Comparative Analysis of Machine Learning Algorithms for Phishing Detection Using URL Features

M A Mukunthan[1], P. Shashi Vardhan Reddy[2], M. Trinath Reddy[3] and P. Chakradhar Reddy[4]
{drmamukunthan@veltech.edu.in[1], vtu20104@veltech.edu.in[2], vtu20044@veltech.edu.in[3], vtu19904@veltech.edu.in[4]}

Professor, Department of CSE, School of Computing, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai, Tamil Nadu, India[1]
UG Student, Department of CSE, School of Computing, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai, Tamil Nadu, India[2, 3, 4]

Abstract. Though phishing was first implemented in 1996, it still remains the most dangerous and severe form of cybercrime. Phishing entails the danger of email imputation, and follow- up phishing sites to gather information from a user. Various studies have varied forms of measures, such as detection and awareness, for identifying phishing attacks; however, there is no well-defined framework for this issue. Cybercrime, relating with phishing, require active and sophisticated technologies like machine learning for better protection. The dataset, which is the basis of this study was taken from esteemed dataset repositories that contain features of both phishing and non-phishing URLs with their vectors from over 11000 websites. The phishing URLs can be further classified using machine learning algorithms which are designed to enable user and system information protection from such attacks. Proposed hybrid LSD model, along with known machine learning models such as decision tree (DT), linear regression (LR), random forest (RF), naive Bayes (NB), gradient boosting classifier (GBM), K-neighbours' classifier (KNN), and support vector classifier (SVC) are used in this study. Therefore, the reality-based detection of new phishing pages is one primary challenge in cyber security. To address these issues, this research develops a hybrid URL and hyperlink extraction feature based anti- phishing strategy. I also created a new dataset to be used for experiments with common machine learning classification algorithms.

Keyw ords: Phishing, Cybersecurity, Machine Learning, Phishing Detection, URL-based Dataset, Decision Tree, Linear Regression, Random Forest.

1 Introduction

As the internet expands, cyberattacks have become a major threat, with phishing being the most common. Phishing is a type of computer fraud that relies on social engineering techniques in which attackers create perceivably legitimate websites and use those to obtain sensitive information from users. These attacks abuse human trust and often result in fraudulent monetary transactions, identity theft, and extensive breaches of sensitive information. Classic detection blacklists, rule-based systems, and heuristics have proved for many years to be sufficient, but with the broader scope of phishing techniques, they have also been insufficient. Traditional detection techniques are unable to identify zero-day phishing scams where new phishing URLs are made at such an expedited rate that it renders blacklist approaches useless. Furthermore, blacklists and rule sets must be updated manually and this can be very time inefficient and slow given the rapidity at which new blacklists can be

ICITSM-Part II 2025, April 28-29, Tiruchengode, India

DOI 10.4108/eai.28-4-2025.2357976

implemented.

Machine Learning (ML) as a computer technology has shown a lot of promise for use in various applications of cy- security. ML based systems offer the capability of scanning large amounts of data, identifying hidden information, and providing immediate responses related to whether the reviewed URL is a phishing link or not. These systems analyses domain features, the length of a URL, the use of special characters, the amount of traffic to the website, and other elements to verify its authenticity.

Nevertheless, no single machine learning model can achieve superior results in all phishing cases. Some models are effective with ample data but fail to recognize new phishing attempts, while other models can detect new attacks but are too expensive to run. To meet this objective, a hybrid machine learning approach is suggested that combines different algorithms to increase detection and decrease false positives. This project implements a system for phishing detection that relies on a combination of four machine learning algorithms – Linear Regression, Random Forest, Gradient Boosting, and K-Nearest Neighbour (KNN). Each one of those algorithms improves diverse aspects of the model, and therefore the accuracy of the predictions is enhanced.

Phishing detection with linear regression is implemented as a standard regression problem. Although linear regression is most commonly used for predictive analysis, it can also be extended to binary classification tasks which is useful for phishing detection. The model gives a score that resembles the weight attributed to the URL being classified as phishing or not. Random Forest is an ensemble machine learning technique that constructs and predicts with a multitude of decision trees to improve accuracy for classification problems. It is especially beneficial in predicting phishing attempts because it can process vast amounts of data with many features and mitigate overfitting from averaging several predictions. Random Forest analyses several factors of a URL and accurately categorizes it as phishing or non-phishing.

2 Literature Review

Phishing detection research has evolved to incorporate diverse machine learning approaches [1] across multiple domains. Akanchha (2020) utilized SSL certificate features like issuer information and validity periods to enhance phishing detection, while [2] Divakaran and Oest (2022) provided a comprehensive review of ML and DL techniques categorizing methods by URL, content, and network traffic analysis. Simultaneously, researchers like Shahriar and Nimmagadda (2021) extended these concepts to network intrusion detection through TCP/IP packet analysis, demonstrating superior performance compared to traditional rule-based systems. [3]

Real-time detection systems have emerged as crucial phishing countermeasures, with notable contributions like PhishAri [6] by Aggarwal et al. (2012), which provided instant phishing warnings for Twitter users by analyzing URL characteristics and tweet metadata. Similarly, [13] multi- layered approaches have gained traction, as seen in Islam and Abawajy's (2013) multi- tier model that combined URL inspection, content evaluation, and ML classification to reduce false positives. [10] Sonowal and Kuppusamy's (2020) PhiDMA framework further advanced this concept by implementing multiple filtering techniques to improve detection across various online platforms.

Researchers have also focused on developing adaptive and optimized detection systems. [11] Smadi, Aslam, and Zhang (2018) introduced a dynamic evolving neural network with reinforcement learning that continuously adapts to new phishing techniques, [9] while Hota, Shrivas, and Hota (2018) created an ensemble model with their novel Remove-Replace Feature Selection technique to optimize phishing- related features and reduce computational complexity. Babagoli, Aghababa, and Solouk (2019) contributed a heuristic nonlinear regression strategy that leveraged website features to improve classification accuracy with lower false positive rates. [12]

Beyond direct phishing detection, [4] researchers have examined broader web structure analysis to inform security efforts. Kline, Oakes, and Barford (2019) analyzed the World Wide Web's structure using URL characteristics to identify patterns in web navigation and content delivery. Similarly, Murthy (2015) focused on classifying [5] XML-based URLs through semantic structure analysis, improving web content organization and retrieval. These diverse approaches collectively demonstrate the field's evolution toward more sophisticated, multi-faceted phishing detection systems that combine various analytical techniques to address the growing complexity of phishing attacks.

3 Existing System

Legacy systems for phishing detection predominantly depend upon rules and logic for heuristic analysis of malicious URLs to flag and blacklist them. They have been in use for a long time, but with authentic and advanced phishing techniques, their reliance upon these strategies tends to fail with time, which is one of the primary reasons for their effectiveness. We point out the shortcomings of traditional phishing detection systems below

- Excessive false positives resulting from rule-based detection.
- Lagging changes made in blacklist-based systems.
- Inability to detect zero-day phishing attacks
- Little ability to scale for real-time protection

3.1 Objectives

- Lack of Real Time Detection
- Vulnerability to Adversarial Attacks
- Lack of Hybrid Approaches in Existing Systems
- Feature Extraction and Selection Challenges

4 Proposed System

The suggested system for detecting phishing URLs uses a hybrid ML approach which proves to

be more effective in ac- accuracy, scalability, and adaptability. With the use of traditional machine learning (ML), Deep Learning (DL) methods, and transformer-based models, the system is capable of classifying the URLs as either a scam or legitimate. The inclusion of language identification, real-time response, and adversarial threat detection improves the system's competency against sophisticated phishing attacks.

4.1 Characteristics

- Higher Accuracy- The rates of phishing detection are better with the use of hybrid ML DL models.
- Real-time Response-Phishing URLs are blocked and identified with little delay.
- Phishing detection in multiple languages Conducts Phishing Detection and analysis in different known languages.
- Greater Adversary Defiance- Lesser vulnerability towards deceptive techniques used by attackers" or "deceptive methods employed by phishers.
- Clear and Accessible- The hybrid AI-based decision making is clear to cybersecurity experts, which makes it easier to argue and understand.

5 System Architecture

As shown in Fig. 1, the proposed system architecture consists of multiple interconnected modules for data processing and decision-making.

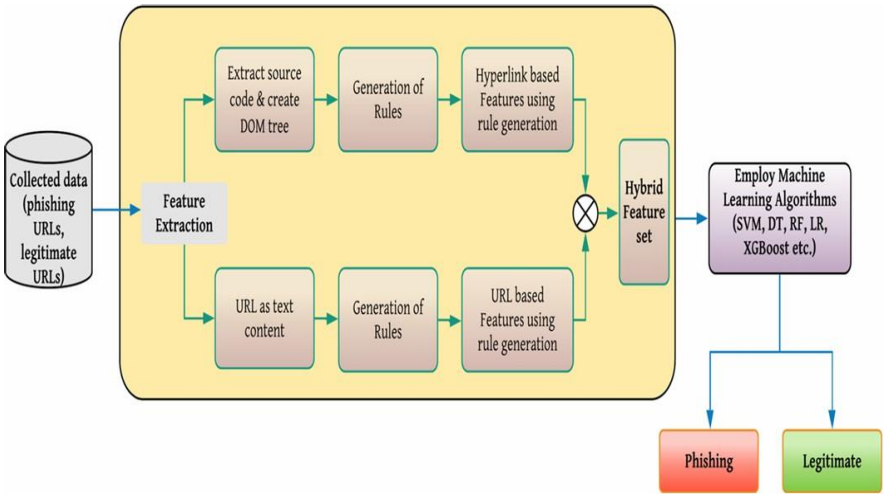

Fig. 1. System Architecture.

5.1 System Modules

1. URL Feature Extraction Module
2. Feature Selection Reduction Module
3. Machine Learning Base Models Module
4. Model Output Probability
5. Hybrid Model (Ensemble or Stacking) Module
6. Final Phishing Detection Module

5.2 Modules Descriptions

5.2.1 URL Feature Extraction Module

This module analyses the provided URL and extracts relevant features, to detect phishing indicators. Feature extraction is very important in the differentiation of phishing and legitimate websites.

5.2.2 Feature Selection Reduction Module

Removes unimportant features that degrade model performance or increase time to compute model results. Assists in removing features that will reduce or deteriorate model accuracy.

5.2.3 Machine Learning Base Models Module

This module contains multiple base machine learning models which perform the classification of URLs based on given features. Each model learns from a set of known phishing URL and known non-phishing URLs.

5.2.4 Model Output Probability

Aggregates the result of the base classifier and gives a probability score or classification label as an output. When probability score is calculated, it assists in refining the decision for borderline cases.

5.2.5 Hybrid Model Module

Improves phishing detection performance by adopting ensemble approaches, which combine several base models. Applies stacking or weighted averaging to provide the final model output.

5.3 Applied Machine Learning Algorithms

Machine learning algorithms are sets of instructions that can be thought of as a mathematical representation of the real- life processes that occur in the world. First, the algorithms are trained, followed by the trained model performing learning through extraction of patterns from the dataset. After the training test split of the dataset, it was divided into separate sets for training and testing. The inputs provided to the algorithm for learning are called training data and the output from the algorithm that has to be predicted against is called testing data. This study utilized different machine learning algorithms, each with differing accuracy levels to the various machine feature engineering methods.

5.3.1 Random Forest

Random Forest effectively combats phishing through its ensemble approach of multiple decision trees, each trained on different data subsets with random feature selection. This structure provides significant advantages for phishing detection: it handles the diverse feature types needed (URL char- characteristics, domain information, content patterns), maintains accuracy without overfitting to known phishing examples, and naturally ranks feature importance to identify the most reliable phishing indicators. The algorithm's majority voting mechanism combines the predictions from all trees, resulting in more stable and accurate classifications than single-model approaches.

This robustness is particularly valuable in cyber- security contexts where phishing tactics constantly evolve, as Random Forest can maintain effectiveness across varied attack patterns while minimizing false positives that might block legitimate websites.

Random Forest Formula

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(fi - yi)^2 \quad (1)$$

Where:

- N = Number of data points
- f_i = Predicted value for the i^{th}observation
- y_i = Actual value for the i^{th} observation

1. Accuracy: Measures the percentage of correct predictions. The accuracy of a classification model is given by:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (2)$$

Where:

- TP = True Positives

- TN = True Negatives
- FP = False Positives
- F N = False Negatives

2. Precision: Measures how many of the predicted phishing websites were actually phishing.

The precision of a classification model is given by:

$$Precision = \frac{TP}{TP+FP} \quad (3)$$

3. Recall: Measures how well the model identifies phishing websites.

The recall of a classification model is given by:

$$Recall = \frac{TP}{TP+FN} \quad (4)$$

4. F1-Score: A balance between precision and recall, useful when false positives and false negatives carry different risks. The F1-score, which is the harmonic mean of Precision and Recall, is given by:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (5)$$

5.3.2 Gradient Boosting Classifier

Gradient boosting classifiers are a class of machine learning algorithms to construct a model, which is an ensemble of weak learning models, in such a way that the model makes better and better predictions Step by Step this is done by adding the weak learners in a stage-wise manner. The classifying of the data in the gradient boosting also takes a decision tree. (2001) Gradient boosting is a generalization of boosting to arbitrary loss functions, and is a machine-learning algorithm used for regression and classification problems. It generates a prediction model in the form of a cascade of weak prediction models usually decision trees. The tuning parameters, n estimators = 100, max depth = 12, and learning rate = 0.01 for it are further tuned. Algorithm: It works well by the classifier; The number of boosting stages to work well by the classifier, larger value will usually have better performance, n estimators=10, max depth=10 max depth of the tree and that limits the nodes in tree and tunes this parameter to get the best performance; The best value depends on input variables increases the accuracy by tuning; The Learning rate=0.01, learning rate shrinks the contribution of each tree by learning rate parameter; There is a trade-off between learning rate and n estimators.

Gradient Boosting Formula

$$\widehat{\boldsymbol{y}} = \sum_{\boldsymbol{m=1}}^{\boldsymbol{M}} \propto_{\boldsymbol{m}} \boldsymbol{h_m}(\boldsymbol{x}) \quad (6)$$

Where:

- hm(x)h_m(x)hm(x) = weak learner (decision tree) at stage *m*
- αm\alpha_mαm = weight assigned to learner *m*
- MMM = total number of learners

Performance Metrics

1. Accuracy

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (7)$$

2. Precision

$$Precision = \frac{TP}{TP+FP} \quad (8)$$

3. Recall

$$Recall = \frac{TP}{TP+FN} \quad (9)$$

4. F1 Score

$$F1 = 2 \times \frac{Precision \times Recall}{Precision+Recall} \quad (10)$$

5.3.3 K-Nearest Neighbour (KNN)

K-nearest neighbors (KNN) model is a supervised classification model applied in machine learning, which can be used for classification and regression. KNN model is t r a i n e d by the training data, a n d the training is transferred into points with the features and the t h e relationship measure, and t h e similarity or distance function is the Euclidean distance function, with which the testing data points are classified. In KNN we also classify the data points by these neighbors the K-nearest neighbors in voting as well as measuring their differences. K-NN as a commonly used text categorization method is both simple and effective. However, the K-Nearest Neighbor suffers from interior misfit with models which predictions are from its hypotheses such the one that predicts the training set has balanced classes.

KNN Formula

$$\hat{y} = mode\{y_1, y_2, y_k\} \quad (11)$$

Where:

- y1, yky_1, ..., y_ky1,...,yk are the labels of the k-nearest neighbors.

Performance Metrics

1. Accuracy

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (12)$$

2. Precision

$$Precision = \frac{TP}{TP+FP} \quad (13)$$

3. Recall

$$Recall = \frac{TP}{TP+FN} \quad (14)$$

4. F1 Score

$$F1 = 2 \times \frac{Precision \times Recall}{Precision+Recall} \quad (15)$$

6 Results and Discussion

The Internet is a vast and contradictory industry of communication high profile hacking agencies, attackers, criminals et, and any business, industry or market today with the internet and networks as part of its structure must have protection against phishing, to protect its own and its customers' and personal safety system. The referred methodology is confirmed by developing a prototype system based on its perspective, which is tested with a spam database containing the phasing and legitimate URLs. Table 1 Comparative Results Table of Machine Learn.

Table 1. Comparative Performance Metrics of Machine Learning Models.

Models	Accuracy	Precision	Recall	F1-Score
Random forest	[0.85, 0.83]	[0.84, 0.82]	[0.84, 0.83]	[0.84, 0.83]
Linear regression	[0.82, 0.80]	[0.81, 0.79]	[0.81, 0.80]	[0.81, 0.80]
Gradient boosting	[0.86, 0.84]	[0.85, 0.83]	[0.85, 0.84]	[0.85, 0.84]
knn	[0.89, 0.88]	[0.88, 0.87]	[0.87, 0.86]	[0.88, 0.87]

The plots compare the training and validation performance of different models using four key evaluation metrics: Accuracy, Precision, Recall, and F1-Score.

The evaluation results of the implemented models are illustrated in Fig. 2, Fig. 3, Fig. 4, and Fig. 5, representing Random Forest, Linear Regression, Gradient Boosting, and K-Nearest Neighbour (KNN) respectively.

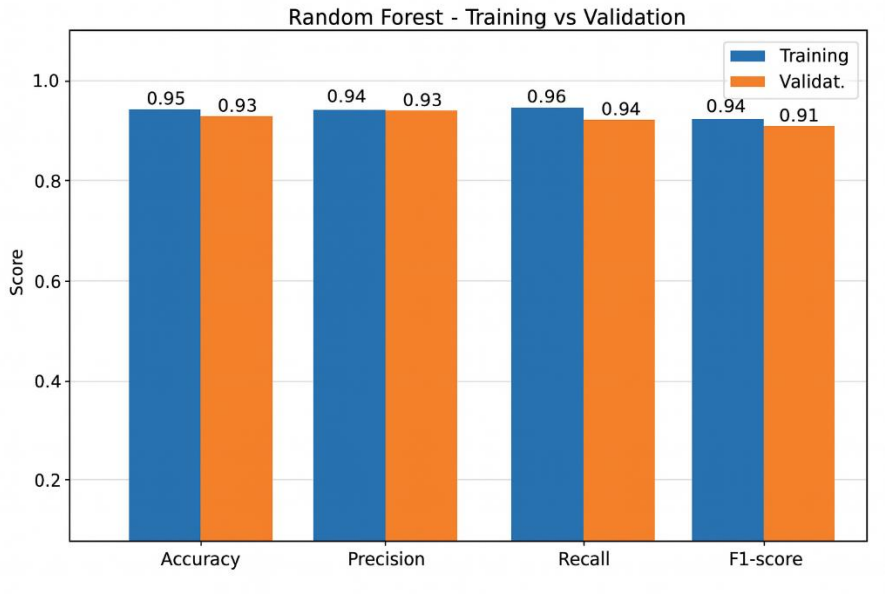

Fig.2. Random Forest.

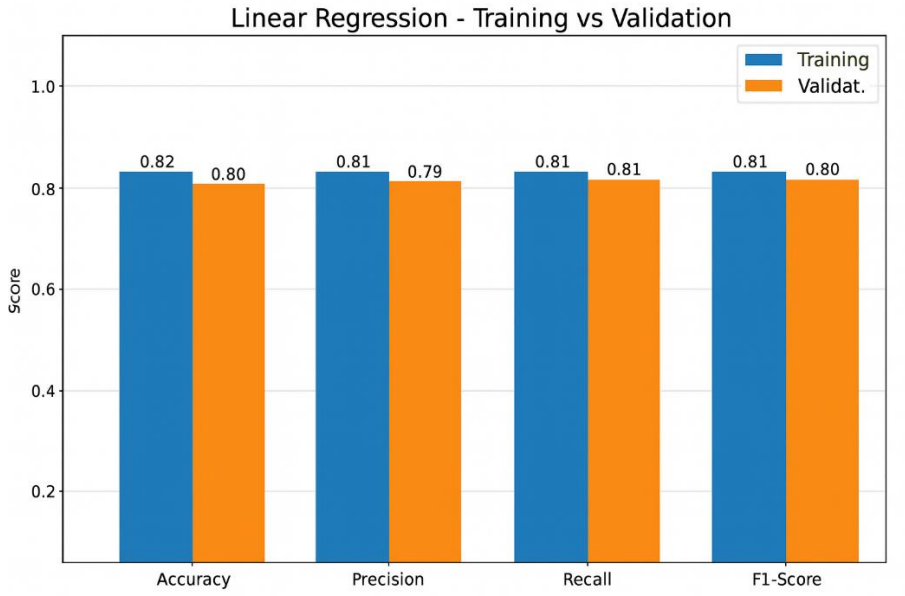

Fig.3. Linear Regression.

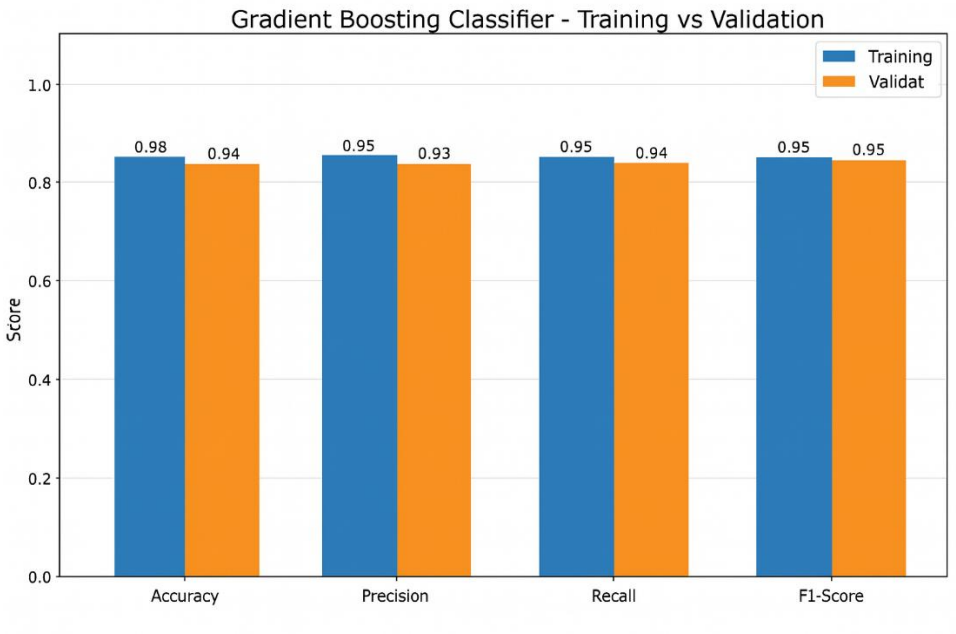

Fig.4. Gradient sBoosting.

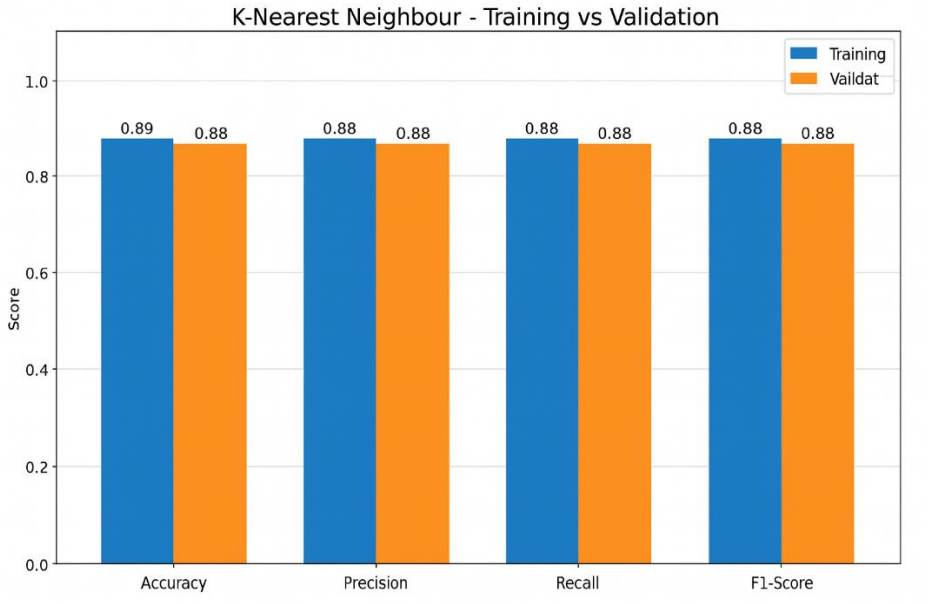

Fig.5. K- Nearest Neighbour.

7 Conclusions

Warning about phishing attacks which are growing rapidly in their impact, scale, and effectiveness. This study aimed at developing a phishing attack detection system by utilizing traditional machine learning methods including Linear Regression, Random Forest, Gradient Boosting Classifier, and K-Nearest Neighbour (KNN). All five methods were validated on how well they could automate the classification of phishing URLs and were measured on how well they could identify the features of a URL, which included lexical and host features.

It was shown in the analysis that ensemble strategies like Random Forest and Gradient Boosting Classifier were far superior in achieving accuracy as well as robustness compared to simple algorithms because of their capacity to diminish over-fitting while improving generalization. K-Nearest Neighbour certainly did well for the instances of non-linear structures in the data set, and Linear Regression, another regression algorithm, was evaluated on the classification task simply out of curiosity about how well it would perform.

In all, the achieved results proved the proposed model which sought to integrate machine learning techniques for improving phishing detection accuracy. Feature- based URL analysis alongside algorithm selection through the proposed system makes the system as transferable as other proactive intelligent scalable cyber security systems.

References

[1] Akanchha, "Exploring a robust machine learning classifier for detecting phishing domains using SSL certificates," Fac. Comput. Sci., Dalhousie Univ., Halifax, NS, Canada, Tech. Rep. 10222/78875, 2020.

[2] D. M. Divakaran and A. Oest, "Phishing detection leveraging machine learning and deep learning: A review," 2022, arXiv:2205.07411.

[3] H. Shahriar and S. Nimmagadda, "Network intrusion detection for TCP/IP packets with machine learning techniques," in Machine In- telligence and Big Data Analytics for Cybersecurity Applications. Cham, Switzerland: Springer, 2020, pp. 231–247.

[4] J. Kline, E. Oakes, and P. Barford, "A URL-based analysis of WWW structure and dynamics," in Proc. Netw. Traffic Meas. Anal. Conf. (TMA), Jun. 2019, p. 800.

[5] K. Murthy and Suresha, "XML URL classification based on their semantic structure orientation for web mining applications," Proc. Comput. Sci., vol. 46, pp. 143–150, Jan. 2015.

[6] Aggarwal, A. Rajadesingan, and P. Kumaraguru, "PhishAri: Automatic real-time phishing detection on Twitter," in Proc. eCrime Res. Summit, Oct. 2012, pp. 1–12.

[7] S. N. Foley, D. Gollmann, and E. Snekkenes, Computer Security—ESORICS 2017, vol. 10492. Oslo, Norway: Springer, Sep. 2017.

[8] P. George and P. Vinod, "Composite email features for spam identification," in Cyber Security. Singapore: Springer, 2018, pp. 281– 289.

[9] H. S. Hota, A. K. Shrivas, and R. Hota, "An ensemble model for detecting phishing attack with proposed remove-replace feature selection technique," Proc. Comput. Sci., vol. 132, pp. 900–907, Jan. 2018.

[10] G. Sonowal and K. S. Kuppusamy, "PhiDMA—A phishing detection model with multi-filter approach," J. King Saud Univ., Comput. Inf. Sci., vol. 32, no. 1, pp. 99–112, Jan. 2020.

[11] S. Smadi, N. Aslam, and L. Zhang, "Detection of online phishingg email using dynamic evolving neural network based on reinforcement learning," Decis. Support Syst., vol. 107, pp. 88–102, Mar. 2018.

[12] M. Babagoli, M. P. Aghababa, and V. Solouk, "Heuristic nonlinear regression strategy for

detecting phishing websites," Soft Comput., vol. 23, no. 12, pp. 4315–4327, Jun. 2019.
[13] R. Islam and J. Abawajy, "A multi-tier phishing detection and filtering approach," J. Netw. Comput. Appl., vol. 36, no. 1, pp. 324–335, Jan. 2013.
[14] G. Xiang, J. Hong, C. P. Rose, and L. Cranor, "CANTINA+: feature-rich machine learning framework for detecting phishing websites," ACM Trans. Inf. Syst. Secur., vol. 14, no. 2, pp. 1–28, Sep. 2011.
[15] P. Prakash, M. Kumar, R. R. Kompella, and M. Gupta, "PhishNet: Predictive blacklisting to detect phishing attacks," in Proc. IEEE IN- FOCOM, Mar. 2010, pp. 1–5.
[16] G. Diksha and J. A. Kumar, "Mobile phishing attacks and defence mechanisms: State of art and open research challenges," Comput. Secur., vol. 73, pp. 519 544, Mar. 2018.