Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 7.056

IJCSMC, Vol. 14, Issue. 4, April 2025, pg.98 – 103

DETECTION OF PHISHING WEBSITES USING MACHINE LEARNING

Mangipudi Lakshmi Sirisha

MTECH CSE – AI & ML
Anil Neerukonda Institute of Technology and Sciences, Andhra Pradesh, Visakhapatnam, INDIA

DOI: https://doi.org/10.47760/ijcsmc.2025.v14i04.008

ABSTRACT:
With raising in-depth amalgamation of the Internet and social life, the Internet is looking differently at how people are learning and working, meanwhile opening us to growing serious security attacks. The ways to recognize various network threats, specifically attacks not seen before, is a primary issue that needs to be looked into immediately. The aim of phishing site URLs is to collect the private information like user's identity, passwords and online money related exchanges. Phishers use the sites which are visibly and semantically like those of authentic websites. Since the majority of the clients go online to get to the administrations given by the government and money related organizations, there has been a vital increment in phishing threats and attacks since some years.

I. INTRODUCTION

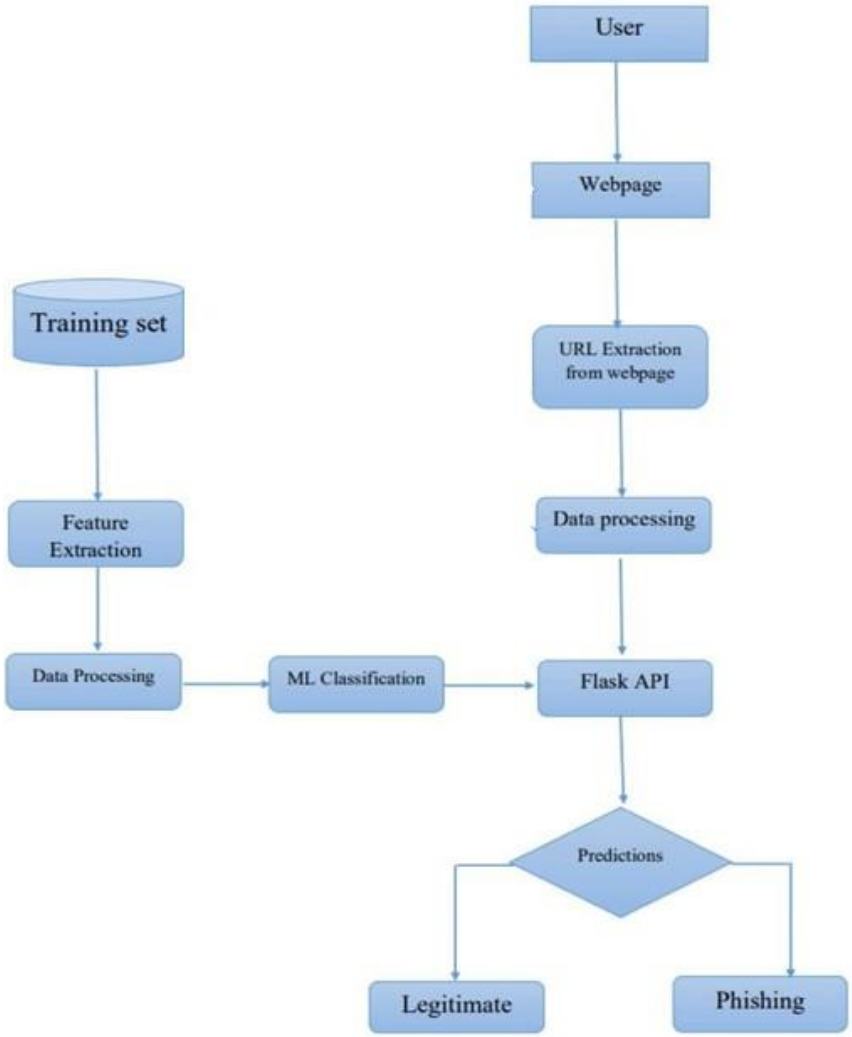
In modern era Phishing becomes a main area of concern for security researchers due to the fact it is not tough to create the fake internet site which looks so close to legitimate internet site. Experts can discover fake web sites however not all the customers can discover the fake website and such customers become the victim of phishing attack. Main purpose of the attacker is to steal banks account credentials. How hackers do their work, they send you just spam mail. In this mail though they will say that this email is mean to inform you that you’re my university network password will expire in

24 hours and they have provided you to update the password and login when we click on that link, we will redirect to that page which is a hacker server and they will be steal your data everything which is online.

II. METHODOLOGY

The methodology for this study involves a series of systematic steps to evaluate and compare various machine learning algorithms for phishing website detection. The process includes data preparation, model selection, training, and evaluation etc. The primary goal is to identify the most effective approach for detecting fraudulent website and to understand the strengths and limitations of each algorithm.

FLOW CHART:



1.Data Collection

Source Identification:

There are many popular open sources for collecting the data. Eg: kaggle.com, UCI repository, etc. In this project we have used phishing.csv data. This data is downloaded from kaggle.com.

Data Quantity:

Aim for a balanced dataset to avoid bias. For example, if you have 10,000 phishing URLs, ensure a similar number of legitimate URLs.

Data Labelling:

Assign a label to each URL:

- 1 for phishing.
- 0 for legitimate.

Data Preprocessing:

The download data set is not suitable for training the machine learning model as it might have so much of randomness so we need to clean the dataset properly in order to fetch good results. This activity includes the following steps.

- Handling missing values
- Handling categorical data
- Handling outliers
- Scaling Techniques
- Splitting dataset into training and test set

Note: These are the general steps of pre-processing the data before using it for machine learning. Depending on the condition of your dataset, you may or may not have to go through all these steps.

2.Feature Extraction

In this phase, relevant features are extracted or constructed from the available data. It involves techniques such as dimensionality reduction, transforming variables, creating new features based on domain knowledge, or incorporating external data sources.

3.Model Building and Comparison of Models

There are two major types of supervised machine learning problems, called classification and regression. Our data set comes under regression problem, as the prediction of suicide rate is a continuous number, or a floating-point number in programming terms. The supervised machine learning models (regression) considered to train the dataset in this notebook are: Logistic Regression, K-Nearest Neighbors, Naive Bayes, Decision Tree, Random Forest, Gradient Boosting, Multi-Layer perceptron Classifier, Support Vector Machine Classifier.

The metrics considered to evaluate the model performance are Accuracy & F1 score.

To compare the model's performance, a data frame is created. The columns of this data frame are the lists created to store the results of the model.

Logistic Regression

Logistic Regression is a linear model used for binary classification tasks. It predicts the probability that an instance belongs to a specific class using the logistic function (sigmoid).

K-Nearest Neighbors (KNN)

KNN is a non-parametric, instance-based algorithm that classifies a data point based on the majority label of its k nearest neighbors in the feature space.

Naive Bayes

Naive Bayes is a probabilistic classifier based on Bayes' Theorem. It assumes that features are conditionally independent given the target class.

Decision Tree

A Decision Tree splits data into subsets based on feature values, forming a tree structure where each node represents a feature, and leaves represent the output class.

Random Forest

Random Forest is an ensemble learning method that builds multiple decision trees and combines their

outputs (via majority voting or averaging).

Gradient Boosting

Gradient Boosting is an ensemble technique where weak learners (typically decision trees) are trained sequentially to correct the errors of the previous learners.

Multi-Layer Perceptron (MLP) Classifier

An MLP is a type of artificial neural network with at least one hidden layer. It learns complex patterns using backpropagation.

Support Vector Machine (SVM) Classifier

SVM is a supervised learning algorithm that finds the hyperplane maximizing the margin between two classes. It can use kernel tricks to handle non-linear data.

4. Model Training and Evaluation

Model training and evaluation in machine learning involves preparing data, training a model, and assessing its performance. The process starts with splitting the dataset into training and test sets to prevent overfitting. The training set is used to teach the model patterns in the data, often with cross-validation to ensure generalizability.

Metrics like accuracy, precision, recall, F1-score, and AUC-ROC are used to evaluate the model's performance on the test set. Hyperparameter tuning techniques such as grid search or random search optimize model parameters, while regularization, pruning, and early stopping prevent overfitting. Once trained, the model is deployed and monitored for real-world performance, with periodic retraining to adapt to new data.

III. IMPLEMENTATION AND RESULT

Scikit-learn tool has been used to import Machine learning algorithms. Dataset is divided into training set and testing set in 50:50, 70:30 and 90:10 ratios respectively. Each classifier is trained using training set and testing set is used to evaluate performance of classifiers. Performance of classifiers has been evaluated by calculating classifier's accuracy score, false negative rate and false positive rate.

Model Comparison Results

Logistic Regression

- Accuracy: **92.22%**
- Balanced performance between classes.
- Precision: ~0.95 (-1), ~0.90 (1)
- Recall: ~0.91 (-1), ~0.94 (1)
- F1-Score: ~0.93 (-1), ~0.92 (1)

K-Nearest Neighbors (KNN)

- Accuracy: 92.22%
- Similar accuracy to Logistic Regression.
- Precision: ~0.92 for both classes.
- Recall: ~0.93 (-1), ~0.91 (1)
- F1-Score: ~0.93 (-1), ~0.92 (1)

Naive Bayes

- Accuracy: 66.24% (lowest among all models)
- Very high recall for class -1: 100%
- Very low recall for class 1: 27%
- Precision skewed: 1.00 for class 1, 0.62 for class -1
- Generally poor performance; struggles with class 1.

Decision Tree

- Accuracy: 91.45%
- Slightly lower than Logistic Regression and KNN.
- Precision: ~0.91 (-1), ~0.92 (1)
- Recall: ~0.94 (-1), ~0.89 (1)
- F1-Score: ~0.92 (-1), ~0.91 (1)

Random Forest

- Accuracy: 92.82%
- Improvement over Decision Tree.
- Precision: ~0.94 (-1), ~0.92 (1)
- Recall: ~0.93 for both classes.
- F1-Score: ~0.93 for both classes.

Gradient Boosting

- Accuracy: 94.53% (highest among all models)
- Strong performance across both classes.
- Precision: ~0.95 (-1), ~0.94 (1)
- Recall: ~0.94 (-1), ~0.95 (1)
- F1-Score: ~0.95 (-1), ~0.94 (1)

Multi-Layer Perceptron (MLP)

- Accuracy: 94.02%
- Excellent performance.
- Precision: ~0.96 (-1), ~0.92 (1)
- Recall: ~0.93 (-1), ~0.95 (1)
- F1-Score: ~0.94 for both classes.

Support Vector Machine (SVM)

- Accuracy: 93.85%
- Very strong performance, close to MLP and Gradient Boosting.
- Precision: ~0.96 (-1), ~0.92 (1)
- Recall: ~0.93 (-1), ~0.95 (1)
- F1-Score: ~0.94 (-1), ~0.93 (1)

IV. CONCLUSION

In conclusion, phishing detection using machine learning offers an efficient and scalable approach to combating online fraud. By leveraging algorithms to analyze URL patterns, domain features, and behavioral attributes, these systems can quickly identify malicious attempts that might evade traditional rule-based detection methods. Machine learning models, such as logistic regression, random forests, or gradient boosting, enable dynamic adaptability to new phishing tactics when trained on diverse and up-to-date datasets. However, the success of these systems depends on addressing challenges like false positives, dataset biases, and computational overhead. Integrating machine learning with real-time monitoring and periodic retraining ensures improved accuracy and robustness, making it a critical tool in enhancing cybersecurity frameworks.

This system is designed for resources are used as intended, prevents from valuable information from leaks out, produce better control mechanism and alerts the user to keep their private information safe. Like any other programs, there are improvements which could be made into this system. Based on the capabilities which the current system processes, text message integration would a great recommendation that could be made to improve the program in the future. The future version of the application could also implement an option to directly notify the blacklisted website with a text

message. The program could be made to access the list as an attachment. This text message integration function would further the usability of the application.

REFERENCES

[1]. A. J. Ashutosh Kumar Singh, and Keshav Singh, "A Survey on Cyber Security Aware-ness and Perception among University Students in India," Journal of Advances in Mathe-matics and Computer Science, November 2021.

[2]. S. Shams Hussein, W. Hashim Abdulsalam, and W. Abed Shukur, "Covid-19 Predic-tion using Machine Learning Methods: An Article Review," Wasit Journal of Pure Sciences, vol. 2, no. 1, pp. 217-230, 03/26 2023, doi 10.31185/wjps.124.

[3]. S. Mahdi Muhammed, G. Abdul-Majeed, and M. Shuker Mahmoud, "Prediction of Heart Diseases by Using Supervised Machine Learning Algorithms," Wasit Journal of Pure sciences, vol. 2, no. 1, pp. 231-243, 03/26 2023, doi: 10.31185/wjps.125.

[4]. N. Kareem, "Afaster Training Algorithm and Genetic Algorithm to Recognize Some of Arabic Phonemes."

[5]. A. S. Hashim, W. A. Awadh, and A. K. Hamoud, "Student performance prediction model based on supervised machine learning algorithms," in IOP Conference Series: Mate-rials Science and Engineering, 2020, vol. 928, no. 3: IOP Publishing, p. 032019.

[6]. H. H. Chinaza Uchechukwu, and Jianguo Ding, "A Survey of Machine Learning Tech-niques for Phishing Detection," IEEE Access, August 2020.

[7]. P. Kalaharsha and B. M. Mehtre, "Detecting Phishing Sites--An Overview," arXiv pre-print arXiv:2103.12739, 2021.

[8]. B. Sabir, M. A. Babar, R. Gaire, and A. Abuadbba, "Reliability and Robustness analy-sis of Machine Learning based Phishing URL Detectors," IEEE Transactions on Dependable and Secure Computing, 2022.

[9]. M. Almousa, T. Zhang, A. Sarrafzadeh, and M. Anwar, "Phishing website detection: How effective are deep learning- based models and hyperparameter optimization?," Security and Privacy, vol. 5, no. 6, p. e256, 2022.

[10]. H. Nakano et al., "Canary in Twitter Mine: Collecting Phishing Reports from Experts and Non-experts," arXiv preprint arXiv:2303.15847, 2023.

[11]. Q. Zhang, "Practical Thinking on Neural Network Phishing Website Detection Re-search Based on Decision Tree and Optimal Feature Selection," in Journal of Physics: Con-ference Series, 2021, vol. 2031, no. 1: IOP Publishing, p. 012062.

[12]. A. AlEroud and G. Karabatis, "Bypassing detection of URL-based phishing attacks using generative adversarial deep neural networks," in Proceedings of the sixth international workshop on security and privacy analytics, 2020, pp. 53-60.

[13]. M. Mijwil, O. J. Unogwu, Y. Filali, I. Bala, and H. Al-Shahwani, "Exploring the Top Five Evolving Threats in Cybersecurity: An In-Depth Overview," Mesopotamian journal of cybersecurity, vol. 2023, pp. 57-63, 2023.

[14]. A. A. E. K. Yassine El Hajjaji, and Abdellah Ezzati, "Phishing Attacks and Counter-measures: A Survey," IEEE Access, 2020.

[15]. P. R. Brandão and G. P. Matos, "Machine Learning and APTs."

[16]. N. Q. Do, A. Selamat, O. Krejcar, E. Herrera-Viedma, and H. Fujita, "Deep learning for phishing detection: Taxonomy, current challenges and future directions," IEEE Access, 2022.

[17]. M. H. A. a. A. A. Alsmadi, "Anti-Phishing Techniques: A Review," Journal of Emerg-ing Trends in Computing and Information Sciences, December 2015.

[18]. S. L. Xu Chen, Wei Wang, and Xiaodan Zhang, "A Real-Time Anti-Phishing Method Based on Online Learning and Semi-Supervised Learning," Journal of Computational Sci-ence, October 2021.

[19]. S. A. Anwekar and V. Agrawal, "PHISHING WEBSITE DETECTION USING MACHINE LEARNING ALGORITHMS."

[20]. S. Jain, "Phishing Websites Detection Using Machine Learning," Available at SSRN 4121102.

