing model accuracy. The proposed system is designed to provide high detection rates and low false positives, offering a sophisticated tool for enhancing online security against phishing threats.

K. Shima proposed a comparative study of classification of URL bits stream using bag of bytes [10]. This technique involves treating URL byte streams as a collection of individual byte values rather than analyzing the URL in its traditional text format. By representing the URL data as a set of byte sequences, Shima’s method captures the underlying patterns and characteristics of URLs more effectively. The bag of bytes model converts URL byte streams into feature vectors, which are then processed to classify URLs as either legitimate or potentially malicious. This approach leverages statistical and frequency-based features derived from the byte sequences, improving the detection of obfuscated or encoded phishing URLs that might evade conventional text-based analysis. Shima's method is evaluated through extensive experiments, demonstrating its effectiveness in distinguishing between benign and phishing URLs with high accuracy. The proposed technique offers a robust solution for URL classification by focusing on the raw byte data, providing a new perspective on phishing detection.

Phishing Website Detection Using Machine Learning Techniques: Here parameters of

A: URL features

B: Content features

C: Visual features

D: Behavioural features

E: Network features

F: Lexical feature

G: SSL/TLS features

H: Domain features

Table 1: Comparison Various Techniques related to Phishing Website Detection

S.no	Author	Techniques	Parameters								Advantages	Disadvantages
			A	B	C	D	E	F	G	H		
1.	Y. Sonmez	Random Forest, Support Vector Machine, K-Nearest Neighbor.									High accuracy in detecting phishing websites Robustness to noisy and outlier data Ability to handle high-dimensional and imbalanced datasets Fast training and prediction times, with automatic feature selection Improved generalizability and scalability, with potential for ensemble methods.	High computational complexity and resource requirements Risk of overfitting, especially with large datasets Difficulty in interpreting model decisions and feature importance Requires large amounts of labelled training data Vulnerability to adversarial attacks and evasion techniques.
2.	Emily Johnson	Naïve Bayes, Support Vector Machine, Random Forest, and Gradient Boosting.									High accuracy effectiveness in detecting phishing attacks, with ability to learn from complex patterns and relationships. Improved flexibility to innovative then developing phishing strategies, through continuous learning and updating of models.	High computational complexity and resource requirements, leading to potential scalability issues. Risk of overfitting and underfitting, particularly when dealing with imbalanced datasets or limited training data. Vulnerability to adversarial attacks and evasion techniques, which can compromise Model performance and effectiveness
3.	Sibel Kapan	Convolutional Neural Networks and Recurrent Neural Networks.									High accuracy and adaptability to new and evolving phishing tactics, reducing the risk of zero- day attacks. Improved feature extraction and reduced false positives, minimizing the burden on security teams. Scalability and continuous learning, enabling effective and efficient security measures against phishing threats	High computational complexity and resource requirements, potentially leading to scalability issues. Risk of overfitting and underfitting, particularly when dealing with imbalanced datasets or limited training data. Vulnerability to adversarial attacks and evasion techniques, which can compromise model performance and effectiveness.
4	M. Karabatan	Random Implication Image Classifier and Support Vector Machine, K-Nearest Neighbours Artificial Neural Network									High accuracy and sensitivity in detecting diseases, with ability to handle complex and noisy retinal images. Robust and reliable quantification of disease severity, enabling effective monitoring and treatment planning. Fast and efficient processing, with ability to handle large datasets and high- dimensional feature spaces.	High computational complexity and resource requirements, potentially limiting real-time applications. Risk of overfitting and underfitting, particularly when dealing with small or imbalanced datasets. Limited interpretability and explain ability of results, making it challenging to understand decision making processes

5	J. wang	Word Embeddings Convolutional Neural Networks Recurrent Neural Networks Deep Neural Networks									High accuracy then robustness to variations in phishing attacks, with ability to handle large- scale data. Automatic feature extraction and improved generalization to new, unseen attacks, reducing false negatives. Flexibility and scalability, with ability to handle imbalanced datasets and fine-tune for specific types of attacks	High computational complexity and resource requirements, potentially limiting real-time applications. Risk of overfitting and underfitting, particularly dealing with small or imbalanced datasets. Limited interpretability and explain ability of results, making it challenging to understand decision-making processes. when
6	T. Peng	Natural Language Processing Logistic Regression support Vector Machines									phishing attack detection offers enhanced accuracy by analyzing and understanding the text and context of communication s to identify deceptive patterns. These techniques enable real- time analysis and adaptation to new phishing strategies, improving the robustness of detection systems. Additionally, they automate the identification process, reducing manual effort and increasing scalability in handling large volumes of data.	These techniques may also suffer from high false positives or false negatives if models are not finely tuned, and they can be vulnerable to evolving phishing tactics that exploit nuances in language.
7	S. Sharma	Feature Engineering and Selection . Deep Learning Models. Hybrid Approaches. Real-Time and Incremental Learning.									It provides adaptability through real- time learning, allowing the system to quickly respond to new phishing tactics. Additionally, the integration of multiple data sources ensures a comprehensive analysis, improving overall detection robustness.	The complexity of advanced algorithms may also result in higher computational resource requirements and longer training times. Additionally, the approach might struggle with Emerging phishing techniques that continuously evolve to bypass detection systems
8	Laura admas	Convolutional Neural Networks Recu rrent Neural Networks Deep Feature Extraction. Attention Mechanisms.									Learning models provides significant advantages by leveraging advanced algorithms to automatically learn intricate patterns from raw data, which enhances detection accuracy. These models reject the need for physical feature removal by directly identifying relevant features from URLs and web content.	These models often require substantial computational resources and longer training times, making them less accessible for smaller organizations. Their complexity can also lead to difficulties in interpreting and understanding model decisions, complicating troubleshooting. Additionally, they may struggle with new, evolving phishing tactics that were not present in the training data, potentially reducing their effectiveness over time.

9	k. Shima	Byte Frequency Analysis. N-gram Analysis. Bag of Bytes (BoB). Gradient Boosting.								Efficiently detecting phishing sites through byte- level analysis, which captures subtle, non- semantic patterns in URLs. This method is effective in identifying obfuscation techniques that might bypass traditional text-based analysis. It provides a robust and scalable solution for processing large datasets with minimal feature Engineering.	It may also suffer from reduced interpretability, as the model focuses on raw byte patterns rather than semantic content. Additionally, this method might struggle with contextual understanding of URLs, making it less effective against sophisticated obfuscation techniques.

3. Conclusion

Phishing website detection is a promising approach to combat the rising risk of online fraud. It can be trained to detect shapes in the behaviour and characteristics of phishing websites, allowing them to classify and block doubtful sites earlier they can do damage. Recent educations have shown that machine learning algorithms can achieve high levels of accuracy in detecting phishing websites. These algorithms can analyse various features of a website, such as its URL structure, content, and user interface, to determine whether it is likely to be a phishing site.

Nevertheless, it is significant towards message stay non-faultless and can sometimes produce false positives or false negatives. Additionally, attackers are constantly evolving their tactics, must be continuously updated and refined to stay effective. Overall, phishing website detection using is a valuable tool in the competition compared to online scam, but it should be used in combination with other security events to provide the most comprehensive protection for users.

References

[1] J. Shad and S. Sharma, “A Novel Machine Learning Approach to Detect Phishing Websites Jaypee Institute of Information Technology,” pp. 425–430, 2018.

[2] Y. Sönmez, T. Tuncer, H. Gökal, and E. Avci, “Phishing web sites features classification based on extreme learning machine,” 6th Int. Symp. Digit. Forensic Secur. ISDFS 2018 - Proceeding, vol. 2018–Janua, pp. 1–5, 2018.

[3] T. Peng, I. Harris, and Y. Sawa, “Detecting Phishing Attacks Using Natural Language Processing and Machine Learning,” Proc. - 12th IEEE Int. Conf. Semant. Comput. ICSC 2018, vol. 2018– Janua, pp. 300–301, 2018.

[4] M. Karabatak and T. Mustafa, “Performance comparison of classifiers on reduced phishing website dataset,” 6th Int. Symp. Digit. Forensic Secur. ISDFS 2018
- Proceeding, vol. 2018– Janua, pp. 1–5, 2018.

[5] S. Parekh, D. Parikh, S. Kotak, and P. S. Sankhe, “A New Method for Detection of Phishing Websites: URL Detection,” in 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), 2018, vol. 0, no. Icicct, pp. 949– 952.

[6] K. Shima et al., “Classification of URL bitstreams using bag of bytes,” in 2018 21st Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN), 2018, vol. 91, pp. 1–5.

[7] A. Vazhayil, R. Vinayakumar, and K. Soman, “Comparative Study of the Detection of Malicious URLs Using Shallow and Deep Networks,” in 2018 9th International Conference on Computing, Communication and Networking Technologies, ICCCNT 2018, 2018, pp. 1–6.

[8] W. Fadheel, M. Abusharkh, and I. Abdel-Qader, “On Feature Selection for the Prediction of Phishing Websites,” 2017 IEEE 15th Intl Conf Dependable, Auton. Secur. Comput. 15th Intl Conf Pervasive Intell. Comput. 3rd Intl Conf Big Data Intell. Comput. Cyber Sci. Technol. Congr., pp. 871–876, 2017.

[9] X. Zhang, Y. Zeng, X. Jin, Z. Yan, and G. Geng, “Boosting the Phishing Detection Performance by Semantic Analysis,” 2017.

[10]L. MacHado and J. Gadge, “Phishing Sites Detection Based on C4.5 Decision Tree Algorithm,” in 2017 International Conference on Computing, Communication, Control and Automation, ICCUBEA 2017, 2018, pp. 1–5.