IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal
International Research Journal IJNRD Research Through Innovation

Phishing URL Detection Using Gradient Boosting: A Machine Learning Approach

M Ratnakar Babu[1], Pulimi Yashwanth[2], K Saketh Raja[3], Katta Ruthvik[4], Suryaneni Rohith[5]

Asst.Prof[1], Student[2], Student[3], Student[4], Student [5]

Artificial Intelligence[1],

Vidya Jyothi Institute Of Technology[1], Hyderabad, India

Abstract

Phishing attacks pose a significant threat, deceiving users into surrendering sensitive information. This study investigates the effectiveness of machine learning in detecting phishing URLs. We train and evaluate various models using a comprehensive dataset encompassing URL structure, website content, and external information. Exploratory data analysis identifies key features, and feature engineering further enhances model capabilities. The Gradient Boosting Classifier achieves a remarkable 97.4% accuracy in identifying phishing attempts. Analysis reveals that HTTPS presence, URL anchor text, and website traffic patterns significantly influence the model's decisions. We acknowledge the need for regular model updates due to evolving phishing tactics and emphasize the importance of user education as a complementary defense strategy. Future research avenues include exploring external data sources, investigating ensemble models, and continuously monitoring phishing trends for improved detection methods.

Key words: Phishing Detection, Machine Learning, Gradient Boosting, Feature Importance

I. INTRODUCTION

Phishing attacks pose both a surprise and a constant threat to the digital space. These attacks use deceptive techniques to trick users into revealing sensitive information or pretending to be legitimate websites to download malware. Modern methods of catching phishing attacks, such as blacklists, often fail to keep up with phishers' changes. In recent years, machine learning has emerged as a promising method to detect phishing URLs by analyzing various web features. In this article, we examine the effectiveness of various machine learning models in detecting phishing attacks. Our research uses extensive data on phishing websites to train and test models and compare their performance to determine the best strategy. Our findings have important implications for the development of more robust phishing systems.

II. RELATED WORKS

Phishing attacks remain a persistent threat in the digital landscape, tricking users into surrendering sensitive information or downloading malware. Traditional detection methods, such as blacklists, often struggle to keep pace with phishers' evolving tactics. This necessitates exploring alternative approaches, and machine learning has emerged as a promising solution.

Several studies have investigated the effectiveness of machine learning models in detecting phishing URLs. Vanhoenshoven et al. [2] explored various techniques, including Support Vector Machines (SVMs), achieving high accuracy in classifying phishing websites based on URL and content characteristics. Similarly, research by Sahoo et al. [3] provided a comprehensive survey of machine learning applications in malicious URL detection, highlighting the potential of this approach.

More recent advancements delve deeper into specific model architectures. Le et al. [4] introduced URLNet, a deep learning model that learns URL representations for improved malicious URL detection. Aljabri et al. [5] further emphasized the need for continuous research in this field, outlining potential future directions like incorporating social network analysis and user behavior data.Beyond specific model types, feature engineering also plays a crucial role in achieving high detection accuracy. Studies by Patil et al. [6] and Ha et al. [11] explored the effectiveness of various features derived from website URLs and content in enhancing the performance of machine learning models.

While these studies demonstrate the promise of machine learning in phishing URL detection, there's still room for improvement. Existing research often focuses on individual models or feature sets. This paper aims to contribute by conducting a comprehensive evaluation of various machine learning models for phishing detection, while simultaneously examining the key features that influence their performance.

III. METHODOLOGY

3.1 Data Acquisition

In our research, we used the phishing website verification database, which contains a comprehensive dataset from legitimate and phishing websites. This dataset can be accessed at data mendeley. This document provides a well-written process suitable for machine learning analysis. This file includes features such as URL templates, content analysis and external files. These features are carefully selected based on their relevance to phishing detection and their ability to distinguish legitimate websites from phishing websites. The file also includes a label for each website, indicating whether it is legitimate or phishing. This allows us to use the monitoring process to train and evaluate our machine learning models. The database contains a total of 11,055 samples, including 6,157 legitimate website samples and 4,898 phishing website samples. The data are well balanced; for example, approximately 55% belong to the legitimate category and 45% belong to the phishing category. This balance is important to ensure that our machine learning models can identify both types of websites.

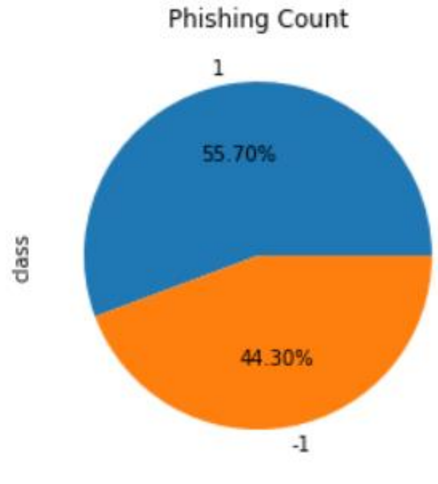

Fig.1 45% Phishing and 55% Non-Phishing (in dataset)

3.2 Exploratory Data Analysis (EDA)

Before deploying machine learning models, we conducted a thorough Exploratory Data Analysis (EDA) to gain a comprehensive understanding of the dataset. This analysis served as the foundation for our research, akin to meticulously examining a crime scene to identify crucial evidence. In this context, the "crime scene" represents the collection of phishing URLs, and the "evidence" represents the factors influencing their distribution.

Distribution Analysis: The initial phase of the EDA involved a detailed analysis of the data distribution. We examined the range and frequency of values for each feature, akin to meticulously sorting a collection of seashells by size and color. This process allows us to identify outliers or intriguing patterns that warrant further investigation. By doing so, we can uncover potential anomalies within the data that might necessitate additional exploration.

Addressing Missing Values: Next, we investigated the presence of missing values. Missing data points, similar to empty seashells on the beach, can significantly impact the performance of machine learning models. We meticulously combed through the data, searching for any missing values. If we encountered substantial gaps, we had contingency plans in place – techniques like imputation (filling in the blanks) or deletion (removing rows with missing data) – to ensure our models wouldn't be misled by incomplete information.

Feature Relationships: Our analysis extended beyond individual features. We delved deeper, investigating the relationships between features, searching for patterns or connections that influence the distribution of phishing URLs. Imagine closely examining the seashells to see if there's a link between their texture and the location where they were found on the beach. To uncover these hidden connections, we created visualizations like scatter plots and heatmaps (a visual map depicting feature interactions – see below). Additionally, we calculated correlation coefficients, which serve as quantitative measures of the strength and direction of these relationships.

Through this comprehensive EDA, we were able to identify the features that play the most critical role in how phishing URLs are distributed within the dataset. This newfound understanding of the data empowered us to not only choose the most relevant features for our models but also inform the specific training methods we employed. These methods will be explored in detail in the next section.

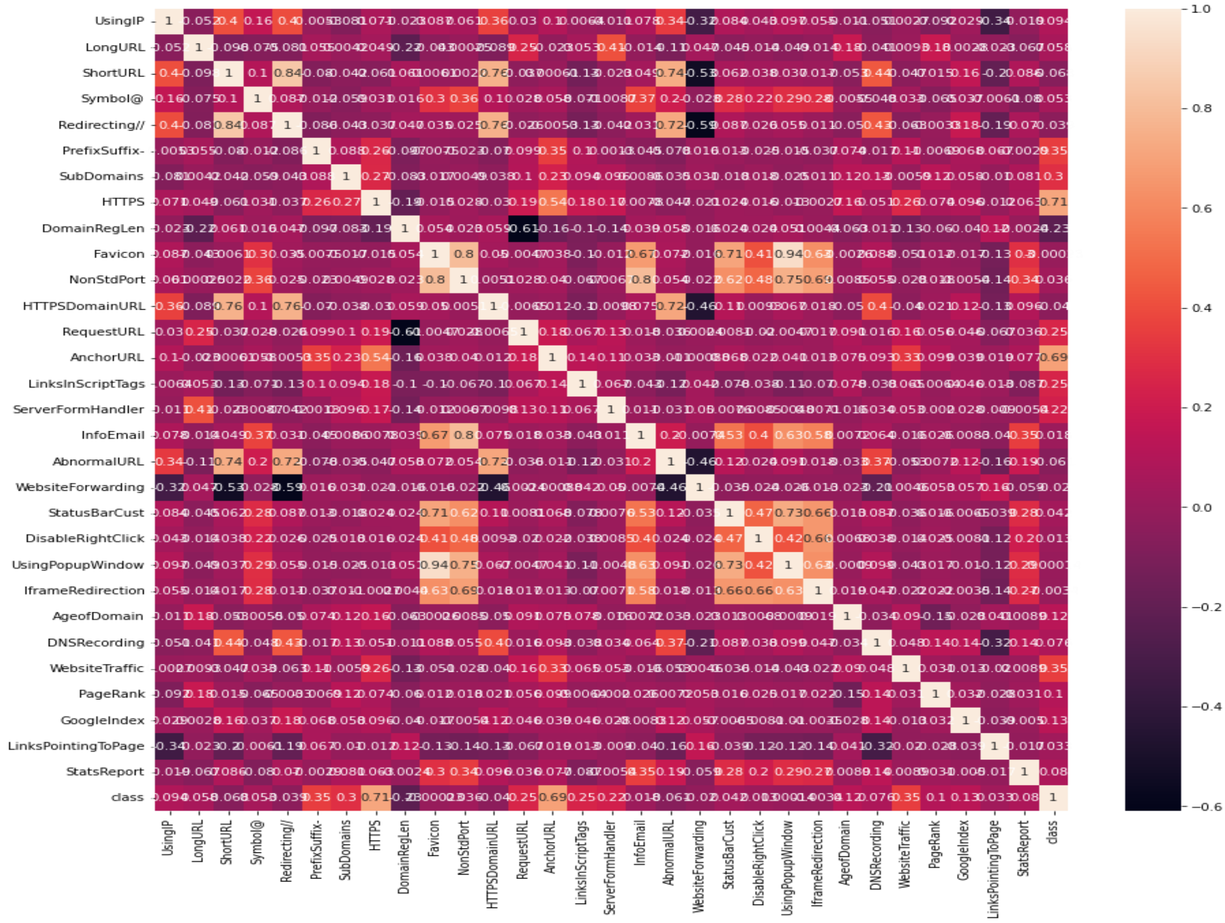

Fig.2 Correlation between features

The heatmap provides a visual representation of the relationships between various features extracted from the URLs. For instance, a dark red square might indicate a strong negative correlation, while a bright lite orange square might show a strong positive correlation. These relationships can provide valuable insights into how features work together to influence whether a URL is likely to be phishing or legitimate.

3.3 Feature Engineering

Our feature engineering strategy focused on extracting valuable characteristics from existing data to improve machine learning model performance for phishing detection. This involved technique's aligned with several references, including textual analysis of URL and HTML content [1, 15, 16]. We analyzed URLs to capture features like length, presence of suspicious keywords, and subdomain count, as mentioned in [1]. Additionally, we examined HTML content to remove text, specific characters, and dubious elements, potentially indicative of phishing attempts [15, 16].

Furthermore, we created new features based on relationships between existing ones. A crucial example is the ratio of internal to external links on a webpage, which can expose phishing sites that often lack substantial internal content [1, 16]. This approach aligns with research highlighting the importance of feature engineering based on these relationships for improved phishing detection [5].

By integrating these features into our machine learning model, we aim to increase its accuracy and robustness. In the following section, we describe selection and training methods for advanced phishing detection.

3.4 Machine Learning Models

To achieve the most effective phishing detection system, we assessed the performance of various machine learning models commonly employed in classification tasks. Our selection included:

Gradient Boosting Classifier (GBC): This ensemble method constructs a robust model by iteratively combining weak decision trees. Each tree focuses on correcting the errors of its predecessor, leading to improved prediction accuracy [18].

Random Forest: This ensemble learning technique generates multiple decision trees and aggregates their predictions for a more robust outcome. By introducing randomness during tree construction, random forests reduce overfitting and enhance generalization capabilities [19].

Support Vector Machine (SVM): A powerful supervised learning algorithm, SVMs excel in both classification and regression tasks. They function by identifying a hyperplane that optimally separates distinct classes within a feature space [14].

Decision Tree: This widely used algorithm leverages a tree-like structure to represent decisions and their consequences. The decision tree partitions the data space into regions and assigns predictions based on the dominant class within each region .

Logistic Regression: This statistical model serves as a foundation for binary classification. It estimates the probability of a data point belonging to a specific class (phishing or legitimate) based on its features [13].

Following feature extraction from the data described in Section 3.1, we trained and evaluated each model. The subsequent sections detail the feature selection and model evaluation processes undertaken to identify the optimal model for phishing URL detection performance.

3.5 Model Evaluation

We use classification metrics such as accuracy, precision, recall, and F1 score to evaluate the performance of each machine learning model. These metrics provide an overall measure of the model's ability to accurately identify phishing URLs.

Performance Metrics:

- **Accuracy:** This metric reflects the overall proportion of correct predictions made by the model. It tells us how often the model correctly classifies a URL as phishing or legitimate.
- **Precision:** Precision focuses specifically on the model's ability to accurately identify phishing URLs. It calculates the percentage of URLs flagged as phishing by the model that are actually malicious.
- **Recall (Sensitivity):** This metric looks at the flip side of precision. It tells us what percentage of actual phishing URLs the model successfully identified. A high recall indicates the model catches most phishing attempts.
- **F1-Score:** This metric strikes a balance between precision and recall, providing a single measure that considers both.

Consequences of Classification Errors:

It's crucial to consider the real-world implications of both misclassified URLs:

- **False Negatives (Missed Phishing Attempts):** Missing a real phishing attempt can be very serious. It could lead to stolen credentials, financial losses, and compromised data.
- **False Positives (Blocking Legitimate URLs):** While inconvenient, mistakenly flagging a legitimate URL as phishing creates user frustration and might require manual intervention. However, the damage is reversible and users can be notified of the mistake.

Feature Importance Analysis: Beyond basic performance metrics, we also conducted an analysis to understand which features in the URL data are most influential in each model's classification decisions. This helps us gain deeper insights into the model's reasoning and identify critical factors for accurate phishing detection.

Ensuring Reliable Results: K-Fold Cross-Validation

To ensure the reliability of our evaluation results, we employed k-fold cross-validation. This technique involves splitting the available data into k equal folds. For each fold, the model is trained on the remaining k-1 folds and evaluated on the held-out fold. This process is repeated k times, providing a more robust evaluation that reduces the impact of any specific data split. The final performance measure is the average of the k individual evaluations.

IV. RESULTS

4.1 Model Performance

Our testing results show that the gradient boost classifier outperforms other models in detecting phishing URLs with 97.4% accuracy. This high accuracy shows that the gradient boost classifier is effective in distinguishing legitimate URLs from phishing URLs.

Other models also perform well, with some models achieving accuracy scores of over 96%. However, the superior performance of the gradient boost classifier makes it a risky candidate for phishing URL detection.

In addition to accuracy, we also evaluated the model's performance based on accuracy, recall and F1 score. The gradient boosting classifier achieves high scores in all three parameters; This shows that classification decisions are not only accurate but also accurate and robust.The performance of the gradient boosting classifier can be attributed to its ability to resolve interactions between features and resistance to overfitting. This makes it a suitable choice for detecting phishing URLs where the location feature can be loud and noisy.

Overall, our testing results demonstrate the potential of machine learning models to detect phishing URLs. By leveraging the power of these models, we can improve effective phishing detection that helps protect users from phishing attacks.

ML Model	Accuracy	f1_score	Recall	Precision
Logistic Regression	0.934	0.941	0.943	0.927
K-Nearest Neighbors	0.956	0.961	0.991	0.989
Support Vector Machine	0.964	0.968	0.98	0.965
Naive Bayes Classifier	0.605	0.454	0.292	0.997
Decision Tree	0.961	0.965	0.991	0.993
Random Forest	0.967	0.97	0.992	0.991
Gradient Boosting Classifier	0.974	0.977	0.994	0.986
CatBoost Classifier	0.972	0.975	0.994	0.989
XGBoost Classifier	0.969	0.973	0.993	0.984
Multi-layer Perceptron	0.971	0.974	0.992	0.985

Table 1 Results of ML Algorithm's

4.2 Feature Importance

Our analysis shows that certain features have a significant impact on the classification decision of the gradient boosting classifier. In particular, the "HTTPS", "AnchorURL" and "WebsiteTraffic" attributes are considered the most important attributes for distinguishing legitimate URLs from phishing URLs.

The presence of HTTPS is often used as an indicator of security. Many users are trained to look for these features before accessing sensitive information. However, our analysis shows that the mere presence of HTTPS is not a reliable indicator of the legitimacy of the URL, as phishing URLs also use HTTPS to be more transparent.

The "AnchorURL" function refers to the text (anchor text) used in the anchor URL format. Our analysis shows that anchor text such as generic phrases like "click here" or "sign up" are frequently used in phishing URLs. Gradient-boosting classifiers can improve the accuracy of phishing detection by identifying anchor text used in URLs.

The "Website Traffic" feature refers to the traffic patterns of a particular website. Our analysis shows that phishing URLs often show poor traffic patterns, such as traffic from unusual sites. Gradient-boosting classifiers can identify phishing websites by analyzing network connection patterns and improve classification accuracy.

Overall, our analysis highlights the importance of considering various factors when checking for phishing URLs. Gradient-boosting classifiers can obtain more accurate search results by analyzing a combination of features such as HTTPS availability, anchor text, and web traffic patterns.

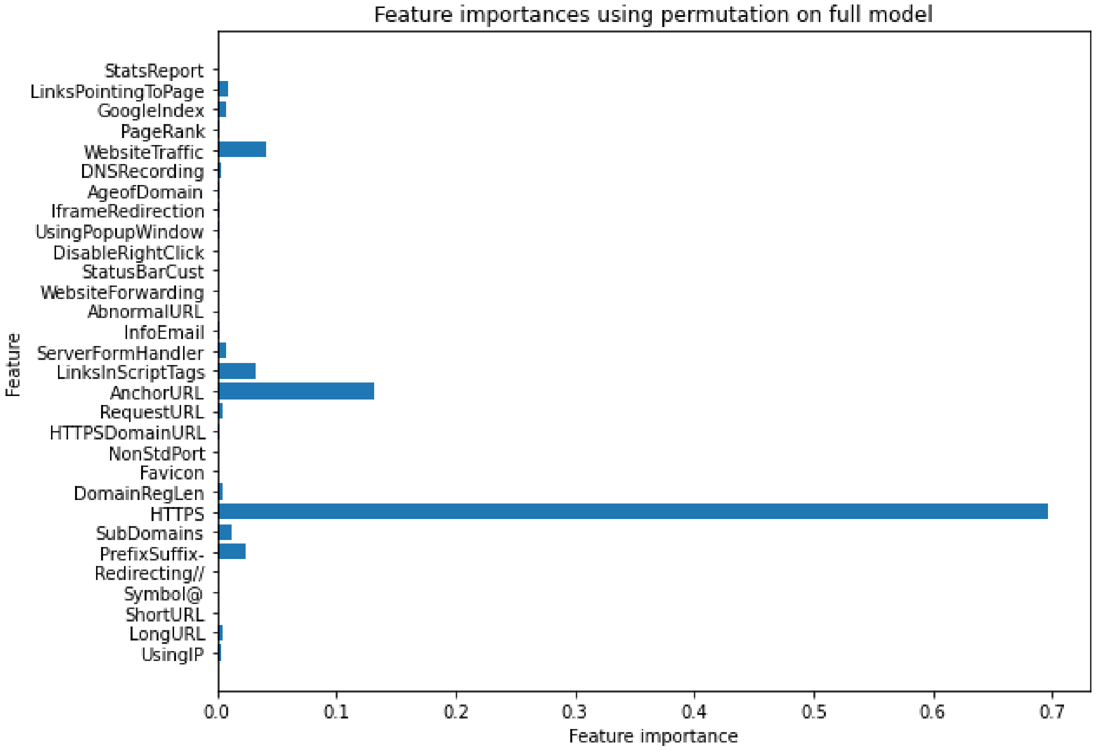

Fig.3 Feature importance compared to all features

V. DISCUSSION

The superior performance of gradient boosting classifiers in detecting phishing URLs demonstrates the potential of machine learning to solve this important security problem. The model can successfully distinguish real URLs from phishing URLs using a comprehensive set of methods.

Attribute key analysis provides insight into the characteristics of phishing URLs and shows that certain features, such as the presence of HTTPS, anchor text, and web traffic patterns, are good indicators of a phishing attempt. This information can inform the development of multi-target search strategies that focus on these key features.

However, it is important to acknowledge the limitations of our study. Phishing tactics continue to evolve, and features that are effective at detecting phishing now may not be as effective in the future. Therefore, it is important to regularly retrain the model with new information to ensure it remains current and effective at detecting phishing exercises.

Additionally, although our research focuses on using machine learning to detect phishing URLs, it is important to remember that this is only one anti-phishing strategy. Other methods such as user education and awareness are also important in preventing phishing attacks.

In summary, our study demonstrates the potential of machine learning in detecting phishing URLs, highlighting the importance of features such as HTTPS availability, transport link articles, and web traffic patterns. However, it is important to regularly update models with new information to maintain accuracy and use other methods as part of effective phishing protection.

VI. CONCLUSION

In this study, we investigate the effectiveness of machine learning in detecting phishing URLs. Our results show that the gradient boosting classifier achieves high accuracy, outperforming other methods in the field. Analyzing the key provides a better understanding of the characteristics of phishing URLs, which can inform the development of more sophisticated detection programs.

Future research may explore better engineering techniques to improve the performance of machine learning models. Combinations combining multiple models can also be checked to get more accurate results. Additionally, because phishing attacks are constantly changing, it is important to monitor the evolution of these trends for new and updated phishing strategies.

Overall, our research demonstrates the potential of machine learning in detecting phishing URLs. Using the power of machine learning, we can create the best and most targeted phishing strategies to protect users from these malicious attacks.

Further Work:

There are many opportunities for future work in this area. One direction is to investigate the impact of integrating data outside the website (such as Alexa Rank) on the performance model. This can provide more insight into the characteristics of phishing URLs and improve the accuracy of the model.

Another direction is to examine the effectiveness of a common system combining different types of machine learning. These combinations have been shown to improve the performance of machine learning models in many domains, and it would be interesting to investigate their potential in detecting phishing URLs.

Finally, using the proposed method is important for future studies. Such a system can help protect users from phishing attacks by quickly identifying and configuring suspicious URLs. This requires resolving issues related to efficiency, effectiveness, and adapting to new and evolving phishing tactics.

Overall, there is great potential for further research in this area with the goal of developing effective and robust phishing detection methods using machine learning.

REFERENCES

1. Fortinet: What is URL phishing? (2023). https://www.fortinet. com/resources/cyberglossary/url-phishing
2. Vanhoenshoven, F., Nápoles, G., Falcon, R., Vanhoof, K., & Köp- pen, M. (2016). Detecting malicious urls using machine learning techniques. In: IEEE Symposium series on computational intel- ligence (SSCI), pp. 1–8
3. Sahoo, D., Liu, C., & Hoi, S.C. (2017). Malicious url detection using machine learning: A survey. arXiv preprint arXiv:1701. 07179
4. Le, H., Pham, Q., Sahoo, D., & Hoi, S.C. (2018). Urlnet: learning a url representation with deep learning for malicious url detection. arXiv preprint arXiv:1802.03162.
5. Aljabri, M., Altamimi, H.S., Albelali, S.A., Maimunah, A.-H., Alhuraib, H.T., Alotaibi, N.K., Alahmadi, A.A., Alhaidari, F., Mohammad, R.M.A., & Salah, K. (2022). Detecting malicious urls using machine learning techniques: review and research direc- tions. IEEE Access.
6. Patil, D. R., & Patil, J. B. (2018). Malicious URLs detection using decision tree classifiers and majority voting technique. *Cybernet-ics and Information Technologies*, *18*(1), 11–29.
7. Hieu Nguyen, H., & Thai Nguyen, D. (2016). Machine learn-ing based phishing web sites detection. In: AETA 2015: Recent advances in electrical engineering and related sciences, pp. 123–131.
8. Yahya, F., Isaac W., Mahibol, R., Kim Ying, C., Bin Anai, M., Frankie, A., Sidney, Ling Nin Wei, E., &

Guntur Utomo, R. (2021). Detection of phising websites using machine learning.

9. Alkhudair, F., Alassaf, M., Khan, U. R., & Alfarraj, S. (2020). Detecting malicious url. In *2020 International conference on com-puting and information technology* 1, 97–101.
10. A. Waheed, M., Gadgay, B., DC, S., P., V., & Ul Ain, Q. (2022). A machine learning approach for detecting malicious url using dif- ferent algorithms and NLP techniques. In: *2022 IEEE North Kar-nataka Subsection Flagship International Conference (NKCon).*
11. Ha, M., Shichkina, Y., Nguyen, N., Phan, T.-S. (2023). Classifica-tion of malicious websites using machine learning based on url characteristics. In *Computational Science and Its Applications - ICCSA 2023 Workshops*, pp. 317–327
12. Urcuqui, C., Navarro, A., Osorio, J., & García, M. (2017). Machine learning classifiers to detect malicious websites. *Pro- ceedings of the Spring School of Networks*, *1950*, 14–17.
13. Chiramdasu, R., Srivastava, G., Bhattacharya, S., Reddy, P.K.,& Gadekallu, T.R. (2021). Malicious url detection using logistic regression. In: *2021 IEEE International conference on omni-layer intelligent systems (COINS)*, pp. 1–6.
14. Cristianini, N., & Ricci, E. (2008). Support vector machines. Springer.
15. Aaron Blum, Brad Wardman, Thamar Solorio, and Gary Warner. 2010. Lexical feature based phishing URL detection using online learning. In Proceedings of the 3rd ACM Workshop on Artificial Intelligence and Security. ACM, 54–60.
16. Davide Canali, Marco Cova, Giovanni Vigna, and Christopher Kruegel. 2011. Prophiler: a fast filter for the large-scale detection of malicious web pages. In Proceedings of the 20th international conference on World wide web. ACM, 197–206.
17. Jian Cao, Qiang Li, Yuede Ji, Yukun He, and Dong Guo. 2014. Detection of Forwarding-Based Malicious URLs in Online Social Networks. International Journal of Parallel Programming (2014), 1–18.
18. Friedman, J. H. (2001). Greedy function approximation (technical report No. Stanford University, Department of Statistics).
19. Breiman, L. (2001). Random forests. Machine learning, 45(3), 5-32.