Enhancing Phishing Universal Resource Locator Detection Systems Using Hybrid Machine Learning Classifiers and Data Balancing Techniques

Mohammed Ibrahim[1], Bello Muhammed Hadi[1], Muhammed Basheer Jasser[2], Samuel-Soma M. Ajibade[2], Athirah Mohd Ramly[3], Yoke Leng Yong[4], Bayan Issa[5]

[1]Dept. of Cyber Security, Nigerian Defence Academy, Kaduna, Nigeria

[2]School of Engineering and Technology, Sunway University, Bandar Sunway, 47500 Selangor Darul Ehsan, Malaysia

[3]School of Architecture, Computing and Engineering, University of East London, University Way, Docklands Campus, London E16 2RD, UK

[4]School of Information and Communication Technology, HELP University, Persiaran Cakerawala, Subang Bestari, Seksyen U4 40150 Shah Alam, Malaysia

[5] Faculty of Informatics Engineering, University of Aleppo, Syria

mohammedi@nda.edu.ng, muhambello@hotmail.com, basheerj@sunway.edu.my, samuelma@sunway.edu.my, a.mohd-ramly@uel.ac.uk, yokeleng.y@help.edu.my, bayan.issa.b@gmail.com

***Abstract*— Phishing attacks have become a significant threat to online security, with cybercriminals using sophisticated tactics to evade detection. Traditional phishing detection systems often relied on rule-based approaches, which can be limited in their effectiveness. This research uses machine learning (ML) to detect phishing universal resource locator (URL) by evaluating the performance of advanced classifier for URL phishing detection using both original and balanced datasets. The classifier assessed is Random Forest. In conjunction with data balancing techniques like SMOTE and Resample, on the original dataset, it achieved an accuracy of 98.10% with a Kappa statistic of 96.20%, requiring 1.93 seconds for training and 0.05 seconds for testing. For the balanced dataset, the performance slightly improved, achieving an accuracy of 98.20% and a Kappa statistic of 96.40%, with reduced training time of 1.30 seconds and testing time of 0.03 seconds. Hence, balancing original dataset can significantly improve URL phishing detections using ML.**

Keywords*— *classifier, dataset, Detection, Machine learning, Phishing.

I. INTRODUCTION

The term "phishing" was first introduced in 1996 to describe a form of online identity theft that emerged following an attack by hackers on America Online (AOL) accounts. The first phishing lawsuit was filed in 2004 against a California teenager who had created a fake version of the AOL website to deceive users into providing sensitive information, such as credit card details, resulting in significant financial losses for the victims[1]. Phishing attacks are a form of social engineering crime where users receive fraudulent emails that appear to be from reputable sources. These emails often request sensitive information, such as login credentials, which are then used by malicious actors to steal financial or personal information [2]. The prevalence of such attacks has been alarming, with the Anti-Phishing Work Group (APWG) reporting over one million phishing incidents recorded in the last three months of 2022. According to APWG data, more than 23% of these attacks targeted the financial sector, highlighting the urgent need for heightened security measures and awareness to combat this growing threat.

URL serves as an address that specifies the location of information and resources on the Internet [3]. With the exponential increase in internet users and the widespread adoption and migration of services to the online realm, the number of URLs has surged accordingly. This growth includes a significant rise in phishing URLs, which are malicious links designed to deceive users into divulging sensitive information.

In the past, phishing websites or URLs were relatively easy to detect, but as technology advances, they have become increasingly sophisticated. The characteristics of URLs have prompted the development of numerous methods for identifying phishing URLs; however, the effectiveness of these techniques varies. Researchers have focused extensively on improving the accuracy of phishing website detection through various approaches. Various classifiers, including K-Nearest Neighbor, Naïve Bayes, Support Vector Machines (SVM), Random Forest, and Artificial Neural Networks, among others, have been employed to train datasets for identifying phishing websites. These classifiers are categorized into probabilistic or machine learning techniques. Researchers have utilized these algorithms to address various challenges in phishing website detection. Evaluation typically involves metrics such as Precision, Recall, F1 Score, and Accuracy.

Phishing has emerged as a pervasive cyber threat in tandem with the rapid expansion of Internet applications, cloud computing, and mobile technologies in the 21st century [4]. This form of social engineering targets users to illicitly obtain sensitive personal information, exploiting the interconnected nature of billions of devices and users worldwide. The consequential rise in e-commerce, online banking, and digital interactions has amplified the incidence of phishing attacks [5], posing significant risks to individuals and organizations alike.

Despite advancements in technology and the deployment of various detection methods, phishing URLs have evolved to evade detection, becoming increasingly sophisticated over time [6]. This evolution necessitates robust defenses capable of accurately identifying and mitigating phishing threats. While machine learning classifiers such as Random Forest,

J48, and others have shown promise in detecting phishing websites [7], challenges persist, particularly in effectively handling imbalanced datasets and improving detection accuracy, because it is more complex to handle.

Furthermore, traditional cybersecurity measures are proving inadequate against the dynamic nature of phishing attacks [8]. The discrepancy between the effectiveness of current detection techniques and the escalating sophistication of phishing strategies underscores the urgent need for enhanced methodologies.

Also, traditional methods of phishing detection, such as rule-based systems and blacklisting, have limitations in keeping up with evolving attack strategies and patterns [9]. Nevertheless, machine learning in recent time offers more dynamic and adaptive approach in identifying phishing attempts by leveraging algorithms that learn from historical data and generalize patterns to detect new, previously unseen attacks [10].

Jain and Gupta [11] introduced a machine learning-based approach to detect phishing attacks, leveraging the analytical power of logistic regression (LR) classifiers. The methodology involves a comprehensive analysis of all hyperlinks within a website. The model achieved an impressive 98.42% accuracy, indicating that it correctly identified the status of websites (phishing or legitimate). The study suffered weaknesses that include limited feature set, data quality and representation as well as data bias.

Rashid et al. [12] proposed an innovative approach to detecting phishing webpages using machine learning techniques. Central to their methodology was the application of Principal Component Analysis (PCA) for feature selection. The selected features were then used to train a Support Vector Machine (SVM) classifier and the results demonstrated an impressive accuracy rate of 95.66%. This high level of accuracy underscores the efficacy of their approach in distinguishing between phishing and legitimate websites. Despite having promising results, the study overrelied on principal component analysis, dataset bias, and lack of hyperparameter tuning.

By combining lexical analysis and machine learning, Abutaha et al. [13] designed a technique that was implemented as a web browser plug-in which alerts users on an attempt to access potential malicious webpages. The plug-in would continue to analyze URLs as users navigate the web, providing real-time alerts and preventing access to potentially dangerous sites. Similarly, the work of Alrefaai et al. [14] which focused on machine learning was proposed to detect phishing websites.

Finally [7] conducted a study that enhanced the detection of phishing websites by identifying the most effective machine learning classifiers and feature selection methods. The findings of the study revealed that the RandomForest, FilteredClassifier, and J-48 classifiers excelled in identifying phishing websites. Specifically, RandomForest achieved an accuracy of 89.948% on the first dataset and 97.259% on the second dataset. However, despite that the results indicated from these classifiers and the InfoGainAttributeEval method are highly effective in the fight against phishing, the authors suggested that integrating the three top-performing classifiers into an ensemble model could further enhance the detection of phishing websites. Additionally, the paper recommend exploring the use of metaheuristic algorithms in future studies that can provide more efficient feature selection algorithms.

Although previous algorithms proofed to be efficient in detecting phishing websites, the algorithms failed to consider Dataset bias, evaluation metric limitations, model interpretability and data quality assumptions. Additionally, current methodologies often struggle with the inherent imbalance in phishing datasets, where the prevalence of legitimate URLs far exceeds that of phishing URLs, leading to skewed model performance and diminished detection accuracy[7].

Therefore, the objective of this paper to proposed a hybrid machine learning model that address these gaps by refining existing methodologies through the integration of advanced machine learning techniques and innovative data balancing strategies, such as SMOTE and Resample for imbalanced datasets.

Specifically, the research will explore a hybrid approach leveraging the best-performing classifiers identified by Alazaidah et al. [7], augmented by SMOTE and Resample techniques to address dataset imbalances. Additionally, the study will employ the InfoGainAttributeEval feature selection technique, with a focus on training and testing time, as timely detection is crucial for real-time phishing prevention. By implementing these methodologies, the study endeavors to enhance the efficacy and reliability of phishing detection systems, thereby bolstering cybersecurity defenses against evolving cyber threats.

The paper is organized into IV sections. Section II explain methodology, section III Outline the results and the conclusion is presented in Section IV.

II. METHODOLOGY

This section outlines the implementation of machine learning algorithms specifically designed for URL phishing detection, incorporating a hybrid approach to class balancing alongside feature selection techniques. The methodological process is as follows:

A. Tools and Environment

The primary tools used for experimentation include the Python programming language, which is known for its extensive libraries and frameworks suitable for machine learning and data analysis. Anaconda 3, a popular distribution of Python, was utilized to manage packages and dependencies efficiently. Jupyter Notebook served as the interactive environment for coding, visualization, and iterative analysis, allowing for seamless integration of code, results, and documentation.

B. System Configuration

The experiments were conducted on an HP Elite Book Gen8 system equipped with 16GB of RAM, Intel Core i7 and clock speed of 2.60GHz. This configuration was chosen to balance computational power and resource efficiency, ensuring that the machine learning models could be trained and tested within a reasonable timeframe without overwhelming system resources.

C. Dataset Source

The dataset used in this study was sourced from the UCI Machine Learning Repository, which is a well-known repository for high-quality datasets. The dataset used in this research is a comprehensive collection of phishing and legitimate URLs. The data is sourced from publicly available repositories and includes various features that characterize URLs. These features are crucial for training machine learning models to distinguish between phishing and legitimate URLs

D. Data Preprocessing

Data preprocessing is a critical step in the research methodology. It involves cleaning the dataset, handling missing values, and transforming the data into a format suitable for machine learning algorithms. Specific preprocessing steps include:

- Dropping Irrelevant Columns: Columns such as 'CLASS_LABEL' and 'id' are removed from the dataset as they do not contribute to the feature set.
- Scaling: Features are scaled using the Standard Scaler to ensure that they have a mean of zero and a standard deviation of one. This step is crucial for algorithms sensitive to feature scaling.
- Handling Class Imbalance: Since phishing datasets are often imbalanced, techniques such as SMOTE (Synthetic Minority Over-sampling Technique) and Resample are employed to balance the classes.

The data underwent extensive preprocessing to address issues such as missing values, noise, and inconsistencies. The Synthetic Minority Oversampling Technique (ADASYN/SMOTE) and under-sampling were employed to tackle the class imbalance problem. SMOTE generated synthetic samples for underrepresented classes, focusing on difficult-to-learn examples while under-sampling reduced the number of majority class instances to balance the dataset effectively.

E. Feature Selection

Feature selection aims to identify the most relevant features that contribute to the accurate detection of phishing URLs. The Info Gain Attribute Eval technique is used to evaluate the importance of each feature based on information gain. This method helps in selecting a subset of features that maximizes the performance of the classifiers while reducing the dimensionality of the dataset.

F. Model Training

Multiple machine learning classifiers are evaluated in this study, including Random Forest, Decision Tree, KNN, Naive Bayes, XGBoost, and ANN. The training process involves:

- Splitting the Data: The dataset is divided into training and testing sets to evaluate the performance of the models.
- Training the Models: Each classifier is trained on the training set using the selected features and balanced dataset.
- Hyperparameter Tuning: Hyperparameters are optimized using techniques such as grid search or random search to enhance the performance of the models.

G. Model Evaluation

The performance of the trained models is evaluated using various metrics, including Precision, Recall, F1 Score, and Accuracy. These metrics provide a comprehensive assessment of the model's ability to correctly classify phishing and legitimate URLs. Additionally, training and testing times are recorded to evaluate the efficiency of the models.

H. Hybrid Approach

Based on the performance evaluation, a hybrid approach is developed by integrating the most effective classifiers and data balancing techniques. This approach aims to combine the strengths of individual models to achieve higher accuracy and robustness in phishing detection. The hybrid model is validated on the testing set to ensure its effectiveness.

I. Implementation and Validation

The final step involves implementing the hybrid model and validating its performance on an independent test set. This validation ensures that the model generalizes well to unseen data and is robust against various types of phishing attacks.

J. Proposed Model

The proposed model implementation performs a comprehensive sequence of steps to handle data preprocessing, feature selection, class balancing, and classification using various machine learning algorithms. The proposed model can be implemented through the process depicted in Fig.1.

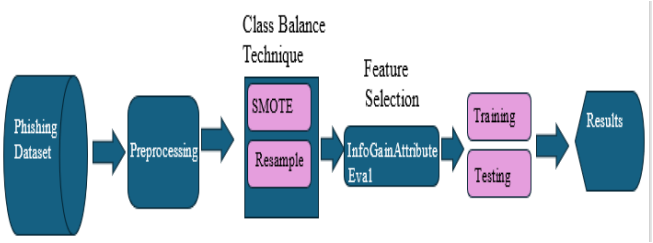

Fig. 1. Model Process Implementation

Initially, the dataset is loaded, and specific columns are dropped, followed by the extraction of the feature set X and target variable y. The features are then standardized using Standard Scaler, which scales the data to have a mean of zero and a standard deviation of one, thus improving the performance and convergence of many machine learning algorithms.

The data is split into training and testing sets using an 80-20 split. To select the most relevant features, 'Select KBest' with 'mutual_info_classif' as the scoring function is used. This technique selects the top k features based on mutual information, where K is set to 20, optimizing the dataset by retaining the most informative features.

Class balancing is crucial in this scenario due to the potential imbalance in the dataset. This is achieved using a combination of SMOTE and Random Under Sampler within a pipeline. SMOTE generates synthetic data for the minority

class to balance the dataset, while Random Under Sampler reduces the number of instances in the majority class, ensuring an equal representation of both classes in the training data.

Various classifiers are trained on the balanced dataset. These include Random Forest Classifier, Decision Tree Classifier, KNeighborsClassifier, GaussianNB, XGBClassifier, and a neural network implemented via Keras Classifier. Each classifier has its own set of parameters and configurations. Random Forest Classifier uses 100 estimators, which are individual decision trees, combined to improve predictive accuracy and control overfitting. Decision Tree Classifier relies on a single decision tree, which is simpler but may be prone to overfitting. KNeighborsClassifier uses a distance metric to classify instances based on the k-nearest neighbors, providing a non-parametric and straightforward approach to classification. GaussianNB applies Bayes' theorem with the assumption of Gaussian distribution of features, which is computationally efficient and effective for high-dimensional datasets. XGBClassifier is a powerful gradient-boosting framework known for its efficiency and performance in both regression and classification tasks. The neural network, implemented using Keras, consists of a Sequential model with three Dense layers. The first Dense layer has 12 nodes and uses the ReLU activation function, the second has 8 nodes also with ReLU activation, and the output layer has 1 node with a sigmoid activation function for binary classification. The model is compiled with binary cross entropy as the loss function and Adam optimizer for training, with metrics tracked on accuracy.

The model is trained on the resampled training data for each classifier, and predictions are made on the test set. Evaluation metrics such as accuracy, Cohen's kappa score, classification report, and confusion matrix are calculated to assess the performance of each classifier. Accuracy measures the proportion of correctly predicted instances, while Cohen's kappa score accounts for the possibility of agreement occurring by chance. The classification report provides precision, recall, and F1-score for both classes, offering a detailed performance breakdown.

Confusion matrices are visualized for each classifier, presenting the counts of true positive, true negative, false positive, and false negative predictions. Additionally, ROC curves are plotted for all classifiers, showcasing the trade-off between true positive rate and false positive rate at various threshold settings. The area under the curve (AUC) is also computed, indicating the classifier's ability to distinguish between classes.

In summary, the implemented model pipeline involves preprocessing, feature selection, class balancing, training multiple classifiers, and extensive performance evaluation, ensuring a thorough and robust approach to classification tasks in the given dataset.

K. Evaluation Metrics

Evaluation metrics play crucial roles in assessing the efficacy and reliability of algorithms across different studies, especially in the domain of URL phishing detection. These metrics serve distinct purposes, each shedding light on different aspects of model performance. Firstly, a true positive (TP) occurs when a prediction model correctly identifies a URL as phishing, aligning with the ground truth. This metric is pivotal in evaluating a model's ability to accurately detect malicious URLs, thereby protecting users from phishing attacks.

Conversely, a true negative (TN) reflects instances where the prediction model correctly identifies a URL as legitimate, meaning the prediction matches the actual absence of phishing. In the context of URL phishing detection, TN would indicate instances where the model correctly identifies URLs as safe, thus ensuring normal internet usage without unwarranted blocking.

On the other hand, a false positive (FP) arises when the model incorrectly predicts a URL as phishing when it is actually legitimate. This could lead to unnecessary blocking of safe websites, potentially disrupting user experience and access to valuable information. Lastly, a false negative (FN) occurs when the model incorrectly predicts a URL as legitimate when it is actually phishing. This would mean failing to protect users from potentially harmful websites, posing significant security risks.

These metrics—TP, TN, FP, and FN—serve as fundamental building blocks for constructing comprehensive evaluation metrics such as accuracy, precision, recall, and F1-score. They provide insights into how well a predictive model performs in distinguishing between phishing and legitimate URLs. By analyzing these metrics, researchers and practitioners can gauge the strengths and weaknesses of their models and make informed decisions regarding model improvements or adjustments in URL phishing detection and other cybersecurity fields.

1) Confusion Matrix

The confusion matrix is a fundamental tool in evaluating the performance of classification models, providing a clear and structured overview of predictions versus actual outcomes across different classes. It consists of four key metrics: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). This is represented in Table I.

TABLE I. CONFUSION MATRIX TABLE

	Predicted Positive	**Predicted Negative**
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

- True Positive (TP) refers to instances correctly predicted as phishing by the model.
- True Negative (TN) pertains to instances correctly predicted as legitimate by the model.
- False Positive (FP) involves instances incorrectly predicted as phishing by the model (actually legitimate).
- False Negative (FN) encompasses instances incorrectly predicted as legitimate by the model (actually phishing).

From the confusion matrix, several key evaluation metrics can be derived:

2) Accuracy Measures the proportion of correct predictions (TP and TN) out of the total predictions made by the model. It is calculated as

$$Accuracy = (TP + TN) / (TP + FN + FP + TN) \quad (1)$$

3) False Positive Rate (FPR): measures the proportion of legitimate URLs incorrectly classified as phishing out of all actual legitimate URLs. It is calculated as

$$FPR = FP / (FP + TN) \quad (2)$$

4) Sensitivity (Recall): measures the proportion of actual phishing URLs that are correctly predicted as phishing by the model. It is calculated as

$$Sensitivity = TP / (TP + FN) \quad (3)$$

5) Precision: Precision measures the proportion of URLs predicted as phishing that are actually phishing. It is calculated as .

$$Precision = TP / (TP + FP) \quad (4)$$

6) Specificity: Specificity measures the proportion of actual legitimate URLs that are correctly predicted as legitimate by the model. It is calculated as

$$Specificity = TN / (TN + FP) \quad (5)$$

7) Recall: synonymously used with sensitivity, measures the model's ability to correctly identify phishing URLs among all actual phishing URLs. It is the same as sensitivity:

$$Recall = TP/ (TP + FN) \quad (6)$$

8) Error Rate: Measures the proportion of incorrect predictions made by the model. It is calculated as It is equivalent to 1 minus Accuracy.

These metrics collectively provide a comprehensive view of a model's performance across different aspects of classification accuracy, making the confusion matrix a foundational tool in assessing and optimizing machine learning models for tasks such as URL phishing detection and beyond. By utilizing these metrics, researchers and cybersecurity experts can ensure that their models are robust and reliable, effectively distinguishing between malicious and safe URLs to protect users from phishing attacks.

III. RESULTS

This study builds upon the methodology of Alazaidah et al. (2024) by implementing SMOTE and under sampling techniques to address the significant class imbalance in the URL phishing detection dataset, complemented by the utilization of the InfoGainAttributeEval feature selection technique. Emphasis is placed on training and testing time, as timely detection of phishing is crucial for real-time applications. By employing these methodologies, the study aims to significantly enhance the effectiveness and reliability of phishing detection systems, thereby strengthening cybersecurity defenses against evolving cyber threats. The result of original and balance dataset is presented in Table II.

TABLE II. RESULT ON ORIGINAL AND BALANCE DATASET

	Original Dataset	**Balanced Dataset**
Random Forest	Accuracy: 0.9810; Kappa: 0.9620; TrainTime:1.9287 seconds; TestTime:0.0527 seconds	Accuracy:0.982 Kappa:0.9640; TrainTime: 1.3020 seconds; TestTime: 0.0271 seconds.

The Random Forest classifier consistently demonstrated robust performance across both datasets. In the original dataset, it achieved an accuracy of 98.10% with a corresponding Kappa statistic of 96.20%. This model's efficiency was notable, requiring 1.93 seconds for training and only 0.05 seconds for testing. Its classification report revealed balanced precision, recall, and F1-scores of 0.98 for both legitimate and phishing URLs

The confusion matrix on Random Forest algorithm on original dataset is depicted in Fig. 2. the confusion matrix for Random Forest on the imbalanced dataset shows that it accurately identifies 939 legitimate URLs and 965 phishing URLs. However, it misclassifies 49 legitimate URLs as phishing and 47 phishing URLs as legitimate, indicating its ability to distinguish between classes with relatively few errors despite the dataset's imbalance.

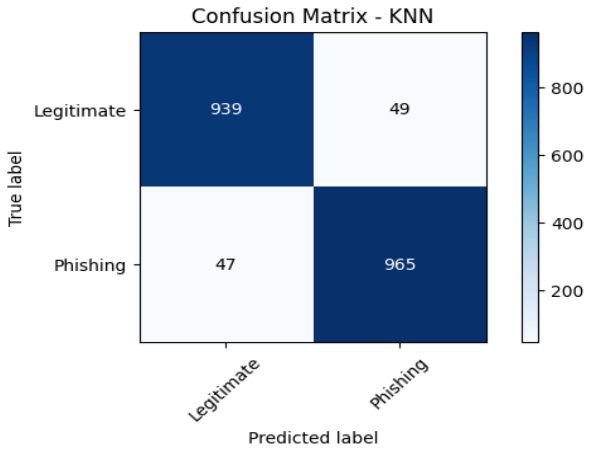

Fig. 2. Confusion matrix for Random Forest on ImBalance Data

Fig. 3. illustrates the confusion matrix for the Random Forest classifier on balanced data. It shows a clear distribution where 969 legitimate URLs are correctly classified, with only 19 misclassified as phishing. Similarly, 995 phishing URLs are correctly identified, with 17 misclassified as legitimate. This matrix underscores the Random Forest's robust performance in accurately distinguishing between legitimate and phishing URLs, with minimal errors.

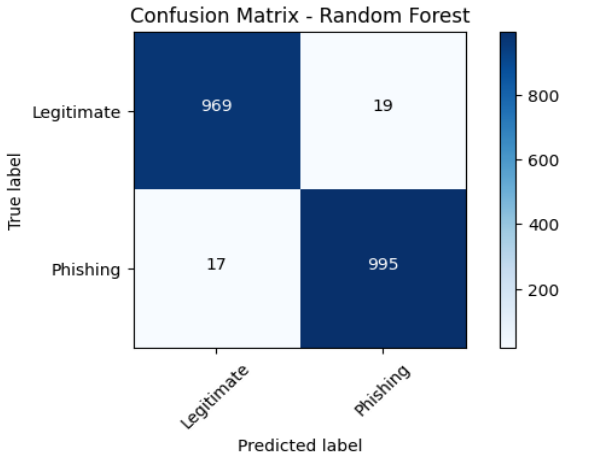

Fig. 3. Confusion matrix for Random Forest on Balance Data

IV. Conclusion

The balanced dataset approach improved the performance of random forest classifier towards URL phishing detection. This highlight the importance of addressing dataset imbalances in phishing detection. Random Forest have shown a robust performance and efficiency, making it suitable for deployment in various phishing detection systems. This model is recommended for URL phishing detection tasks, given its ability to effectively differentiate between legitimate and phishing URLs while maintaining high reliability in classification outcomes.

For future work, to improve the ability of our model to handle imbalanced datasets, we could use metaheuristic approach. As it proves to be effective in several domains [15, 16]. We have employed several swarm intelligence algorithms in several domains such as HHO [17, 18], GTO [19], DA [20], and Bat [21]. We could employ those algorithms to obtain better results in the application domain of this work.

References

[1] Musa, H., Gital, D. A. Y., & Zambuk, F. U. (2019). A Comparative Analysis Of Phishing Website Detection Using Xgboost Algorithm 1. Journal of Theoretical and Applied Information Technology, 15(5). www.jatit.org

[2] Junoh, A. K., AlZoubi, W. A., Alazaidah, R., & Al-luwaici, W. (2020). New features selection method for multi-label classification based on the positive dependencies among labels. Solid State Technology, 63(2s).

[3] Osho, O., Oluyomi, A., Misra, S., Ahuja, R., Damasevicius, R., & Maskeliunas, R. (2019). Comparative evaluation of techniques for detection of phishing URLs. Communications in Computer and Information Science, 1051 CCIS, 385–394. https://doi.org/10.1007/978-3-030-32475-9_28

[4] Alazaidah, R., Samara, G., Almatarneh, S., Hassan, M., Aljaidi, M., & Mansur, H. (2023). Multi-Label Classification Based on Associations. Applied Sciences, 13(8), 5081.

[5] Do, N. Q., Selamat, A., Krejcar, O., Herrera-Viedma, E., & Fujita, H. (2022). Deep learning for phishing detection: Taxonomy, current challenges and future directions. Ieee Access, 10, 36429-36463.

[6] Bhardwaj, A., Al-Turjman, F., Sapra, V., Kumar, M., & Stephan, T. (2021). Privacy-aware detection framework to mitigate new-age phishing attacks. Computers & Electrical Engineering, 96, 107546.

[7] Alazaidah, R., Alshaikh, A., Almousa, M. R., & Samara, G. (2024). Website phishing detection using machine learning techniques. Journal of Intelligent Information Systems, 63(1), 147-161.

[8] Nti, I. K., Narko-Boateng, O., Adekoya, A. F., & Somanathan, A. R. (2022). Stacknet based decision fusion classifier for network intrusion detection. Int. Arab J. Inf. Technol., 19(3A), 478-490.

[9] Gupta, B. B., Arachchilage, N. A., & Psannis, K. E. (2018). Defending against phishing attacks: taxonomy of methods, current issues and future directions. Telecommunication Systems, 67, 247-267.

[10] Sagar, R., Jhaveri, R., & Borrego, C. (2020). Applications in security and evasions in machine learning: a survey. Electronics, 9(1), 97.

[11] Jain, A. K., & Gupta, B. B. (2019). A machine learning based approach for phishing detection using hyperlinks information. Journal of Ambient Intelligence and Humanized Computing, 10, 2015-2028.

[12] Rashid, J., Mahmood, T., Nisar, M. W., & Nazir, T. (2020). Phishing detection using machine learning technique. In 2020 first international conference of smart systems and emerging technologies (SMARTTECH) (pp. 43-46). IEEE.

[13] Abutaha, M., Ababneh, M., Mahmoud, K., & Baddar, S. A. H. (2021, May). URL phishing detection using machine learning techniques based on URLs lexical analysis. In 2021 12th International Conference on Information and Communication Systems (ICICS) (pp. 147-152). IEEE.

[14] Alrefaai, S., Özdemir, G., & Mohamed, A. (2022, June). Detecting phishing websites using machine learning. In 2022 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA) (pp. 1-6). IEEE.

[15] L.S. Kong, M.B. Jasser, S.S.M. Ajibade and A.W. Mohamed, "A systematic review on software reliability prediction via swarm intelligence algorithms". Journal of King Saud University-Computer and Information Sciences, p.102132, 2024.

[16] B. A. S. Emambocus, M. B. Jasser and A. Amphawan, "A Survey on the Optimization of Artificial Neural Networks Using Swarm Intelligence Algorithms," in IEEE Access, vol. 11, pp. 1280-1294, 2023, doi: 10.1109/ACCESS.2022.3233596

[17] K. Jain, M.B. Jasser, M. Hamzah, A. Saxena and A.W. Mohamed, "Harris hawk optimization-based deep neural networks architecture for optimal bidding in the electricity market." Mathematics, 10(12), p.2094, 2022.

[18] B. A. S. Emambocus, M.B. Jasser, S.H. Tan, S.M. Ajibade, H.N. Chua, R.T. Wong and A.S. Rafsanjani, "An Optimized Harris Hawks Algorithm for Enhancing ANN Performance in Prediction Tasks Applied in Sales Domain," 2024 IEEE 14th International Conference on Control System, Computing and Engineering (ICCSCE), Penang, Malaysia, 2024, pp. 112-117.

[19] M. Abdel-Basset, R. Mohamed, M.B. Jasser, I.M. Hezam and A.W. Mohamed, "Developments on metaheuristic-based optimization for numerical and engineering optimization problems: Analysis, design, validation, and applications." Alexandria Engineering Journal, 78, pp.175-212, 2023.

[20] B.A.S. Emambocus, M.B. Jasser and A. Amphawan, "An optimized continuous dragonfly algorithm using hill climbing local search to tackle the low exploitation problem." IEEE Access, 10, pp.95030-95045, 2022.

[21] P. Kumar Mohapatra, S. Kumar Rout, S. Kishoro Bisoy, S. Kautish, M. Hamzah, M.B. Jasser and A.W. Mohamed, "Application of Bat algorithm and its modified form trained with ANN in channel equalization." Symmetry, 14(10), p.2078, 2022.