Vol. 3 No. 3 2025

RECOGNIZED IN "Y" CATEGORY BY

Research Consortium Archive
P(ISSN) : 3007-0031
E(ISSN) : 3007-004X
https://rc-archive.com/index.php/Journal/about

Web Attack Detection Using Machine Learning-Phishing Attack Detection

Hafsa Badar
Govt. of Punjab in School Education Department as SST(CS),
MS Computer Sciences, Institute of Southern Punjab, Multan, Punjab, Pakistan.
E-mail: hafsa.04@gmail.com
ORCID: https://orcid.org/0009-0007-8321-570X

Publisher: EDUCATION GENIUS SOLUTIONS
Review Type: Double Blind Peer Review

ABSTRACT

Website network assaultive techniques, presently, along the expeditious evolution of Internet, the use of web is increasing and it has become an important part of our daily life. Web based susceptibility represents a substantial portion of the security. In our experiment, we used the Dataset named Phishing URLs taken from a Cyber Security Dept. of IT/ Telecom Company which is used as an input of the model. The proposed ML-PAD model will analyze the URLs and converting the URLs into a model and analyzing them using the keywords. In order to distinguish between true URLs and phishing URLs, segregation is carried out that determines the allowed or denied assignment tags. The performance of the machine learning algorithm in detection of phishing URLs was evaluated by accuracy. A Comparison is conducted with other data analytics techniques such as Naïve Bayes and SVM to validate the model performance. Our experiments show that the selection of the LDA model and the implementation of the LDA model with the existing methods outperformed ML-PAD with 100% accuracy and with 0% error rate far out performing existing methodologies.

Keywords: Web Attacks, Text Preprocessing, LDA, Topic Modeling, Natural Language Processing, ML-PAD

GEL Classification Code: T1.3.
ACM Computing Classification System (CCS):
· Security and privacy →Intrusion detection systems
· Security and privacy →Phishing
· Computing methodologies →Machine learning →Machine learning applications

INTRODUCTION

Cyber security is crucial in today's web-based world, articulated by (Alani & Tawfik, 2022). Organizations now rely on online data systems, Cyber-attack detection systems gather network data, yet improving accuracy and reducing false alarms remain difficult, said by researchers (Arman & Bairstow, 2022). This study assesses the effectiveness of machine learning algorithms in accurately identifying phishing URLs and distinguishing normal from anomalous data, stated by (Sundararaj & Kul, 2021). It could be very essential for the major projects and initiatives of the state government as well as in the enterprises to acclimatize the state-run and for the acceleration of the cyber security to protect from the anti-state activists, said by, (Ismail, Zohaib, & Tahir, 2025). The major form of web attack is phishing emails containing URLs offering free products or services, its researched by (Saleem, 2021). These confuse users about authenticity, to identify and reject such emails and links, there should be a process for early segregation of fake and real emails to minimize phishing attacks at early stages, (Boyle & Shepherd, 2021).

Research Objectives

- To investigate the challenges of cyber-attacks detection using machine learning.
- To investigate the trade-off between latency and accuracy of model.
- To study the effect of optimization parameter of machine learning model.

Research Questions

RQ1: What are challenges of detecting cyber-attacks using machine learning model?
RQ2: What is an efficient machine learning model for detection of cyber-attacks?
RQ3: Which optimizing technique will improve the accuracy of our ML-PAD?

Research Methodology

A cyberattack detection system collects network data. Improving performance, accuracy, and reducing false alarms remains challenging. Machine learning algorithms can identify normal and anomalous data with high accuracy. Though generalizability is lower for unknown attacks, deep learning shows remarkable precision. This study assessed phishing URL detection using machine learning accuracy.

LITERATURE REVIEW

(Feng, Zou, Ye, & Han, 2020) proposed EDL-WADS using three deep learning models to detect web attacks with high accuracy. (Mahmud, Prince, Ali, Hossain, & Andersson, 2024) used LSTM with hyper-parameter tuning to detect network attacks using the CICIDS 2017 dataset. (Apruzzese, Conti, & Yuan, 2022) used HMM with BOW to detect XSS and SQL attacks with low cost and high accuracy. (SRUTHI, 2023) proposed LSTM-RNN with Adam optimizer achieving 0.9944 accuracy for web intrusion detection. (Alazaidah et al., 2024) used HTTP log analysis with clustering to detect network anomalies with high accuracy. (Apruzzese et al., 2022) used KDD 1999 dataset with LSTM-RNN, achieving higher detection rate and accuracy than other classifiers. (Bountakas, 2023) used GAN and CNN with 2D data mapping to improve intrusion detection accuracy on four datasets.

(Gautam & Bansal, 2023) used CNN with NSL-KDD and UNSW-NB 15 datasets, achieving 91.14% and 94.9% accuracy rates. (2020) (Gautam & Bansal, 2023) introduced Tiki-Taka using multi-layered neural networks for intrusion detection, addressing feature quality issues and enabling end-to-end learning with improved feature representation and detection accuracy. (Ahmad, Ismail, Sutoyo, Kasim, & Mohamad, 2020) used sparse autoencoder with XG Boost on NSL-KDD, achieving high accuracy across five attack categories using SMOTE sampling. (Apruzzese et al., 2022) compared four

Decision Tree algorithms using hold-out and cross-validation for ordinal data learning and pairwise preferences. (Deshpande, Pedamkar, Chaudhary, & Borde, 2021) applied decision tree technique in university financial systems, effectively predicting financial situations and enhancing service levels through integrated management strategies.

(Alzahrani, 2021) evaluated decision tree architectures, showing reduced resource use compared to Random Forest, Compact Random Forest, and AdaBoost. (Atlam & Oluwatimilehin, 2022) used Fast Text and Sentence-LDA for effective short text analysis with external corpora. (Nagy et al., 2023) stated that LDA on Digital Economy Dataset, generating semantic word clouds for trend analysis in business economy. (Ashour, Marzouk, & AbdElhalim, 2024) proposed EETM combining topic and word embedding, performing efficiently on multiple datasets. (Do, Selamat, Krejcar, Herrera-Viedma, & Fujita, 2022) enhanced MODSECURITY WAF using machine learning, one-class classification, and n-gram analysis for detection.

(Ige, Kiekintveld, & Piplai, 2024) developed ADADM using MKL techniques to detect early-stage DDoS attacks accurately and quickly. (Alani & Tawfik, 2022) proposed deep learning frameworks reducing false positives by 99.3% in detecting application-layer DDoS attacks. (Tanimu & Shiaeles, 2022) achieved 99.6% accuracy by reducing CICIDS2017 features from 81 to 10 for intrusion detection. (Atlam & Oluwatimilehin, 2022) found social engineering a major attack source, with high accuracy on real and semi-synthetic datasets. (Veach & Abualkibash, 2022)found Decision Tree more robust than Bayesian Network for DDoS detection with effective attribute selection.

(Hamaidi, 2021) introduced iSSo-SGD, a hybrid algorithm outperforming Adam, RMS Prop, and others across five image datasets. (Igwilo & Odumuyiwa, 2022)developed a CNN based model using BOW and HTTP CSIC 2010 dataset, achieving high accuracy, TPR, and low FPR in detecting anomalous HTTP traffic. (Fan, 2021)introduced a machine learning-based intrusion detection model using web server logs. The model classifies logs as normal or attack types, with a multiclass labeled dataset generated via a private Apache WAMP server and DVWA. Text-based classification outperformed simple classification in detection accuracy.

(Do et al., 2022) proposed a detection mechanism aimed at minimizing false positives and false negatives. Using the CISC 2010 HTTP dataset, the study distinguished normal from anomalous traffic by employing a fine-tuned feature set. Experimental results using J48, Naive Bayes, and One R machine learning algorithms demonstrated effective web-based attack detection. Among these, the J48 decision tree algorithm achieved the highest performance with a 94.5% attack detection rate.

As per the above studies, many deep learning techniques are used to solve the Phishing URLs management issues but the

problem is there is no high-performance mechanism to manage, categorize and prioritize Phishing URLs data. So, there is a strong need of developing a model that can manage, categorize and prioritize Phishing URLs data with enhanced accuracy.

Problem Statement

Emails are the major source of communication from decades from now. Almost every internet user uses an email ID to communicate. But as it has many benefits, it also has some harmful aspects like phishing emails or emails containing phishing URLs that can manipulate users and can harm them financially as well as leak their private information. The above arguments and the background study refer the need of a model that is integrated with the natural language preprocessing techniques and data mining models to predict the results. This creates a strong foundation for creating a model that reduces that vulnerability in emails to avoid any breaches.

Methodology

The dataset contains the iterations or samples that reflect the various features. To acquire the dataset, a Cyber Security Department of IT Company was approached. The department provided a sample of URLs that are representative of phishing attempts encountered by the company's users and customers. These URLs serve as the input for the ML-PAD model, which is trained based on specific criteria derived from the dataset. By utilizing the titles of phishing URLs received by the company, the model is able to learn and analyze the patterns and characteristics associated with such malicious URLs. This dataset acts as a foundation for training the model and enhancing its ability to accurately detect and classify phishing attempts.

The data preprocessing phase encompasses a series of sequential steps, including the applications of various techniques which are as follows:

1. Number filter
2. Punctuation erasure
3. Stop word filter
4. Case converter
5. Stemmer

Model Formation

The well-known machine learning approach known as Latent Dirichlet Allocation (LDA) has been incorporated in the ML-PAD (Machine Learning-based Phishing URL Detection) framework. By utilizing the distinctive properties displayed by the term’s "http" or "https" inside the dataset, LDA is used to create two separate groups or clusters. This rigorous clustering is essential for correctly categorizing and identifying the preprocessed dataset, which includes both legitimate and phishing URLs. This step

provides a clear difference between the two categories based on their respective distinct characteristics and properties by successfully separating true URLs from phishing URLs. This kind of approach offers a promising way to increase online protection and resilience while strengthening security precautions and preventing phishing assaults.

ML-PAD ML-PAD

Machine Learning-based Phishing URL Detection (ML-PAD) identifies phishing URLs in emails to mitigate web-based attacks. Phishing URLs often use HTTP due to its lack of SSL/TLS certificates, avoiding HTTPS encryption and authentication. The thesis processes URLs using keywords http and https, enabling segregation between phishing and legitimate URLs. A dataset supports ML-PAD model training and testing. Natural language preprocessing is applied, including Latent Dirichlet Allocation (LDA), a statistical modeling method. LDA extracts meaningful topics from processed text. These techniques improve the quality of data and enhance accuracy and effectiveness of phishing URL detection.

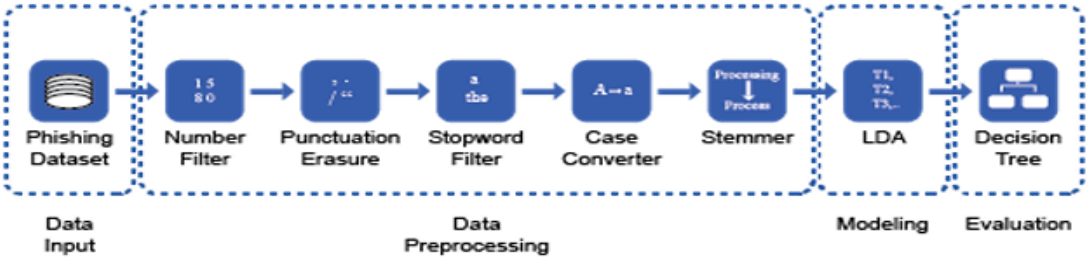

Figure. 1. "Phishing URL Detection Workflow Using Machine Learning Techniques"

Conceptual framework of ML-PA

A dataset containing phishing URLs are introduced as data input. After that a series of data preprocessing steps are performed to create clean data so it can be introduced to the main model. Among the various machine learning algorithms available, Latent Dirichlet Allocation (LDA) is chosen for its widespread popularity and effectiveness in natural language processing tasks. The LDA model plays a pivotal role in the research methodology as it generates two distinct groups or clusters based on the presence of the keyword's "http" or "https" within the URLs. These clusters enable the identification and categorization of preprocessed phishing URLs, effectively segregating them from legitimate URLs. Consequently, this approach enables a clear differentiation between phishing and authentic URLs. To assess the performance and accuracy of the generated clusters, a robust data analytics technique known as the decision tree is employed. The decision tree analysis evaluates the clustering results. The outcome of this evaluation exhibits impressive results, showcasing the strength and efficacy of the

proposed approach in identifying and categorizing phishing URLs.

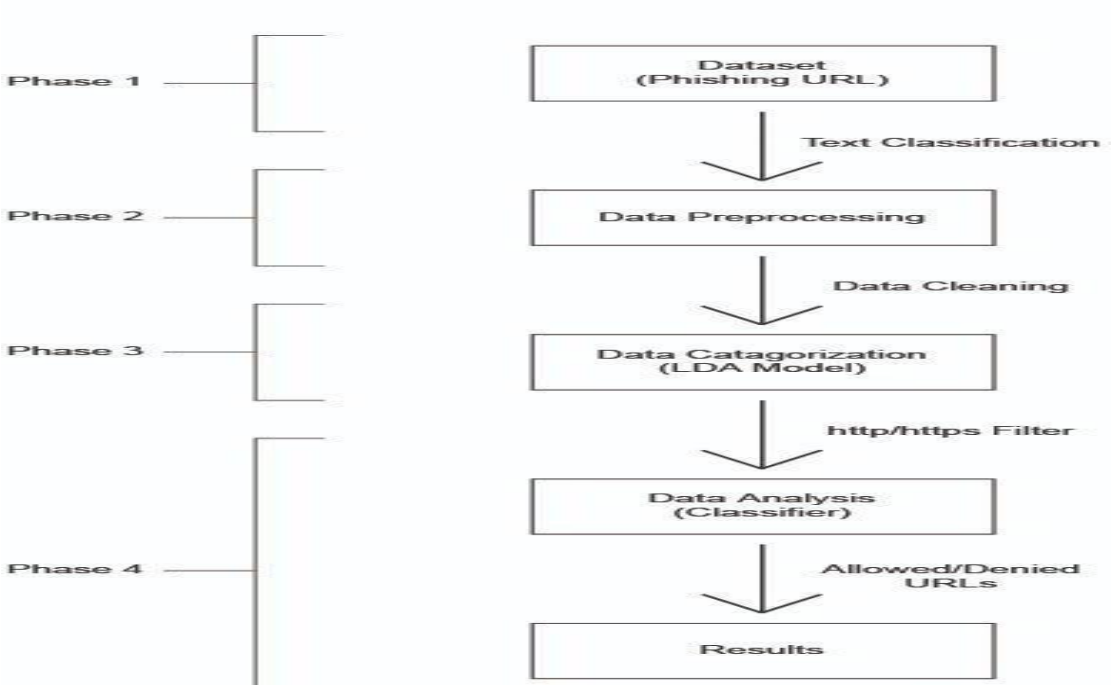

Figure. 2. Phase Diagram

Figure. 2 illustrates the Phase diagram of ML-PAD, a comprehensive framework comprising four distinct phases. In the first phase, the initial dataset, denoted as Phishing URLs, is introduced into the system. The second phase focuses on data preprocessing using techniques such as Number filter, Punctuation erasure, Stop word filter, Case converter, and Stemmer. This process removes irrelevant information, transforming the text into a standardized format. Phase 3 integrates the Latent Dirichlet Allocation (LDA) algorithm to create two clusters using keywords "http" or "https," categorizing the phishing URLs from true URLs. In the final phase, a decision tree evaluates the clusters. The decision tree learner trains on the data, while the predictor forecasts the topics associated with the training data. This enables ML-PAD to perform accurate topic classification. The Phase diagram of ML-PAD outlines data preprocessing, LDA-based clustering, and decision tree evaluation for efficient dataset categorization and analysis.

Simulation and Testing

Simulation platforms enable simulations, testing, and offer precise readings. They facilitate model outcome estimations before production. By simulating scenarios, researchers optimize models and reduce risks. ML-PAD model testing uses the KNIME simulation tool for its expansive library and stable performance, even with large input or training data. The nodes which are used in the simulation process and model testing are as follows:

Description about each node involved in the simulation of ML-PAD is below:

Nodes	Description
CSV Reader	To load the dataset in CSV extension
Strings to Document	To convert the sample entries in strings to a document form
Number filter	To remove numbers from the textual data samples
Punctuation Erasure	To remove the punctuations from the textual data samples
Stop words Filter	To remove the stop words from the textual data samples
Case Converter	To convert textual data samples into small alphabets
Stemmer	To convert the textual data sample words into their root form
Topic Extractor LDA Model	To create number of topics according to our requirement based on the collection of words present in textual data samples
Decision tree Learner	To load training data provided by the LDA
Decision Tree Predictor	To load testing data to predict model output
Scorer	**To visualize results**

Table. 1 Description of nodes using KNIME simulation

RESULTS

The experiment is performed using phishing URL dataset. The dataset is collected from the Cyber Security Dept. of an IT company and provide the required results. The simulations sequence is as follows

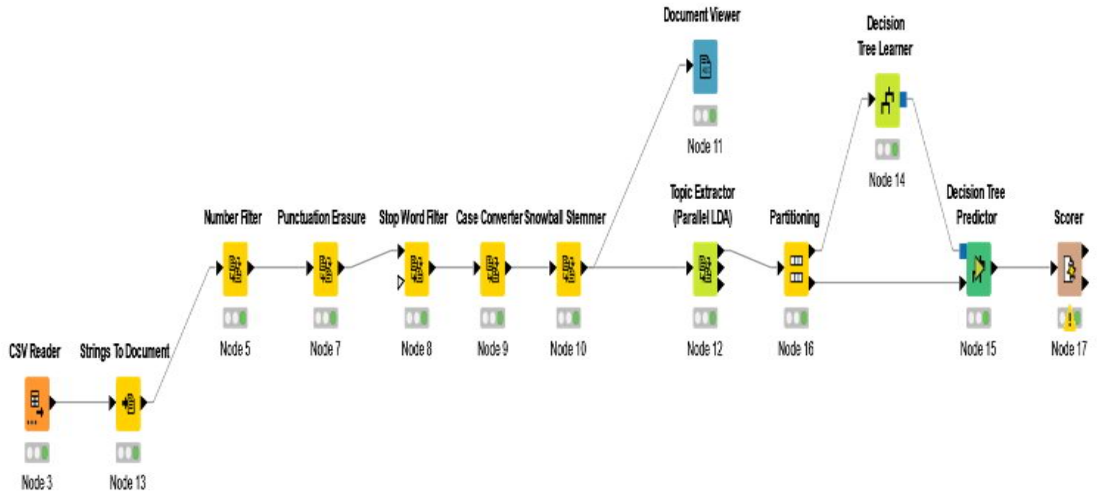

Figure. 3. Simulation of ML-PAD using KNIME

The ML-PAD is tested using Decision Tree algorithm. Decision

Tree Learner learns the train data and Decision Tree Predictor predicts the topics based on the training data. The results generated after evaluation are considered to impressive that can be measured by the following factors

The ratio of correct predictions (both true positives and true negatives) To the total number of samples under consideration.

$$\text{Accuracy} = \frac{True\ Positive + True\ Negative}{All\ Samples}$$

True Positive = Correct prediction of true values True Negative = Correct prediction of negative

Following are the results obtained after introducing phishing URLs dataset to the ML- PAD.

Table. 1. Confusion Matrix LDA Model with Decision Tree Accuracy.

Assigned Topic/Preduciton (Assigned topc)	**Topic_1**	**Topic_0**
Topic_1	666	0
Topic_0	0	508
Correct classified: 1,174	Wrong classified: 0	
Accuracy: 100 %	Error: %	
Cohen's Kappa (k) 1		

The accuracy on the phishing URLs dataset is 100 %.

Error Rate

The error rate on the phishing URLs dataset is 0 %.

Sensitivity

ROW ID	**SENSITIVITY**
TOPIC_1	**1.0**
TOPIC_0	**1.0**

Table. 1.2.

Specificity

ROW ID	**SPECIFICITY**
TOPIC_1	**1.0**
TOPIC_0	**1.0**

Table. 1.3

Results obtained from above experimentation shows that our model ML-PAD outperformed under the given datasets. The ML-PAD proves a useful technique in segregating the rejected/fake or approved/true URLs in the most effective way. As per our experiment, the selection of LDA model, implementation of LDA model in existing techniques ML-PAD outperformed with the accuracy of 100%which far better than the existing methodology.

Comparison with existing Model

This study examines the existing literature on alert prioritization using data mining approaches. The paper digs into the discussion of these strategies, providing significant insights into prediction methods for determining priority levels of input data, with a particular focus on Phishing URLs that are supplied into the model. This study obtains a deeper understanding of alert prioritizing and how data mining techniques can successfully enhance the process of assigning priority levels by drawing on the richness of knowledge offered in the existing literature. The insights obtained from the literature serve as a firm foundation for the development and evaluation of the model proposed in this study, ensuring that it is based on proven methodology and best practices.

Our proposed approach for detecting True URLs and Phishing URLs using ML-PAD was compared with existing phishing detection techniques in the literature. To validate the model, performance is compared to other data analytics approaches such as Naive Bayes and SVM. The results of the accuracy and error rate are mentioned below.

The simulation results of the phishing URLs result with Naïve Bayes evaluation method.

CONFUSION MATRIC – 0:55 - SCORER		
ASSIGNED TOPIC / PREDICTION (ASSIGNED TOPIC)	Topic_1	Topic_0
TOPIC_1	685	485
TOPIC_0		
CORRECT CLASSIFIED: 1.1770	Wrong classified: 4	
ACCURACY: 99.659 %	Error: 0.341 %	
COHEN'S KAPPA (K) 0.993		

Table. 2. Confusion Matrix LDA Model (4Topics) with Naïve Bayes

The comparison table of the Phishing_URLs dataset result with Decision Tree evaluation method with the Phishing_URLs dataset result with Naïve Bayes evaluation method.

CONSTRAINTS	**PHISHING_URLS DATA SET (DECISION TREE)**	**PHISHING_URLS DATASET (NB)**
ACCURACY	99.78%	**98.65%**
ERROR RATE	**0.22%**	**1.35%**

Table. 3. Comparison table (Decision Tree) with Naïve Bayes

The simulation results of the phishing_URLs dataset result with SVM evaluation method.

CONFUSION MATRIX – 0:14 – SCORER	
ASSIGNED TOPIC / PREDICTION (ASSIGNED TOPIC)	Topic_

TOPIC_1	683	2
TOPIC_0	487	2
CORRECT CLASSIFIED: 685 **ACCURACY: 58.348 %** **COHEN'S KAPPA (K) 0.001**	Wrong classified: 489 Error: 41.652 %	

Table. 4. Confusion Matric LDA Model (4 Topics) with SVM.

The comparison table of the phishing URLs dataset result with Decision Tree evaluation method with the phishing URLs dataset result with Support Vector Machine (SVM) evaluation method.

CONSTRAINTS	**PHISHING_URLS DATASET (DECISION TREE)**	**PHISHING_URLS DATASET (SVM)**
ACCURACY	100%	**58.34%**
ERROR RATE	**0%**	**44.65%**

Table: 5 Comparison Table (Decision Tree) with SVM

CONCLUSION

By summing up the whole work, the research questions presented at the start of the study have been effectively addressed. Focused experimentation and simulation were conducted to obtain results. The research aimed to deliver ML-PAD and test its validity based on its performance.

RQ1: What are the challenges of detecting cyber-attacks using machine learning models?

Cyber-attacks attempt to steal, alter, or destroy critical data. Web attacks, especially phishing emails with misleading URLs, are becoming increasingly complex. These attacks often confuse users about the authenticity of emails. To identify and reject phishing emails and URLs, an early segregation process is needed. The main challenge is the continuous evolution of fraudulent techniques and URL masking. Data cleaning in NLP is crucial to remove irrelevant content from textual data, making the input more meaningful and improving output accuracy.

RQ2: What is an efficient machine learning model for detection of cyber-attacks?

Topic Modeling is used due to its ability to handle both structured and unstructured data. The model identifies phishing emails by assigning topics using LDA after preprocessing with NLP techniques. The preprocessing stage handles stop words, punctuation, numeric, and missing values, cleaning the data before topic assignment.

RQ3: Which optimization technique improves the accuracy of ML-PAD?

The Decision Tree model is integrated with LDA to improve prediction accuracy. It serves as an effective classifier, reducing the error rate. Cleaned data enhances LDA's output, which is then used as training data for the Decision Tree. The simulation used

partitioning, Decision Tree Learner, and Predictor nodes. The result was 100% accuracy on the Phishing URLs dataset with 0% misclassification. The ML-PAD proves to be an effective approach for detecting and segregating phishing URLs from emails, reducing phishing attacks at early stages.

LIMITATIONS

1. Every research study is conducted under certain constraints and limitations, such as financial resources, data collection processes, and availability of suitable equipment.
2. These limitations provide direction for future researchers to explore patterns and answers not addressed in the current study.
3. The current research was limited to performance testing using only the Decision Tree analysis technique.
4. Other analysis techniques such as Random Forest or Artificial Neural Networks (ANN) were not applied or tested in this study.
5. The dataset used was collected solely from the cybersecurity department of an IT company, excluding other sectors like healthcare.

FUTURE WORK

It's not only important due to its directional characteristic but also the improvement in the current study made to create a sounder and more efficient model that have less error rate and have promising facts to perform well in real life scenario. The future work can be performed by the researchers on the implementation and evaluation of ML-PAD using other analysis techniques like ANN and Random Forest. Also, datasets from other industries like healthcare and other security environments can be introduce to the ML-PAD to monitor its diversity.

REFERENCES

Ahmad, S. W., Ismail, M. A., Sutoyo, E., Kasim, S., & Mohamad, M. S. (2020). Comparative performance of machine learning methods for classification on phishing attack detection. *International Journal of Advanced Trends in Computer Science and Engineering*.

Alani, M. M., & Tawfik, H. (2022). PhishNot: a cloud-based machine-learning approach to phishing URL detection. *Computer Networks, 218*, 109407.

Alazaidah, R., Al-Shaikh, A., Al-Mousa, M., Khafajah, H., Samara, G., Alzyoud, M., . . . Almatarneh, S. (2024). Website phishing detection using machine learning techniques. *Journal of Statistics Applications & Probability, 13*(1), 119-129.

Alzahrani, S. M. (2021). Phishing attack detection using deep learning. *International Journal of Computer Science & Network Security, 21*(12), 213-218.

Apruzzese, G., Conti, M., & Yuan, Y. (2022). *Spacephish: The evasion-space of adversarial attacks against phishing website detectors using machine learning.* Paper presented at the Proceedings of the 38th annual computer security applications conference.

Arman, J., & Bairstow, J. (2022). The Role of Natural Language Processing in Enhancing Network Security and Preventing Cyber Attacks.

Ashour, M. M., Marzouk, E. S. A., & AbdElhalim, E. (2024). Anti-Phishing approach for IoT system in Fog networks based on machine learning algorithms. *Mansoura Engineering Journal, 49*(3), 13.

Atlam, H. F., & Oluwatimilehin, O. (2022). Business email compromise phishing detection based on machine learning: a systematic literature review. *Electronics, 12*(1), 42.

Bountakas, P. (2023). Implementing AI-driven methodologies for cyberattack detection.

Boyle, P., & Shepherd, L. A. (2021). *Mailtrout: a machine learning browser extension for detecting phishing emails.* Paper presented at the 34th British HCI Conference.

Deshpande, A., Pedamkar, O., Chaudhary, N., & Borde, S. (2021). Detection of phishing websites using Machine Learning. *International Journal of Engineering Research & Technology (IJERT), 10*(05), 430-434.

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.

Fan, Z. (2021). *Detecting and classifying phishing websites by machine learning.* Paper presented at the 2021 3rd International Conference on Applied Machine Learning (ICAML).

Feng, J., Zou, L., Ye, O., & Han, J. (2020). Web2vec: Phishing webpage detection method based on multidimensional features driven by deep learning. *IEEE Access, 8*, 221214-221224.

Gautam, A. K., & Bansal, A. (2023). Email-based cyberstalking detection on textual data using multi-model soft voting technique of machine learning approach. *Journal of Computer Information Systems, 63*(6), 1362-1381.

Hamaidi, K. (2021). *A Predictive Model for Phishing Detection Based on Convolutional Neural Networks.* University of Badji Mokhtar.

Ige, T., Kiekintveld, C., & Piplai, A. (2024). Deep learning-based speech and vision synthesis to improve phishing attack detection through a multi-layer adaptive framework. *arXiv preprint arXiv:2402.17249*.

Igwilo, C. M., & Odumuyiwa, V. T. (2022). Comparative analysis of ensemble learning and non-ensemble machine learning algorithms for phishing URL detection. *FUOYE Journal of Engineering and Technology, 7*(3), 305-312.

Ismail, M., Zohaib, M., & Tahir, M. Y. (2025). Management Perspective Of a Hybrid Leadership And Inter Institutional Coordination, For 'Special Investment Facilitation Council'(SIFC). *Research Consortium Archive, 3*(3), 72-85.

Mahmud, T., Prince, M. A. H., Ali, M. H., Hossain, M. S., & Andersson, K. (2024). Enhancing cybersecurity: Hybrid deep learning approaches to smishing attack detection. *Systems, 12*(11), 490.

Nagy, N., Aljabri, M., Shaahid, A., Ahmed, A. A., Alnasser, F., Almakramy, L., . . . Alfaddagh, S. (2023). Phishing urls detection using sequential and parallel ml techniques: comparative analysis. *Sensors, 23*(7), 3467.

Saleem, B. M. (2021). *The p-fryer: Using machine learning and classification to effectively detect phishing emails.* Marymount University.

SRUTHI, K. (2023). Real-Time Phishing Threat Detection using Lexical URL Features and Machine Learning Techniques.

Sundararaj, A., & Kul, G. (2021). Impact Analysis of Training Data Characteristics for Phishing Email Classification. *J. Wirel. Mob. Networks Ubiquitous Comput. Dependable Appl., 12*(2), 85-98.

Tanimu, J., & Shiaeles, S. (2022). *Phishing detection using machine learning algorithm.* Paper presented at the 2022 IEEE international conference on cyber security and resilience (CSR).

Veach, A. M., & Abualkibash, M. (2022). Phishing website detection using several machine learning algorithms: a review paper. *International Journal of Informatics, Information System and Computer Engineering (INJIISCOM), 3*(2), 219-230.