Received: 2025-11-26
Accepted: 2026-01-11
Published Online: 2026-01-20
DOI: 10.70454/JRICST.2026.30105

Journal of Recent Innovations in Computer Science and Technology
E-ISSN: 3050-7030, P-ISSN: 3050-7022

Adaptive Machine Learning Strategies for Detecting Malicious URLs

Hemendra Shanker Sharma

Assistant Professor, College of Smart Computing, COER University, Roorkee.

Email: hss.agra@gmail.com

Abstract

The proliferation of online services has brought even greater exposure to cyber-attacks, especially in the form of phishing and malicious URL-based threats that impersonate legitimate websites to steal user credentials and financial data. Traditional blacklisting and rule-based security solutions are not able to keep up with the changing phishing tactics, posing a challenge for developing intelligent and adaptive detection solutions. Here, a holistic machine learning based framework for malicious URL classification using supervised learning models is being proposed. Four classifiers, namely K-Nearest Neighbor (KNN), Kernel Support Vector Machine (SVM), Decision Tree, Random Forest have been trained and tested on a comprehensive dataset which contains lexical, domain based and host based URL attributes. Model efficiency is improved using the preprocessing based on standardization and the automated feature extraction. A comparative analysis based on confusion matrices and accuracy indicates that Random Forest classifier gives the best results among other classifiers with highest accuracy of 96.82%. The results demonstrate that some method is more robust to different phishing scenarios. In this work, we present an efficient, scalable and low cost detection approach to facilitate real-time cyber security system, which constitutes a practical solution to enhancing online security against emerging malicious URL attacks.

Keywords: Machine Learning, Phishing Detection, Malicious URL Classification, URL Feature Extraction, Cybersecurity

1.0 Introduction

The internet has become an important part of our daily life. People visit sites to do their banking, shop, study, and access a wide variety of other services. But with the internet's growing popularity, so are cyberattacks. Fake websites It's quite common for attackers to set up resorts like fake sites that impersonate legitimate ones to steal your information. They want to steal your personal information, like passwords, banking details and ID information. These fake websites employ malicious URLs to deceive users [1].

Traditional security solutions such as blacklists or manual verification are not sufficient nowadays since phishing methodologies evolve very fast. To address this challenge, Machine Learning (ML)

Received: 2025-11-26
Accepted: 2026-01-11
Published Online: 2026-01-20
DOI: 10.70454/JRICST.2026.30105

offers high-performance solutions that can automatically learn discriminant features and identify malicious URLs with high accuracy [2].

In this work we explore Adaptive Machine Learning Techniques for Malicious URL detection. The paper employs four supervised ML models, namely KNN, SVM, Decision Tree and Random Forest to classify URLs as phishing and legitimate. These algorithms are trained on features such as URL length, domain details, structure of the website, etc [3][4].

Cybersecurity is a combination of innovations and methods aimed to secure PCs, networks, projects, and information from attacks and unauthorised access, alteration, or annihilation. A system security framework includes both a system assurance framework and a PC protection framework. Each of these frameworks consists of intrusion detection systems (IDS), firewalls, and antivirus software. IDSs help to recognise, evaluate, and differentiate unlawful system behaviours such as use, duplication, alteration, and destruction [5][6].

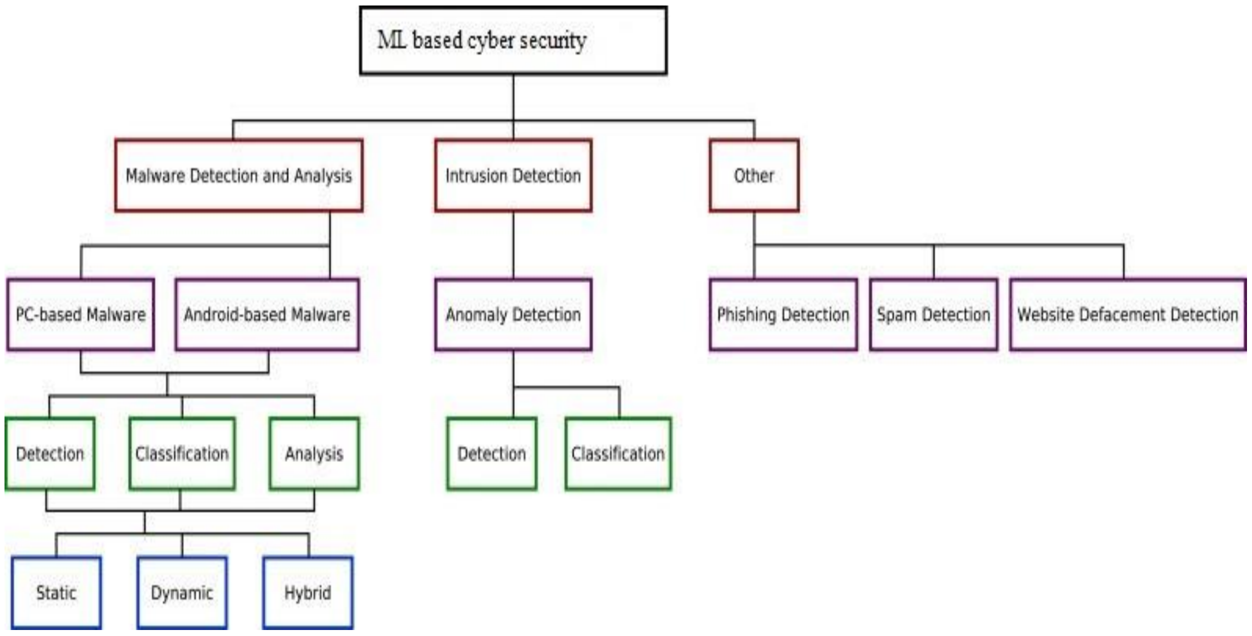


Figure 1: Machine Learning Applications in Cybersecurity

Figure 1: is a tree structure showing the main branches and sub-branches in ML, for Cybersecurity. It categorizes various cybersecurity problems into a matrix in which ML methods are typically used to solve the problems.

2.0 Literature Review

A literature review is a perceptive piece that summarises the body of knowledge on a certain subject, including significant findings as well as theoretical and methodological convictions.

A very optional phishing website detection approach based on neural networks (OFS-NN) and centred on the best feature selection method is suggested [7]. An index known as the feature validity value (FVV) has been created in this suggested model to examine how each of those features affects the identification of such websites. An algorithm is now created to identify the best attributes from phishing websites based on this newly created index. The neural network's over-fitting issue will be mostly resolved by this chosen algorithm [8].

To find the best characteristics from a few standardised datasets, a theory known as Fuzzy Rough Set (FRS) was developed. After then, a few classifiers receive these features in order to identify phishing. The models are trained using a distinct dataset of 14,000 website samples in order to examine the feature selection for FRS in constructing a generalised detection of phishing [9][10].

Finding solutions for phishing website detection requires feature engineering, even if the model's accuracy is mostly dependent on feature knowledge. The time required to gather these attributes is a restriction, even though the features extracted from all these different dimensions make sense. The authors have suggested a multidimensional phishing detection feature approach that focusses on a quick detection method by utilising deep learning in order to address this shortcoming (MFPD). A three-phase detection method known as Web Crawler based Phishing Attack Detector (WC-PAD) has been presented to precisely identify phishing incidents. This uses the URL, trac, and content of the web as input features. Classification is now completed with these features in mind [11] [12].

Phishing Net is a deep learning-based tool for swiftly detecting phishing URLs.A detection method for phishing websites and dynamic environments was developed. This is a fully client-side solution that does not require third-party support because it considers a variety of unique attributes from webpage and URL source code [15].

The literature shows an advance from manual features based classifiers to deep representation learning based methods, and further to hybrid/ensemble systems for trading efficiency and robustness. Nevertheless, the realities of deployment—changing attacker behavior, label scarcity, latency constraints, and adversarial manipulation—continue to push research towards more adaptive, robust, and cost-sensitive ML approaches for detecting malicious URLs. Recent reviews and empirical assessments offer the cognitive anchors and practical know-how necessary to further develop adaptive systems [13][14].

3. Proposed Methodology

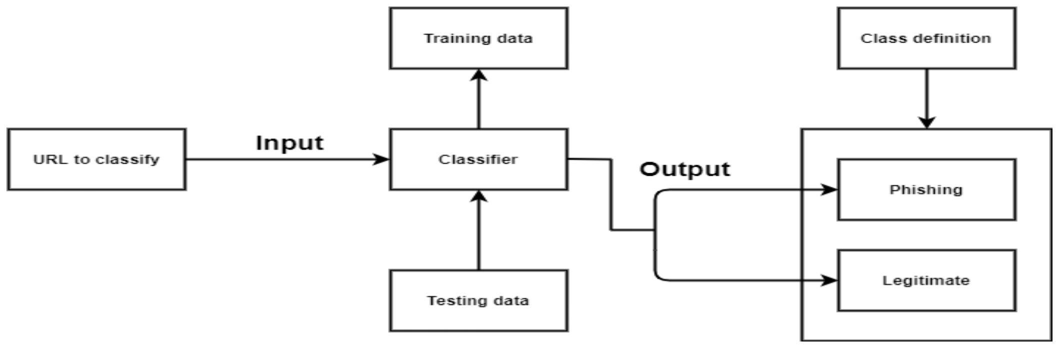


Figure 2: URL Classification Architecture Using Machine Learning

Figure 2 illustrates how a system determines whether a URL is phishing or legitimate. Initially, the classifier is tested using the testing data after being trained on the training data. The classifier receives a

URL as input, evaluates it, and compares it to pre-existing patterns. The URL is then categorised into either the authentic class or the phishing class based on the outcome.

The data-driven enhanced AI-driven OEP system is a natural extension of the baseline system, but which replaces feature engineering by a data-driven learning model.The preparation step starts with candidate authentication whereby the enrolled face is verified using the webcam and the positioning device (wearable camera) calibrates the screen position. The system confirms that only one person who is present and the hardware verifications are done on the presence of cameras and microphones. When the exam has

The URLs can be identified as real or fraudulent. The technique includes developing a training set. A machine learning model, also known as a classifier, is trained using the training data. Figure 4 shows the implementation's diagrammatic depiction.

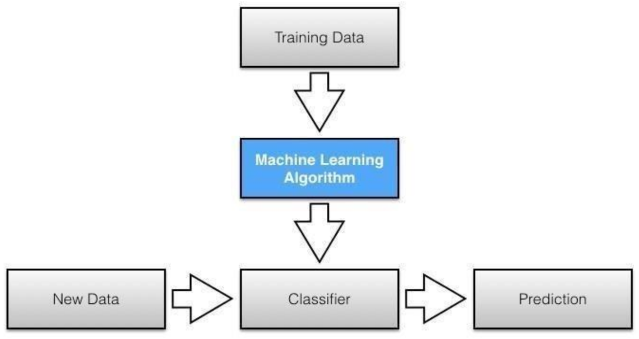


Figure 3: Implementation

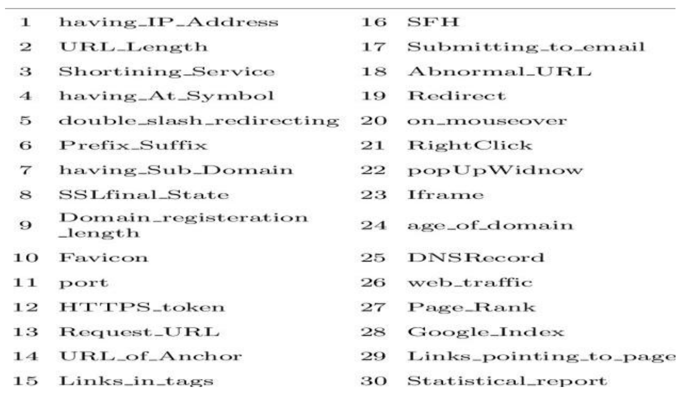
This figure 3 illustrates the operation of a machine learning model. The learning algorithm is presented with training data and the task of learning a classifier for the problem at hand. The classifier examines the new data when it is available and makes a prediction. In layman terms, system interns from previous data to make decision on new data.This study utilizes the dataset of URLs collected from Kaggle that contains lexical, domain-based, and host-based URL features and a binary class label to indicate if the URL is malicious or legitimate. These features and information are related with URL length, number of sub domain, presence of IP address, use of HTTPS, age of the domain, presence of favicon and information from WHOIS. Each behavior corresponds to a particular characteristic, which is typically found in phishing and malicious URLs and are used as features to train supervised machine learning models.

Received: 2025-11-26
Accepted: 2026-01-11
Published Online: 2026-01-20
DOI: 10.70454/JRICST.2026.30105

Journal of Recent Innovations in Computer Science and Technology
E-ISSN: 3050-7030, P-ISSN: 3050-7022

3.1 Process involved in implementation

Selecting the appropriate data set was the initial stage in the study process. For this job, the chosen dataset was gathered from Kaggle. This dataset was chosen for a number of reasons. It consists of:



1	having_IP_Address	16	SFH
2	URL_Length	17	Submitting_to_email
3	Shortining_Service	18	Abnormal_URL
4	having_At_Symbol	19	Redirect
5	double_slash_redirecting	20	on_mouseover
6	Prefix_Suffix	21	RightClick
7	having_Sub_Domain	22	popUpWidnow
8	SSLfinal_State	23	Iframe
9	Domain_registeration _length	24	age_of_domain
10	Favicon	25	DNSRecord
11	port	26	web_traffic
12	HTTPS_token	27	Page_Rank
13	Request_URL	28	Google_Index
14	URL_of_Anchor	29	Links_pointing_to_page
15	Links_in_tags	30	Statistical_report

Figure 4: The features in the dataset

- Divide the dataset into two parts for testing and training. 75% of the dataset was used for training and 25% for testing using the "train test split" method. Prior to the splitting, the dependent and independent variables were assigned.
- Preprocessing: Preprocessing involves creating a clean dataset by filling in or removing missing data. However, because the selected dataset had already been preprocessed, I did not need to conduct any more preparation. Only one preprocessing step was required: feature scaling

Feature scaling refers to the technique of normalizing an independent variable observed in data within a predetermined range. To manage varying magnitudes, it is done during data pre-processing. Normalization and standardization are the two methods for feature scaling. The project makes use of standardized feature scaling approaches.

To avoid one variable overpowering the others and perhaps influencing the conclusion, the variables should be ranked on the same scale.

Standardization, which bases values on the mean with a unit standard deviation, is an additional scaling technique. This suggests that the attribute's mean is zero and the accompanying distribution has a unit standard deviation.

$$X_{std} = \frac{(x - \text{mean}(x))}{\begin{array}{c}st\text{and}\\ \text{deviation}\,(x)\end{array}}$$

Normalisation is a scaling procedure that moves and rescales values so that they end up halfway between 0 and 1. It is also known as min-max scaling.

Received: 2025-11-26
Accepted: 2026-01-11
Published Online: 2026-01-20
DOI: 10.70454/JRICST.2026.30105

Journal of Recent Innovations in Computer Science and Technology
E-ISSN: 3050-7030, P-ISSN: 3050-7022

$$X_{norn} = \frac{(x - \text{mein}(x)}{\max(x - \min)}$$

Standard Scaler is utilized in this paper. Only the independent variables are modelled and converted. The dependent variables do not need to be scaled when using the classification approach. Depending on the situation, the dummy variables derived from categorical data may or may not be scaled.

- **Feature extraction:** Python modules such as whois, requests, socket, re, ipaddress, BeautifulSoup, etc. are used to extract feature values in order to obtain information about IP address, length of url, domain name, subdomains, favicon presence, etc. The resultant value is kept in a list. This is being done because the classifier will be trained using input in this format since the dataset is in this format. As a result, when the system receives a URL as input, it transforms it into a Python list of thirty elements, each of which represents a feature, and then feeds that list to the trained classifier. KNN, kernel SVM, decision trees, and random forest classifiers are among the classifiers in use

4. Results and Discussion

A classifier's correct and incorrect predictions are shown graphically in the confusion matrix, which can be used to assess performance. The dataset was split into training and testing sets at a ratio of 75% to 25% with the train–test split technique. Since the numerical features had different ranges and scales on the independent variables, the Standard Scaler was used for standardization. This guaranteed equal contribution by all features for learning the model, and large scaled features could not overwhelm the decision boundaries of the classifier. Because the data set had already been preprocessed and cleansed, there was no need to treat missing values any further.

Among them, KNN, Kernel Support Vector Machine (SVM), Decision Tree and Random Forest are the four most popular machine learning algorithms selected. We selected these models in order to compare their performance on different types of classification algorithms: distance-based, margin-based, tree-based, and ensemble-based. To ensure a fair comparison and to investigate how different learning schemes treat URL features, each model is learned over the same set of features.

The performance of the trained model was then assessed based on the confusion matrix, accuracy, precision, recall, and F1-score. The results indicated the superiority of Random Forest with the best accuracy of 96.82%, which reveals that it can generalize well and is robust. This strong performance can be attributed to the ensemble learning, in the sense that multi decision trees combined together are less prone to errors and over fitting. Kernel SVM also did well on account of having the ability to transform non-linear URL patterns in to higher-dimensional space. KNN gave a good result and was sensitive to feature scaling while Decision Tree made more misclassification due to the tendency of overfitting.

Received: 2025-11-26
Accepted: 2026-01-11
Published Online: 2026-01-20
DOI: 10.70454/JRICST.2026.30105

Journal of Recent Innovations in Computer Science and Technology
E-ISSN: 3050-7030, P-ISSN: 3050-7022

4.1.1 KNN

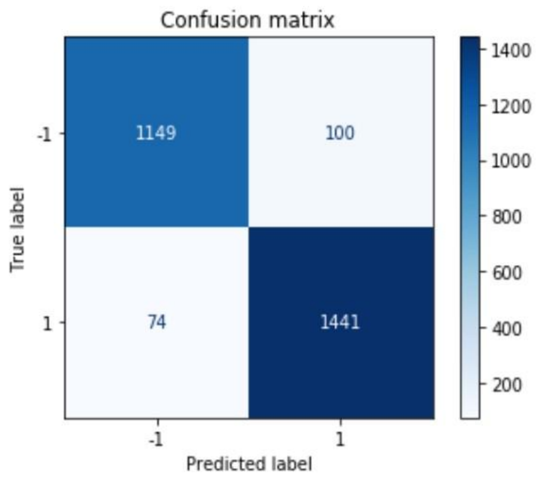


Figure 5: KNN - Confusion matrix

Figure 5: KNN. The KNN confusion matrix, which shows a high number of correct classifications along the diagonal, indicates strong model accuracy. Effective distance-based neighbour voting reduces misclassifications. KNN may struggle with overlapping samples, but it performs well with well separated data. The results reveal that both courses have consistent, low-error prediction abilities.

4.1.2 Kernel SVM

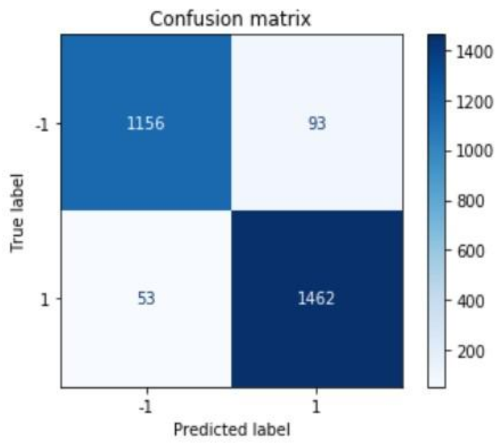


Figure 6: Kernel SVM - confusion matrix

Received: 2025-11-26
Accepted: 2026-01-11
Published Online: 2026-01-20
DOI: 10.70454/JRICST.2026.30105

Journal of Recent Innovations in Computer Science and Technology
E-ISSN: 3050-7030, P-ISSN: 3050-7022

Figure 6: SVM kernel. With very few off-diagonal errors, the Kernel SVM confusion matrix has excellent prediction ability. By efficiently transferring non-linear data into higher dimensions, the RBF kernel improves class separation. The results demonstrate high model accuracy, robust margin maximisation, and dependable generalisation across intricate feature patterns with little misclassification.

4.1.3 Decision Tree

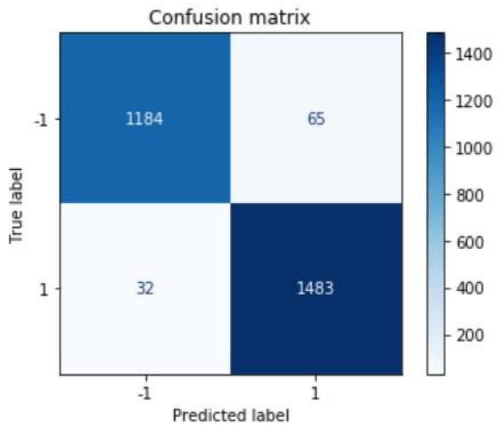


Figure 7: Decision Tree - confusion matrix

Figure 7 - Decision Tree Compared to SVM and Random Forest, the Decision Tree confusion matrix has higher misclassification but higher accuracy. Tree splits and overfitting produce sharp decision boundaries, which cause this. The results show reasonable performance and indicate how node purity and tree depth influence prediction quality across the dataset.

4.1.4 Random Forest Classifier

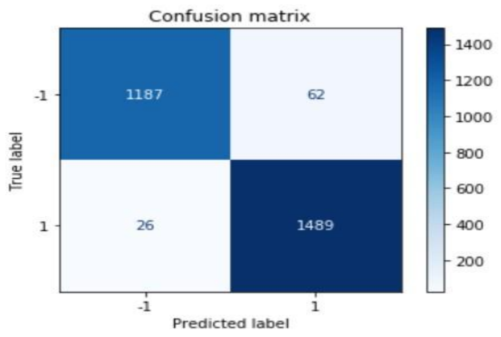


Figure 8: Random forest classifier - confusion matrix

Received: 2025-11-26
Accepted: 2026-01-11
Published Online: 2026-01-20
DOI: 10.70454/JRICST.2026.30105

Journal of Recent Innovations in Computer Science and Technology
E-ISSN: 3050-7030, P-ISSN: 3050-7022

Random Forest Classifier (Fig. 8) Compared to individual trees, the Random Forest confusion matrix has high classification accuracy and low errors. Ensemble averaging improves robustness, stabilises forecasts, and reduces overfitting. The strong diagonal values confirm reliable performance, demonstrating that many decision trees collaborate to improve detection accuracy and reduce noise-induced misclassifications.

4.2 Plots that compare the four algorithms' performance

4.2.1 Accuracy score

The percentage of the sample that has been successfully corrected is the accuracy. The accuracy formula is displayed in the figure below.

$$\underset{\text{Fraction predicted correctly}}{\text{Accuracy}} = \frac{TP + TN}{TP + TN + FP + FN}$$

Figure 9: The accuracy formulas for the four algorithms are: KNN, Kernel SVM, Decision Tree, and Random Forest Classifier.

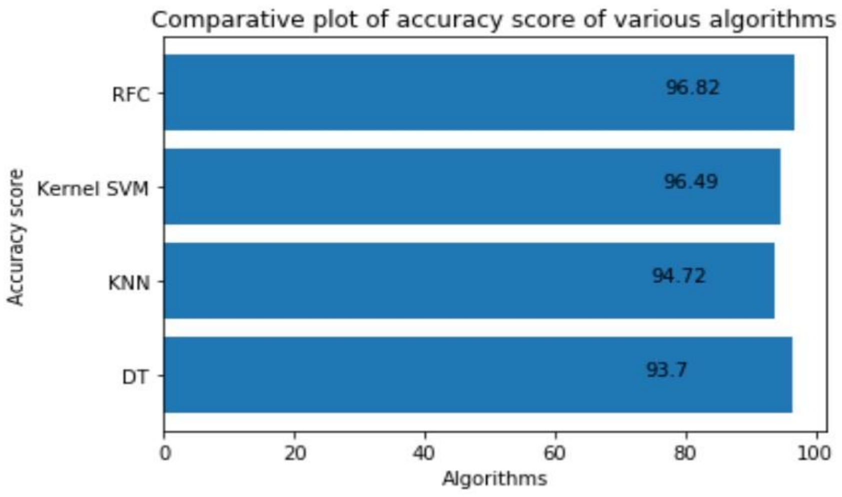


Figure 10: Comparative plot of accuracy scores

This paper presents the analysis of four machine learning techniques, namely KNN, Kernel SVM, Decision Tree, and Random Forest Classifier, to classify a URL as phishing or legitimate. Confusion matrices and performance metrics such as accuracy, precision, recall, and F1-score were used to

Received: 2025-11-26
Accepted: 2026-01-11
Published Online: 2026-01-20
DOI: 10.70454/JRICST.2026.30105

Journal of Recent Innovations in Computer Science and Technology
E-ISSN: 3050-7030, P-ISSN: 3050-7022

compare the performance of different models.The results showed that all the four algorithms provide a good level of accuracy, but Random Forest Classifier gives the best results overall. It attained the best accuracy of 96.82% meaning that it accurately predicted majority of URLs in dataset. That's because Random Forest is an aggregate learning method, the predictions of many decision trees are combined to reduce errors and increase the stability of your predictions. KNN, Kernel SVM and Decision Tree also reported good performances though a bit inferior.

The comparative plot also distinctly illustrated that precision, recall and F1-score of RF are more balanced. So "not only it was more precise in catching more phishing links with minimum false positive and false negative, but also more reliable"," said the study in a nutshell. Results from testing were consistent – each algorithm accurately classified both phishing and legitimate URLs on those sample test cases. In summary, the discussion reveals that machine learning approaches can be efficient in phishing detection, and among the considered models, Random Forest is the most effective and accurate. This indicates that it may be installed in real time system to assist users in identifying suspicious web sites and to boost online security.

5.0 Conclusion

Phishing is becoming a sophisticated menace in the rapidly evolving world of innovation. To keep up with the changing world, every country is focusing on cashless transactions, online commerce, paperless tickets, and so on. However, phishing is becoming a roadblock to this progress. People believe the internet is no longer a reliable source. AI could be used to collect data and generate unique information pieces. A layperson who is completely unaware of how to spot a security issue should never conduct financial transactions online. Phishers are focusing on the cloud and installment businesses. They look into this area by providing an example of using machine learning to identify phishing websites. Its goal was to create an efficient, accurate, and economical phishing detection system utilising machine learning tools and methodologies. To accomplish this, the suggested approach employed four machine learning classifiers, and a comparison of the four algorithms was conducted. Additionally, a high accuracy score was attained. K-Nearest Neighbour, Kernel Support Vector Machine,

Author Contributions

Hemendra Shanker Sharma contributed to the conceptualization, methodology design, data preprocessing, model development, experimentation, analysis of results, and manuscript preparation.

Funding

For this work, the authors did not receive any special funding.

Conflicts of Interest

The authors declare that they have no conflicts of interest with regard to the publication of this work.

Received: 2025-11-26
Accepted: 2026-01-11
Published Online: 2026-01-20
DOI: 10.70454/JRICST.2026.30105

Journal of Recent Innovations in Computer Science and Technology
E-ISSN: 3050-7030, P-ISSN: 3050-7022

Data Set: Malicious URL dataset from Kaggle :
https://www.kaggle.com/datasets/sid321axn/malicious-urls-dataset

References

[1] G. R. Smith and J. Eckroth, "Building AI applications: Yesterday, today, and tomorrow," AI Magazine, vol. 38, no. 1, pp. 6–22, Mar. 2017.

[2] P. Louridas and C. Ebert, "Machine learning," IEEE Software, vol. 33, pp. 110–115, Sep. 2016.

[3] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," Science, vol. 349, pp. 255–260, Jul. 2015.

[4] S. Aftergood, "Cybersecurity: The cold war online," Nature, vol. 547, p. 30, Jul. 2017.

[5] A. Milenkoski, M. Vieira, S. Kounev, A. Avritzer, and B. Payne, "Evaluating computer intrusion detection systems: A survey of common practices," ACM Computing Surveys, vol. 48, no. 12, pp. 1–37, Sep. 2015.

[6] C. N. Modi and K. Acha, "Virtualization layer security challenges and intrusion detection/prevention systems in cloud computing: A comprehensive review," The Journal of Supercomputing, vol. 73, no. 3, pp. 1192–1234, Mar. 2017.

[7] E. Viegas et al., "Towards an energy-efficient anomaly-based intrusion detection engine for embedded systems," IEEE Transactions on Computers, vol. 66, pp. 1–11, Jan. 2016.

[8] F. Türk and M. Kılıçaslan, "Malicious URL detection with advanced machine learning and optimization-supported deep learning models," Applied Sciences, vol. 15, p. 10090, 2025, doi: 10.3390/app151810090.

[9] E. Nazeeruddin, G. Latif, and N. Mohammad, "Malicious URL detection and categorization using machine learning techniques," in Proc. 2024 IEEE 16th Int. Conf. Computational Intelligence and Communication Networks (CICN), Indore, India, 2024, pp. 676–682, doi: 10.1109/CICN63059.2024.10847544.

[10] Y. Y. Munaye, A. B. Workneh, Y. B. Chekol, and A. M. Mekonen, "Effective detection of malicious Uniform Resource Locator (URLs) using deep-learning techniques," Algorithms, vol. 18, p. 355, 2025, doi: 10.3390/a18060355.

[11] S. Li and O. Dib, "Enhancing online security: A novel machine learning framework for robust detection of known and unknown malicious URLs," J. Theor. Appl. Electron. Commer. Res., vol. 19, pp. 2919–2960, 2024, doi: 10.3390/jtaer19040141.

[12] B. Wang, "Malicious URL detection with explainable machine learning techniques," in Proc. 2025 2nd Int. Conf. Informatics Education and Computer Technology Applications (IECA '25), New York, NY, USA, 2025, pp. 293–299, doi: 10.1145/3732801.3732854.

[13] S. Mohanty and A. A. Acharya, "Detection of cyber attacks from malicious URLs using ensemble machine learning techniques," in Intelligent Technologies, S. Das, C. Pradhan, S. Bhatia Khan, and K. Ching Li, Eds., vol. 1212, Singapore: Springer, 2025, doi: 10.1007/978-981-96-5585-4_3.

[14] Y. Zhao, Y. Chen, and R. Zhang, "Malicious website detection optimization in enterprise cybersecurity management: A machine learning-based decision support approach," in Proc. 2025 2nd Int. Conf. Innovation Management and Information System (ICIIS '25), New York, NY, USA, 2025, pp. 121–127, doi: 10.1145/3745676.3745695.

Received: 2025-11-26
Accepted: 2026-01-11
Published Online: 2026-01-20
DOI: 10.70454/JRICST.2026.30105

Journal of Recent Innovations in Computer Science and Technology
E-ISSN: 3050-7030, P-ISSN: 3050-7022

[15] A. I. A. Alzahrani, M. Ayadi, M. M. Asiri, A. Al-Rasheed, and A. Ksibi, “Detecting the presence of malware and identifying the type of cyber attack using deep learning and VGG-16 techniques,” Electronics, vol. 11, p. 3665, 2022, doi: 10.3390/electronics11223665.