Phishing Detection in Customer Support E-mails

Aroob Mukhtar
Dept. of Information and Communication Engineering
The Islamia University of Bahawalpur
Bahawalpur, Pakistan
aroobmukhtar850@gmail.com

M. Madni
Dept. of Information and Communication Engineering
The Islamia University of Bahawalpur
Bahawalpur, Pakistan
armanmanni186@gmail.com

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

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

Abstract—Phishing is the most widespread cyberattack in the world, and email remains the most frequently used medium. Traditional machine learning models are one method for phishing detection. Many studies have been conducted to identify phishing in customer service emails. However, no one has looked into combining the Transformer and Graph Neural Network (GNN) to identify fraudulent emails sent to customer service. This study proposes a hybrid model that combines the GNN for graphical features with the RoBERTa for textual features. The experiments were performed with the CEAS-08 phishing email dataset, an openly available benchmark released in the 2008 CEAS Live Spam Challenge. The dataset is partitioned into 70% for training, 30% for validation, and testing to evaluate performance. The Hybrid RoBERTa + GNN model developed here attained an astonishing accuracy of 98.40% with the frozen RoBERTa base on Google Colab Free. In comparison with other traditional classifiers like SVM, KNN, and Naive Bayes, the new hybrid model provides more contextualized knowledge with RoBERTa and relation knowledge with GNN. Future work may investigate fine-tuning the RoBERTa base to further improve its performance. While the main objective is to open a new research pathway that offers deep contextual and structural insights, the suggested model achieves this aim, unlike conventional models that fall short in this regard.

Keywords: Phishing Detection, Customer Support Emails, Roberta-GNN, Email Security

I. Introduction

As the world begins to progress, cybercrime continues to evolve alongside technological progress. People use digital media, assets, and communication platforms to perform crimes like cyberattacks. Phishing is also one of these social engineering attacks [1]. The main aim of these attacks is to exploit the weakness found in system processes caused by system users [2]–[4].

Phishing is a cybercrime that convinces people to provide sensitive information, for instance, account IDs, passwords, and bank details. Emails, instant messages, and phone calls are widely used to launch such a cyber attack [5]. Mostly, attackers use phishing emails and disguise themselves as legitimate emails to get sensitive information and login credentials.

In industries, cybercriminals use phishing emails to compromise the security of organizations by tricking employees into revealing sensitive information or clicking on malicious links, potentially leading to data breaches and financial losses [6], [7].

Data breaches in the United States were recorded about 1001 times in 2020 [8]. Consequently, more than 155.8 million individuals were impacted by these breaches that year. As reported by the Identity Theft Resource Center's (ITRC) data breach study, there were 1,291 breaches from September 2020 to September 2021. Compared to the 1,108 data breaches reported in 2019, this statistic represents an 8 percent rise [8].

Phishing detection becomes more complex because attackers continually change their strategies to bypass security measures. With time, they change their tactics and make the previous phishing detection methods obsolete. Cybercriminals widely use clever tricks like misspelling, hidden text, and professional-sounding language to make it more difficult to detect or analyze phishing emails accurately [9]. Traditional models used for phishing detection are KNN, SVM, Naive Bayes, logistic regression, and linear regression.KNN leverages the model's ability to identify similarities and patterns within sentences to detect potentially dangerous content. The performance of this process is dependent on attribute quality, 'K' selection, and the distance measure, and these can potentially need to be adjusted to work best [10].

KNN is not the best model due to several issues, including the need for feature scaling and k-selection, and many other issues in handling large or complex data.

The Naive Bayes classifier can effectively classify phishing sites using its probabilistic approach in classifying sites as legitimate or malicious based on the number of attributes. Common characteristics may be the presence of specific keywords, length of URL, domain age, status of the SSL certificate, etc [11].

Naive Bayes has certain limitations, like poor perfor-

DOI: https://doi.org/10.67535/tsp.000001.011
Published under the ICACNC 2025 Proceedings

mance with complex language patterns, assuming equal importance of features, difficulty in handling imbalanced data, and limited handling of feature data.

SVM is a supervised machine learning technique applied to both regression and classification tasks to identify an optimal hyperplane that divides data points belonging to distinct classes. This approach is especially useful for data that cannot be separated linearly, utilizing an n-dimensional space to strategically position each data point to determine the most suitable hyperplane for classification [12].

There are several limitations to support vector machines, including their incompatibility with real-time detection, sensitivity to noisy data, scalability issues, difficulty in handling imbalanced data, complex parameter tuning, and inefficiency with high-dimensional data [13]–[16].

Among the various machine learning algorithms categorized under Supervised Learning, Logistic Regression (LR) is particularly notable. It utilizes a set of different variables to predict the outcome of the dependent categorical factor. If the dependent variable is categorical, LR can assist in predicting its value. As a result, the outcome will be a discrete or categorical value. Instead of providing exact answers of yes or no, true or false, etc. It generates estimates ranging between 0 and 1. Logistic and linear regression are the appropriate choices for classification challenges [17]. There are some limitations of logistic regression, like other ML algorithms.

In addition, modern transformer models like BERT, RoBERTa, DistilRoBERTa, and XLNet also provide good and high accuracy greater than 95 percent in capturing complex contextual patterns.

There are certain limitations of modern BERT transformers: they take higher Computational costs, are memory-intensive, require large labeled datasets, and context understanding can be misleading. In this paper, we examine the accuracy of the Roberta and GNNs hybrid models and provide a new research path.

II. Related Work

In starting the phishing emails, structural, semantic features, and text-based features are detected by the conventional ML models like Naive Bayes, Linear Regression, Logistic Regression, and Support Vector Machine to classify the emails [18]. They have achieved a lot of success, but there are certain limitations due to predefined features and new engineering tactics [19].

Lightweight Ethereum transaction networks were built from a suggested end-to-end architecture of Phishing Detection Graph Neural Network (PDGNN), and this graph differentiates subgraphs of phishing accounts. Applying a Chebyshev-GCN-based detection model, PDGNN outperforms the classical methods in massive classical Networks and can well separate normal accounts from phishing accounts [20], [21].

KS Jishnu and B. Arthi contrasted Roberta and LSTM for the detection of phishing URLs. This work introduces a novel method to identify phishing URLs through RoBERTa transformer-based feature extraction and LSTM classification. RoBERTa extracts the contextual and semantic information of URLs when extracting the features. With the embedding of URLs into contextual representations, RoBERTa can learn to represent the URLs in a manner that reflects the deep meaning and context of the URLs. The LSTM layer of the model accurately classifies URLs by capturing their sequential connections with the extracted features. The data set is large, with 300,000 URLs. The suggested system distinguishes phishing and valid URLs efficiently with 97.14% accuracy following extensive training and testing [22].

For the Ethereum fraud detection, Kanezashi et al studied the application of heterogeneous GNNs, and during their study, they also compared the performance of homogeneous and heterogeneous models. They discovered that diverse models—especially the relational graph convolutional network (RGCN)—achieved better accuracy in spotting phishing operations [23].

Izotova and Lavrova investigate the use of GNNs to detect fake posts on social networks. A new graph-based model that investigates the relationship between posts and users, allowing them to detect fraudulent content with high precision [24].

All the algorithms and techniques are not enough to detect phishing with perfect accuracy as needed because there are some challenges like imbalanced data, adversarial attacks, and higher computational costs [25].

The researcher presents a simple and fast method of generating adversarial examples to mitigate adversarial attack issues. Using this approach to provide examples for adversarial training, we reduce the test set error of a maxout network on the MNIST dataset [26]. The researcher further proposed Eda: Easy data augmentation techniques for boosting performance on text classification tasks. However, further research is underway to mitigate all the above challenges [27].

While some earlier studies have attained good performance by employing transformers or GNNs alone, none have harnessed their strengths together for phishing detection. The majority of models are concerned with only textual or structural features, ignoring the email content–sender–receiver relationship interaction. In an effort to rectify this shortcoming, the Hybrid RoBERTa + GNN model in this work integrates semantic and graph representation to achieve better detection accuracy and generalization.

III. Methodology

In the Methodology section, first, we briefly introduce our hybrid model, the integration of Roberta for text analysis, and GNN for the email sender-receiver relationship.

We use the CEAS-08 dataset to train the model that contains legitimate and phishing emails.

3.1 Dataset Description

The proposed model was tested using the CEAS-08 phishing email dataset. The dataset was designed for the 2008 CEAS Live Spam Challenge and contains both ham (legitimate) as well as spam (phishing) emails. While this dataset is fixed, it is small in size relative to and mainly used for spam-v-ham classification and not suitable for large-scale fine-tuning of RoBERTa or learning complexity graphs.

3.1.1 Dataset Features:

The following are the features of the dataset CEAS-08:

- Sender: Sender's email address.
- Receiver: Receiver's email address.
- Subject: The Main Email Subject Line.
- Body: Main text content of the email.
- URLs: Links found in the email body.
- Timestamp: Email sending date and time.
- Label: Represents whether the email is ham (0) or spam (1).

3.2 Data Preprocessing

As Fig. 1 shows that we have to preprocess the emails to clean the text and construct graphs before training the model.

3.2.1 Text Processing: Since we are utilizing the Roberta tokenizer, there is no requirement to manually perform preprocessing tasks such as: Eliminating special characters, Converting to lowercase, Removing HTML tags.

We apply the following steps to prepare the email text for Roberta:

- By using byte-level BPE, we perform subword tokenization
- (CLS, SEP, etc) Special tokens are added
- Generation of input_ids and attention mask
- Pad & truncate tokens (128 tokens max)

$$T = \text{Tokenizer(Email Body)} \tag{1}$$

As Eq.1 shows, following tokenization, the text is processed through Roberta to retrieve the CLS token embedding (768 dimensions).

3.2.2 Email Data Preprocessing: In this step, we model communication patterns:

- Initially, we convert the email addresses of both the sender and receiver into integer IDs through a process called factorization.
- To represent the email interactions, we create graph edges using the IDs generated in the first step.
- To ensure consistency within the graph, we split the dataset into train/val/test; each split contains its encoding.

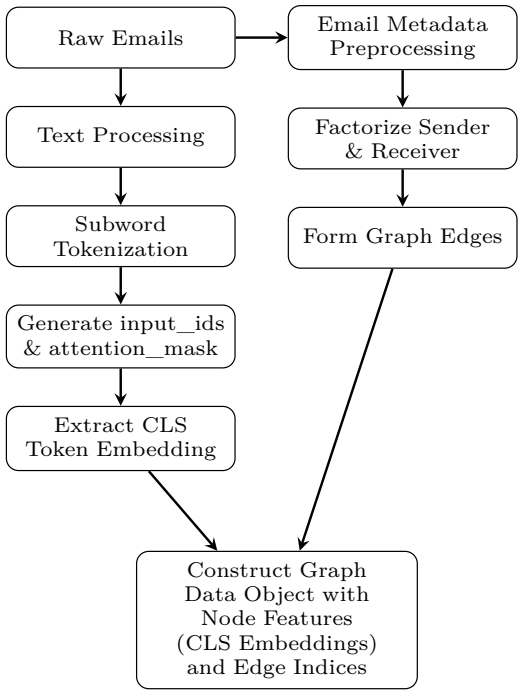

Fig. 1. Flowchart of Data Preprocessing Pipeline

3.3 Graph Construction

Sender and recipient email addresses were converted into numeric IDs in order to build a reduced graph indicating communication patterns. Yet, since the CEAS-08 dataset does not contain sender–receiver inherent network relationships, the graph is an approximation for phishing detection test material of graph-based reasoning.

As Eq.2 shows, the graph is represented as G.

So the graph G is defined as:

$$G = (V, E) \tag{2}$$

Where:

$$V = \text{set of nodes (email addresses)}$$

$$E = \text{set of connections (edges)}$$

These connections are directed from sender to receiver. Each node within the graph holds the feature vector generated by Roberta, while the edges represent the communication between the sender and the receiver. For the graph's implementation, we utilize the Geometric.data.Data class, where the edge indexes illustrate the dir

3.4 Hybrid RoBERTa+GNN Model

In this paper, the hybrid model of RoBERTa and GNN is used, which uses both text features and graph features. Fig.2. shows the hybrid architecture of RoBERTa-GCN, and below is a detailed description of this architecture:

3.4.1 RoBERTa for Feature Extraction

The frozen Roberta base model is used for obtaining CLS embeddings. As shown in Eq.3, the Roberta model extracts the email text into 768-dimensional embeddings:

$$E = \text{Roberta}(T) \tag{3}$$

Where:

- E = extracted features
- Roberta = pre-trained Roberta model
- T = input text (email body)

After passing these embeddings to the GCN layer (GCNConv) to reduce them to 256 dimensions.

3.4.2 GNN for Graph Encoding

In this subsection, as shown in Eq.4, we use a single-layer GCNConv with 768 inputs and 256 output dimensions. The model makes use of the Graph Convolutional Network (GCN) operator from PyTorch Geometric (PyG) for edge processing of graph edges as well as node relationships.

$$H^{(1)} = \text{ReLU}(\text{GCNConv}(X, \text{edge\_index})) \tag{4}$$

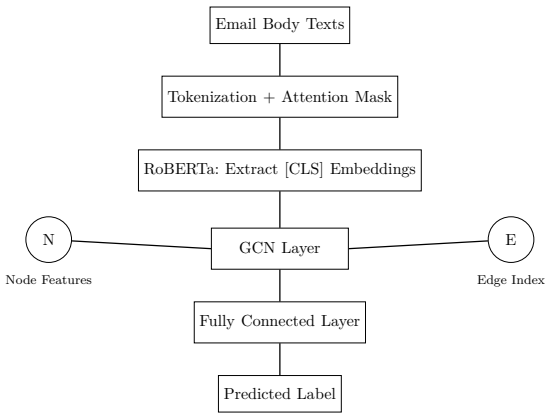

Fig. 2. Architecture of the Hybrid RoBERTa-GCN Model

3.5 Training and Evaluation

In Fig.3, Training and Evaluation with Auto-Resume training mechanisms is shown:

3.5.1 Experimental setup:

The following are the requirements:

- Platform: Google Colab free
- Hardware: The Tesla T4 GPU is used.
- Optimizer: AdamW
- Learning Rate: optimized via Optuna (1e-4)
- Loss Function: Binary Cross-Entropy
- Batch Size: 16 emails
- Epochs: 10

During training:

The CEAS-08 dataset is split into 70% for training, 15% for validation, and 15% for testing to evaluate performance.

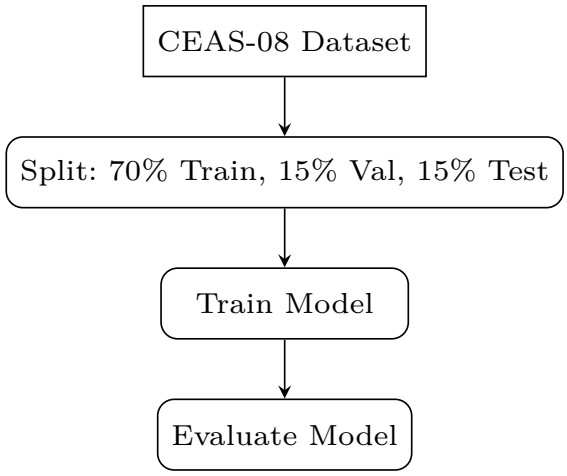

Fig. 3. Training and evaluation pipeline for the proposed model.

3.5.2 Evaluation:

Model Performance Metrics:

- Accuracy
- Precision
- Recall
- F1 Score
- Confusion Matrix

To perform the evaluation, we use a test and a validation dataset.

Hyperparameter Tuning with Optuna:

Optuna is used to optimize the model parameters:

- Hidden dimensions: 64 to 256

IV. Result and Discussion

	Validation	Test
Accuracy	98.62%	98.40%
Precision	0.98	0.98
Recall	0.99	0.99
F1 Score	0.99	0.99

Fig. 4. Results of the Hybrid RoBERTa-GNN Model on Validation and Test Sets

As shown in Fig.4, the proposed model achieves high accuracy on the CEAS-08 dataset: The model achieves 98.62 percent accuracy on the validation dataset and 98.40 percent accuracy on the test dataset. It not only gives high accuracy but also provides good precision in both validation and test data sets. The model achieves 0.98 precision on validation and 0.98 precision on the test dataset.

While these findings are encouraging, such near-optimal performance on a small data set can be a sign of overfitting or insufficiency of data sets. Hence, future testing needs to include cross-validation and testing on larger, more

heterogeneous corpora of phishing attacks in order to evaluate model stability

These findings show that the combination of contextual representations from RoBERTa with structural graph features from the GNN clearly improves phishing detection performance. Utilizing a frozen RoBERTa base enhances the efficiency of computation while preserving high-text embeddings. Complete fine-tuning of RoBERTa on a large dataset would further enhance adaptability and real-world performance.

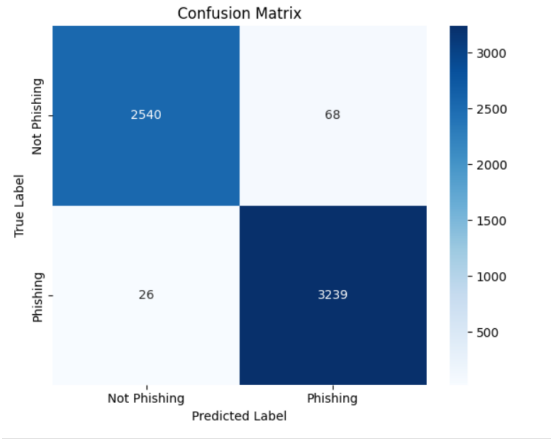

Fig. 5. Confusion matrix of the phishing detection model.

As shown in Fig.5, the confusion matrix of the test dataset. Out of all " phishing emails, " 3239 were correctly identified while only 26 emails were misclassified as "not phishing ". For "not phishing " emails, 2540 are correctly identified, while 68 emails are incorrectly classified as phishing.

V. Comparison of Different Models

TABLE I
Comparison of Model Performance Metrics

Model	Accuracy	Precision	Recall	F1-score
Proposed	0.9840	0.980	0.990	0.990
SVM	0.9946	0.9991	0.9912	0.9951
KNN	0.8383	0.7745	0.9993	0.8727
Naive Bayes	0.9769	0.9965	0.9627	0.9788
RoBERTa (est.)	0.9805	0.9832	0.9760	0.9796
GNN (est.)	0.9512	0.9631	0.9405	0.9516

The above table shows the comparative analysis of the models' performance and their results.

The Support Vector Machine (SVM) used the linear kernel because it works very well in text classification due to high-dimensional data that is ideal for a linear kernel. Using poly or RBF is more computationally expensive and not well generalized for this type of dataset.

But such a result can be due to dataset properties or training and test example overlap. Also, while SVM classifies text perfectly, it cannot learn contextual or semantic relations, which deep models like transformers can learn easily

The next is K Nearest Neighbor (KNN). The KNN classifier used K=5. It classified the majority of labels that are among the k Nearest Neighbor data points. Its performance tends to be slower when dealing with large datasets, is less effective with high-dimensional texts, and requires a significant amount of memory during inference.

The Naive Bayes (Multinomial): This model mainly uses the probabilities of the features based on class labels using Bayes' theorem. It assumes feature independence that doesn't hold in the language, but it still performs well sometimes. This model gives lower accuracy than SVM, but it is still fast and lightweight and is often used as a baseline for NLP tasks.

The suggested Hybrid RoBERTa–GNN model combines the RoBERTa transformer to learn contextualized rich representations from the content of the email with a Graph Neural Network (GNN) on the basis of Graph Convolutional Network (GCN) layers to learn sender–receiver relationships. The combination allows the model to learn semantic as well as structural features jointly and discover textual content and patterns of communication, which machine learning cannot otherwise identify. RoBERTa base was used in a frozen state as a computationally efficient feature extractor to provide a balance between contextual quality and computational efficiency.

Meanwhile, this GNN module represented inter-email relations, embedding relational knowledge into textual embeddings, and boosting the overall phishing detection capability of the model. But as CEAS-08 is comparatively small and was mostly tested for filtering spam, there is a need for future comparisons on larger phishing-specific datasets to validate these results.

VI. Conclusion

This research introduces an innovative method that combines Roberta and GNN for detecting phishing in customer support emails using the CEAS-08 dataset. The proposed framework presents a novel and competitive strategy compared with conventional models such as SVM, KNN, and Naïve Bayes. Although the SVM baseline achieved marginally higher accuracy than that of the proposed model. The primary objective is to explore new research opportunities by combining deep contextual insights with the structural aspects of an email.

Overall, the above hybrid model addresses semantic and relational aspects that are usually neglected by conventional approaches, thus being highly suitable for real-world phishing detection. Its robust performance also attests to the appeal of blending transformer-based and graph-based learning towards pushing forward with intelligent email security systems

The recent study has certain limitations that open up opportunities for future research:

- The base RoBERTa was left frozen, constraining more profound contextual adaptation; follow-up research will seek fine-tuning for better linguistic representation.
- The graph topology is impoverished because CEAS-08 does not have explicit sender-receiver relations; future research should utilize richer, more real-world-like communication graphs.
- CEAS-08 is a small, partially unbalanced, and spam-dominated dataset, which may constrain generalizability; larger phishing-domain datasets are encouraged.
- Irregular or noisy email content might have affected tokenization; noise-robust preprocessing will be included in upcoming experiments.
- Cross-validation and wider evaluation measures (e.g., ROC-AUC, loss curves) will be incorporated to improve model validation and scalability.

References

[1] I. Haider, M. A. Haider, and A. Saeed, "Big Data in Internet of Things: Architecture and Open Research Challenges," Proc. 2020 IEEE 23rd International Multitopic Conference (INMIC), pp. 1–6, 2020.

[2] M. Khonji, Y. Iraqi, and A. Jones, "Phishing detection: A literature survey," IEEE Communications Surveys & Tutorials, vol. 15, no. 4, pp. 2091–2121, 2013.

[3] R. Jabir, J. Le, and C. Nguyen, "Phishing Attacks in the Age of Generative Artificial Intelligence: A Systematic Review of Human Factors," AI, vol. 8, article no. 174, 2025.

[4] M. R. Haque, S. C. Tan, Z. Yusoff, K. Nisar, R. Kaspin, I. Haider, S. Nisar, J. P. C. Rodrigues, B. S. Chowdhry, M. A. Uqaili, et al., "Unprecedented Smart Algorithm for Uninterrupted SDN Services During DDoS Attack," Computers, Materials & Continua, vol. 70, no. 1, pp. xxx–yyy, 2022.

[5] S. Salloum, T. Gaber, S. Vadera, and K. Shaalan, "Phishing email detection using natural language processing techniques: A literature survey," Procedia Computer Science, vol. 189, pp. 19–28, 2021.

[6] 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)," Proc. 2020 IEEE 23rd International Multitopic Conference (INMIC), pp. 1–6, 2020.

[7] P. K. K. Loh, A. Z. Y. Lee, and V. Balachandran, "Towards a Hybrid Security Framework for Phishing Awareness Education and Defense," Future Internet, vol. 16, no. 3, article no. 86, 2024.

[8] M. Alawida, A. E. Omolara, O. I. Abiodun, and M. Al-Rajab, "A deeper look into cybersecurity issues in the wake of COVID-19: A survey," Journal of King Saud University — Computer and Information Sciences, vol. 34, no. 10, pp. 8176–8206, 2022.

[9] R. S. Rokade and D. D. Doye, "Spelled sentence recognition using Radon transform," Proc. 2014 Science and Information Conference, pp. 351–354, 2014.

[10] L. Sawe, J. Gikandi, J. Kamau, and D. Njuguna, "Sentence level analysis model for phishing detection using KNN," Journal of Cybersecurity, vol. 6, pp. 2579–0072, 2024.

[11] A. N. Lone, M. Alam, S. Mustajab, M. Mustaqeem, M. Shahid, and F. Ahmad, "Performance Evaluation on Detection of Phishing Websites Using Machine Learning Techniques," Proc. 2024 International Conference on Electrical Electronics and Computing Technologies (ICEECT), vol. 1, pp. 1–6, 2024.

[12] R. Prasad et al., "Phishing email detection using machine learning: A critical review," Proc. 2024 IEEE International Conference on Computing, Power and Communication Technologies (IC2PCT), vol. 5, pp. 1176–1180, 2024.

[13] 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, pp. 108792, 2024, doi: 10.1016/j.fss.2023.108792.

[14] I. Haider, M. A. Mehdi, A. Amin, and K. Nisar, "A Hand Gesture Recognition Based Communication System for Mute People," Proc. 2020 IEEE 23rd International Multitopic Conference (INMIC), pp. 1–6, 2020.

[15] S. Tariq and A. Amin, "Detection of DNA-Protein Binding Using Deep Learning," Proc. 2023 IEEE International Conference on Emerging Trends in Engineering, Sciences and Technology (ICES&T), pp. 1–4, 2023.

[16] 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.

[17] D. P. Garapati, L. V. A. Priya, K. P. Swaroop, B. Samyuktha, G. H. Sowmya, and B. H. N. Valli, "A Comparative Analysis of Logistic Regression, Support Vector Machines, and Random Forest for Phishing Website Identification," Proc. 2024 International Conference on Computational Intelligence for Green and Sustainable Technologies (ICCIGST), pp. 1–5, 2024.

[18] N. Altwaijry, I. Al-Turaiki, R. Alotaibi, and F. Alakeel, "Advancing phishing email detection: A comparative study of deep learning models," Sensors, vol. 24, no. 7, p. 2077, 2024.

[19] S. Sarkar, A. Yadav, and T. Balachander, "Email Phishing Detection Using AI and ML," Proc. International Conference on Deep Sciences for Computing and Communications, pp. 357–377, 2023.

[20] X. Li, Y. Wang, and J. Zhang, "PDGNN: A Phishing Detection Graph Neural Network for Ethereum," arXiv:2204.08194, 2022.

[21] Z. Ahmad, A. S. Khan, K. Nisar, I. Haider, R. Hassan, M. R. Haque, S. Tarmizi, and J. J. P. C. Rodrigues, "Anomaly Detection Using Deep Neural Network for IoT Architecture," Applied Sciences, vol. 11, no. 15, p. 7050, 2021.

[22] K. S. Jishnu and B. Arthi, "Phishing URL detection by leveraging RoBERTa for feature extraction and LSTM for classification," in Proc. 2023 Second International Conference on Augmented Intelligence and Sustainable Systems (ICAISS), pp. 972–977, 2023.

[23] H. Kanezashi, K. Nakamura, and H. Sasaki, "Ethereum Phishing Detection with Heterogeneous Graph Neural Networks," arXiv:2203.12363, 2022.

[24] A. Izotova and E. Lavrova, "Detection of Fake Posts Using Graph Neural Networks," Automatic Control and Computer Sciences, vol. 55, no. 8, pp. 1034–1043, 2021.

[25] K. Grosse, N. Papernot, P. Manoharan, M. Backes, and P. McDaniel, "Adversarial Examples for Malware Detection," in Proc. ESORICS 2017 — 22nd European Symposium on Research in Computer Security, pp. 62–79, 2017.

[26] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and Harnessing Adversarial Examples," arXiv:1412.6572, 2014.

[27] J. Wei and K. Zou, "EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks," Proc. ACL, 2019.

[28] I. Haider, M. A. Qureshi, A. Saeed, and M. A. Haider, "Enhanced Model for Data Security in Cyberspace Using Combined Steganographic and Encryption Techniques," Proc. 2023 IEEE International Conference on Emerging Trends in Engineering, Sciences and Technology (ICES&T), pp. 1–6, 2023.