IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 11, April 2025
Impact Factor: 7.67

Detecting Phishing Attacks using NLP and Machine Learning

Priyanshu Bhandari[1], Vastav Taneja[2], Satyansh Rai[3] , Ashima Mehta[4]
Students, Department of Computer Science and Engineering[1,2,3]
Professor, Department of Computer Science and Engineering[4]
Dronacharya College of Engineering, Gurugram, India

Abstract: *Phishing emails remain one of the most pervasive threats to cybersecurity, often leading to unauthorized access, data breaches, and financial losses. Due to their deceptive nature and close resemblance to legitimate communication, identifying phishing attempts automatically poses a significant challenge. This study presents a machine learning-based approach to detect phishing attacks by analyzing various features of emails, including text content, subject lines, and metadata. Using Natural Language Processing (NLP) techniques and neural network models, each component of an email is independently processed to capture linguistic and structural patterns commonly found in phishing messages. An ensemble of machine learning algorithms is then employed to combine these insights and classify emails as phishing or legitimate. This multi-faceted framework emphasizes the importance of analyzing composite email features for robust phishing detection and provides a foundation for real-world deployment of intelligent anti-phishing systems*

Keywords: Machine Learning, Natural Language Processing (NLP), Phishing Detection, Cybersecurity, Spam Filtering

I. INTRODUCTION

Phishing is a form of cyberattack in which malicious actors deceive users into revealing sensitive informationsuch as login credentials, credit card numbers, or personal detailsby impersonating trusted entities through emails, websites, or messages. These attacks exploit human trust and are often crafted to appear indistinguishable fromlegitimate communication. The ultimate goal of phishing is to gain unauthorized access to systems, commit fraud, or distribute malware.

Over the years, phishing has evolved in both scale and sophistication. Attackers now employ advanced social engineering tactics and language manipulation to craft emails that can bypass traditional filters and deceive even cautious users. As a result, phishing remains one of the most successful and prevalent methods of cyber intrusion, posing significant risks to individuals, organizations, and governments alike.

Machine learning emerges as a promising solution in the ongoing battle against such deceptive attacks. Traditional rule-based approaches struggle to keep up with the continuously evolving strategies employed by attackers, often proving ineffective in real-time threat scenarios. In contrast, machine learning provides a proactive defense mechanism by learning from vast datasets, discerning patterns, and adapting to new phishing strategies, offering a dynamic shield against future threats.

Numerous studies have delved into employing machine learning for phishing detection, utilizing predictive analytics and Natural Language Processing (NLP) to enhance the accuracy of these systems. However, the rapid evolution of phishing tactics necessitates ongoing exploration and refinement of state-of-the-art machine learning methods. By continuously updating and learning from diverse data sources, these intelligent systems can offer scalable, real-time, and robust phishing detection mechanisms.

This research seeks to address existing challenges by thoroughly analyzing modern machine learning models and methods used in detecting phishing websites. It focuses on evaluating the advantages, limitations, and practical relevance of each approach to extract meaningful insights that can guide future advancements and real-world

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 11, April 2025
Impact Factor: 7.67

implementations. By incorporating cutting-edge machine learning and NLP techniques, the objective is to develop resilient detection systems capable of not only identifying phishing attempts but also predicting and mitigating them proactively, ultimately contributing to a more secure and trustworthy online environment.

II. LITERATURE REVIEW

Historically, research on phishing detection has primarily centered around developing automated techniques to identify phishing attempts. This section reviews related work, highlighting various approaches to tackling phishing detection. It starts by briefly tracing the evolution of phishing and outlines the most widely used detection strategies. Over time, researchers have adopted diverse methodologies, some emphasizing machine learning models, while others have explored manual tools and natural language processing techniques applied to email content.

Origin and Types of Phishing

According to Cybersecurity & Infrastructure Security Agency (CISA), phishing is "a form of social engineering in which attackers deceive individuals into providing sensitive information by masquerading as a trustworthy entity in digital communication."

Type of Phishing	Description
Email Phishing	Attackers send fake emails posing as legitimate sources to trick users into revealing sensitive information.
Spear Phishing	A highly targeted phishing attack tailored to a specific individual or organization.
Whaling	Spear phishing aimed at high-profile individuals like CEOs or government officials.
Smishing	Phishing via SMS messages that include malicious links or deceptive alerts.
Vishing	Fraudulent phone calls attempting to extract sensitive information under false pretenses.
Pharming	Redirects users from real websites to malicious ones without their knowledge.
Clone Phishing	A copy of a previously sent legitimate email is modified with malicious links or attachments.
Malvertising	Involves placing malicious ads on legitimate websites that can infect users' systems or redirect them to phishing sites.
Man-in-the-Middle (MITM)	Attackers intercept communication between two parties to eavesdrop or alter information.
Malware Phishing	Phishing emails or links that trick users into downloading malware, which then steals data or gives control to attackers.
Business Email Compromise (BEC)	Involves compromising a business email to manipulate employees into transferring money or data.

The term “phishing” originated around 1996 and is believed to have been coined by hackers attempting to steal AOL (America Online) accounts. The term is a play on the word "fishing", symbolizing the use of bait (fraudulent emails or messages) to lure victims into giving up sensitive information. The "ph" is thought to be inspired by early hacker culture, referencing "phreaking" a form of hacking that involved manipulating telephone systems.

Attackers would send messages impersonating AOL staff, asking users to verify their login credentials for security purposes. Unsuspecting users who complied were tricked into giving up their account access. These techniques laid the groundwork for what is now known as phishing, a broad term encompassing various online fraud tactics aimed at stealing sensitive data through deception.

Humans remain the most vulnerable element in phishing attacks, as they can be easily tricked into revealing sensitive information or clicking on harmful links through social engineering tactics.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 11, April 2025
Impact Factor: 7.67

III. RELATED MODELS AND METHODS

Phishing is a cybercrime where attackers impersonate legitimate websites or trusted entities to deceive users into revealing sensitive information such as usernames, passwords, or financial data. These malicious websites are carefully crafted to appear authentic, employing techniques like URL manipulation, social engineering, and email spoofing to mislead unsuspecting users. The consequences of phishing attacks can be severe, including financial loss, identity theft, and compromised security. Over time, the field of phishing detection has seen significant progress, particularly with the adoption of machine learning (ML) techniques to enhance detection accuracy and efficiency. These algorithms analyze vast datasets to identify patterns and features that distinguish legitimate websites from fraudulent ones.

This research explores a range of machine learning models and techniques commonly used in phishing detection. By studying these approaches, the goal is to advance our understanding of effective countermeasures against phishing threats, contributing to stronger cybersecurity frameworks for individuals and organizations alike. As phishing strategies continue to evolve, so must the tools we use to combat them. Machine learning, with its adaptability and predictive capabilities, offers a powerful solution in the fight against phishing.

To further enhance anti-phishing systems, machine learning-based methods can be optimized through the careful selection of feature vectors derived from online elements such as URLs, webpage structures, and behavioral attributes.

This study implements a modular, component-based framework where these features are fed into predictive models using algorithms like Support Vector Machine (SVM) and Naïve Bayes (NB). These models are trained on both phishing and benign datasets to evaluate their effectiveness, demonstrating strong potential in accurately identifying phishing threats.

One of the primary challenges faced by many anti-phishing systems, as noted by researchers, lies in the significant computational overhead, vulnerability to zero-day attacks, and high false positive or false negative rates. Although various machine learning models have demonstrated promising accuracy in detecting phishing websites, their overall effectiveness often hinges on the selection and performance of the feature vectors used. To overcome these limitations, the development of an optimized feature selection mechanism becomes essential. Such a module should focus on extracting the most relevant features from URL structures, webpage attributes, and behavioral indicatorsensuring that the final set of features provided to the predictive model is both efficient and impactful.

While the methodology and results presented in the study are comprehensive, there is a noticeable gap regarding explicit details on the computational cost of the proposed approach. Computational overhead plays a critical role in the deployment of phishing detection models, particularly in environments with limited resources or where real-time detection is necessary. For a thorough evaluation of a model's practical applicability, factors such as model size, inference time, and memory usage during training and deployment must be considered. Assessing these aspects is key to understanding the feasibility of implementing the model in real-world cybersecurity systems.

Logistic Regression

The logistic regression model estimates the probability of an event (in this case, phishing) using the logistic function, which maps the linear combination of input features (x) to a probability (p).

$$p(x) = \frac{1}{1 + e^{-(x-u)/s}}$$

Where

u is location parameter (Midpoint curve p(u) = ½)

s is scale parameter for sigmoid function is also called as activation function for logistic regression.

$$p(x) = \frac{1}{1 + e^{-x}}$$

Where

e = base of natural algorithms.

The equation below represents logistic regression.

$$y = \frac{e^{(b_0 + b_1 x)}}{1 + e^{(b_0 + b_1 x)}}$$

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 11, April 2025
Impact Factor: 7.67

Where
x = input value, y = predicted output, b_0 = intercept term , b_1 = coefficient for input x.

Nearest Neighbors

KNN calculates distances between the target sample and its neighboring samples and assigns the most prevalent class label among the K nearest neighbors.

Given two feature vectors with numeric value

$A=(a_1,a_2,...,a_n)$ and $B=(b_1,b_2,...,b_n)$

Below is the distance measure (Euclidean Distance) formula where R_i is the range of the ith Component:

$$d = \sqrt{\sum_{i=1}^{n} \frac{(a_i - b_i)^2}{R_i^2}} = \sqrt{\frac{(a_1 - b_1)^2}{R_1^2} + \frac{(a_2 - b_2)^2}{R_2^2} + \cdots + \frac{(a_n - b_n)^2}{R_n^2}}$$

Decision Tree

Decision tree algorithms construct a tree-like model to classify data by making a series of hierarchical decisions. In this structure, internal nodes correspond to decisions based on specific features, and leaf nodes indicate the final class labels. These models are easy to understand and interpret, and they can effectively manage both categorical and numerical data. Despite their strengths, decision trees are susceptible to overfitting and may struggle to capture complex patterns in the data.

The feature space is divided through successive decisions made at each level of the tree, and classification is achieved by following a path from the root to a leaf node, guided by the input feature values. This study utilizes the ID3 algorithm (an enhancement of D3) for tree construction. ID3 determines whether a node should become a leaf based on entropy: if the entropy is zero, the node is considered pure and becomes a leaf; if it's greater than zero, further splitting

is required. Entropy measures the uncertainty or impurity in the data and is calculated using logarithmic functions, making it computationally intensive.

$$\text{Entropy} = \sum_{i=1}^{C} -p_i * \log_2(p_i)$$

Where p_i represents relative frequency and c represents number of classes.

Information gain is another attribute used to measure how well a attribute divides the training example according to target classification. Developing a decision tree is all about identifying an attribute that returns high information gain and low entropy.

IG (T/X) = Entropy(T) − Entropy (T,X)

Gini index is used which is the formula used to decide how nodes on a Decision tree branch.

$$\text{Gini} = 1 - \sum_{i=1}^{C} (p_i)^2$$

Where
p_i represents relative frequency and c represents number of classes.

Random Forest

Random Forest is an ensemble-based technique that builds a collection of decision trees, where each tree is trained on a randomly selected subset of the dataset and feature set. The final prediction is made by combining the outputs of these individual trees, typically through majority voting for classification tasks. This method reduces the risk of overfitting and enhances the model's ability to generalize to new data. It is effective in dealing with noisy inputs and performs well even with large feature spaces. However, it can be resource-intensive in terms of computation. In regression tasks,

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 11, April 2025 Impact Factor: 7.67

Random Forest leverages the mean squared error (MSE) to evaluate the quality of splits, helping determine the most optimal branching at each node for better predictive performance.

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(f_i - y_i)^2$$

Where

N = datapoints

f_i= Value returned by model

y_i= Actual Value of Data point

When RF performed on classification data Gini index is used which is the formula used to decide how nodes on a Decision tree branch.

$$\text{Gini} = 1 - \sum_{i=1}^{C}(p_i)^2$$

Where

p_i represents relative frequency and c represents number of classes.

Entropy avails the probability of definite outcome in order to make decision on how the node should branch. Entropy is mathematically intensive due to log functions used in it.

$$\text{Entropy} = \sum_{i=1}^{C} -p_i * \log_2(p_i)$$

Support Vector Classifier

The Support Vector Classifier (SVC) is a binary classification model that works by identifying a hyperplane that best separates two classes with the widest possible margin. It transforms data into a higher-dimensional space to make it more separable and determines the optimal hyperplane in that space. SVC is well-suited for high-dimensional datasets and can model complex patterns using various kernel functions. However, its performance can be highly dependent on the choice of hyperparameters, and training can be time-consuming, especially on large datasets.

The SVC finds the optimal hyperplane that maximizes the margin between the two classes. The decision function for classification is given by:

$f(\text{x}) = \text{sgn}(\text{w}^T\text{x} + b)$

Where

$\text{w} = \sum \alpha\text{i}\ \text{y}_\text{i}\ \text{x}_\text{i}$, αi is zero for all cases and $\text{y}_\text{i} \in \{1,-1\}$ are the labels.

Linear Support Vector Classifier

Linear SVC is a specialized form of Support Vector Classifier that employs a linear kernel, making it ideal for datasets that are linearly separable. It is computationally efficient, scales effectively to large datasets, and tends to generalize well. However, it may struggle with data that is not linearly separable. Compared to standard SVC, Linear SVC typically converges faster on large datasets. While SVC minimizes the regular hinge loss, Linear SVC minimizes the squared hinge loss. Additionally, Linear SVC uses a one-vs-rest strategy for multiclass classification, whereas SVC adopts a one-vs-one approach.

Naïve Bayes

Naïve Bayes is a probabilistic classification algorithm based on Bayes' theorem, which assumes that features are conditionally independent given the class label. It computes the posterior probability of a class by combining prior knowledge with the likelihood of observed features. Naïve Bayes is known for its computational efficiency, minimal training data requirements, and ability to handle high-dimensional feature spaces. However, its strong independence assumption can lead to oversimplification of feature relationships.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 11, April 2025
Impact Factor: 7.67

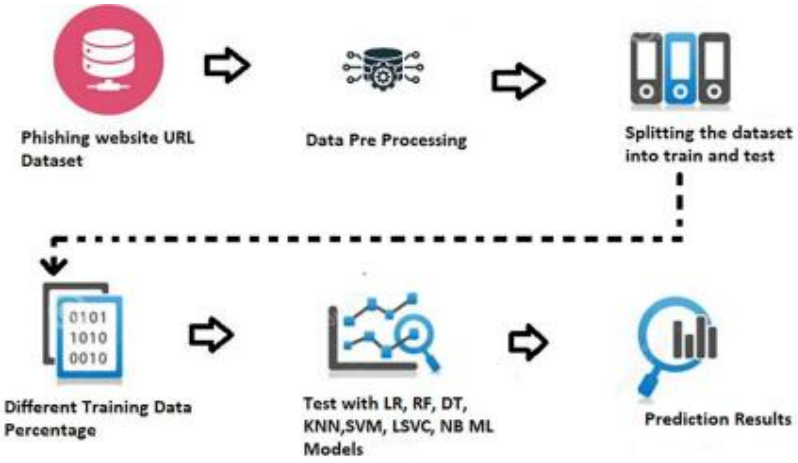
In the context of phishing website detection, various machine learning models have been explored with differing levels of success. For instance, Machan applied logistic regression using URL-based features to identify phishing sites. Other studies have utilized algorithms like K-Nearest Neighbors (KNN) and decision trees to analyze website characteristics. Random Forests have also been employed to detect phishing by leveraging both website content and URL features. The literature underscores a diverse range of models, each with its own strengths, limitations, and supporting research. Naïve Bayes, in particular, is valued for its simplicity and effectiveness on large datasets. Despite its simplicity, it can outperform more complex classification methods, as it leverages Bayes' theorem to estimate the posterior probability P(c|X) using the prior P(c), likelihood P(X|c), and evidence P(X).

$$P(c \mid \mathrm{X}) = \frac{P(X \mid c) * P(c)}{P(\mathrm{X})}$$

Here P(c) and P(X) are prior probability of class and predictor, respectively.
P(X/c) is Probability of predictor given class.
P(c/X) is posterior probability of class C given predictor (x, attribute).

IV. ARCHITECTURE OF ML MODELS TESTING AND METHODOLOGY

The first step in this research involves collecting a labeled dataset of phishing aThe first step in this research involves collecting a labeled dataset of phishing and legitimate website URLs. The dataset, sourced from Kaggle, includes over 500,000 entries around 150,000 phishing (fake) URLs and 360,000 legitimate ones. It consists of two columns: the URL and its label (Fake or Legitimate).



Data Pre-Processing

At this stage, the dataset is tested for issues like imbalance, duplicates, and null values, which are cleaned if found. The data is then shuffled and a representative sample is selected to match the overall distribution. If needed, the dataset is updated and split into features and labels for further processing.

Split Data into Train & Test Datasets

In this phase, the data is split into training and testing sets. The training set is used to train the ML models, while the test set evaluates their performance. Both are subsets of the main dataset. The test set should be representative and large enough for meaningful results, while the training set must be sufficiently large to ensure effective model learning without data leakage.

Import and Initialize Models

At this stage, machine learning models such as Logistic Regression, Random Forest, Decision Tree, KNN, SVM, Linear SVC, and Naïve Bayes are imported and initialized using the sklearn library. Below is a sample code snippet for importing these models:

```
from sklearn.ensemble import RandomForestClassifier
```

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 11, April 2025
Impact Factor: 7.67

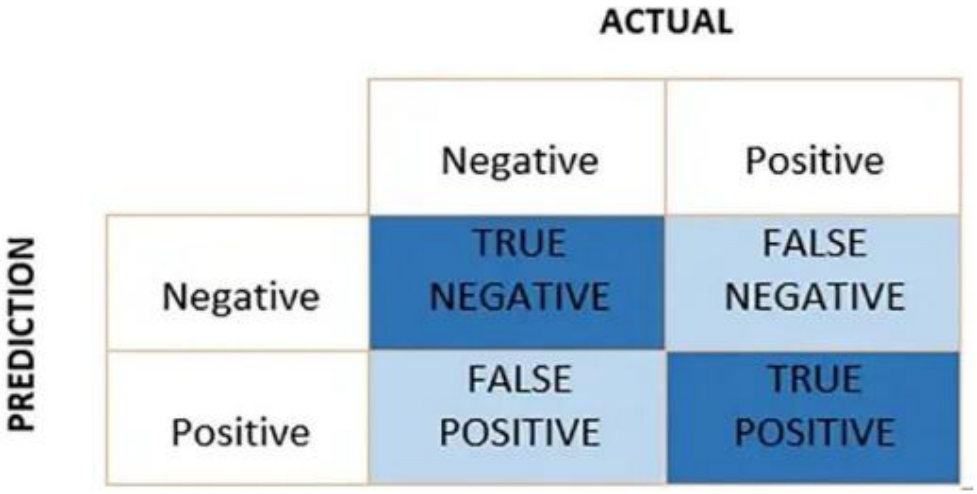
```
from sklearn.svm import SVC, LinearSVC
from sklearn.naive_bayes import MultinomialNB
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
```

Train ML Model with Train Data

After importing the ML models, they are trained using various training percentages. It's important to ensure the training data is large, diverse, and distinct from the test data. A well-sized and representative training set improves model accuracy and should include a wide range of phishing URL patterns for better prediction performance.

Test ML Models with Test Data and Using Different Training Percentage

In the final stage, the experiment is tested using a test dataset and various training percentages. The results will include the model's accuracy, precision, and recall, along with a confusion matrix generated through Python. This stage evaluates the ML models' performance to ensure optimal results. Testing requires large datasets and long training cycles. If the test results are unsatisfactory, the models return to training with additional phishing site data. This phase helps the organization select the best models to combat phishing, a critical issue for both employees and the organization.



$$\text{Accuracy} = \frac{\text{True Positive} + \text{True Negative}}{\text{True Positive} + \text{False Positive} + \text{True Negative} + \text{FalseNegative}}$$

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$$

V. METHODOLOGY

We recognized the importance of feature engineering and extracted a mix of structural, lexical, and aggregated features from URLs to evaluate prediction models using both qualitative and quantitative methods. The dataset was preprocessed to remove redundant or missing values, and then split using stratified sampling to ensure both the training and test sets reflected the same distribution of target classes.

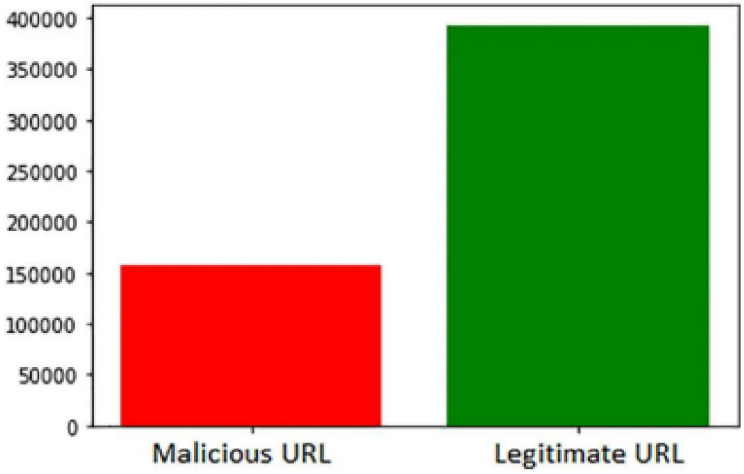
To optimize model performance, hyperparameter tuning was performed for each model. For logistic regression, we tuned regularization strength and type, for KNN, the number of neighbors and distance metric, and for decision trees, the tree depth, minimum samples to split, and the criterion (Gini or Entropy). Models were evaluated using accuracy,

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 11, April 2025
Impact Factor: 7.67

precision, recall, F1 score, and AUC to assess performance comprehensively. After training, an ensemble approach was explored to combine the strengths of individual models, aiming to improve prediction accuracy and reduce overfitting. These methodologies offer valuable insights into the models' ability to classify phishing websites and provide a foundation for practical cybersecurity applications.

VI. DATA ANALYSIS AND EXPERIMENTAL RESULTS

Phishing URL detection involves identifying and classifying URLs that deceive users to steal sensitive information. With phishing attacks becoming more sophisticated, detecting these URLs is vital to prevent harm to individuals, businesses, and organizations. Various techniques, including machine learning, natural language processing, and behavioral analysis, are used to analyze URL characteristics and identify malicious intent. Effective detection helps protect users, safeguard privacy, and maintain the security of online platforms.



Below Fig. 4 is a cloud of words used in legitimate and fake website URLs, with the respective word count represented by color bars. The red bar represents the number of words in malicious URLs, totaling 150,000, while the green bar represents the word count in legitimate URLs, which stands at 400,000.

The figure displays the domains of both fake and genuine websites. Legitimate website domains typically include terms such as "tryout,""media,""edu,""eira,""ye,""youthleaguesusa,""fmesxcc,""nvinip,""salesecureweb,""index,""html," and "end." In contrast, fake website domains often feature words like "paypal,""skype,""bin,""login,""cd,""org,""webscrc,""widescreen,""mail,""nobel," and "icloud."

Table 1 shows the accuracy of various machine learning models at different training percentages. The models listed include Logistic Regression, Random Forest, Decision Tree, K-Nearest Neighbors (KNN), Support Vector Machine

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 11, April 2025
Impact Factor: 7.67

(SVM), Linear Support Vector Classifier (LSVC), and Naive Bayes (NB). Each row represents a different training percentage, ranging from 10% to 90%, and each column corresponds to a specific model.

Table 1: Accuracies of different machine learning models

Training %	Logistic regression	Random forest	Decision tree	KNN	SVM	LSVC	NB
10	0.83	0.83	0.82	0.74	0.78	0.87	0.86
20	0.86	0.87	0.85	0.73	0.83	0.88	0.88
30	0.88	0.88	0.85	0.73	0.85	0.90	0.88
40	0.88	0.88	0.85	0.74	0.86	0.90	0.89
50	0.89	0.89	0.86	0.75	0.86	0.91	0.90
60	0.90	0.89	0.87	0.75	0.87	0.91	0.90
70	0.90	0.90	0.87	0.76	0.88	0.91	0.91
80	0.90	0.90	0.87	0.74	0.88	0.91	0.91
90	0.90	0.89	0.88	0.74	0.87	0.90	0.90

Table 2 presents the precision of various machine learning models at different training percentages. Models include Logistic Regression, Random Forest, Decision Tree, K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Linear Support Vector Classifier (LSVC), and Naive Bayes (NB). Each row represents a training percentage from 10% to 90%, and each column corresponds to a model. As the training percentage increases, most models show improved precision, with Logistic Regression, Random Forest, KNN, SVM, and LSVC achieving high precision at 90%. Decision Tree and NB also show relatively high precision, though slightly lower. However, precision alone may not fully evaluate model performance, and other metrics like recall, F1 score, and accuracy should also be considered.

Table 2: Precisions of different machine learning models

Training %	Logistic regression	Random forest	Decision tree	KNN	SVM	LSVC	NB
10	0.91	0.89	0.78	0.56	0.95	0.91	0.96
20	0.91	0.91	0.84	0.53	0.92	0.89	0.94
30	0.92	0.93	0.82	0.52	0.92	0.90	0.93
40	0.90	0.92	0.81	0.54	0.89	0.89	0.91
50	0.89	0.94	0.81	0.56	0.89	0.90	0.92
60	0.92	0.93	0.82	0.56	0.94	0.91	0.92
70	0.93	0.93	0.83	0.59	0.94	0.91	0.90
80	0.92	0.93	0.82	0.55	0.93	0.90	0.89
90	0.93	0.94	0.85	0.54	0.92	0.92	0.89

Table 3 shows the recall of different machine learning models at varying training percentages. The models listed include Logistic Regression, Random Forest, Decision Tree, K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Linear Support Vector Machine (LSVM), and Naive Bayes (NB). Each row represents a different training percentage from 10% to 90%, and each column corresponds to a specific model.

Table 3: Recalls of different machine learning models

Training %	Logistic regression	Random forest	Decision tree	KNN	SVM	LSVM	NB
10	0.45	0.49	0.56	0.46	0.26	0.62	0.52
20	0.58	0.60	0.60	0.53	0.45	0.67	0.60
30	0.64	0.64	0.61	0.59	0.51	0.70	0.65
40	0.68	0.65	0.67	0.57	0.57	0.75	0.70
50	0.71	0.70	0.68	0.59	0.60	0.76	0.72
60	0.72	0.68	0.69	0.60	0.59	0.75	0.73
70	0.74	0.70	0.70	0.60	0.63	0.78	0.77
80	0.72	0.70	0.74	0.61	0.62	0.75	0.78
90	0.69	0.67	0.69	0.62	0.60	0.71	0.74

VII. CONCLUSION

This paper explores a range of algorithms and machine learning models for the detection of phishing website URLs. The models analyzed include Logistic Regression, K-Nearest Neighbors (KNN), Decision Tree, Random Forest,

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 11, April 2025
Impact Factor: 7.67

Support Vector Classifier (SVC), Linear SVC, and Naïve Bayes. Each model was evaluated using various training set proportions. Among them, Linear SVC consistently demonstrated high accuracy, while Random Forest showed a significant improvement in precision as the training data size increased. Naïve Bayes achieved the highest true positive rate, followed by Linear SVC, Logistic Regression, and other models. Logistic Regression produced stable and adaptable results across accuracy, precision, and recall metrics. In contrast, KNN and Decision Tree models performed comparatively poorly, with lower accuracy and true positive rates.

Given the increasing prevalence of phishing attacks, there is a growing need to continually refine and adapt these models to counter emerging phishing tactics and trends. The algorithms discussed serve as effective tools for distinguishing between legitimate and malicious websites, helping to prevent potential cybersecurity threats. By leveraging these models, cybersecurity professionals can gain valuable insights to proactively detect and mitigate risks. Furthermore, integrating these models into web browsers, email filters, and security applications can enhance real-time protection for users.

The study highlights the importance of machine learning in cybersecurity and suggests several avenues for future research. Ensemble approaches, which combine multiple machine learning algorithms, may offer improved reliability and detection accuracy. Additionally, incorporating advanced techniques such as deep learning and natural language processing (NLP) could lead to more robust solutions for phishing detection. Evaluating these models across diverse datasets and scenarios can provide deeper insights into their strengths and limitations

In summary, this study underscores the critical role of machine learning in combating phishing threats. The models and methodologies presented form a solid foundation for future innovations, contributing to a more secure digital ecosystem and enabling the development of smarter defenses against phishing attacks.

REFERENCES

[1] A. K. Dutta (2021) Detecting phishing websites using machine learning technique.

[2] F. Mbachan, “Phishing URL prediction using logistic regression,” 2022.

[3] A. Akinyelu, and A. Adewumi (2014). Classification of Phishing Email Using RandomForest Machine Learning Technique. Journal of Applied Mathematics. 2014.

[4] A. Alhogail and A. Alsabih (2021). Applying machine learning and natural languageprocessing to detect phishing emails. Computers & Security, 110, 102414.

[5] O.K. Sahingoz, E. Buber, O. Demir, and B. Diri (2019). Machine learning-based phishing detection from URLs. Expert Systems with Applications, 117, 345–357.

[6] Shah, Varun. "Machine Learning Algorithms for Cybersecurity: Detecting andPreventing Threats." Revista Espanola de Documentacion Cientifica 15.4 (2021): 42-66.

[7] A. Kulkarni and L. L. Brown, “Phishing websites detection using machine learning,” International Journal of Advanced Computer Science and Applications, vol. 10, 2019.

[8] A. Safi and S. Singh, “A systematic literature review on phishing website detection techniques,” Journal of King Saud University—Computer and Information Sciences, 2023.

[9] S. Das Guptta, K. T. Shahriar, H. Alqahtani, D. Alsalman and I. H. Sarker, “Modeling hybrid feature-based phishing websites detection using machine learning techniques,” Annals of Data Science, 2022.

[10] P. Gupta and A. Mahajan, “Phishing website detection and prevention based on logistic regression,” International Journal of Creative Research Thoughts, vol. 10, pp. 2320–2882, 2022.

[11] T. A. Assegie, “K-nearest neighbor based URL identification model for phishing attack detection,” Indian Journal of Artificial Intelligence and Neural Networking, vol. 1, no. 2, pp. 18–21, 2021.

[12] D. Ahmed, K. Hussein, H. Abed and A. Abed, “Phishing websites detection model based on decision tree algorithm and best feature selection method,” Turkish Journal of Computer and Mathematics Education, vol. 13, no. 1, pp. 100–107, 2022.

[13] G. Ramesh, R. Lokitha, R. Monisha and N. Neha, “Phishing detection system using random forest algorithm,” International Journal for Research Trends and Innovation, vol. 8, pp. 510, 2023.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 11, April 2025
Impact Factor: 7.67

[14] D. Aksu, A. Abdulwakil and M. A. Aydin, "Detecting phishing websites using support vector machine algorithm," Pressacademia, vol. 5, no. 1, pp. 139–142, 2017.
[15] G. Kamal and M. Manna, "Detection of phishing websites using Naïve bayes algorithms," International Journal of Recent Research and Review, vol. XI, no. 4, pp. 34–38, 2018.
[16] F. Mbachan, "Phishing URL prediction using logistic regression," 2022.
[17] S. Hutchinson, Z. Zhang and Q. Liu, "Detecting phishing websites with random forest," Machine Learning and Intelligent Communications, pp. 470–479, 2018.
[18] A. A. Orunsolu, A. S. Sodiya and A. T. Akinwale, "A predictive model for phishing detection," Journal of King Saud University—Computer and Information Sciences, vol. 34, no. 2, pp. 232–247, 2022.
[19] M. Shoaib and M. S. Umar, "URL based phishing detection using machine learning," in 2023 6th Int. Conf. on Information Systems and Computer Networks (ISCON), Mathura, India, pp. 1–7, 2023.
[20] S. Alnemari and M. Alshammari, "Detecting phishing domains using machine learning," Applied Sciences, vol. 13, no. 8, pp. 4649, 2023.
[21] J. Bharadiya, "Machine learning in cybersecurity: Techniques and challenges," European Journal of Technology, vol. 7, no. 2, pp. 1–14, 2023.
[22] P. Dhanavanthini and S. S. Chakkravarthy, "Phish-armour: Phishing detection using deep recurrent neural networks," Soft Computing, pp. 1–13, 2023.
[23] P. P. Kumar, T. Jaya and V. Rajendran, "SI-BBA-A novel phishing website detection based on swarm intelligence with deep learning," Materials Today: Proceedings, vol. 80, pp. 3129–3139, 2023