

Enhancing Web Security: An Efficient URL Phishing Classifier based on Deep Learning

Abhijna S¹ and Vinay Prasad M S²

¹⁻²JSS Science and Technology University, Mysore, India
Email: sabhijna8@gmail.com, vpms1408@sjce.ac.in

Abstract— To procure sensitive data such as account Identification details, usernames, passwords, etc Attackers often develop new strategies, such as phishing, to fool users by utilizing fake websites. This paper presents and illustrates various methods for addressing phishing detection. To enhance user security and privacy, a web application is developed to determine if the URLs are malicious or benign. Malicious website detection can significantly decrease the probability of phishing attempts. In this paper, two distinct feature extraction techniques, Natural Language Processing (NLP) and TF-IDF vectorization, are employed to capture the intricate patterns within URLs. Evaluation and training are carried out by feeding the NLP features into Convolutional Neural Networks (CNN), Multilayer Perceptron's (MLP), Long Short-Term Memory Networks (LSTM), and Deep Neural Networks (DNN). Initial experimentation with NLP features yields promising results, achieving up to 98.78% accuracy with CNN and 95.32% with LSTM. An accuracy of 94.03% and 94.64% is obtained by exploring TF-IDF vectorization using CNN and MLP models respectively. Precision, F1-score, recall, ROC and various metrics for evaluation provide a holistic assessment of the model's performance. Oversampling and class weight adjustment techniques address the implications of class imbalance. To enhance model generalization, Hyperparameter tuning, cross validation, and ensemble methods are emphasized. Overfitting concerns are mitigated using regularization techniques such as dropout. This approach involves Principal component analysis (PCA), a feature reduction technique to lower the number of features present.

Index Terms— URL phishing, Phishing attack, Deep learning, Cybersecurity

I. INTRODUCTION

Nowadays, phishing has gained a primary concern for security researchers as creating fraudulent websites look similar to genuine ones. It can be difficult to determine whether a website is fraudulent or not, especially for those who are not experts in the field. It's important to be cautious and aware of the signs of fraudulent websites. Phishing attacks hit an all-time high in 2022, with over 4.7 million attacks and 1.35 million in Q4 alone, according to APWG's phishing activity trend report. This represents a 150% growth rate per year since 2019. The success of phishing attempts is increasing as a result of user ignorance. It can be challenging to combat phishing assaults since they often exploit user vulnerabilities. However, it is crucial to continuously improve phishing detection methods. An efficient intrusion detection solution is required to recognize and mitigate the impacts of phishing. Machine learning and deep learning techniques are being implemented to enhance cybersecurity. These techniques require past data to decide or predict future data. Artificial intelligence (AI) is a rapidly evolving intelligence method that helps oversee cybersecurity and provide protection for computer

activities. It has become an essential component due to its broad spectrum of abilities, ranging from pattern recognition to security. The advancements in deep learning have made it possible for machines to learn on their own and extract features automatically without human intervention making the entire process more efficient and cost-effective. Deep learning has revolutionized the way to approach complex problems by providing faster and more accurate solutions, especially while dealing with large datasets and tasks in phishing detection. It is not an easy task to choose the appropriate algorithm or method for any application. It is crucial to highlight that while deep learning offers numerous benefits, the selection of the appropriate algorithm plays a pivotal role in maintaining accuracy and efficiency. This is particularly important in phishing detection, a task that demands continuous adaptation and evolution due to the ever-changing tactics employed by attackers to exploit system vulnerabilities and users. By making algorithmic choices, the potential to protect against phishing threats could be increased. In combating the growing threat of Phishing, deep learning mechanisms have proven to be an effective solution. Through learning by example, deep learning trains machines to imitate human brains, enabling them to perform classification tasks on large data sets, including text, sound, and image. In contemporary times, different strategies have been put forth to combat fraudulent websites through the extraction of URL features for classification purposes. Deep learning algorithms have demonstrated immense robustness, enabling them to identify patterns in vast datasets and make precise predictions. In this paper, a real-time phishing detection system has been proposed, utilizing deep learning algorithms like Multilayer Perceptron, Convolutional Neural Network, Long Term Short Memory, and Deep Neural Networks by collecting a significant amount of authentic and deceitful web page URLs. From this data, three distinct feature sets: NLP feature, TF-IDF, and hybrid features are defined and tested to assess the efficacy of the proposed system.

II. RELATED WORKS

With regard to the detection of phishing URLs, we found a few publications concerning URL-based phishing. In this paper [1], an approach to classify URLs automatically based on their host and lexical features was explored. Incorporating labels as predictive features enhances the efficiency and effectiveness of the classification systems in their tasks. This model boasts a high degree of accuracy and is capable of detecting numerous phishing hosts while keeping false positives to a minimum.

Mehmet Korkmaz et al. (2020) [2] discussed a phishing detection system that utilizes machine learning algorithms to analyse URLs and compare results from three datasets. The system leverages anomaly detection, which is particularly efficacious for identifying zero-day attacks.

Buket Geyik et al. (2021)[3] developed an anti-phishing method to detect various phishing attacks and prevent intrusion using the WEKA tool.

An anti-phishing system [4] utilizing seven distinct classification algorithms and features on basis of natural language processing (NLP) was put forth in real time, employing feature-rich classifiers. Based on the results obtained from the implemented algorithms, the Random Forest algorithm achieves the highest performance.

In order to identify phishing websites and their targets,

Huaping Yuan et al. (2018) [5] utilized lexical features extracted from the webpage links, URL & web page content, including the textual content & title. Moreover, the researchers explored multiple machine learning models for phishing detection and identified that the Deep Forest model delivered the most competitive performance.

Kudirat Oyewumi jimoh et al. (2022) [6] presented a system that utilizes rules to classify phishing websites. This system accurately identifies and categorizes websites as either phishing or non-phishing by analysing specific design and implementation-related features.

A recent study incorporates natural language processing techniques [7] to identify visual similarities. As phishing attacks can be quite intricate, the system addresses the semantics-based attack that exploits users' vulnerabilities. Deep learning frameworks and neural networks can be utilized to detect inappropriate language that is indicative of phishing attacks. This led to the development of a system called Phishector [8], which takes an email as input and determines the accuracy of predicting whether the email is genuine or fraudulent. The system's effectiveness was tested on two benchmark datasets, SpamAssassin and Ham Spam, utilizing various machine learning techniques. During the testing process, neural networks outperformed other techniques.

Another system to detect phishing attacks is CBR-PDS [9]. Its adaptive and dynamic nature confers a significant advantage over other classifiers. It has proven to be highly accurate.

Mohammad Siful Islam Mamun et al. (2016) [10] proposed a lightweight approach to detect and categorize malicious URLs based on their attack type. The approach uses lexical analysis for proactive detection. The study also examines the impact of obfuscation techniques on malicious URLs.

CANTINA is a novel method for identifying fraudulent websites, utilizing the TF-IDF algorithm. Additionally

CANTINA+ uses an extensive feature-based approach that incorporates eight innovative features. These results were compared to a classification model proposed by Routhu Srinivasa Rao et.al (2018) [11] which is based on deriving heuristic features from the URL.

PhishWHO[12] is a phishing detection technique that leverages the disparity between the actual and target identities of a webpage.

URLNet[13] is an innovative deep learning framework that directly learns a nonlinear URL embedding technique. This advanced feature allows the model to capture different types of semantics that were previously unattainable with existing models.

A new framework called the Hybrid Ensemble Feature Selection (HEFS) [14] is proposed by Kang Leng Chiew et al. (2019). It uses Cumulative Distribution Function gradient (CDF-g) to generate subsets for predominant features in the initial phase. Subsets for subordinate features are produced by utilizing previously obtained subsets in the data perturbation ensemble.

Ankit Kumar Jain et al. (2019) [15] proposed a new method that analyses hyperlinks in HTML source code to identify fraudulent attacks. This method employs 12 categories of features that are specific to hyperlink in order to train Machine learning algorithms, resulting in high accuracy. It is independent of language and is capable of detecting websites that use any language.

PHISH-SAFE[16] utilizes URL features to detect fraudulent websites. The system is designed to evaluate websites as either fraudulent or genuine by employing fourteen features. The results demonstrate that SVM achieved a high degree of accuracy.

III. PROPOSED METHODOLOGY

The proposed model consists of various stages such as preprocessing, feature extraction, feature reduction and Model classification using different deep learning techniques. The architecture of the proposed model is given in Fig 1.

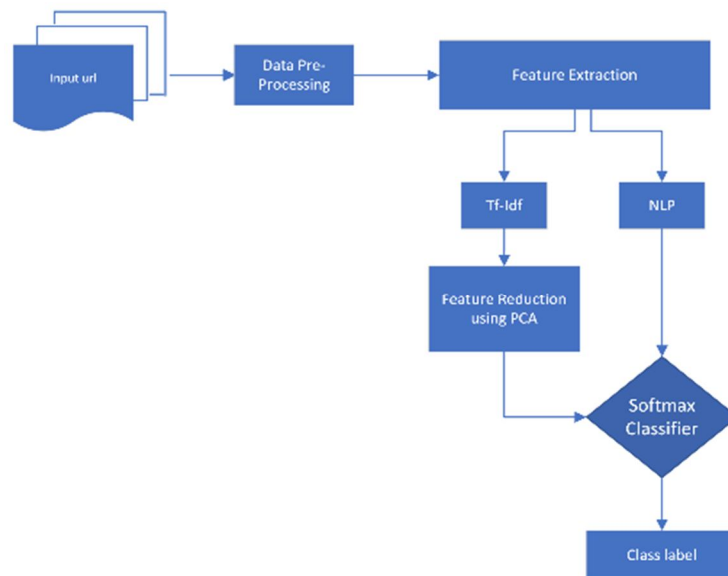


Fig 1. General architecture of the proposed model

A. Pre-processing

The data preprocessing phase includes the detection of null values, conversion of URLs to lowercase, the removal of non-alphanumeric characters and digits from URLs, and so on. This stage also involves operations like removing “https://”, “http://”, “www.” and so on. Consequently, every word is collected from a URL before being processed as part of the data pre-processing phase and then evaluated at each stage of execution. This is followed by a series of experiments conducted on the MLP, LSTM, CNN, and DNN models.

B. Feature Extraction

This methodology offers a comprehensive analysis of the URL’s structure and content by using two different feature extraction techniques such as NLP & TF-IDF.

1. Natural language processing features

In this decomposition process, the URL was first parsed while accounting for special characters like “?”, “/”, “.”, “,”, “=” and “&”. Attackers may employ a variety of strategies to trick users. Based on the attack type, brand names like Google, Apple etc., or some crucial phrases like login, secure, account, server, etc. are used. The proposed system uses ten NLP features, and they are detailed as follows.

- Raw-word-count: The total number of terms in the URL. This could be an indicator of URL complexity or length.
- Spitted-word-count: It refers to amount of words after splitting the URL.
- Average word length: It is the mean length of words in the URL.
- Longest-word-length: It refers to the length of the longest word in the URL. This gives insights into the presence of long strings, which could indicate obfuscation.
- Shortest-word-length: The length of the shortest term in a URL. This gives insights into the presence of short strings, which could indicate obfuscation.
- Std word-length: The standard deviation of length of a word. This captures variations in word lengths and could be indicative of unusual structures.
- Compound word-count: The number of terms grouped under a specific criterion (e.g., length greater than 7).
- Keyword count: The number of keywords extracted from the URL.
- Brand name-count: The amount of brand names extracted from the URL.
- Negligible word-count: The number of terms considered negligible based on some analysis (e.g., determined to be not informative).

These features provide additional insights into the structure and content of the URL thereby distinguishing between legitimate and phishing URLs.

2. TF-IDF

TF-IDF involves converting each URL into a numerical vector representing the frequency of each word in the URL relative to its frequency in the entire dataset of URLs. The output vector can be used as input to a deep learning algorithm for classification. TF-IDF process involves:

- Calculate Term Frequency (TF): It is imperative to take into consideration the frequency of specific terms that appear within contents. This metric, referred to as Term Frequency is determined by dividing the count of the word or token by the total amount of terms in that document.

$$TF(t) = \text{Number of times term } t \text{ occurs in a document} / \text{Total number of terms in the document} \quad (1)$$

- Calculate Inverse Document Frequency (IDF): Inverse Document Frequency calculates the rarity of a word or token in whole dataset or corpus. It is calculated as logarithmic ratio of overall number of documents to the quantity of documents containing the term. The goal of IDF is to give more weight to terms that are rare across documents:

$$IDF(t) = \log_e(T/N) \quad (2)$$

where T = Total number of documents & N = Number of documents containing term t

- Compute TF-IDF Scores: The TF-IDF score for a term in a specific document is the product of its TF value and its IDF value. TF-IDF scores capture the importance of terms within a document while also considering their rarity across the entire dataset:

$$TF - IDF(t, d) = TF(t, d) * IDF(t) \quad (3)$$

C. Feature Reduction

Dimensionality reduction is an essential technique for improving model performance and preventing overfitting. These techniques enable the model to focus on the most important features, resulting in improved accuracy and efficiency. Principal component analysis is a method used for dimensionality reduction. In this work, the top features for TF-IDF are selected using PCA. It is observed that accuracy of proposed model increases with the PCA value upto 100, after which the accuracy remains stable, resulting in no further significant improvement in accuracy. These features serve as valuable input for machine learning models, allowing them to learn and make accurate predictions based on the unique characteristics of the URLs.

IV. EXPERIMENTATION AND RESULTS

A. Dataset

In this paper, we aim to obtain a dataset that is globally acceptable. Ebbu2017 Phishing Dataset, 2017 is utilized. In total, this dataset contains 36,400 genuine URLs and 37,175 fraudulent URLs.

B. Training and Testing

The data set is segmented into 20% for testing purpose and 80% for training. Various parameters are taken into account, including but not limited to the number of layers, neurons within each layer, batch size, learning rate, dropout rate, epochs, activation function, and optimizer.

C. Results

Four distinct Deep Learning Algorithms are used to perform each test set and statistical measures are used to calculate algorithm efficiency by applying the values from the confusion matrix.

1. NLP feature

In this paper, a total of 10 NLP features are used to classify URL. Fig 2 illustrates the relationship between predicted and actual class labels of NLP feature confusion matrix. This serves as a foundation for calculating various performance metrics. According to Table I, the Convolutional Neural Network (CNN) algorithm with NLP features demonstrated strong capabilities in detecting phishing URLs by exploiting local patterns, achieving an accuracy of approximately 98.78%. Meanwhile, MLPs leveraged their depth to capture complex relationships between features, achieving a competitive accuracy of around 90.56%. LSTM showcased its effectiveness in handling sequential data, contributing to an accuracy of 95.32%, While DNN has the lowest classification performance with an accuracy of 74.50%. Deep learning algorithm models shows better accuracy than machine learning models like Logistic regression & XGBoost with accuracy of 61.6% and 90.04%. Fig 3 and 4 represent accuracy, precision, recall and F1 score for various classification algorithms using NLP.

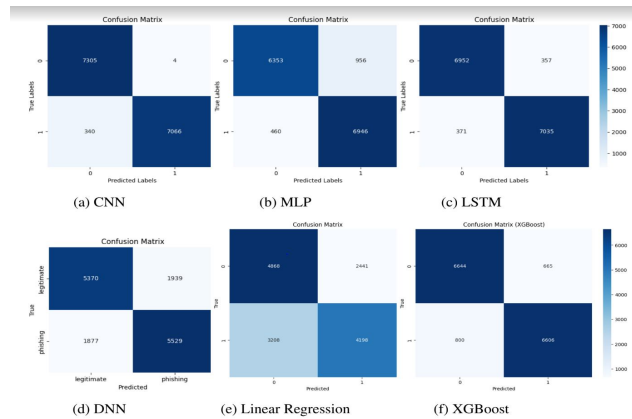


Fig 2. NLP feature confusion matrix

TABLE I TEST RESULTS OF CLASSIFICATION ALGORITHMS FOR NLP FEATURE

Algorithm	Features	Precision	Recall	F1 Score	Accuracy
CNN	NLP Features	0.989	0.985	0.986	98.78%
MLP	NLP Features	0.889	0.928	0.908	90.56%
LSTM	NLP Features	0.952	0.954	0.953	95.32%
DNN	NLP Features	0.770	0.687	0.726	74.50%
Logistic Regression	NLP Features	0.60	0.567	0.579	61.6%
XGBoost	NLP Features	0.908	0.892	0.899	90.04%

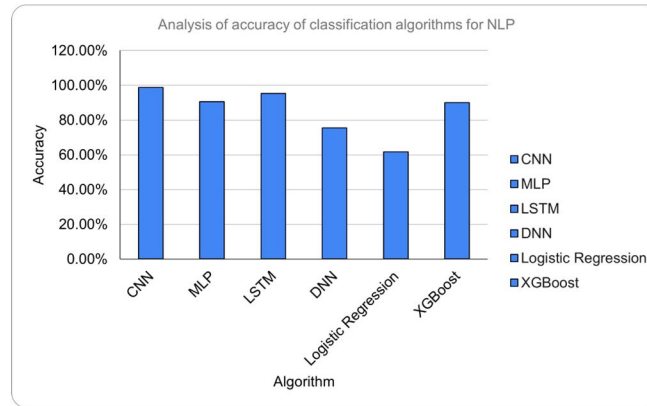


Fig 3. Analysis of accuracy of classification algorithms for NLP.

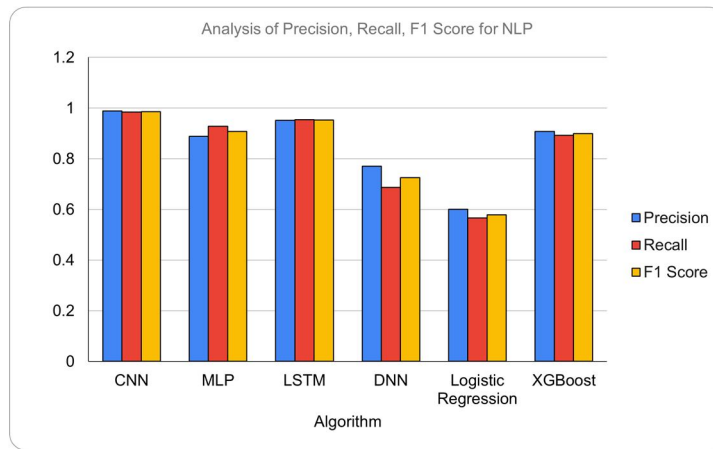


Fig 4. Analysis of various evaluation metrics for NLP.

2. TF-IDF Feature

Performance metrics for several deep learning classification algorithms with TF-IDF features are presented in Table II. The TF-IDF feature extraction technique gives a performance of 94.03% and 94.64% upon using Convolutional Neural Network and Multi-Layer Perceptron respectively. However, LSTM and DNN models lack a sufficient complex or deep architecture to capture the intricate patterns with a limited accuracy of only 50%. Several potential reasons for this discrepancy exist. One factor to consider is the limited size of the dataset, which may not provide enough examples for these models to generalize effectively. Fig 5 and 6 represent accuracy, precision, recall and F1 score for various classification algorithms using TF-IDF.

TABLE II TEST RESULTS OF CLASSIFICATION ALGORITHMS FOR TF-IDF

Algorithm	Feature	Precision	Recall	F1 Score	FeaturesPrecisi Accuracy
CNN	TF-IDF	0.939	0.943	0.941	94.03%
MLP	TF-IDF	0.933	0.963	0.948	94.64%
LSTM	TF-IDF	0.50	1.00	0.66	50.33%
DNN	TF-IDF	0.50	1.00	0.67	50.33%
Logistic Regression	TF-IDF	0.912	0.918	0.915	91.1%
XGBoost	TF-IDF	0.942	0.969	0.955	95.5%

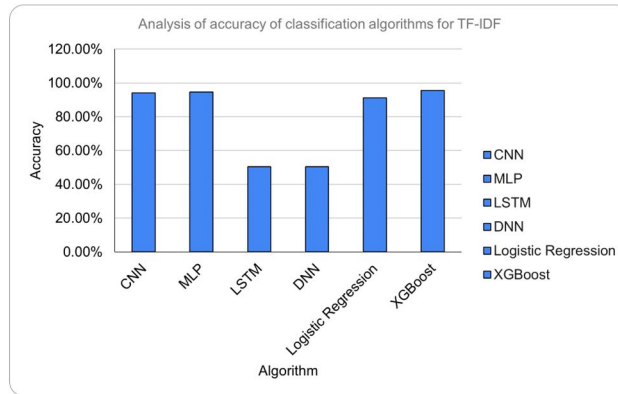


Fig 5. Analysis of accuracy of classification algorithms for TF-IDF.

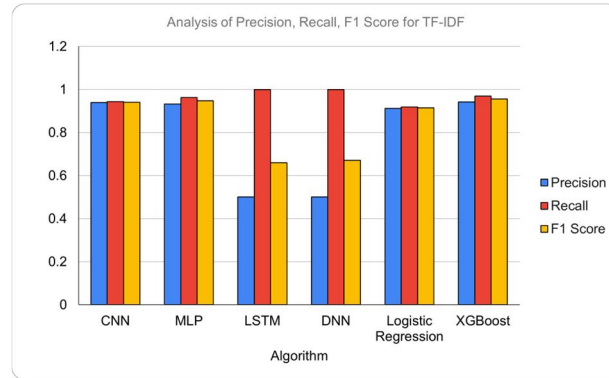


Fig 6. Analysis of various evaluation metric for TF-IDF.

Overfitting is also a critical concern, as it can lead to a model performing well on training data but poorly on new data. This may be particularly relevant to those models, especially if their architectures are excessively complex given the limited amount of data available. Furthermore, the vanishing gradient problem, which can present significant challenges for LSTM networks, may be impeding the learning process. Differences in data representation and preprocessing techniques also contributes to the lower performance of model. The classification results indicate that the NLP feature Deep learning models have a significantly better performance overall. This table gives valuable insights into a model's strengths and performance of constructed models using NLP features and TF-IDF.

3. NLP & TF-IDF features

Natural Language Processing (NLP) and TF-IDF (Term Frequency-Inverse Document Frequency) features are used for text analysis and classification of URLs. The raw URL data is transformed into text data using NLP techniques (stopwords removal and tokenization) and then the processed text data is converted into TF-IDF features using the "TF-IDF vectorizer."

TABLE III NLP & TF-IDF FEATURES TEST RESULTS OF CLASSIFICATION ALGORITHMS

Algorithm	Precision	Recall	F1-Score	Accuracy
CNN	0.954	0.970	0.962	96.18%
MLP	0.956	0.969	0.962	96.22%
DNN	0.973	0.910	0.939	94.23%

These features are used for dimensionality reduction (PCA) and training machine learning models like CNN, MLP & LSTM to classify URLs as legitimate or phishing. From Table III, Algorithms such as MLP, CNN and DNN

show promising results with a good accuracy of 96.22% ,96.18%, and 94.23% respectively. Fig 7 represents precision, recall, F1 score for NLP & TF-IDF.

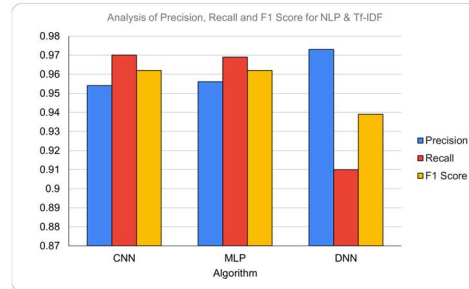


Fig 7. Analysis of various evaluation of metrics for NLP & TF-IDF.

V. COMPARITIVE ANALYSIS

The Existing model [4] uses various machine learning algorithms, among which the random forest algorithm with NLP feature achieved an accuracy of 97.98%. Comparatively, our proposed model obtains a higher accuracy of 98.78% using CNN with NLP feature, also resulting in a lower response time. From the Table IV, it is observed that there is a huge reduction in the number of features in our model which is a noteworthy aspect of our novel approach.

TABLE IV COMPARITIVE ANALYSIS OF EXISTING MODEL

Model	Dataset	<i>NLP features</i>
Ozgur.et.al (2018)	(ebbu 2017 Phishing dataset, 2017)	40
Proposed Model	(ebbu 2017 Phishing dataset, 2017)	10

VI. CONCLUSION

The potential of combining deep learning and feature ex traction techniques for URL classification is demonstrated in this study. Increasing sophistication of phishing attacks demands robust and accurate detection mechanisms, and using deep learning models has proven to be a promising approach. Specifically, we have designed real-time phishing detection system that employs two crucial feature extraction techniques: NLP and TF-IDF. NLP enables the capture of linguistic patterns and context, while TF-IDF facilitates the numeric representation of text features. These features were used as inputs for various Deep learning and Machine learning architectures. According to the experimental results, Adopting NLP features with CNN classifier yields better performance compared to another classifier. Experiments were conducted using these models, and their performance was evaluated meticulously. This paper emphasizes on achieving high accuracy, precision, recall, and F1-score values, all essential metrics for phishing attack detection. As cybersecurity evolves and phishing attacks become more sophisticated, there are several avenues for future enhancement by incorporating more advanced deep learning architectures, such as Transformer-based models like BERT or GPT, that offer improved feature extraction. Additionally, integrating this work with ensemble learning techniques, combining the strengths of multiple models could enhance overall accuracy and robustness. Finally, real-time deployment and integration of the developed models within web browser extensions or email clients could provide users immediate protection against phishing threats, contributing to a safer online experience.

REFERENCES

- [1] Mehmet Korkmaz, Ozgur Koray Sahingoz, Banu Diri ,(2020) "Detection of Phishing Websites by using Machine Learning Based URL Analysis", 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT).
- [2] Buket Geyik, Kubra Erensoy, Emre Kocyigit ,(2021) "Detection of Phishing Websites from URLs by using Classification Techniques on WEKA", 2021 6th International Conference on Inventive Computation Technologies (ICICT).
- [3] Ozgur Koray Sahingoz, Ebubekir Buber, Onder Demir, Banu Diri, (2018),"Machine Learning Based Phishing Detec-tion from URLs".

- [4] Huaping Yuan, Xu Chen, Yukun Li, Zhenguo Yang, Wenyin Li ,(2018) “Detecting Phishing Websites and Targets Based on URLs and Webpage Links”, 2018 24th International Conference on Pattern Recognition (ICPR).
- [5] Kudirat Oyewumi jimoh, Ibrahim Kazeem, Adepeju Abeke Adigum ,(2022) “Development of a Rule Based Phishing Website Classification System”, International Journal of Engineering Research & Technology (IJERT).
- [6] Buber et al ,(2018) “NLP Based Phishing Attack Detection from URLs”, International Journal of Engineering Research & Technology (IJERT).
- [7] Peng et al ,(2018) “Detecting Phishing Attacks using Natural Language Processing & Machine Learning”, IEEE 12th International Conference on Semantic Computing (ICSC).
- [8] Hassan Y A Abutai, Abdelfettah Belghith ,(2017) “Using a Case Based Reasoning for Phishing Detection”, 8th International Conference on Ambient Systems, Networks and Technologies.
- [9] Mohammad Siful Islam Mamun, Mohammada hmad Rathore, Arash Habibi Lakshkari, Natalia Stakhanova ,(2016) “Detecting Malicious URLs Using Lexical Analysis”, Network and System Security.
- [10] Routhu Srinivasa Rao and Alwyn R. Pais, (2017) " Detection of Phishing Websites using an Efficient Feature-Based Machine Learning Framework", Neural Computing and Applications.
- [11] Choon Lin Tan and Kang Leng Chiew and KokSheik Wong and San Nah Sze,(2016) "PhishWHO: Phishing Webpage Detection via Identity Keywords Extraction and Target Domain Name Finder", Decision Support Systems Journal.
- [12] Hung Le and Quang Pham and Doyen Sahoo and Steven C. H. Hoi,(2018) "URLNet: Learning a URL Representation with Deep Learning for Malicious URL Detection", arXiv: Cryptography and Security
- [13] Kang Leng Chiew and Choon Lin Tan and KokSheik Wong and Kelvin S. C. Yong and Wei King Tiong, (2019) "A New Hybrid Ensemble Feature Selection Framework for Machine Learning-Based Phishing Detection System", In-formation Sciences.
- [14] Ankit Kumar Jain and Brij B. Gupta, (2019) "A Machine Learning Based Approach for Phishing Detection Using Hyperlinks Information", Journal of Ambient Intelligence and Humanized Computing.
- [15] Ankit Kumar Jain and Brij B. Gupta,(2018) "PHISH-SAFE: URL Features-Based Phishing Detection System Using Machine Learning" , Advances in Intelligent Systems and Computing, vol 729. Springer, Singapore..