International Journal of Scientific Research and Engineering Development-– Volume 9 Issue 2, Mar-Apr 2026
Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

A Feature-Free Deep Learning Approach for Detecting Phishing Attacks

Somendra Pratap Singh, Vinit Khandelwal , Ajit, Nitin Phulwani
(Department of Computer Science, Poornima Institute Of Engineering and Technology Jaipur , Rajasthan, India
Email: 2022pietcssomendra164@poornima.org)
(Department of Computer Science, Poornima Institute Of Engineering and Technology Jaipur , Rajasthan, India
Email: 2022pietcsvinit181@poornima.org)

----------**********----------

Abstract:

Phishing attack is a prevalent and a growing cyber threat that targets internet users, governments and service pro-viding organizations. In these attacks, attackers are trying to gain access to sensitive user information-including, but not limited to, login credentials, bank or card information, and email addresses-using deceptive email messages or fraudulent websites. Such attacks are frequently based on a social engineering approach, where hackers mimic legitimate web sites and deliver harmful URLs as spam messages, text messages, or in social media. In order to offer a holistic insight into the phishing attacks, this paper offers a literature review of diverse Artificial Intelligence (AI) techniques applied in detecting them such as: Machine Learning, Deep learning, Hybrid learning and Scenario based methods. It also juxtaposes various studies in phishing detection, outlining each of the methods, their advantages, and drawbacks. Also, the paper determines the current limitations of phishing detection and provides the possible future research directions in this field.

Keywords — **Artificial intelligence (AI), cyber security, extra trees, phishing, phishing website detection, meta– learners.**

----------**********----------

I. INTRODUCTION

This document is a template. An electronic copy can be downloaded from the conference website. For questions on paper guidelines, please contact the conference publications committee as indicated on the conference website. Information about final paper submission is available from the conference website.

The unique aspects of machine learning, including between identifying and extrapolating trends and changing to a new one environment, so that it can be an important component of technological systems like nuclear power plants monitoring, cyber and home-computer vision, land security and IoT (Internet of Things), to name a few. The concept of cyber security is used to refer to the management and design of technologies, tools and methods needed to protect against data, devices and information. It covers different issues of computer and network security such as Intrusion Detection System (IDS), Anti-virus, Phishing etc. Phishing is an irresponsible manner of cyber-hacking that is plaguing the internet which has a direct effect on physical world. Phishing is now a familiar subject-matter and the impact of successfully conducting phishing attacks has been reported to be disastrous. A number of studies have been carried out to prevent, mitigate and even to undo phishing attacks. Majority of the researches are concentrated on running alternative machine learning models, deep learning models and/or the combinations of the models. In the authors carried out a study in detail on how to build [3].Potential cyber security systems that are based on machine learning. Re-phishing URLs are likely to get enhanced by searchers and

security analysts detection systems based on machine learning and deep learning models algorithms. In the long-term, phishing opponents have too crossed their horizons (i.e. targeted other end equipments) and enhanced their offensive strategies. In [2], the authors conducted a study to put into the limelight the phishing attacks implemented both on mobile devices and defence mechanisms and existing challenges. More so, machine learning and deep learning models do so analytically with the aim of utilizing statistical models to perform some work well without being given any external direction, they nevertheless inaccuracy in the tasks of those particular types, which leads to misclassifications.

II. RELATED WORK

An In this section, the phishing attack would be discussed in the cyber security environment. Other than that, a detailed review of existing tools which have been used to detect phishing activities is discussed. Phishing detection systems typically employ a number of methods of detection and prediction phishing sites. The main techniques are the following:

A. Lists and heuristic-based techniques.

In such strategies, the detection systems are based on the set of rules and have white lists or blacklists to identify a URL as phishing or a legit one. The biggest weakness of such techniques is, however, that they cannot observe newly created phishing sites, also referred to as zero-day attacks. Moreover, these systems need to be updated on their rules and lists regularly so that they can be effective.

The other negative effect is that websites that look or have similar content to those in blacklists or white lists are easily misclassified because of limitations in heuristics. In a study, the authors suggested a phishing URL detection technique, grounded on the string matching, which placed a lot of reliance on blacklists and was costly to compute. Designed as the solution to blacklist based detection systems, the method failed to work with zero-day attacks and took a lot of time to calculate results and produced rather poor performance.

B. Content based techniques

This method examines the text of a webpage to determine it to be phishing or legitimate. One of the main weaknesses is however that it needs access to the source code or entire content of the site; both text and images to extract and analyze features. This dependency causes the approach to be computationally expensive and less practical in most practical cases, with increased total time consumption.

The authors came up with a phishing detection mechanism based on webpage, which demonstrated enhanced performance, but at a high cost in terms of time. Moreover, they used stacked models i.e. Gradient Boosting Decision Trees (GBDT) to make initial and final predictions and this created a possibility of bias in the results. The other method was based on the extraction of hyperlinks-based attributes of webpage source codes. Because this technique is solely based on examining source code properties, bigger source code leads to a higher time complexity, which in turn has an additional impact on efficiency.

C. Third-party based techniques

There are phishing detection methods that are based on third-party functionalities and external services. A significant disadvantage with these methods though is they have high misclassification. This is the main problem, as they are greatly affected by such factors as domain age or frequency of search results. Therefore, legitimate websites that have just been created can be mistakenly regarded as phishing sites.

Moreover, such methods are susceptible to threats of third-party services, including bias or compromise. As an illustration, some attacks such as the DNS (Domain Name System) spoofing attacks may tamper with information supplied by these services thus lowering the level of reliability of the detection system.

D. Lexical features-based techniques

Moreover, such methods are susceptible to threats of third-party services, including bias or compromise. As an illustration, some attacks such as the DNS (Domain Name System) spoofing attacks may tamper with information supplied by

these services thus lowering the level of reliability of the detection system.

The authors in one of the studies were able to come up with a phishing detection system based on the lexical feature analysis with an accuracy of 94.91 percent. An additional piece of work also enhanced performance of phishing URL detection with better lexical features. Their model was highly accurate and less time consuming because it did not rely on third-party services or source code analysis and was able to achieve a high accuracy of approximately 95 percent. Moreover, a detection system that utilized seven various machine learning algorithms performed well. Through selectivity of type of features, their methodology overcame issues like language dependence, using third party information and facilitated successful identification of real time and zero-day phishing attacks.

E. Hybrid features-based techniques

Attempts to enhance blacklist maintenance, as well as methods relying on HTML content analysis, external features and TF-IDF, continue to have a number of limitations. In one of the methods, the authors used a combination of several techniques, such as analysis of the source code, white list verification, third-party services and page similarity with the help of screenshots. None the less, this high dependency on various parts rendered the system time consuming and less effective.

In a different work, even though the authors wanted to improve traditional blacklisting and heuristic-based methods of detection, their solution relied on host-based properties, thus inheriting the shortcomings of host-based solutions.

Forming these observations, one may conclude that methods based on lexical features are more effective and have less limitations. Thus, the suggested detection system will be developed based on lexical feature extraction and analysis.

III. PROPOSED METHODOLOGY

A number of phishing detection methods are based on third-party features and third-party services. Nevertheless, one of the major drawbacks of these approaches is they misclassify.

The main problem with such methods is that they rely largely on such variables as domain age, frequency of search engine ranking. Consequently, legitimate websites that have just been created can be wrongly perceived as phishing sites.

Furthermore, the use of third-party services brings with itself other risks, such as the possibility of bias or even a security breach. Examples of these attacks include DNS (Domain Name System) spoofing, which can affect the integrity of these systems.

In order to reduce these shortcomings, lexical feature based approaches are embraced. The techniques are aimed at studying the structural features of URLs, removing the reliance on external services and enhancing efficiency and reliability in detecting phishing.

A. Stage 1: URL HIT

Upon input of a URL, it gets redirected to the proposed detection system to be initially processed. The system then creates a reply to the specified URL. When the URL is a short (mini) link, it is elongated to the original one, and in other cases, the URL is kept the same.

After this pre processing is done, the processed URL is sent to the next component which is the Feature Extractor to further analyse it.

B. Stage 2: FEATURES EXTRACTOR

At this stage, the system carries out extraction of features on the input URL. The original URL is first used to extract features, then a larger (long URL) resulting in the first extraction is used to extract features.

This expanded URL is then broken down into five major parts; segment, netloc, path, query and fragment. All these parts are further broken down and applied to extract any features.

To achieve this, regular expressions are used in order to extract and find useful patterns out of the various components of the URL. The end result of this subcomponent is a well-organized collection of features that are extracted and are further processed in the detection system.

C. Stage 3: MODELICS

During this phase, there is the formation and execution of several machine learning models as parallel threads. The results of the last step (Feature Extractor) are given as input to this component.

This module is also known as Model Implementation and it has ten various machine learning models. These models are categorized into three models which include boosting based models, learning-based models and non-learning-based models. This classification should add some diversity to the decision-making process and selection strategies of features, which will contribute to the overall strength and functioning of the detection system.

• Boosting-based approach

This category of classifiers has its foundation on ensemble learning which involves the integration of several weak learners to create a strong predictive model in the form of voting.The main aim of boosting techniques is to enhance overall model performance by concentrating on poorly classified ones in the past. AdaBoost and Gradient Boosting classifiers are used in this study because these are effective.

AdaBoost Classifier operates through repetitively modifying the weighting of samples in training. It gives greater weight to cases that are hard to classify and the next weak learners can pay more attention on these hard cases. A weighted majority vote by all weak learners is used to get the final prediction with each learner contributing based on its accuracy.

Gradient Boosting Classifier tries to reduce the total error of prediction by successively adding weak learners. It employs gradient optimisation to minimize the loss function such that each subsequent model aims at correcting the mistakes committed by the previous ones. This refinement process will better handle misclassified samples and will result in an overall increase in the accuracy of the model.

• Non-learning-based approach

The classifiers in this category tend to be less overfitting and are stable in performance across datasets. The models that have been taken into consideration in this study include Decision Tree, Random Forest, Extra Trees, Bagging and K-Nearest Neighbour (KNN) as they are effective and various in terms of data handling.

Decision Tree takes into account all the characteristics of the data when building the model. It divides the data according to the feature values to create a tree like structure. Though interpretable and simple, it is likely to be highly variable.

Random Forest is an ensemble approach which constructs a series of decision trees using random selections of a subset of features and data samples (with replacement, i.e. bootstrap ping). The last prediction is made by taking the combination of the output of all trees leading to lower variance as compared to using a single decision tree.

Extra Trees Classifier (Extremely Randomized Trees) also uses random feature subsets; it however, unlike Random Forest, splits are selected randomly and most do not use boot-strapping. This randomization also enhances generalization and decreases variance.

Bagging Classifier splits the data into several subsets, which are sampled with replacement and models are trained on each subset. The predictions are aggregated to obtain the final output and this assists in lowering the variances and enhancing stability.

K-Nearest Neighbour (KNN) is a distance-based algorithm which classifies data points in clusters depending on a similarity of features. In case of a new sample, it classifies it using the majority of its closest neighbours in the feature space.

D. Stage 4: DECISION MAKER

At this step, the results of the different classifiers are fused to come up with the final decision on whether a URL is phishing and legitimate or not. The models in the earlier stage are diversified and used in the decision-making process to enhance the overall accuracy and robustness.

Extra Trees models, Bagging and K-Nearest Neighbour (KNN) models are some of the models that contribute to this process in their unique ways. The Extra Trees classifier adds some more randomness by randomly choosing features and split points without replacement, leading to low variance, and improved generalization. The Bagging classifier minimizes variance by using more than one model that is trained on various segments of the data and then combines the predictions. Meanwhile, KNN uses the similarities of a new sample to the existing data points in the feature space to classify it.

Combining the forecasts of these various models, the Decision Maker will have a more reliable and accurate classification, which will effectively minimize the mistake and enhance the system to recognize phishing URLs. In the last step, the prediction results of the last subcomponent (Model Implementation) are given as input to the Decision Maker. The results achieved by all the applied models are summed up by this subcomponent.

The system makes the final decision based on the composite predictions of whether the input URL is a phishing or legitimate URL. This step makes the overall detection process more reliable and accurate since the results of various models are incorporated.

IV. EXPERIMENTAL SETUP

A. Simulation Environment

The tests were done on Linux Ubuntu 16.04. Python 3.5.4 was used to develop the proposed detection system. To extract features and analyze the data, a number of Python libraries were used such as nltk, re, numpy, matplotlib, wordcloud, collections (Counter), plotly, urlparse, and dnstwist.

To execute the Model Implementation (Modelics) component, libraries like pandas, sklearn (sklearn), threading, and seaborn were used to facilitate machine learning processes and parallel processing.

B. Data Preparation

In order to attain the main goal of modeling AI-generated phishing URLs, 50,000 phishing URLs were generated using the model.. In the case of valid URLs 50,000 normal URLs were gathered out of Alexa rankings. Moreover, 50,000 easy phishing URLs were retrieved on PhishTank to measure performance on the traditional phishing attacks.

To eliminate bias, the datasets were initially pooled together and randomly shuffled. The data was then split into training and testing sets, through the hold-out method. Cross-validation based on K-folds was not used because there was a limited variability in the pattern of URLs generated by AI which would result in a biased generalization. The hold out method contributed to the introduction of diversity and mitigating overfitting and over-generalization risks.

To ensure consistency in assessment, the same experimental procedure was used when dealing with the AI-generated phishing URLs and with simple phishing URLs.

C. EVALUATION METRICS

Risk reduction is an imperative issue in cybersecurity. Thus, it is necessary to choose the right evaluation metrics that will help to determine the performance of the models. As phishing attacks may result in a great loss of finances and data, the system focuses on identifying malicious URLs in the best way possible.

In order to measure performance, the Sensitivity (True Positive Rate or Hit Rate) is selected as one of the key measures as it provides the potential of the system to detect the phishing URLs correctly.

1) THEORETICAL ANALYSIS :

The theoretical analysis is aimed at confirming the hypothesis, presented in the introduction, that the system, PhishHaven is capable of identifying AI-generated phishing URLs.

In order to prove this hypothesis, two propositions are created:

• Proposition 1: The system can be able to identify accurately shortened (mini) URLs.

• Proposition 2: The chosen lexical properties have invariance that allows the system to be dependable in detecting AI-generated phishing URLs.

These hypotheses are the theoretical basis of the efficiency of the suggested strategy.

2) EXPERIMENTAL ANALYSIS:

The theoretical results are tested by the experimental analysis. The experiments will aim to:

• Test the performance of the chosen machine learning models in an ensemble framework with the help of the multi-threading.

• Test the system with an AI-generated phishing URLs and the simple phishing URLs.

• Compare the suggested method with the current lexical feature-based phishing detection systems.

Moreover, the system is enhanced by parallel training and testing of ensemble models to enhance the computational efficiency and speed up the general URL classification.

V. RESULTS AND DISCUSSIONS

In this part, the results obtained based on the experimental framework are examined theoretically to determine the diagnostic efficiency, interpretability, and formal reliability of the presented multi-stage phishing detection system. The talk highlights logical modeling, invariance proofs, and the shift towards interpretable artificial intelligence methods to subjugate the changing phishing threats.

A. AI Meta-Learners Framework.

The system suggests that phishing is a classification problem that can be addressed using AI because the goal here is to effectively differentiate the legitimate and malicious entities by an optimized decision-making process.

The weakness of single classifiers is one of the main theoretical assumptions of this study[13].The conventional standalone models are not effective enough to deal with dynamically changing phishing patterns, which tend to produce less detection and increased false positives. On the contrary, ensemble and hybridized methods exhibit greater strength and flexibility.The other important feature is the transition to interpretable AI. In contrast to traditional black-box models, interpretable systems present the transparency of decision-making, which is becoming a key requirement in contemporary cyber security models. This transformation guarantees accuracy and, most importantly, trust and explainability in real-life applications[14].The framework also explains how phishing attack vectors have evolved, now being found not just in emails and SMS, but in QR codes, faked mobile apps, and AI-generated malicious URLs. This requires intelligent and dynamic detection systems.

B. Formal Propositions of Theory.

In order to confirm the soundness of the offered system, the formal theoretical propositions are made:

Propositions 1: URL Hit Bijection.

The URL Hit mechanism is presented as a bijective function on which shortened URLs and expanded versions are in the one-to-one correspondence. This makes it possible to reliably map every obfuscated or small URL to its original destination.

The algorithm works without any previous information about URL shortening services, and it uses the standard browser redirection. The likelihood of retrieving the same expanded URL is consistent (near to 1) under the assumption of an active server thus making sure that its behavior is deterministic.

Proposition 2: Invariant Lexical Features

The study introduce s the idea of lexical feature invariance, holds that there are structural components that are universal to all permissible URLs. The paper questions the current methods and determines that they have a number of theoretical drawbacks:

Model Constraints: Statistical models fail to induce non- linear relationship, which limits their application to complex phishing environment. Black-Box Nature: Not all AI systems are interpretable, and thus their decision-making process is not transparent and is hard to trust. Single Classifier Weakness[7].Single classifiers have lesser adaptability to changing threats. Heuristic and List-Based Failures: Heuristic-based and List Based failures are also opposed to zero-day attacks and need frequent manual updates. Theoretical properties of phishing attacks.

Phishing is conceptualised as a kind of social web engineering assault, where the attackers manipulate the behaviour of users to obtain sensitive data.

The attack environment has drastically changed and now involves sophisticated methods that include AI-generated URLs, phishing with the help of QR codes, and bad apps. One of the paradigm shifts is the creation of the so-called Offensive AI, where hackers use deep learning models to create ultra realistic and adaptive phishing materials.

Such AI-generated URL possesses specific theoretical features, such as:

Absence of port numbers Listing of the use of the double forward slashes. Very complicated query parameters with false tokens.

These trends underscore the growing complexity of phishing.

C. URL Structural and Logical Analysis.

The suggested model uses the structural integrity of URLs to be detected.

Lexical invariance makes sure that there are elements that are always the same in all of the valid URLs and they are used to create strong classification. This method is further increased by the URL Hit functionality that ensures the expansion of shortened links is always consistent.

Besides, the theoretical difference between human-made and AI-generated URLs is also found by analyzing the distributions of features, which allow finding them more accurately.

D. Metacognition : Theory of Meta-Learner.

The use of AI-driven meta-learning is set as the core of the present-day phishing mitigation practices. Meta-learners improve the accuracy and generalization abilities by aggregating a number of models.

Machine learning techniques are more efficient in being able to deal with changing and dynamic patterns of attack than traditional methods. The implementation of meta-learners allows detecting efficiently and at a large scale, thus they are very applicable in cyber security applications in the real world.

E. Discussion of Machine Ethics.

Although AI has a vital role in cyber security, the paper accepts that machine ethics is not the solution to solve all social issues of phishing attacks. Human consciousness and conduct are part and parcel of integrated security solutions[6].

Theoretical Analysis Conclusion.

Altogether, the theoretical framework justifies the strength, flexibility, and scalability of the suggested system. Through combining feature analysis invariants, formal logic proofs, and interpretable AI principles, the model puts in place a formidable basis in identifying present and future phishing threats by the use of AI.

VI. LIMITATIONS AND FUTURE WORK

The research is able to establish a very powerful methodology. to define black-box model and be formal correct guarantees, but there are still a number of vital restrictions.

Theoretical Foundations of Detection the study changes the focus of a basic pattern matching to a complicated AI-based classification issue[8]. It is aimed at replacing the black-box models with the interpretable AI that is becoming an increasingly important concern to the cyber security community. The Logic of URL Invariance One of the key theoretical pillars of the second study is the invariance of specific lexical features. Necessary Elements: These elements are taken to be building blocks which are important in building any working URL. Set Theory Proof: The theory assumes that these elements are necessary to exist even when it comes to a zero-day attack since it treats those properties of a future attack as a subset of these invariant properties. Accuracy: According to this rationale, the framework in theory will have 100 percent accuracy in identifying future AI generated URLs. The URL Hit Bijection method views URL redirection as a mathematical one-to-one correspondence (bijection).Mapping: it presupposes that all shortened URLs.(IURLs) can be mapped to real, long-length response URLs of a server. Probability: This theory can be detected without any prior information about particular shortening algorithms or a third-party service. Weak to Strong Learners: Meta-learners such as LogitBoost-Extra Tree (LBET) and AdaBoost-Extra Tree (ABET) are operated by iterating on data distributions to convert weak hypotheses into one strong predictive model. Boundary of Decision: It is theoretically justified based on the fact that Support Vector Machines (SVM) can be used to create a decision boundary on a high-dimensional hyperplane to distinguish between classes. Complexity Analysis: In parallel execution of the models, theoretical time complexity can be computed as follows: The following is what the study refers to as the existing ceiling: Generator Dependency: The first constraint is that these systems are frequently trained on the example of particular AI generators, including

DeepPhish Architecture Restrictions: Some types of complex features by nature can be prohibited by the internal structures of some of the models.

Interpretability Gap: The issues of high false alarm rate and lack of transparency in the operation of many modern AI countermeasures continue to plague AI countermeasures[9].Maintenance Overhead: Conventional list and heuristic approaches are conceptually flawed since such systems have to be manually maintained and up to date with zero-day threats .Future Theoretical WorkTo stretch the fabric, the researchers propose a number of directions: Unsupervised Deep Learning: The researchers should incorporate unsupervised models to work with data without an existing labelling of an object. Hybridization and Feature Jettisoning: Integrating meta-learners with more complex feature selection to save computation at accuracy. Input Scalability: Scaling one of the input units with multi-threading to enable parallel classification of many URLs.

VII. CONCLUSION

This paper is a strong and scalable defense that can be used against the growing threat of phishing attacks synthesized by AI. The new system under focus focuses on lexical features analysis, in addition to the new introduction of URL HTML Encoding that helps in real-time detection of malicious patterns. Moreover, the URL Hit method is successfully implemented to reduce the risks of using shortened or obfuscated URLs that the use of these strategies does not escape the detection.

One of the main contributions of this study is the optimization of the classification structure by means of accepting a parallel multi-threaded system[15]. This is in contrast to the conventional sequential processing which severely slows down the speed of execution thus the system can only be used in real time situations which must run in high traffic where quick response is of paramount importance. The proposed model is effective as shown by the experimental assessment. It is reported to have an accuracy rate of 98 percent and F1-score of 98 per strong potential to identify genuine URLs with the true negative rate declared as 99.17 percent[10]. Correspondingly, in the case of standard phishing datasets, the model has an accuracy of 98 percent and an F1- score of 98 percent, which is independent of phishing types and stable and reliable performance.

The ensemble based voting mechanism is used which ensures that there is a balanced decision making and therefore high precision and also low false positives. These findings verify that parallel processing with ensemble learning gives the best trade off between speed and accuracy[11].

In the future, this framework should be evolved continuously. The improvements that can be made in future consist of unsupervised deep learning solutions that enable adapting to a changing pattern of phishing, model optimization and reduction tactics used to reduce the calculation load, and extending the multi-threading architecture to handle numerous URLs at the same time[12]. These innovations will further reinforce the capacity of the system to fight against existing and future computer-based crimes caused by AI.

REFERENCES

[1] Anti-Phishing Working Group (APWG). (2019). Phishing Activity Trends Report—Third Quarter 2019.
[2] D. Goel and A. K. Jain, "Mobile phishing attacks and defence mechanisms: State of art and open research challenges," Comput. Secur., vol.73, pp. 519–544, Mar. 2018.
[3] T. Thomas, P. A. Vijayaraghavan, and S. Emmanuel, "Machine Learning and Cybersecurity," in Machine Learning Approaches in Cyber Security Analytics. Singapore: Springer, 2020.
[4] D. Abraham and N. S. Raj, "Approximate string matching algorithm for phishing detection," in Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI), Sep. 2014, pp. 2285–2290.
[5] Patil, S. and Shekokar, N.M., 2023. A study of recent techniques to detect zero-day phishing attacks. In Intelligent approaches to cyber security (pp. 71-83). Chapman and Hall/CRC.
[6] Siponen, M.T., 2000. A conceptual foundation for organizational information security awareness. Information management and computer security, 8(1), pp.31-41.
[7] Von Eschenbach, W.J., 2021. Transparency and the black box problem:Why we do not trust AI. Philosophy and technology, 34(4), pp.1607-1622.
[8] Aloqaily, M., Otoum, S., Al Ridhawi, I. and Jararweh, Y., 2019. An intrusion detection system for connected vehicles in smart cities. Ad Hoc Networks, 90, p.101842.
[9] Brundage, M., Avin, S., Clark, J., Toner, H., Eckersley, P., Garfinkel, B.,Dafoe, A., Scharre, P., Zeitzoff, T., Filar, B. and Anderson, H., 2018.The malicious use of artificial intelligence: Forecasting, prevention, and mitigation. arXiv preprint arXiv:1802.07228.
[10] Mohanty, S. and Ambhakar, A., 2024. A study on machine learning and deep learning techniques for identifying malicious web content. SN Computer Science, 5(7), p.800.
[11] Al-Andoli, M.N., Tan, S.C., Sim, K.S., Seera, M. and Lim, C.P., 2023. A parallel ensemble learning model for fault detection and diagnosis of industrial machinery. IEEE Access, 11, pp.39866-39878.
[12] Xu, S.S.D., Hsu, T.Y., Shih, C.H., Ho, Y.T., Chen, Y.C., Lee, S.C., Yu, K. and Mumtaz, S., 2024. Design and implementation of a multi-threading architecture for downlink process in software-defined

physical layer over long-term evolution networks. IEEE Transactions on VehicularTechnology, 74(4), pp.5914-5927.

[13] Kittler, J., 1998. Combining classifiers: A theoretical framework. Pattern analysis and Applications, 1(1), pp.18-27.

[14] Kalasampath, K., Spoorthi, K.N., Sajeev, S., Kuppa, S.S., Ajay, K. and Maruthamuthu, A., 2025. A literature review on applications of explainable artificial intelligence (XAI). IEEE access, 13, pp.41111-41140.

[15] Deniz, E. and Sen, A., 2016. Using machine learning techniques to detect parallel patterns of multi-threaded applications. International Journal of Parallel Programming, 44(4), pp.867-900.

