Detection of Homograph Attacks in Phishing URLs using XGBoost model

Muhammad Ahmad
Department of Information and Communication Engineering
The Islamia University of Bahawalpur
Bahawalpur, Pakistan
ahmad.zahoor303303@gmail.com

Ali Shan
Department of Information and Communication Engineering
The Islamia University of Bahawalpur
Bahawalpur, Pakistan
alirao7778324@gmail.com

Sana Tariq
Department of Information and Communication Engineering
The Islamia University of Bahawalpur
Bahawalpur, Pakistan
sana.tariq@iub.edu.pk

Iram Haider
Department of Information and Communication Engineering
The Islamia University of Bahawalpur
Bahawalpur, Pakistan
iramhaider765@yahoo.com

***Abstract*—Phishing has become one of the most common and dangerous cyber threats. The invaders use fake websites or links that look very similar to the real website so that users can be tricked into sharing personal information, such as passwords, banking details, or login credentials. There is a modern form of phishing known as a homograph attack. In this attack, cyber attackers register URLs that look similar to real websites by using the same characters from different languages or scripts. For example, the Latin letter "a" and the Cyrillic letter "a" look the same but are actually different characters. For a common user, it is difficult to feel the difference, which makes such a type of attack more effective and dangerous. Traditional URL filters and blacklists are often not enough to detect homograph attacks. This system usually checks whether the URL matches the list of harmful links, but the homograph attacks use new or altered URLs that pass such checking. Therefore, in real time, more advanced techniques are required to detect these risks.In this study, we aimed to develop a machine learning-based framework for detecting homograph attacks in phishing URLs. We use both the Natural Language Processing and String Similarity steps to study the URL structure, to indicate suspicious patterns, and to capture difficult manipulation at the character level. We trained our system using a phishing dataset, which has a wide range of fraudulent and harmless URLs. We extract crucial features that are used to train and test machine learning models. From the models we tested, including Random Forest, XGBoost, Long Short-Term Memory (LSTM) networks, and Convolutional Neural Networks (CNN), we determined which method performs best in identifying homograph-based phishing URLs. Thus, XGBoost provides the best accuracy of all models, which makes it a perfect choice for practical or automated security tools.**

***Index Terms*—Phishing detection, Homograph attacks, Machine Learning (ML), URL analysis, Cyber security, XGBoost, Phishing URL dataset, URL based threat analysis, Intelligent URL filtering,**

I. Introduction

Phishing attacks have become one of the most spreading attacks and have created a cyber security risk, affecting people and global organizations.As people rely more on online communication, banking, and shopping, strong security measures are becoming more necessary. [1]–[3]. In various types of phishing attacks, homograph attacks are especially dangerous threats. These attacks exploit different types of visually similar characters (e.g. Cyrillic and Latin alphabet) to help users deceive the websites that appear legitimate. It is difficult to find such attacks. Because the changed URLs appear almost identical which makes it difficult for users to distinguish them without close inspection [4], [5].

There is a challenge to deal with the homograph attacks which is the detection of visualizing similar URLs which are used for fraud. Alternative traditional security measures, such as blacklist-based detection and heuristic-based systems, often fail to find out the latest attacks [6]. Standard URL Filtering Systems are unable to recognize this secret manipulation due to the characters used in homograph attacks looking like Latin letters, making it almost impossible for human consumers to notice without using advanced tools [7]. The increasing experience of invaders complicates the problem and highlights the need for a new detection system that can identify these fraudulent URLs at a deeper level [8].

With time, different ways have been suggested to find the Phishing URL. The initial solutions used to be primarily dependent on Blacklist or Heuristic rules, where known phishing URLs are manually recorded and flagged. While these methods are effective against various types of attacks, these methods struggle with zero-day attacks which are based on new and previously unknown URL conditions [9], [10] .In previous years, researchers have used machine learning (ML) and natural language processing (NLP) techniques to increase detection accuracy. Multiple studies apply a feature-based approach using URL lexical features, as well as character-level analysis. However, these solutions often ignore the homograph-based

DOI: https://doi.org/10.67535/tsp.000001.006
Published under the ICACNC 2025 Proceedings

phishing problem, for which both visual similarity and URL structure need highly developed understanding [11], [12]. Moreover, without integration of deep learning techniques and dependence only on traditional machine learning algorithms limits their ability to catch the complex patterns involved in the homograph attacks [13]–[15].

The gap in the existing research is the lack of using specific advanced ML and NLP techniques to detect homograph-based phishing attacks. While some work focuses on the detection of common phishing attacks, there is a major gap in dealing with the homograph attack, which requires analysis at the character level, Unicode Normalization and visual similarity [16]–[18]. Furthermore, the current solution fails to handle real-time detection of these attacks, especially in an expandable and user-friendly style. The purpose of this research is to overcome this gap by especially focusing on homograph detection using an advanced combination of feature engineering, machine learning models, and deep learning techniques [19].

We propose a novel detection system for homograph attacks in phishing URLs, utilizing a combination of NLP techniques and machine learning models, including Random Forest, XGBoost, LSTM, and CNN. Our focus is on Unicode Normalization and string similarity matrices based on visual and structural samples to analyze and detect fraudulent URLs [20], [21]. Additionally, we review and modify these models on the Phishing URL dataset. Machine learning (ML) and deep learning (DL) models perform better than traditional methods in detecting homograph-based phishing URLs. [22].

Our main contributions are as follows:

- We build a hybrid model with the combination of NLP techniques and machine learning models to detect homograph attacks in phishing URLs.
- We provide a detailed analysis of different machine learning and deep learning models such as Random Forest, XGBoost, LSTM, and CNN.
- We tried to get the best result in over current solution with machine learning and deep learning models in terms of detection accuracy.

The rest of the paper is organized as follows: Section 2 offers a detailed review of the relevant work, features current solution and their limits. Section 3 describes the proposed procedure, including dataset details, pre-processing steps, feature engineering, and model diagnosis. Section 4 offers experimental results and discusses them. Performance of different models. Finally, the section ends the paper and future work.

II. Related Work

Many recent studies have worked toward phishing URL detection because they use machine learning and deep learning models. However, many of them do not focus on homograph attacks using fake lookalike characters within URLs to trick users. Our method targets these specific homograph attacks effectively. Features at the character level and also features that are Unicode-based are used. While explaining our work's improvements, we review key studies below.

Different feature selection techniques in conjunction with SMOTE were employed. Study [23] intended for improved classification. This helps balance the dataset, but it does not check for Unicode manipulation or visual character tricks. Our method normalizes Unicode besides detecting scripts also measures string similarity to better detect such homograph patterns.

A public phishing URL dataset came from study [24]. Detection models are trained using it frequently. However, labeled homograph URLs are not there. For the training, we extracted and identified homograph-like samples by applying preprocessing techniques taken from this dataset.

Study [25] used CNN and RNN deep learning to find general malware. It works well though it lacks homograph tricks or phishing URLs. We adapted deep learning models specifically for phishing URL detection since we targeted visual deception.

Study [26] proposed Shamfinder as a tool for the detection of IDN homograph attacks using domain structure and visual similarity. It handles the normal ASCII-based lookalike attacks with less facility than it handles the powerful internationalized domain names that are non-ASCII. Our system has been designed so it can handle ASCII and IDN-based homographs. It uses NLP along with Unicode analysis for this purpose.

TABLE I
Comparison of Related Studies with Our Work

Study Method	Limitation	Our Improvement
Feature selection [23]	No Unicode or visual similarity check	Added Unicode normalization and character-level features
UCI / Kaggle dataset [24]	No labeled homograph samples	Applied preprocessing to detect homograph patterns
Deep learning for malware [25]	Not focused on phishing or homograph URLs	Used DL models trained on phishing URLs with homograph tricks
Shamfinder for IDN domains [26]	Focus only on non-ASCII homographs	Handles both ASCII and IDN homographs

III. Problem Statement and Motivation

The phishing URL in the homograph attacks offers a big challenge for the traditional detection system. These are the attacks that used similar Unicode Characters to create fraud URLs that look real to the human eye but cause malicious attacks. There is a need to analyze the URL structure and visual similarity to detect such attacks, which is ahead of the ability of traditional URL filters and blacklists [22], [8].

With the increasing digitization of services, users are more exposed to phishing attacks. While many phishing attacks have been detected using signature-based methods. While the homograph attacks in automated systems are left undiscovered.

We need a good system that uses NLP and ML, like analyzing text and models such as CNN and Random Forest, to catch homograph phishing scams. [9], [8].

IV. Methodology

Fig. 1 shows the overall architecture and block diagram of our proposed method, with three main building blocks. i) Data Preprocessing ii) Feature engineering iii) Machine Learning algorithms. Fig. 2 shows shows how many phishing and real URLs are in the dataset. Label 0 means phishing URLs along with legitimate URLs means label 1. The data do contain a greater number of legitimate URLs than phishing ones.

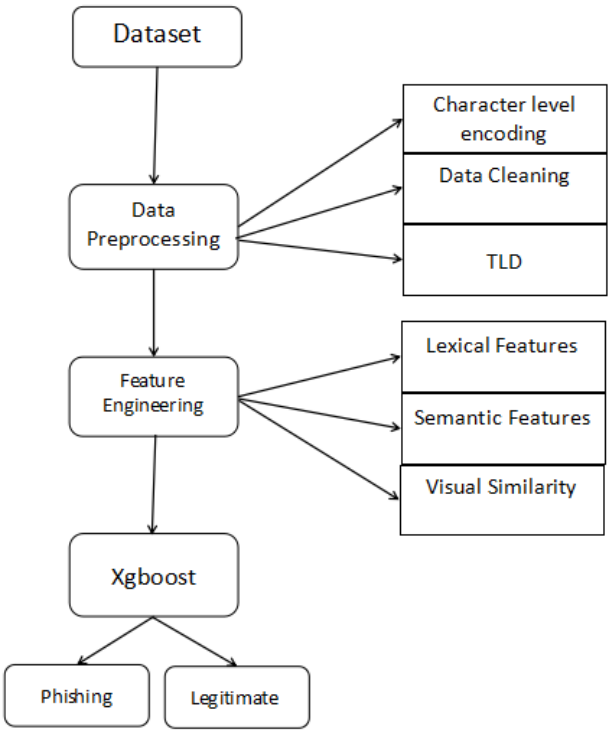

Fig. 1. Flowchart of the Detection System

A. Dataset

Our study uses the publicly available phishing URL dataset from the UCI Machine Learning Repository. This dataset includes a mix of phishing and legitimate URLs for training and testing machine learning and deep learning models. Separate labels regarding homograph attacks are not included in the dataset. We applied pre-processing techniques to extract features with relation to homograph characteristics. These techniques identified Unicode characters, lookalike letters, also abnormal patterns in URLs. This allowed us to study homograph attacks using the original dataset without manually creating new samples.

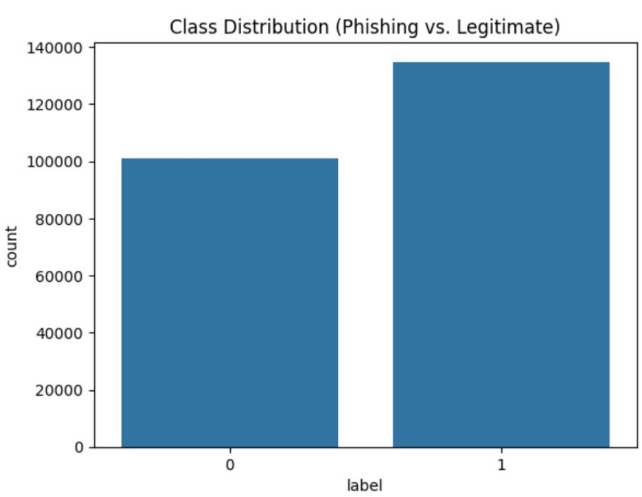

Fig. 2. Class Distribution

B. Data pre-processing

Preprocessing plays an important role in converting the raw URL data for machine learning models into suitable formats for models. The pre-processing steps are as follows:

- **Unicode Normalization**: We apply Unicode Normalization techniques that can be visible to the same standard character but the encoding is different. To decrease noise in the dataset we used the NFC (Normalization Form C) technique to make sure that characters like highlighted letters and special symbols are represented similarly.
- **Character-level Encoding**: As deep learning models subset of machine learning models work best with numerical data therefore we convert URLs into sequences of characters. Each character has been created with a unique integer value, which allows to take action on the data. This encoding captures precise variations in URLs that can indicate fraudulent ways, such as homograph manipulation.

C. Feature Engineering

Feature engineering is important for providing machine learning models with relevant input that can effectively make a difference between legitimate and phishing URLs. We extract the following features to increase model performance:

- **Lexical Features**: We catch basic structural properties of the URL, like its overall length, the number of special characters (e.g., hyphens, underscores, slashes), and digits that can count within the URL. These attributes can be typical of suspicious URLs, as phishing URLs often display unusual lengths or unusual character distributions.
- **Semantic Features**: We analyze the semantic content of the URL, such as the existence of brand names or well-known keywords (e.g., ”bank”, ”login”, ”secure”). Additionally, we review these keyword positions within the URL structure, because phishing invaders often place

suspicious space at specific places (e.g., after domain name or subdomains).

- **Visual Similarity**: According to homograph attacks, we measure string similarity metrics such as Levenshtein distance and Jaccard similarity. These metrics compute how visually similar two URLs are based on their role sequence. A high matching score between a legitimate URL and a homograph URL can indicate an attempted attack.

D. Machine Learning Models

We train and evaluate different machine learning models for the detection of homograph attacks. Models are selected for their capacity to handle different types of data, like sequence data and structured information. There are following models we used :

- **Random Forest**: This learning technique is supported by a combination of decision trees for the prediction. Each tree is trained on a random subset of data, and the final output determines the majority vote of all trees. The tree Random forest is efficient for manipulating complex features.
- **XGBoost**: XGBoost is an advanced grade enhancement model that makes a pair of decisive trees set. XGBoost is known for its high performance and regulatory capabilities, including both ideas ideal and intelligent features to deal with different overfitting models to increase the accuracy of the model.
- **LSTM (Long Short-Term Memory)**: This type of recurrent neural network (RNN) is especially suitable for sequence data. LSTM networks are designed to capture long-term dependencies within the character sequences of URLs. They are effective for modeling URLs that can be fine but important differences, such as homograph attacks, require that the model remembered inputs for the detection of attacks for high accuracy.
- **CNN (Convolutional Neural Network)**: While CNNs are usually used in imaging processing, they are also effective for analyzing sequential data. The CNN can extract the N-gram patterns and structural features. This model helps to identify repeated patterns or structural features that are indicative of phishing efforts.

Each model is trained and tested using a combination of the lexical, semantic, and visual similarity features extracted during the preprocessing and feature engineering steps. The performance of the models is evaluated based on key metrics such as accuracy, precision and F1-score, with special attention on reducing false positives (i.e., legitimate URLs incorrectly flagged as phishing).

V. Experiments and Results

A. Experimental Setup

- **Environment**: The experiments we performed in a Google Colab environment with necessary libraries such as sci-kit-learn, Keras, and TensorFlow. These libraries provided important tools for model creation and training for the detection of deceptive URLs.
- **Evaluation Metrics**: For the evaluation of models was used important key performance metrics: Accuracy, Precision, F1-Score, and AUC-ROC, to ensure a detailed evaluation of model performance among different components of classification. These metrics are very useful in detecting phishing URLs to understand the efficiency of the model.
- **Train-Test Split**: We split the data into 80:20 train-tests which make sure fine model training and testing. This standard ratio allows the model to learn from a enough amount of data for the evaluation of its performance on unseen data, to make sure better generalization and reliability.

B. Result

XGBoost provides the highest performance overall evaluation metrics, which make it a highly efficient model for the detection of homograph attacks in phishing URLs. It fulfilled 99% accuracy, 98% precision, and 99% F1-score which displays its credibility that it is effective to catch difficult patterns in the data. The high AUC-ROC score of 99% also confirms its high accuracy in the difference between legitimate and phishing URLs.

TABLE II
XGBoost Model Performance based on Evaluation Metrics

Metric	Score
Accuracy	99%
Precision	98%
F1-Score	99%
AUC-ROC	99%

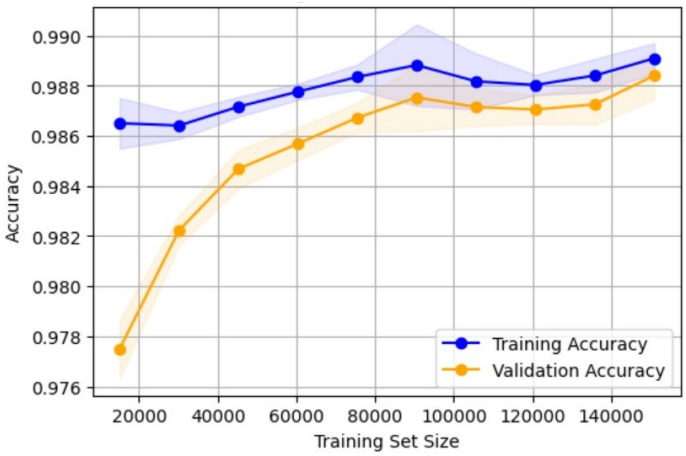

Fig. 3. Learning Curve of the XGBoost Model

Fig. 3 shows the learning curve that is for the XGBoost model strongly shows generalization capabilities. Throughout the training process, the training, as well as validation accuracies consistently remained above 99%. This indicated that the learning was effective indeed. Validation performance and

training show a small gap so the model is not overfitting much. That the two curves align reflects how the model can capture the underlying patterns within the data while it performs strongly on unseen URLs, thus validating it suits the phishing URL detection task.

C. Results Summary

TABLE III
PERFORMANCE COMPARISON OF VARIOUS MODELS

Model	Accuracy	Precision	F1-Score	AUC-ROC
XGBoost	**99%**	**98%**	**99%**	**99%**
LSTM	97%	96%	97%	97%
CNN	95%	93%	96%	96%
Random Forest	82%	78%	86%	79%

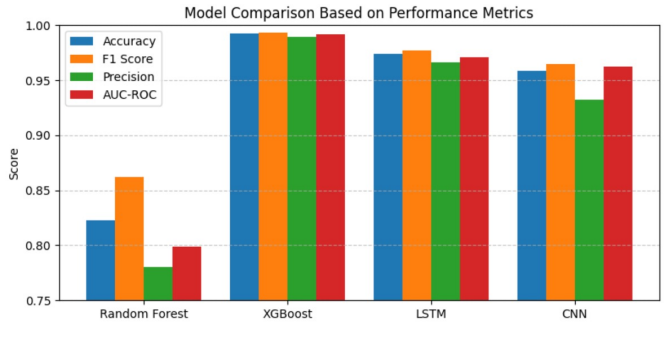

Fig. 4. Model comparison based on performance metrics

Fig. 4 compares Random Forest, XGBoost, and LSTM with CNN models. It uses accuracy and F1 score as well as precision and AUC-ROC as comparison metrics. Random Forest has the lowest scores, while XGBoost performs best with scores near 1.0 in every metric. CNN and LSTM perform strongly also but XGBoost exceeds them slightly.

VI. DISCUSSION

In this study, we enforced different machine learning and deep learning models for the detection of homograph attacks in phishing URLs. From of all of them, the Extreme Gradient Boosting (XGBoost) model displays the highest accuracy and complete performance. This model is extremely ideal because it efficiently catches the fine lexical and character-level patterns that are present in homograph attacks and makes it well suited for these types of cyber threat detection.

Normalization and gradient boosting provide benefits to the XGBoost model by making it useful in decreasing overfitting and enhancing generalization. XGBoost regularly provides higher accuracy, F1-score and precision than the other model during testing which makes it reliable in dealing with imbalanced data and detecting phishing URL structures.

XGBoost can handle a wide number of input features specifically those which are engineered from URL structures like length, character frequency, and Unicode similarity. XGBoost is a powerful tool for real-time detection systems because it allows well-organized tree boosting and parallel computation.

The high-level performance of XGBoost in our project validates that it has the ability for practical deployment in deceptive URL detection tools, especially for identifying homograph attacks that use traditional rule-based or blacklist systems.

VII. CONCLUSION AND FUTURE WORK

This study offers a method to find homograph attacks that use machine learning within phishing URLs. Learning models, include Random Forest, LSTM, and XGBoost. High detection accuracy across all models was achieved by combining lexical, semantic, along character-level features. XGBoost was the most accurate of the models tested since it strongly handled structured data with complex patterns. Because it has robustness and efficiency, it is the most suitable choice for detecting homograph-based phishing URLs in our study.

The future work to do involves expanding on the dataset. For that, we will source more real-world homograph attacks from threat intelligence platforms. We also seek to increase model performance as we test advanced deep learning models like transformer-based architectures (e.g., BERT) to better find contextual patterns. Lastly, we have an intention for the deployment of the model trained in applications for real-time use. This offers protection that is practical against phishing attacks from browser extensions and cloud-based APIs. This research expands the current collection of work when detecting phishing through specific homograph attack targets plus when showing the XGBoost model is effective.

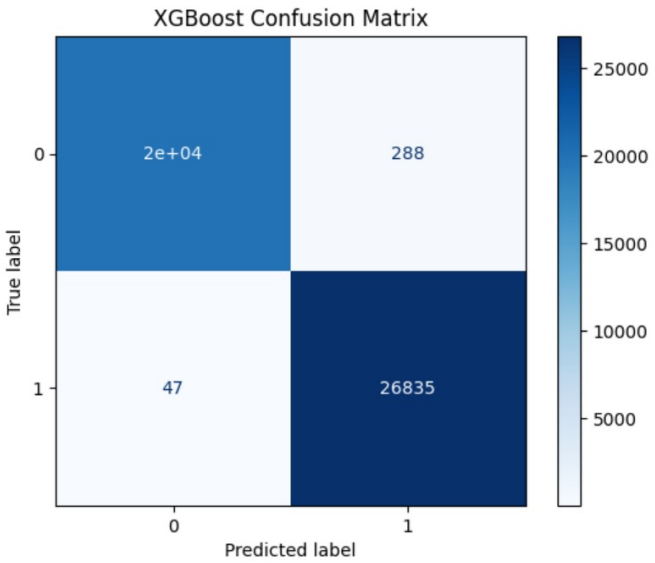

Fig. 5. XGBoost confusion matrices

Fig. 5 shows the XGBoost confusion matrix for our homograph phishing URL detection. It tells us the model caught 26,835 phishing URLs right and only missed 47. It also got

20,000 legit URLs correct but wrongly flagged 288. This shows it's pretty good at spotting scams, with few mistakes.

REFERENCES

[1] I. Haider, K. B. Khan, M. A. Haider, A. Saeed, and K. Nisar, "Automated robotic system for assistance of isolated patients of coronavirus (covid-19)," in *2020 IEEE 23rd International Multitopic Conference (INMIC)*, 2020, pp. 1–6.

[2] I. Haider, M. A. Haider, and A. Saeed, "Big data in internet of things: Architecture and open research challenges," in *2020 IEEE 23rd International Multitopic Conference (INMIC)*, 2020, pp. 1–6.

[3] M. K. Khan, K. Nisar, Y. Farooq, S. Habib, M. Danish, and I. Hyder, "An improved banking application model using blockchain," 2021.

[4] A. Karim, M. Shahroz, K. Mustofa, S. B. Belhaouari, and S. R. K. Joga, "Phishing detection system through hybrid machine learning based on url," *IEEE Access*, vol. 11, pp. 36 805–36 822, 2023.

[5] A. M. Almuhaideb, N. Aslam, A. Alabdullatif, S. Altamimi, S. Alothman, A. Alhussain, W. Aldosari, S. J. Alsunaidi, and K. A. Alissa, "Homoglyph attack detection model using machine learning and hash function," *Journal of Sensor and Actuator Networks*, vol. 11, no. 3, p. 54, 2022.

[6] I. Haider, M. A. Qureshi, A. Amin, and A. Saeed, "An efficient cyber security framework for network intrusion detection using hybrid classifier," in *2023 25th International Multitopic Conference (INMIC)*, 2023, pp. 1–6.

[7] E. Uçar, M. Ucar, and M. O. İncetaş, "A deep learning approach for detection of malicious urls," in *6th international management information systems conference*, 2019, pp. 12–20.

[8] J. Sern Lee, G. P. D. Yam, and J. Hao Chan, "Phishgan: Data augmentation and identification of homoglpyh attacks," *arXiv e-prints*, pp. arXiv–2006, 2020.

[9] T. P. Thao, "Improving homograph attack classification," *arXiv preprint arXiv:2009.08006*, 2020.

[10] T. Phuong Thao, "Improving homograph attack classification," *arXiv e-prints*, pp. arXiv–2009, 2020.

[11] D. Viji, V. Dixit, and V. Jha, "Phishing website detection and classification," in *Proceedings of International Conference on Deep Learning, Computing and Intelligence: ICDCI 2021*. Springer, 2022, pp. 401–411.

[12] G. Sonowal and K. Kuppusamy, "Phidma–a phishing detection model with multi-filter approach," *Journal of King Saud University-Computer and Information Sciences*, vol. 32, no. 1, pp. 99–112, 2020.

[13] A. Ozcan, C. Catal, E. Donmez, and B. Senturk, "A hybrid dnn–lstm model for detecting phishing urls," *Neural Computing and Applications*, pp. 1–17, 2023.

[14] S. Tariq and A. Amin, "Deepctf: transcription factor binding specificity prediction using dna sequence plus shape in an attention-based deep learning model," *Signal, Image and Video Processing*, vol. 18, no. 6, pp. 5239–5251, 2024.

[15] A. Farooq and K. H. Memon, "Kernel possibilistic fuzzy c-means clustering algorithm based on morphological reconstruction and membership filtering," *Fuzzy Sets and Systems*, vol. 477, p. 108792, 2024.

[16] R. Liu, Y. Wang, H. Xu, Z. Qin, Y. Liu, and Z. Cao, "Malicious url detection via pretrained language model guided multi-level feature attention network," *arXiv preprint arXiv:2311.12372*, 2023.

[17] T. P. Thao, Y. Sawaya, H.-Q. Nguyen-Son, A. Yamada, A. Kubota, T. Van Sang, and R. S. Yamaguchi, "Influences of human demographics, brand familiarity and security backgrounds on homograph recognition," *arXiv preprint arXiv:1904.10595*, 2019.

[18] Z. Ahmad, A. Shahid Khan, K. Nisar, I. Haider, R. Hassan, M. R. Haque, S. Tarmizi, and J. J. Rodrigues, "Anomaly detection using deep neural network for iot architecture," *Applied Sciences*, vol. 11, no. 15, p. 7050, 2021.

[19] M. Z. Ansari, T. Ahmad, S. Khan, F. Mabood, and M. Faizan, "Homograph language identification using machine learning techniques," in *Proceedings of International Conference on Data Science and Applications: ICDSA 2022, Volume 1*. Springer, 2023, pp. 863–873.

[20] F. Z. Qachfar and R. M. Verma, "Homograph attacks on maghreb sentiment analyzers," *arXiv preprint arXiv:2402.03171*, 2024.

[21] J. Woodbridge, H. S. Anderson, A. Ahuja, and D. Grant, "Detecting homoglyph attacks with a siamese neural network," in *2018 IEEE Security and Privacy Workshops (SPW)*. IEEE, 2018, pp. 22–28.

[22] J. Nowak, M. Korytkowski, M. Wozniak, and R. Scherer, "Url-based phishing attack detection by convolutional neural networks." *Aust. J. Intell. Inf. Process. Syst.*, vol. 15, no. 2, pp. 60–67, 2019.

[23] N. Bhoj, A. Tripathi, G. S. Bisht, A. R. Dwivedi, B. Pandey, and N. Chhimwal, "Comparative analysis of feature selection techniques for malicious website detection in smote balanced data," *RS Open Journal on Innovative Communication Technologies*, vol. 2, no. 3, 2021.

[24] M. Siddhartha, "Malicious urls dataset," *Kaggle.[Online]. Available: https://www. kaggle. com/datasets/sid321axn/malicious-urls-dataset.[Accessed: 05-May-2023]*, 2021.

[25] R. Vinayakumar, M. Alazab, K. Soman, P. Poornachandran, and S. Venkatraman, "Robust intelligent malware detection using deep learning," *IEEE access*, vol. 7, pp. 46 717–46 738, 2019.

[26] H. Suzuki, D. Chiba, Y. Yoneya, T. Mori, and S. Goto, "Shamfinder: An automated framework for detecting idn homographs," in *Proceedings of the Internet Measurement Conference*, 2019, pp. 449–462.