iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 Issue: XII Month of publication: December 2024

DOI: https://doi.org/10.22214/ijraset.2024.65888

Analysis of Machine Learning Techniques to Identify and Mitigate Phishing Websites

Soumya Ranjan Das[1], Ramireddy Nithileswar Reddy[2], Yogesh Kumar[3], Khudaija Nazhath
Computer Science Department, SVCE Bengaluru, Karnataka, India

Abstract: This paper focuses on understanding and solving the problem of phishing websites. Phishing is a type of cyber attack that creates fake websites and URLs that look like real people and tricks people into sharing sensitive information, such as passwords or financial details. These attacks target unsuspecting users and can have serious consequences. In this study, we discuss how AI can help identify and prevent phishing websites. We also explore different AI models that could create smarter, more effective anti-phishing systems in the future. While phishing is quite a common problem, no single method can address all the vulnerabilities. Instead, an amalgamation of techniques is often needed to combat different types of attacks. Machine learning, in particular, is a powerful tool that can mitigate phishing attempts and ensure online safety for everyone.

As technology advances, phishing approaches are becoming more sophisticated and often outpacing the potency of present anti-phishing tools, which often have limitations. This paper explores how to use machine learning controls to combat phishing. The goal is to create an efficient, accurate, and effective system. To achieve this, we use four machine learning models: K- Nearest Neighbors (KNN), Kernel-SVM, Random Forest Classifier, and Decision Tree. These models rely on labeled datasets to learn and improve their ability to detect phishing attempts. Each of these algorithms brings unique strengths to the classification process, helping to identify and prevent malicious activity. Machine learning allows cybersecurity systems to analyze patterns, adapt to new threats, and take proactive steps to stop future attacks. By integrating machine learning into security, we can take important steps to protect users from these attacks.

Keywords: Web Service, Cryptography, Web Security, Machine Learning, Classification, Clustering.

I. INTRODUCTION

Phishing is a sophisticated type of cyber-attack which has been present since 1996, starting with a Usenet newsgroup called AOHELL [1]. It involves tricking users into sharing personal information by using fake emails or websites designed to look trustworthy. The goal of phishing is to make the victim believe the message is legitimate—like an alert from their bank or receiving a message from a colleague—so they click the link or download an attachment.

In these attacks, the scammer pretends to be a reliable source, such as a person or company the victim knows or does business with. By using social engineering, they manipulate the victim into taking actions that benefit the attacker, often leading to financial loss or exposure of sensitive data [2].

As more of our personal information — like Aadhaar numbers, addresses, and phone numbers—is stored online, the risks increase. Attackers may target even more critical details like passwords or bank account credentials, leading to serious consequences. Thankfully, advancements in technology, such as machine learning, offer ways to detect phishing URLs and prevent these attacks [3]. Traditionally, machines operated based on instructions given by humans. Now, with machine learning, machines can analyze past data, learn from it, and make predictions much faster [4]. Machine learning uses tools and technology to make tasks simpler and more efficient, enabling systems to process data, identify patterns, and respond intelligently [5].

The process starts by training a machine using existing data. This involves feeding it information, refining its understanding, and creating a model trained using algorithms to predict outcomes [6]. Over time, as more data is added, these models improve, adapting to new challenges and becoming more accurate. By leveraging machine learning, we can develop systems that help detect phishing websites, offering a powerful tool to combat cyber-attacks and protect users [7].

II. BACKGROUND

Phishing is an online scam in which attackers deceit people into providing sensitive information like passwords, personal banking related details, and online credentials. These scams usually happen through emails that lure victims into clicking on malicious links or downloading harmful attachments [8].

At its core, phishing is a deceptive scheme where fake emails and websites are used as bait to steal personal information. The term "phishing" reflects this idea of "fishing" for data, with the "ph" coming from "phreaking." Phreaking, a term from the early days of hacking, refers to experimenting with and manipulating communication systems. Hackers, also called "phreaks," adopted the term to link phishing attacks to their underground community roots [9, 10].

The first known instance of phishing appeared on January 2, 1996, in a Usenet newspaper called "AOHell." During this time, AOL was the leading internet service provider, with millions of users logging in daily. Its popularity made it an ideal target for cybercriminals. Hackers and sellers of malicious software, known as the "warez community," began using AOL's platforms to carry out phishing attacks. They would impersonate AOL employees, sending fraudulent messages to trick users into verifying their accounts or payment information. Since phishing was a relatively new concept, many people unknowingly fell victim to these scams.

Over time, the attacks became more sophisticated. Hackers began using AIM accounts that were harder to track and bypassed AOL's security measures. Although AOL issued warnings to its users about these fraudulent activities, the attacks persisted and evolved.

By 2020, phishing scams had become even more advanced. Microsoft identified new techniques, such as linking emails to pirated Google search results that redirected users to malware-infected sites. Some scams used error pages (like 404 pages) as fake login portals for official websites, while others created realistic-looking login pages for services like Office 365 to deceive users. These tactics made phishing emails appear highly convincing, increasing the likelihood of people falling for them.

Phishing continues to grow more sophisticated, posing significant challenges for online security and highlighting the need for better awareness and defenses against these attacks.

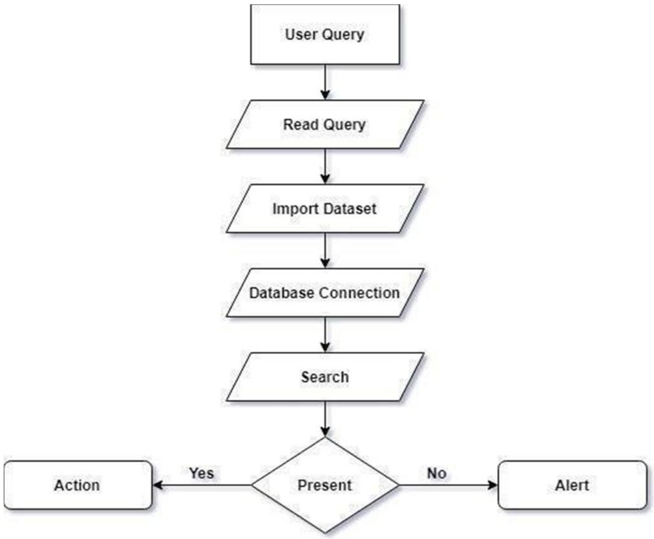


Fig.1. Data Flow Diagram

III. RELATED WORK

A phishing detection model is proposed that combines the best options with neural networks to increase accuracy. This method specifies a special metric called feature validity value, which contains the appropriate value. This metric helps to measure how certain factors affect the detection of phishing websites. Using this understanding, we developed an algorithm to find the most important features required to identify phishing websites.

To achieve this goal, the fuzzy set method is utilized to extract the most important features present in the dataset. These features are now further split into separate categories to determine whether a website is phishing or legitimate. Another method is called the "web crawler" based phishing scam detector, which is a three-layer detection model that evaluates network content, traffic data, and URL features. The system identifies phishing websites based on these techniques.

A dynamic detection system is also designed to match the constantly evolving nature of phishing threats. This system operates entirely on the client side, making it independent of third-party services and ensuring user privacy [11].

To better understand how the system works, Data Flow Diagrams (DFD) and Use-Case Diagrams are provided (Fig. 1). These diagrams illustrate the system's data flow and user/administrator interactions.

The diagram shows a visual representation of data moving through the system, showing each step of the process. It begins with the user submitting a query as input:

1) The system collects the input for processing.
2) A phishing URL dataset is loaded into the system's database.
3) A connection is established with the database to perform necessary operations.
4) The system checks key features of the input URL, such as its IP address and length.
5) If the features match known characteristics of phishing websites, the system alerts the user.

IV. DETECTION APPROACH

Phishing scams take advantage of inexperienced people and often exploit their dishonesty when dealing with online platforms such as email or websites. As these attacks continue, it becomes more difficult to find permanent solutions to stop them. The goal of all solutions is to reduce the impact and damage caused by phishing. Theoretically, there are two primary strategies that could be used to reduce phishing attacks: [12]

A. Challenges with these strategies

User Education: Many non-technical users struggle to learn about online threats or may forget important security practices over a period of time. Hence users need to be educated continuously so that they stay informed and vigilant.

Software Solutions: Although there are tools developed to protect users, such as authentication systems and security warnings, they rely on users following the instructions. If a user ignores security alerts, these tools lose their effectiveness.

B. Shortcomings of Existing Systems:

This section examines previous studies that utilized machine learning techniques to identify malicious phishing websites. A method explored by combining neural networks with custom selection. Important here is the introduction of a new measure, the “Feature Effectiveness Value”, to help quantify how different website features affect the detection of phishing sites. Using this index as a foundation, an algorithm was developed to pinpoint the most distinctive features for accurately identifying malicious websites.

To determine the most important ones, a technique called fuzzy rough set theory is used in various data processing. The classifier then performs feature selection to detect phishing websites. Another solution is a web browser-based phishing attack detector, which is a three-level testing model that evaluates web content, traffic, and URL features to classify a website as a phishing or legitimate site.

A detection system has been suggested that is more dynamic, one that adapts to the constantly changing nature of phishing websites. This system works directly on the user's device, eliminating the need for third-party support.

Additionally, a technique called Parse Tree validation has been proposed, which uses the Google API to analyze hyperlinks on a page. The system creates a parse tree, using a Depth-First Search algorithm to check if any of the links lead to suspicious content, helping to identify phishing websites more effectively.

V. REQUIRED MODEL

The project focuses on utilizing machine learning methods to detect phishing attempts. Supervised learning is a method of training models on recorded data to classify data or predict outcomes. In this case, the task is to classify a URL as phishing (malicious) or legitimate. The algorithms used include:

A. Kernel Support Vector Machine (SVM)

Kernel SVM is a robust classification algorithm. It works by mapping features onto an n- dimensional space and identifying a hyperplane (or boundary) that best separates the given data into distinct categories. The objective is the maximization of the margin between the two groups, ensuring a clear distinction. Mathematically, the decision function for SVM is defined as: $F(x,w,b)=\mathrm{sgn}((w\cdot xi)+b)$

Here, xi represents the input data, w is the hyperplane's direction, and b is the threshold. The algorithm effectively classifies data based on these principles.

B. Decision Tree

Decision trees are intuitive and easy-to-use tools for classification. They mimic human decision-making by splitting data into branches based on specific conditions or features. At each step, the tree determines the most relevant feature and condition to make accurate predictions. Their straightforward nature makes them a popular choice for analyzing and classifying datasets.

C. Random Forest Classifier

The random forest algorithm generates an ensemble of decision trees and combines their results to increase accuracy. It is like consulting many experts before making a final decision. This integration method is particularly helpful for big data because it reduces the risk of making mistakes from a single tree and leads to better results.

D. K-Nearest Neighbor (KNN)

KNN is a yet another effective algorithm which classifies data based on its proximity to other data points. It groups similar items by evaluating their distance, often using Euclidean distance, and assigns the class based on the majority class of its nearest neighbors.
$d(a,b)= \sqrt{\sum}(bi-ai)^2$
Here, a and b are data points, and d is the distance between them. KNN is particularly useful when data naturally forms clusters or groups.

VI. DETECTION PROCESS

To develop a machine learning-based solution for detecting phishing threats, the system must be trained on datasets that include both phishing and legitimate domains. This training data helps the system learn patterns and make accurate predictions. Once collected, the data is processed and prepared for analysis to ensure it can be effectively utilized by the model.

A. Decision Tree Algorithm

A decision tree works like a flowchart, where each step involves testing a specific feature to make a decision. In this case, the dataset includes classes such as phishing and legitimate domains, along with features like:

1) Domain name
2) Page title
3) Subdomain name
4) URL length
5) Number of digits in the URL
6) Occurrence of "www" in the URL
7) Frequency of specific keywords
8) Second-level domain details

The decision tree selects the most relevant features based on information gain; a metric that measures how well a feature separates the dataset into distinct classes. Information gain is calculated using this formula:
$Gain(S,A)=Entropy(S)-\sum |Sv|/|S| Entropy(Sv)$
Where:

- S is the dataset.
- A is the feature being evaluated.
- Sv represents the subset of data where A has a specific value v.

Entropy, which determines the impurity or randomness in the data, is calculated as:
$H(S) = -\sum pi \log2 pi$
The tree first selects the feature that has the highest data as the root and divides the data further into smaller groups using the next feature. As the tree grows, the data becomes “purer,” meaning that each leaf has a higher probability of representing a cluster. Once the tree is built, the training process is complete. Well- designed decision trees trained on many different datasets can achieve good results and work well in real-world situations.

B. Support Vector Machine (SVM)

Support Vector Machine (SVM) is another popular learning algorithm used to identify phishing websites. It classifies the URL as phishing or legitimate in two steps:

1) Training Phase: During this phase, the SVM model learns to distinguish between phishing and legitimate URLs using the training dataset.
2) Testing Phase: In this phase, the model tests its predictions on unseen data by analyzing features extracted from test URLs.

SVM finds the best possible "hyperplane," or boundary, that separates the two classes, ensuring accurate classification. This makes it an effective tool to pinpoint malicious URLs.

C. System Architecture

The system combines both existing and proposed detection methods, with the following key components:

1) *Feature Extractor*: This module extracts relevant details from phishing websites, such as keywords, URL structure, and page content. These features are converted into numerical representations, or vectors, which serve as input for machine learning algorithms.
2) *Classifier:* Algorithms like Decision Tree and SVM analyze the extracted features to classify URLs as phishing or legitimate. By training the system on diverse datasets and incorporating robust models, the solution can achieve high accuracy and reliability in identifying phishing threats.

By leveraging an ensemble of decision trees, SVMs, and comprehensive feature extraction, this system offers a powerful approach to safeguarding users from phishing attacks. The components ofthese system are briefly described in Fig. 2:

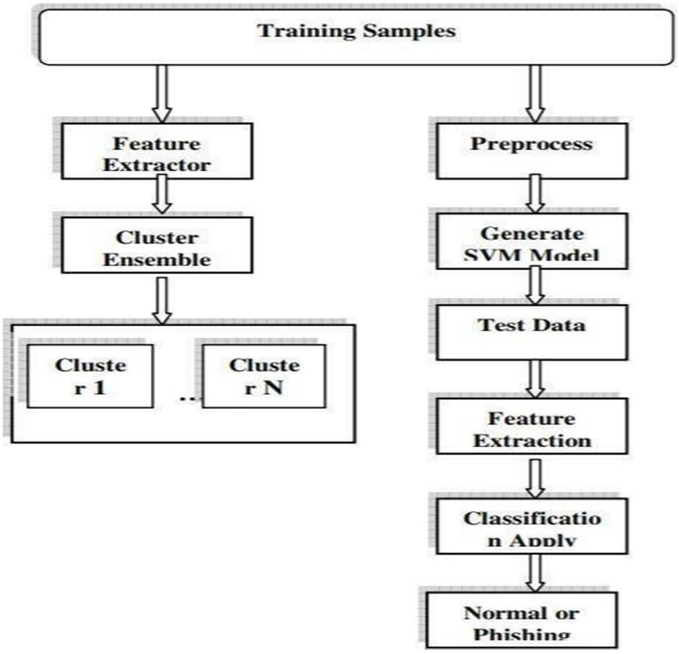


Fig. 2. System Architecture

3) *Support Vector Machine (SVM)*: Support Vector Machine (SVM) is a popular supervised learning algorithm which classifies URLs as safe or malicious. The process includes:

a) *Training Phase:*
- The SVM trained model is built by making use of a training dataset, teaching it how to classify URLs.
- Kernel functions (such as linear, polynomial, or sigmoid) are used for mapping the data to a further higher-dimensional space, making classification more precise.

b) *Testing Phase:*
- Test data, including URLs, is analyzed to extract relevant features.
- These features are categorized into two groups: legitimate websites or phishing websites depending on the trained model.

4) *Preprocessing :* This step ensures the data is clean and usable:

- Records with missing or incomplete values are removed.
- The remaining data is processed and prepared for training, enabling the system to work effectively with reliable inputs.

5) *Random Forest Classifier:* The Random Forest Classifier is a powerful algorithm that detects phishing websites by leveraging key features of each URL. Here's how it works:

a) *Data Preparation:*

- Input data is categorized into training and testing sets:
  - ➢ Training input and output for teaching the classifier.
  - ➢ Testing input and output for evaluating its performance.

b) *Training:*

- The classifier utilizes training data to learn patterns and make predictions based on the selected features.

6) *Testing and Evaluation:*

- The trained classifier analyzes the testing data and predicts whether a URL is phishing or legitimate.
- Performance metrics including precision, recall, accuracy and F-score are employed to assess its effectiveness.

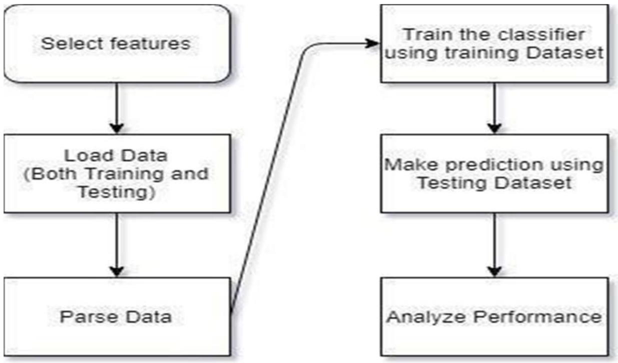

Select features
Load Data (Both Training and Testing)
Parse Data
Train the classifier using training Dataset
Make prediction using Testing Dataset
Analyze Performance

Fig. 3. Processed flowchart for each selected feature set.

VII. EXPERIMENTAL RESULT

The effectiveness, accuracy, and performance of the models mentioned earlier are evaluated using specific mathematical metrics. These metrics rely on four key factors: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) scores. Here's a simplified explanation of how each measure contributes:

1) *Precision*

$$\text{Precision} = TP / TP + FP$$

Precision tells us how many of the URLs flagged as phishing are actually phishing URLs. It prioritizes the system's accuracy in identifying phishing websites among those flagged.

2) *Recall*

$$\text{Recall} = TP / TP + FN$$

Recall measures the system's ability to correctly identify phishing URLs. It calculates the ratio of correctly identified phishing URLs to the total number of actual phishing URLs in the dataset.

3) *F1 Score*

F1 Score = 2 × Precision × Recall / Precision + Recall The F1 Score balances Recall as well as Precision, combining them into one metric. It is also useful when
there's an imbalance between correctly flagged phishing
URLs and legitimate ones that were incorrectly flagged.

4) *Accuracy*

Accuracy = TP+TN / TP+TN+FP+FN

Accuracy computes the overall success of the system, showing the percentage of correct predictions (both phishing and legitimate) out of all tested URLs.

Definitions:

- True Positive (TP): Phishing websites correctly identified as phishing.
- False Positive (FP): Legitimate websites mistakenly flagged as phishing.
- True Negative (TN): Legitimate websites correctly classified as legitimate.
- False Negative (FN): Phishing websites incorrectly classified as legitimate.

These metrics together provide a better overview of the model's performance, helping to refine the system to better detect phishing threats while minimizing errors.

VIII. CONCLUSION

This paper focuses on building quality, accuracy, and cost- effectiveness. To reach this goal, four supervised machine learning models were used: K-Nearest Neighbors (KNN), Nuclear Support Vector Machine (SVM), Random Forest classifiers and Decision Trees. Each model is carefully analyzed and compared, highlighting its strengths, weaknesses, and overall performance.

REFERENCES

[1] Lokesh Harinahalli, G., and Gowda, B., "Detection of phishing websites using an advanced machine learning methodology," Journal of Cyber Security Technology, pp. 1- 14, 2020.

[2] Mahajan, R., and Siddavatam, I., "Utilizing machine learning techniques for identifying phishing websites," International Journal of Computer Applications, vol. 181, no. 23, pp. 45-47, 2018.

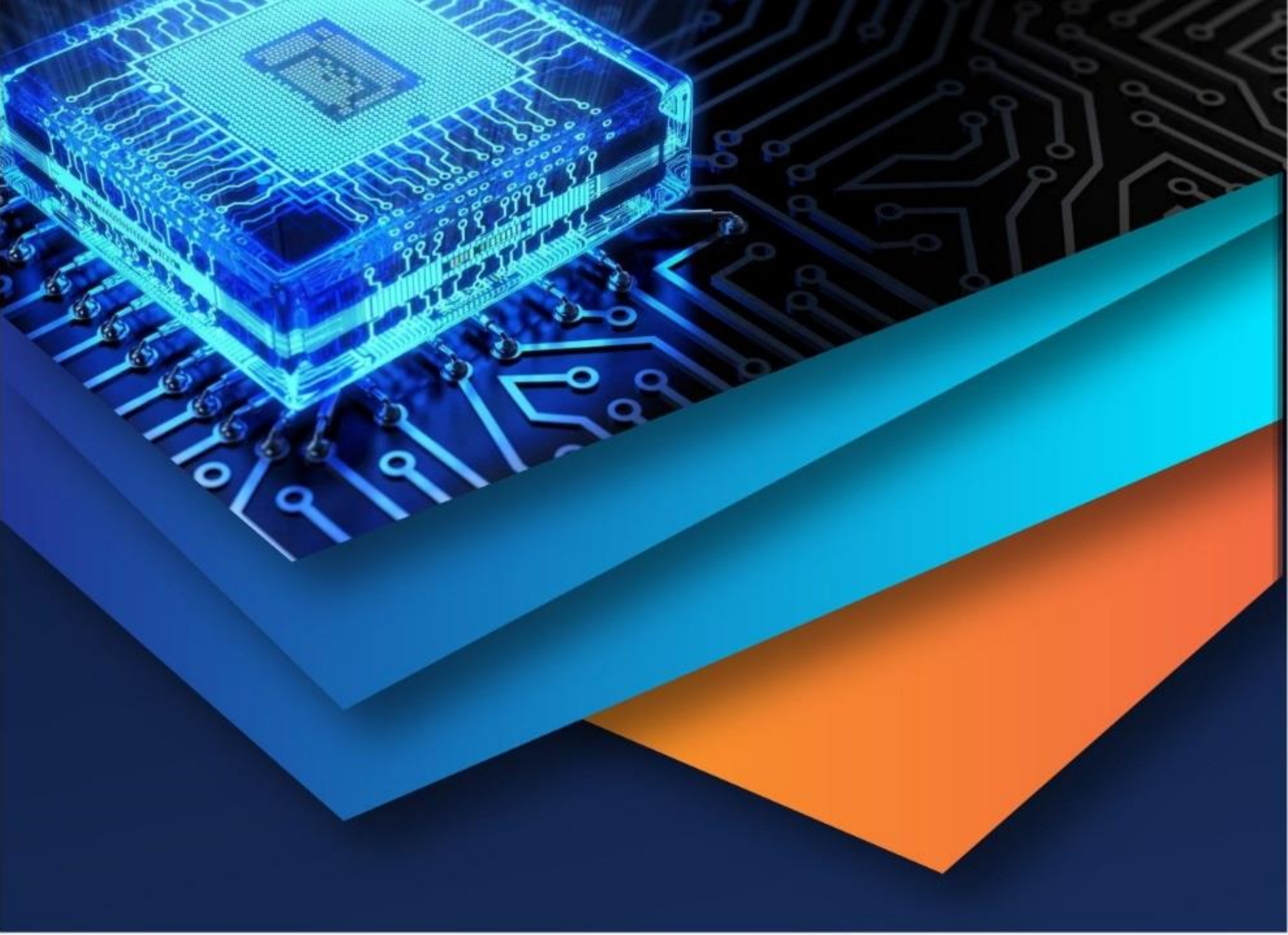
[3] MIT Technology Review, "An introduction to machine learning: A visual guide." Available at: https://www.technologyreview.com/2018/11/17/103781/wha t-is-machine-learning-we-drew-you-another-flowchart/, 2020.

[4] Odeh, A., Keshta, I., and Abdelfattah, E., "Multilayer perceptron-based efficient phishing website detection," International Journal of Interactive Mobile Technologies (iJIM, vol. 14, no. 11, p. 22, 2020.

[5] Ahmed, K., and Naaz, S., "A machine learning-based solution for phishing website detection," SSRN Electronic Journal, 2019.

[6] Altaher, A., "Classifying phishing websites using a hybrid SVM and KNN model," International Journal of Advanced Computer Science and Applications, vol. 8, no. 6, 2019.

[7] HR, M., MV, A., S, G., and S, V., "Building an anti- phishing browser utilizing a random forest framework and rule extraction," Cybersecurity, vol. 3, no. 1, 2020.



10.22214/IJRASET

INDEX COPERNICUS
45.98

IMPACT FACTOR:
7.129

TOGETHER WE REACH THE GOAL
IMPACT FACTOR:
7.429