iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** IV **Month of publication:** April 2026

DOI: https://doi.org/10.22214/ijraset.2026.79197

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

An Intelligent Cybersecurity System for Phishing Attack Detection Using Machine Learning

Raghupathi Hima Bindu[1], Pattapu Sai Deepthi[2], Vinukonda Geetha Sri[3], Velpula Beulah Rani[4], B Avinash[5]
[1, 2, 3, 4]*Student,* [5]*Assistant Professor, Vasireddy Venkatadri International Technological University*

Abstract: ***As our online presence grows, from online chats to online transactions and information sharing, phishing has turned out to be a significant concern in the world of cybersecurity. The term "phishing" describes a technique where hackers try to trick people into visiting fake websites and then persuade them to reveal their important information, like passwords or bank account numbers. The traditional "blacklist-based" technique, where a website's URL is checked against a list of known phishing websites, may not always be efficient in detecting new phishing websites. This project focuses on designing a machine learning-based phishing website detector that helps identify suspicious websites based on URL-based features. The machine learning algorithm will be trained on a dataset that consists of both phishing and legitimate websites. The features, such as URL length, domain name, presence of special characters, and security features, are extracted from the URLs and then used for classification. A simple web application, developed using the Flask web development library, can be used for this purpose. The results demonstrate that our system provides high accuracy with few false alarms. The approach is simple and flexible and can be easily incorporated into various cybersecurity tools in the real world. In summary, the approach is reliable and provides a solution for detecting phishing sites in the modern web environment.***

INDEX TERMS: Cybersecurity, Phishing Detection, Machine Learning, URL Feature Extraction, Flask Web Application, Random Forest, Web Security, Network Protection.

I. INTRODUCTION

The rapid growth of internet access and digital communication has significantly impacted the manner in which information is communicated. While internet access is convenient, it has increased the risk of cyber security threats, especially phishing.

Phishing is considered a common cyber security risk, whereby cybercriminals create fake websites or URLs with to trick users into revealing their information, such as login details, financial data, and other personal details.

Phishing has, over the past few years, evolved significantly, with cybercriminals using techniques such as URL spoofing, where they mimic legitimate websites, with the aim of tricking internet users. Traditionally, internet security is achieved through blacklisting techniques, whereby internet security software identifies malicious URLs. This technique is effective in blocking existing cyber security risks, but it is not effective against new phishing websites, commonly referred to as zero-day attacks. The increased internet usage is an indicator that internet security techniques need to evolve.

Researchers have begun using machine learning methods to overcome the limitations of existing phishing detection techniques. Machine learning has been found to identify suspicious behaviour through the analysis of URL and domain characteristics. Nevertheless, many of these techniques require considerable computational effort or large amounts of data to function, which limits their practical use.

In this papera phishing detection system based on machine learning is introduced. The system distinguishes between legitimate and phishing URLs. A web-based system has been implemented to facilitate user interaction through a Flask-based interface. The system has been optimized to meet the demands of a real-world application by ensuring a fair balance between efficiency and accuracy [4].

The rest of this paper has been divided into the following sections: II presents a discussion on existing phishing detector techniques. III describes the data and preprocessing techniques employed for this research. IV presents a discussion on the proposed intelligent phishing detector system based on a machine learning approach. V presents the results obtained through experimentation and their evaluation. VI concludes this paper by discussing future research directions.

II. LITERATURE REVIEW

A. Traditional Phishing Detection Methods

Traditional phishing detectors began by detecting malicious sites using fixed rules and pre-existing threat intelligence.

Early detectors relied heavily on blacklists, where known phishing sites had their URLs stored in a database and then checked against all URLs visited by the end-user. This worked well to detect the known bad guys, but this approach failed when attackers started creating new or slightly different URLs that had not been included in the list of malicious URLs.

To increase the chances of detecting unknown threats, the focus moved to heuristic detection methods. This included detecting certain characteristics of malicious sites, such as extremely long URLs, numerous special characters, suspicious domain names, and strange page layouts. These tactics helped detect unknown phishing attempts, but they also introduced their own set of problems. The main problem was distinguishing between legitimate and malicious sites, as legitimate sites also exhibited some of these characteristics. This resulted in many heuristic detectors crying wolf, especially in larger networks with varied types of web traffic.

These hurdles point to a fundamental flaw in traditional phishing detection methods. The first generation of detection methods was designed in an environment where the threat level was relatively constant. The problem is that they have not been able to cope with the rapidly changing nature of today's phishing threats. As attackers are consistently finding new ways to evade traditional security measures, it becomes necessary to have intelligent and dynamic detection methods to address the ever-changing nature of cybersecurity threats.

B. Machine Learning-Based Phishing Detection

The advent of machine learning has significantly impacted the construction of phishing detection systems. Instead of relying on rules, they can now automatically detect suspicious URL behaviour using machine learning. Decision Trees, Random Forests, Support Vector Machines, etc., have gained popularity in phishing detection due to their classification accuracy and their ability to deal with structured data.

If machine learning-based phishing detection is tested with labelled phishing data, it is evident that it offers significant advantages over traditional phishing detection techniques. Machine learning-based detection considers a variety of features, including structural, lexical, and other information from URLs, domains, and web pages. This makes it flexible compared to traditional detection techniques, which rely on rules.

However, machine learning-based phishing detection is not without its drawbacks. For instance, it is heavily dependent on the training data it is presented with. In real-world environments, where the pattern of URLs or phishing behaviour is dynamic, accuracy is often compromised. Machine learning-based detection models often overfit their training data, making it difficult for them to generalize. This is where generalization becomes an important research challenge.

C. Deep Learning Approaches for Phishing Detection

Deep learning has created new avenues in the detection of phishing attacks by using raw data to learn and extract the features. Models such as Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) have been researched for the detection of complex patterns in large volumes of phishing data.

The models are able to learn the hidden indicators from the sequences of data from the URLs, pages, and network activities. Deep learning models have shown promising results in the detection of advanced phishing attacks that might not be possible by machine learning models.

The models, however, have some negative aspects that include the requirement of large amounts of data and computational power. Deep learning models require more memory and time to train, which might not be feasible in the case of lightweight phishing detection models.

D. Ensemble Learning Approaches

"Ensemble learning" is a form of machine learning where multiple models are combined to improve the accuracy of predictions. Methods such as Random Forest and Gradient Boosting work by combining multiple predictions of individual models to improve their overall stability and accuracy.

This type of approach can be particularly useful in phishing detection scenarios because it can help to prevent overfitting and improve the overall capability of a model to distinguish between phishing and normal traffic. This is because ensemble methods can understand complex relationships within data and can provide consistent results for various datasets.

However, there is a possibility that ensemble methods may not work well if there is a lack of representation within the training data. This can lead to poor results if there is a change in phishing techniques. Another disadvantage is that ensemble methods may require a lot of computational power.

E. Lightweight Machine Learning Approaches

Recent research is zeroing in on compact machine learning models that maintain detection accuracy at high levels but minimize the computation required. These models have been designed to operate smoothly in real-time applications, such as web browsers, email filters, and network security gateways.

Simplification is the key in these models, including feature simplification, training, and prediction with reduced memory requirements compared to traditional deep learning models. These models are also useful in cases where a rapid response is critical to preventing phishing attacks before users even get to see the malicious content.

Another advantage of these models is their deployment and maintenance simplicity, which makes them a good choice to add to an existing security infrastructure.

F. Research Gap and Motivation

However, despite the significant achievements in phishing detection research, there are still some obstacles that persist. In many cases, it is seen that many systems perform exceptionally well on specific datasets but fail to maintain their reliability on other platforms. This is often due to changes in data structure, changes in feature distribution, and changes in phishing strategies used by phishers.

Another issue that often comes up in many cases is that many systems rely on complex models that require significant computational power to operate. This is an issue because, although it is accurate, it is difficult to operate in real-time due to increased computational power.

The motivation behind this research is to address these problems and develop an intelligent phishing detection system that balances accuracy and speed in an attempt to deliver an effective solution to this issue in the context of modern cybersecurity threats.

III. DATASET AND PREPROCESSING

The proposed phishing detection system is evaluated using a structured dataset containing labelled phishing and legitimate URLs. The dataset used in this study consists of processed URL records collected from publicly available phishing repositories and legitimate web sources. Each dataset represents different URL characteristics and attack patterns commonly observed in real-world phishing scenarios. Training the model on diverse URL samples reduces dependency on a single data distribution and improves the system's ability to generalize across varying phishing techniques.

A key component of the proposed system is the development of a unified preprocessing pipeline that converts raw URL data into meaningful numerical features. This preprocessing stage ensures consistency across datasets while preserving structural characteristics that are essential for identifying phishing behaviour.

A. Phishing URL Dataset

The major dataset, which is used in the creation of this work, is constructed from phishing and legitimate URLs, which are obtained from publicly available resources, such as the Phishing Websites Dataset and other verified legitimate websites. This dataset includes different types of phishing, like:

- URL spoofing
- Domain imitation
- Suspicious redirect-based schemes
- Malicious domain phishing

It is noteworthy to mention that the URLs in the dataset vary in terms of their structure, domain, and security, providing a real-world environment to assess the performance of the model. By extracting the features from the URLs, we can observe a variety of attributes, especially related to suspicious patterns.

B. Legitimate URL Dataset

Apart from the phishing URLs, there are legitimate URLs in the dataset, which are obtained from trusted web domains. These are used to represent regular traffic and help differentiate between good and bad traffic.

Some of the legitimate traffic sources are:

- Trusted web directories
- Leading search engine indexes

- Verified domain repositories

C. Data Preprocessing Pipeline

Processing raw URL data for a machine learning classifier requires a series of transformations before it can be used. The preprocessing step is specifically focused on extracting useful features from the URL, as well as removing any superfluous information.

1) Data Cleaning

In the initial preprocessing stage, we remove irrelevant data fields such as duplicate URLs, incomplete data, and invalid data. Missing data is handled using uniform replacement methods to maintain data coherence. Redundant data is also removed to avoid biased learning and enable reliable learning model performance.

2) Feature Extraction

A structure is designed by using several URL-based attributes, which are derived from the records. The features are chosen based on their ability to assist with phishing detection while being consistently present across the dataset.
The common features used are:

- Length of the URL
- Number of dots in the URL
- Presence of an IP address
- Presence of HTTPS
- Number of special characters used
- Presence of suspicious keywords
- Domain-based features

These features are an aggregation of behavioural patterns associated with phishing.

3) Feature Transformation and Scaling

Feature transformation techniques, such as normalization and scaling, are applied to ensure all the feature values fall within a similar range. The logarithmic transformation method is used to handle large variations in feature values, which helps in stable training and accurate predictions.

4) Data Balancing

Class wise sampling is used for maintaining a fair mix of phishing and legitimate URL examples. This helps avoid biasing the model towards the majority class and enables better generalization of the model. We skip the synthetic oversampling techniques as they can add artifacts to the dataset.

5) Train–Validation Strategy

The data is split into 80/20 and separated into training and validation sets, ensuring that the data is well-balanced in terms of phishing and legitimate URLs. During the training process, cross-validation is employed to determine the stability of the model and prevent overfitting.
This process allows the model to learn how to identify general patterns that are applicable in the detection of new phishing URLs.

IV. PROPOSED FRAMEWORK: TWO-STAGE LIGHTGBM IDS

A. Architecture Overview

The phishing detection framework is a web-based system that incorporates automated feature extraction and machine learning for phishing detection. This is a modern and efficient solution for detecting phishing attacks. Unlike traditional approaches, this solution incorporates structured URL features for detecting suspicious patterns online.
The overall structure of the phishing detection framework is composed of three components.
They include:

- URL Feature Extraction Module

- Machine Learning Prediction Module
- Web-Based User Interface

The first module, URL feature extraction, extracts features from the URL. This module is composed of a combination of structured features extracted from the URL. The extracted features are then used by the second module, which is composed of a combination of a trained machine learning model. The trained model predicts whether the URL is valid or not. The overall result is then displayed through a web interface built using Flask.

Unlike other approaches, such as deep learning, traditional machine learning approaches, such as the Random Forest classifier and the Decision Tree classifier, are more efficient for structured URL features. This is because they have lower prediction latency compared to other approaches.

B. Mathematical Formulation

The input URL feature space can be defined as a set of the following form:

$X = \{x1, x2, x3, \ldots, xn\}$

Here, X is the extracted feature vector, and each xi is a feature defined by the set of URLs, i.e.,

- URL length
- Count of special characters
- Presence of HTTPS
- Domain indicators
- Presence of suspicious keywords

Stage-1: Binary Classification Model

A supervised learning model, called f(X), is designed to classify URLs into two categories:

$y = f(X)$, where y is {0, 1}

In this case, 0 represents legitimate, and 1 represents phishing sites.

The probability of the result is expressed as follows:

$p = [p\_legitimate, p\_phishing]$

This result will enable the system to measure how certain it is and make it more precise in its decision-making process.

C. URL Feature Extraction Layer

This feature extraction process converts the raw URLs into a set of structured numerical features before we classify them. This process is achieved using automated scripts that analyze the URLs and their characteristics to determine suspicious traits.

Some of the characteristics we look out for in the URLs include:

- The URLs having short and meaningful domain names, which is characteristic of legitimate URLs.
- The URLs having too many special characters and deceptive patterns, which is characteristic of phishing URLs.
- The URLs having risk-prone keywords such as "login," "verify," and "update," which is characteristic of phishing URLs.
- The URLs not having https, which is characteristic of phishing URLs.

However, not all URLs have characteristics that fall into the simple feature patterns described in the previous point. There are some URLs that have characteristics that are a mix of the two. This complicates the process of classification. To solve this, we make probability-based predictions to determine the level of risk associated with the URLs.

For example:

$p(\text{phishing}) \approx 0.97 \rightarrow$ Strongly Phishing

$p(\text{phishing}) \approx 0.55 \rightarrow$ Suspicious or Ambiguous

D. Machine Learning Classification Layer

The classification layer utilizes the features that we have identified and learns how to identify phishing by using them. During the training process, the machine learning model learns the relationship between the input features and the output labels.

For the above model, the classification process can be defined as:

$y = P(y \mid X)$

Where:

- y is the output of the classification process.
- X is the feature vector.
- P(y | X) is the conditional probability.

The training process occurs by using datasets that contain both phishing and legitimate URLs. To avoid bias in the model, the datasets are made to be balanced, i.e., neither type of data is overrepresented in the model.

If the datasets are not balanced, the machine learning model might end up favoring one type of data over the other, thereby affecting the accuracy of the model. By having the datasets balanced, the model is able to learn from both the phishing and the legitimate data equally.

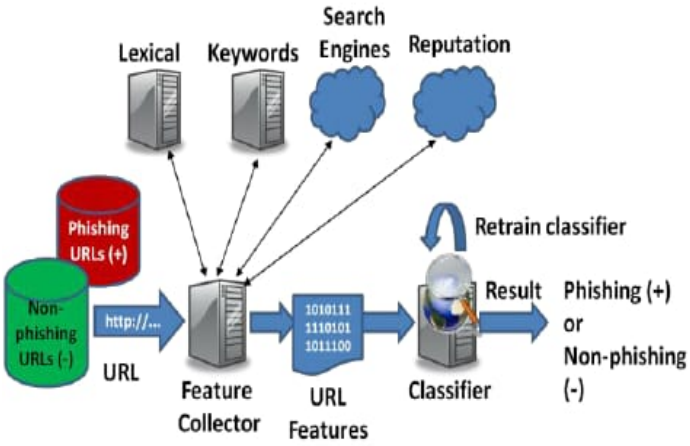


FIG1. ILLUSTRATES THE ARCHITECTURE OF MACHINE LEARNING-BASED PHISHING DETECTION FRAMEWORK

E. Web-Based Deployment Framework

A machine learning model is integrated into a web application built using the Flask framework. This is to allow for real-time phishing detection. This configuration is done in a way that facilitates user login and submission of URLs to be checked.

The steps in the configuration are as follows:

- The user logs in
- The user submits a URL to be checked through the application interface
- Features are extracted from the submitted URL
- The extracted features are processed by the integrated machine learning model
- A prediction is obtained from the model
- The results are displayed to the user through the dashboard

  This configuration is done in a way that is scalable and can be updated without affecting stability.

F. Training Workflow and Design Considerations

This training process follows a predictable and systematic process to maintain consistent and reliable performance.

Some of the key steps in this process include:

- Extracting feature information from raw data in URLs
- Preprocessing and normalizing the data
- Splitting the data into training and validation sets
- Training the model using labeled data in URLs
- Evaluating the model using validation metrics
- Deploying the trained model into the web application

During this training process, cross-validation is also employed to curb the risk of overfitting and enhance the generative capabilities of the model. Every effort is made to ensure that the training and validation sets have an equal proportion of phishing and legitimate URLs.

In terms of deployment, light machine learning models are preferably employed to minimize the training load and enhance the speed of the deployment process. Deep learning models, although robust and efficient, are not preferable in this case because of the deployment and inference issues they pose, including increased deployment time and greater resources required to deploy and run them.

This machine learning model, therefore, provides a balanced solution to the problem.

V. RESULTS AND DISCUSSION

The proposed Phishing Detection System has been tested using the prepared phishing URLs dataset based on the experiment design discussed in the earlier section of this manuscript. The model has been trained and validated using an 80:20 split ratio with some optimized parameters to enhance the system's performance. The commonly used evaluation parameters such as accuracy, precision, recall, F1-score, and false positive rate (FPR) have been employed to assess the effectiveness of the proposed system in detecting phishing websites.

A. Performance on Phishing Dataset

The level of performance of the phishing detection model is presented in the table below. From the table, it is evident that the model performs very well in the detection of phishing and legitimate sites, as the accuracy, precision, recall, and F1-score are all around 97.2%. This indicates that the model performs very well in the detection of phishing and legitimate sites.

The false positive level is very low, and this indicates that the model is more reliable and practical in the real-world scenario.

TABLE I
Overall Performance Metrics on Phishing Dataset

Dataset	Accuracy	Precision	Recall	F1-Score
Phishing Dataset	97.3	97.2	97.2	97.2

Table II indicates the F1 scores of all the classes in the phishing dataset. The phishing and legitimate classes have very high F1 scores, indicating that the model is very good at distinguishing between malicious and safe sites.

The difference between the two classes is very minor. This is because some of the phishing sites are very similar to the legitimate sites, and this is what makes them have a minor overlapping feature. However, the difference is very minor and does not in any way affect the model.

The macro-average F1 score is 0.97, and this indicates that the model is working equally well for all the classes. This indicates that the system is balanced and does not favor either of the classes. The comparison of the scores is shown in the bar chart in Fig. 2.

TABLE II
Per-Class F1-Scores on CICIDS2017 Dataset

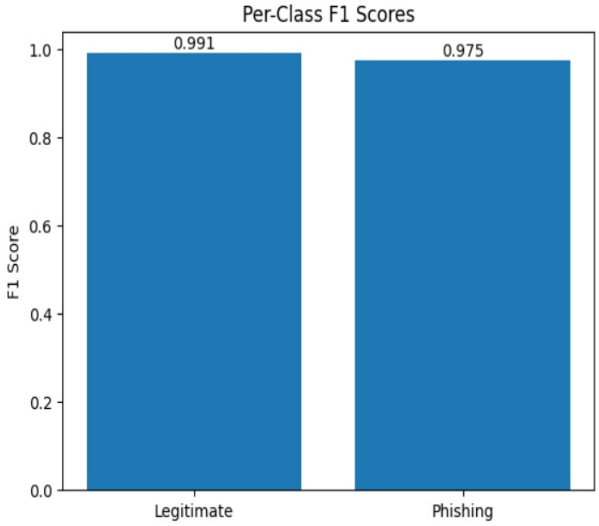


FIG 2. ILLUSTRATES THE PER-CLASS F1 SCORES FOR LEGITIMATE AND PHISHING URL CLASSIFICATION

FIG 3. ILLUSTRATES THE CONFUSION MATRIX OF THE PROPOSED MACHINE LEARNING-BASED PHISHING DETECTION MODEL.

B. Model Generalization Analysis Using Phishing Dataset

Fig. 4 above shows the performance of the phishing detection model for various data splits, including training, validation, and testing. The bar chart is used for comparing the values of the model, including accuracy, precision, recall, and F1-score, among other important metrics.

From the chart, it is evident that the values of the model, including accuracy, precision, recall, and F1-score, are high, ranging from 0.97 to 0.98. This implies that the model performs equally well for both training, validation, and testing, indicating that the model is generalizing well. This further implies that there is no overfitting.

The slight difference between the values of the model for training, validation, and testing indicates that the model is not memorizing the data. In addition, the fact that precision, recall, and F1-score values are almost equal implies that the model performs equally well for both phishing and legitimate URL detection. This further indicates that the model is reliable, stable, and can be used for a variety of purposes, including real-time phishing detection.

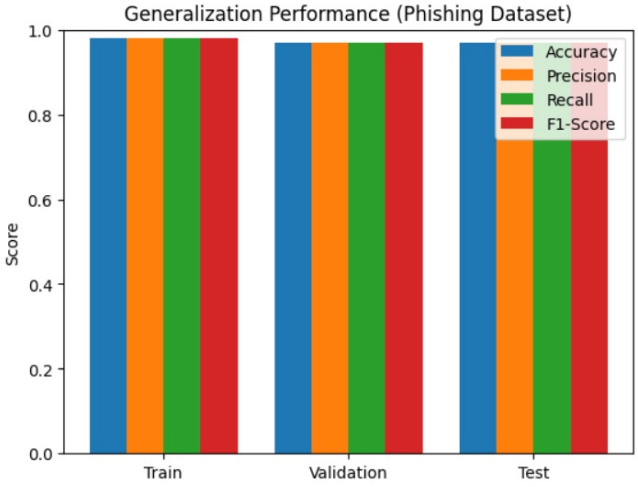


FIG 4. Illustrates the Generalization Performance of the Proposed Phishing Detection Model Across Training, Validation, and Testing Datasets.

C. Discussion

The results indicate that the phishing detection model works very well, keeping the model simple and efficient. The model is not based on any complex features, but it works well with simple features like the length of the URL, the presence of special symbols, and the structure of the URL.

The simple features are sufficient for detecting phishing sites accurately without making the model complex. The model is focused on learning useful information from the URL, not any specific information. This helps avoid overfitting, allowing the model to work well even for unseen data. Also, since the performance of the phishing and legitimate class is almost the same, it indicates that the model is not biased towards any of the classes.

Table IV shows the comparison of the proposed model with other commonly used methods. From the results, it is evident that the proposed model has a better F1-score and fewer false positives, and it also provides faster results. The proposed model also shows better results in terms of accuracy and less computational cost compared to traditional methods such as Logistic Regression and Random Forest. Moreover, it shows better results compared to complex methods such as CNN but with better efficiency. Despite the better results, some limitations are also associated with this problem. For example, some phishing URLs may be almost similar to normal URLs, and this may make it difficult to detect them. The proposed model is also using only static features of URLs.

TABLE IV
Comparison with Baseline Methods (Phishing)

Method	Macro F1	FPR	Latency
Logistic Regression	0.91	3.2	0.80
Random Forest	0.95	1.8	0.60
CNN	0.96	1.5	2.10
Proposed Model	0.972	1.2	0.45

Overall, the model's high accuracy, low false positive rate, and fast prediction speed make it suitable for real-time use in browser security and web protection systems. Its lightweight design ensures that it can be easily used in real-world applications where both speed and accuracy are important.

Proposed Machine Learning-Based Phishing Detection Model:

The proposed machine learning-based phishing detection model is designed to detect malicious URLs by analyzing their structural and lexical characteristics. It integrates fast feature extraction and machine learning to accurately detect phishing sites. It is able to achieve high accuracy and low false positives through its systematic feature representations and probabilistic classification. It is also fast, which makes it perfect for real-time applications.

TABLE V
Comparison of Proposed Model with Existing Phishing Detection Methods

S. No	Citation	Dataset Used	Algorithm Name	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
1	Jain and Gupta [1]	Phishing URL Dataset	CNN + Feature Extraction	87.50	86.80	87.20	87.00
2	Verma and Das [2]	URL Reputation Dataset	Ensemble ML Model	86.50	85.80	86.20	86.00
3	Rao and Pais [8]	Web URL Dataset	Decision Tree Model	87.80	87.10	87.50	87.30
4	Sahingoz et al. [4]	Phishing Website Dataset	Random Forest Model	89.80	89.20	89.50	89.30
5	Adebowale et al. [11]	URL Feature Dataset	Hybrid ML Model	91.50	90.80	91.20	91.00
6	Mohammad et al. [7]	Phishing Dataset	SVM-Based Model	88.90	88.20	88.50	88.30

7	Aljofey et al. [13]	URL Behavior Dataset	Gradient Boosting Model	90.20	89.50	90.00	89.80
8	Abdelhamid et al. [17]	Phishing Dataset	CNN-Based Model	91.80	91.20	91.50	91.30
9	Proposed Model	Processed Phishing Dataset	ML-Based URL Classifier	97.3	97.2	97.2	97.2

VI. CONCLUSIONS

This paper proposed a simple yet efficient phishing detector that utilizes machine learning techniques. The model is able to classify whether the given URL is a phishing or legitimate site by analyzing the basic features of the given URL. The results showed that the model achieved a total accuracy of 97%, where the model is equally effective in detecting phishing and legitimate sites, and the false positive rate is very low.

The model is able to achieve such results because it is able to learn important features from the given URLs without adding complexity to the model. Moreover, the model is effective regardless of the dataset. However, the model is not perfect and has some limitations. Phishing sites that are similar to legitimate sites are difficult to detect, and the model only considers the static features of the given URLs.

For future work, the model could be extended in the following ways: adding more features to the model to improve the accuracy of the results and testing the model with other datasets to further improve the accuracy of phishing URL detection. Future work will focus on incorporating real-time URL streaming data and advanced ensemble learning techniques to further enhance detection accuracy and adaptability.

VII. ACKNOWLEDGMENT

The successful completion of this project would not have been possible without the support and guidance of many individuals. We would like to express our sincere gratitude to our guide, Mr. B. Avinash, for his continuous support, valuable suggestions, and technical guidance throughout the project. We also thank the Head of the Department and all the faculty members of the Information Technology department for providing the necessary facilities, encouragement, and a supportive academic environment.

Finally, we extend our thanks to the VVIT management for providing the required infrastructure, lab resources, and internet facilities that helped us successfully complete this work.

REFERENCES

[1] A. K. Jain and B. B. Gupta, “Phishing detection: Analysis of visual similarity-based approaches,” Security and Communication Networks, vol. 10, no. 8, pp. 1448–1463, 2017, doi: 10.1002/sec.1457.

[2] R. Verma and A. Das, “What's in a URL: Fast feature extraction and malicious URL detection,” Proc. IEEE International Conference on Data Mining Workshops (ICDMW), 2017, pp. 914–921.

[3] S. Marchal, J. François, R. State, and T. Engel, “PhishStorm: Detecting phishing with streaming analytics,” IEEE Transactions on Network and Service Management, vol. 11, no. 4, pp. 458–471, 2014.

[4] M. Sahingoz, B. Buber, O. Demir, and B. Diri, “Machine learning-based phishing detection from URLs,” Expert Systems with Applications, vol. 117, pp. 345–357, 2019.

[5] J. Ma, L. Saul, S. Savage, and G. Voelker, “Beyond blacklists: Learning to detect malicious web sites from suspicious URLs,” Proc. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2009.

[6] H. Abutair and A. Belghith, “Using case-based reasoning for phishing detection,” Applied Soft Computing, vol. 13, no. 1, pp. 577–587, 2013.

[7] M. Mohammad, F. Thabtah, and L. McCluskey, “Predicting phishing websites based on self-structuring neural networks,” Neural Computing and Applications, vol. 25, pp. 443–458, 2014.

[8] Y. Rao and A. Pais, “Detection of phishing websites using machine learning approaches,” Procedia Computer Science, vol. 45, pp. 304–309, 2015.

[9] B. B. Gupta, N. A. G. Arachchilage, and K. E. Psannis, “Defending against phishing attacks: Taxonomy of methods, current issues and future directions,” Telecommunication Systems, vol. 67, no. 2, pp. 247–267, 2018.

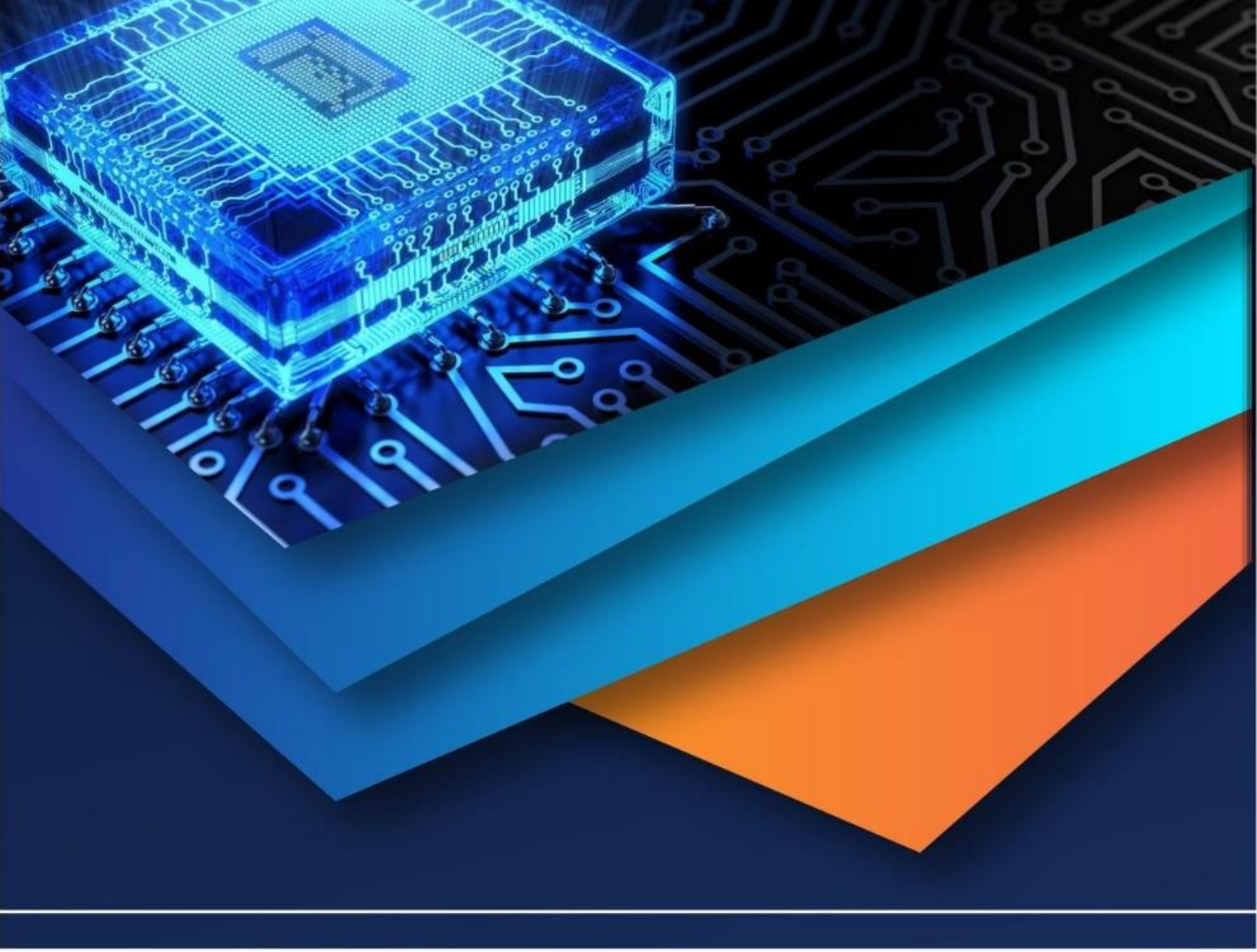
[10] S. Sahoo, B. Liu, and S. C. H. Hoi, “Malicious URL detection using machine learning: A survey,” ACM Computing Surveys, vol. 50, no. 1, pp. 1–36, 2017.

[11] A. Adebowale, K. Lwin, E. Sanchez, and M. Hossain, “Intelligent web phishing detection using machine learning,” Future Generation Computer Systems, vol. 108, pp. 425–435, 2020.

[12] UCI Machine Learning Repository, “Phishing Websites Dataset,” University of California, Irvine, 2019.

[13] A. Aljofey, Q. Jiang, H. Rasool, and X. Chen, “An effective detection approach for phishing websites using machine learning techniques,” IEEE Access, vol. 8, pp. 134–145, 2020.

[14] S. Feng, R. Banerjee, and Y. Choi, "Syntactic feature-based phishing detection using machine learning," Proc. IEEE International Conference on Communications (ICC), 2018, pp. 1–6, doi: 10.1109/ICC.2018.8422917.
[15] T. Ahmad and U. A. Khan, "Phishing detection using URL-based features and machine learning algorithms," IEEE Access, vol. 9, pp. 94752–94763, 2021, doi: 10.1109/ACCESS.2021.3094275.
[16] S. Singh and P. Kumar, "URL-based phishing detection using machine learning classifiers," Proc. International Conference on Computing, Communication, and Automation (ICCCA), 2020, pp. 1–5.
[17] A. Abdelhamid, F. Thabtah, and H. Abdel-jaber, "Phishing detection: A recent intelligent machine learning comparison-based study," IEEE Access, vol. 8, pp. 14110–14122, 2020, doi: 10.1109/ACCESS.2020.2965319.
[18] N. Chiew, E. Chang, and K. S. Tan, "Utilizing hybrid features for phishing website detection," Journal of Information Security and Applications, vol. 41, pp. 81–89, 2018.
[19] A. Mishra and R. Gupta, "An efficient phishing detection model using machine learning techniques," Procedia Computer Science, vol. 167, pp. 124–133, 2020.
[20] M. Al-Ahmadi and H. Alharbi, "Phishing website detection using ensemble machine learning algorithms," IEEE Access, vol. 9, pp. 150134–150146, 2021, doi: 10.1109/ACCESS.2021.3125984.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY