Optimizing Phishing URL Detection with TF-IDF, M-Relief, and RoBERTa: A Deep Learning Approach

Vijayalakshmi.V[1] Suguna.S[2]

[1]*Research Scholar, PG and Research Department of Computer Science, Sri Meenakshi Government arts and science college for Women*

[2]*Associate Professor, PG and Research Department of Computer Science, Sri Meenakshi Government arts and science college for Women*

Abstract- **Malicious URLs are a major cyber security threat, enabling attacks like phishing and malware. Traditional detection methods, such as blacklists and heuristics, often miss new or disguised threats. To improve detection, machine learning and deep learning are increasingly used, though they depend on large, regularly updated datasets. This study introduces a novel phishing URL classification method that combines TF-IDF for feature extraction, Label Encoding for transforming categorical data, Borderline SMOTE to address class imbalance, M-Relief for feature selection, and RoBERTa, a transformer-based deep learning model, for final classification. The dataset includes a diverse mix of phishing and legitimate URLs. The effectiveness of the models is assessed by measuring their accuracy, analyzing precision, recall, confidence score, confusion matrix, histogram and AUC-ROC specifically for the classification of malware attacks. The fine-tuned RoBERTa model demonstrates superior performance in phishing detection, achieving 98.3% accuracy on the test set. Compared to traditional classifiers like Random Forest, SVM, and XGBoost, RoBERTa excels in identifying phishing URLs with higher precision and recall. The proposed approach proves effective for real-time phishing detection, enhancing overall cyber security protection.**

Keywords: **Borderline SMOTE, TF-IDF, XGBoost, RoBERTa, Label encoding**

I. INTRODUCTION

The rapid expansion of the internet has led to a rise in cyber threats, with phishing attacks being one of the most widespread and deceptive forms of cybercrime. Phishing involves deceiving users into disclosing sensitive information, such as login credentials and financial details, through fraudulent websites that imitate legitimate ones.

Cybersecurity reports indicate a global surge in phishing attacks, with cybercriminals employing increasingly sophisticated techniques to evade detection. Traditional rule-based filtering and blacklist-based detection methods struggle to keep up with these evolving threats. As a result, machine learning (ML) and deep learning (DL) approaches have gained popularity for phishing URL detection due to their ability to recognize patterns in data and identify new threats in real time.

This study introduces an advanced phishing URL classification framework that integrates multiple techniques to improve detection accuracy and robustness. The Term Frequency-Inverse Document Frequency (TF-IDF) method is used to extract meaningful textual features from URLs, while Label Encoding converts categorical data into a numerical format for better representation in ML models. To address data imbalance, the Borderline Synthetic Minority Over-sampling Technique (Borderline SMOTE) ensures a balanced distribution of phishing and legitimate URLs. Additionally, M-Relief is employed for feature selection, enhancing model efficiency by filtering out irrelevant features. Finally, the RoBERTa (Robustly Optimized BERT Pretraining Approach) model, a transformer-based deep learning framework, is fine-tuned for phishing detection.

Extensive experiments are conducted to assess the proposed method against traditional ML classifiers such as Support Vector Machine (SVM), Random Forest (RF), and XGBoost. The results demonstrate that RoBERTa achieves a superior accuracy of 98.3%, outperforming conventional models in phishing detection. This research highlights the importance of integrating feature extraction, resampling techniques, and deep learning to enhance phishing URL classification.

The rest of this paper is organized as follows: Section 2 discusses related work on phishing detection. Section 3 presents the proposed methodology, covering the dataset, feature extraction, and model training. Section 4 provides experimental results and performance evaluation. Finally, Section 5 concludes the paper with key findings and future research directions.

II. RELATED WORK

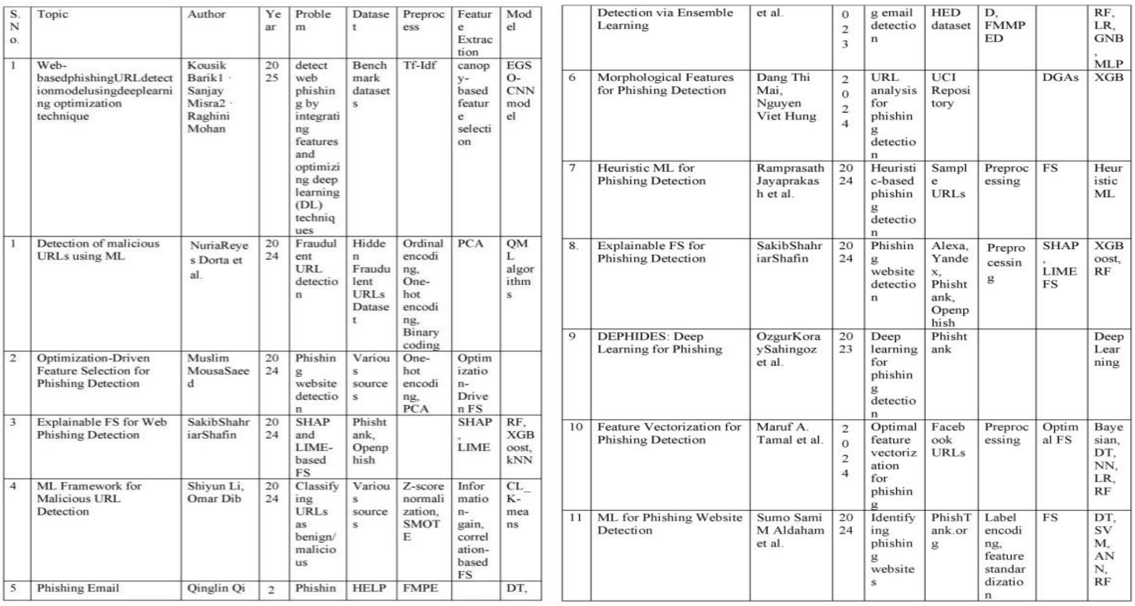
1. Barik et al.[1] (2025) proposed a Web-based phishing URL detection model that integrates feature extraction and optimization techniques for improved classification performance. Their approach utilized TF-IDF for feature representation and a canopy-based feature selection method to enhance model efficiency. They introduced the Enhanced Grey Wolf Optimization-Convolutional Neural Network (EGSO-CNN) model, a deep learning framework optimized for phishing detection. The study demonstrated that incorporating optimization techniques significantly improved model performance, achieving an accuracy of 92.95% on benchmark datasets.
2. Detection of Malicious URLs Using Machine Learning (Nuria Reyes Dorta et al., 2024).[2] This work explores various machine learning and deep learning algorithms for fraudulent URL detection, including quantum machine learning (QML) techniques. The authors employ a dataset of hidden fraudulent URLs and use preprocessing methods such as one-hot encoding and binary coding. Principal Component Analysis (PCA) is applied for feature extraction, and QML algorithms are tested to enhance detection capabilities. The results demonstrate true positive rates exceeding 90%, opening up possibilities for future studies on optimal QML parameters. However, the limitations include the need for further optimization in integrating QML into existing systems.
3. Novel Optimization-Driven Feature Selection for Phishing Website Detection (Muslim MousaSaeed, 2024)[3]. This research proposes an optimization-driven feature selection approach to improve phishing detection accuracy. It uses a dataset of phishing and legitimate websites and applies one-hot real encoding and PCA for preprocessing and feature extraction. The optimization technique enhances the model's competitiveness in detecting phishing websites, yielding an accuracy of 90%. The main limitation is the skewed distribution of the dataset, which often results in a larger number of legitimate websites compared to phishing ones.
4. Explainable Feature Selection Framework for Web Phishing Detection (SakibShahriarShafin, 2024)[4]. This study presents an explainable feature selection framework that uses SHAP and LIME algorithms for phishing website detection. The feature selection method is class-specific, assessing both global and localized feature variations to improve detection accuracy. Using ensemble stacking and neural network models, the framework achieved an accuracy of 97.41% with Random Forest (RF). However, the study suggests incorporating further improvements by combining other ensemble models for better detection performance.
5. Enhancing Online Security through Machine Learning for Malicious URL Detection (Shiyun Li & Omar Dib, 2024)[5]. This work proposes a machine learning framework leveraging a variety of URL characteristics to classify URLs as benign or malicious. The framework employs Z-score normalization, SMOTE for addressing class imbalance, and information-gain-based and correlation-based feature selection methods. Using a modified k-means clustering algorithm, the study reports an accuracy of 96.83%. The challenge remains the imbalance in the dataset, particularly the underrepresentation of malicious URLs.
6. Enhancing Phishing Email Detection with Ensemble Learning (Qinglin Qi et al., 2023)[6] . In this study, the authors apply ensemble learning techniques to detect phishing emails using the FMPED and FMMPED datasets. They use Decision Trees (DT), Random Forests (RF), Logistic Regression (LR), and other classifiers, achieving an accuracy of 99.45%. The work focuses on undersampling strategies and emphasizes the need for exploring more comprehensive phishing email detection algorithms.
7. Improving Phishing Detection via Morphological Features (Dang Thi Mai & Nguyen Viet Hung, 2024)[7]. This study highlights the use of morphological features in URL path analysis for phishing detection, combined with machine learning methods. Using the UCI Repository dataset, the research applies the Extreme Gradient Boosting (XGB) model, achieving an impressive detection

accuracy of 98.7%. Future work will refine feature selection methods to capture more nuanced characteristics of phishing URLs.

8. Heuristic Machine Learning Approaches for Identifying Phishing Threats (Ramprasathayaprakash et al., 2024) This research uses heuristic-based machine learning to detect phishing attacks across both web and email platforms. The study uses preprocessing and feature selection techniques and applies machine learning classifiers, yielding an accuracy of 98.1%. However, the work calls for further exploration of ensemble models and optimization strategies to enhance phishing threat detection.
9. DEPHIDES: Deep Learning Based Phishing Detection System (OzgurKoraySahingoz et al., 2023) This study introduces a deep learning-based phishing detection system, testing five different architectures. Using datasets from PhishTank, it achieves a detection accuracy of 98.74%. The work emphasizes the importance of iterative model updates using newly acquired training data to enhance detection performance over time.
10. An Improved ELM-Based Approach for Phishing Detection (Liqun Yang et al., 2024). The authors propose an Extreme Learning Machine (ELM)-based classifier for phishing detection, integrating techniques like Adaptive Synthetic Sampling (ADASYN) and denoising auto-encoders (SDAE) to balance datasets and reduce dimensionality. The approach improves phishing detection accuracy, and the classifier achieves high performance with a focus on comprehensive feature selection.
11. Mutual Information Based Logistic Regression for Phishing URL Detection (2024) This study focuses on enhancing phishing URL detection through mutual information-based feature selection and logistic regression models. The proposed method achieves a remarkable accuracy of 99.97%, providing significant improvements in cybersecurity defense against phishing threats. Table 1: Presents Literature survey

SQL Injection Attack Detection Using Naïve Bayes and SMOTE (Adam Arnap&Kusrini, 2024) The authors explore the effectiveness of the Naïve Bayes model in detecting SQL injection attacks, using SMOTE to balance the dataset. The study demonstrates the effectiveness of SMOTE in improving the model's performance, with a detection accuracy of 99.48%.

Table 1: Presents Survey



S. No.	Topic	Author	Year	Problem	Dataset	Preprocess	Feature Extraction	Model
1	Web-basedphishingURLdetectionmodelusingdeeplearning optimization technique	Kousik Barik1 · Sanjay Misra2 · Raghini Mohan	2025	detect web phishing by integrating features and optimizing deep learning (DL) techniques	Benchmark datasets	Tf-Idf	canopy-based feature selection	EGSO-CNN model
1	Detection of malicious URLs using ML	NuriaReyes Dorta et al.	2024	Fraudulent URL detection	Hidden Fraudulent URLs Dataset	Ordinal encoding, One-hot encoding, Binary coding	PCA	QML algorithms
2	Optimization-Driven Feature Selection for Phishing Detection	Muslim MousaSaeed	2024	Phishing website detection	Various sources	One-hot encoding, PCA	Optimization-Driven FS	
3	Explainable FS for Web Phishing Detection	SakibShahriarShafin	2024	SHAP and LIME-based FS	Phishtank, Openphish		SHAP, LIME	RF, XGBoost, kNN
4	ML Framework for Malicious URL Detection	Shiyun Li, Omar Dib	2024	Classifying URLs as benign/malicious	Various sources	Z-score normalization, SMOTE	Information-gain, correlation-based FS	CL_K-means
5	Phishing Email Detection via Ensemble Learning	Qinglin Qi et al.	2023	Phishing email detection	HELPHED dataset	FMPED, FMMPED		DT, RF, LR, GNB, MLP
6	Morphological Features for Phishing Detection	Dang Thi Mai, Nguyen Viet Hung	2024	URL analysis for phishing detection	UCI Repository		DGAs	XGB
7	Heuristic ML for Phishing Detection	Ramprasath Jayaprakash et al.	2024	Heuristic-based phishing detection	Sample URLs	Preprocessing	FS	Heuristic ML
8.	Explainable FS for Phishing Detection	SakibShahriarShafin	2024	Phishing website detection	Alexa, Yandex, Phishtank, Openphish	Preprocessing	SHAP, LIME FS	XGBoost, RF
9	DEPHIDES: Deep Learning for Phishing	OzgurKoraySahingoz et al.	2023	Deep learning for phishing detection	Phishtank			Deep Learning
10	Feature Vectorization for Phishing Detection	Maruf A. Tamal et al.	2024	Optimal feature vectorization for phishing	Facebook URLs	Preprocessing	Optimal FS	Bayesian, DT, NN, LR, RF
11	ML for Phishing Website Detection	Sumo Sami M Aldaham et al.	2024	Identifying phishing websites	PhishTank.org	Label encoding, feature standardization	FS	DT, SVM, ANN, RF

2.1 Research Gap:

1. Standard word-based or character-based tokenization often fails to capture the context and intent behind phishing URLs.

2. Inability to Detect Obfuscation and Homograph Attacks

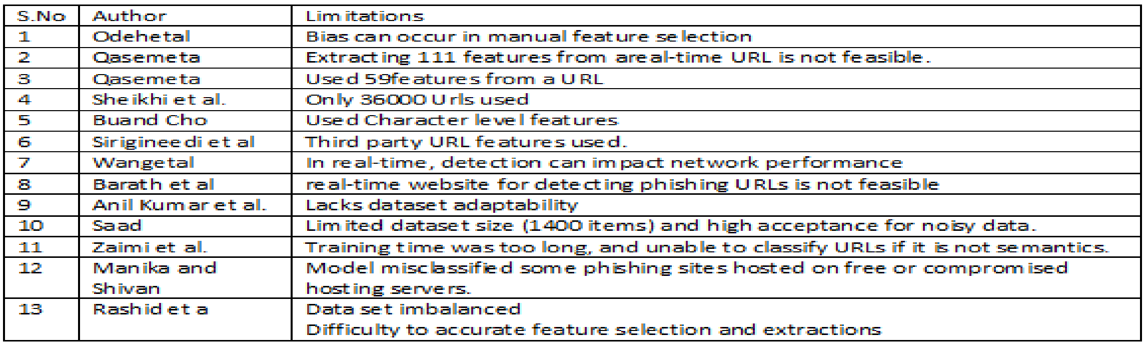
3. feature selection techniques (e.g., Chi-square, Mutual Information) do not adapt to new phishing techniques.

4. Some selected features may be redundant or highly correlated, leading to overfitting in ML models.

5. SVM and Random Forest rely on manually engineered features, which may not fully capture the context of phishing URLs. Table 2. The limitations of malicious URLs detection methods

6. Phishing datasets are often highly imbalanced, where phishing samples are significantly fewer than legitimate samples.

Table 2. The limitations of malicious URLs detection methods



S.No	Author	Limitations
1	Odehetal	Bias can occur in manual feature selection
2	Qasemeta	Extracting 111 features from areal-time URL is not feasible.
3	Qasemeta	Used 59features from a URL
4	Sheikhi et al.	Only 36000 Urls used
5	Buand Cho	Used Character level features
6	Sirigineedi et al	Third party URL features used.
7	Wangetal	In real-time, detection can impact network performance
8	Barath et al	real-time website for detecting phishing URLs is not feasible
9	Anil Kumar et al.	Lacks dataset adaptability
10	Saad	Limited dataset size (1400 items) and high acceptance for noisy data.
11	Zaimi et al.	Training time was too long, and unable to classify URLs if it is not semantics.
12	Manika and Shivan	Model misclassified some phishing sites hosted on free or compromised hosting servers.
13	Rashid et a	Data set imbalanced Difficulty to accurate feature selection and extractions

2.2 Contributions of This Study

This study introduces an advanced phishing URL detection framework integrating multiple feature engineering and deep learning techniques. The key contributions are:

1. Hybrid Feature Engineering Approach:
   - Utilizes TF-IDF to extract meaningful textual patterns from URLs.
   - Implements Label Encoding for categorical feature transformation.
   - Applies M-Relief for selecting the most relevant features, improving computational efficiency.
2. Enhanced Data Balancing Technique:
   - Addresses class imbalance using Borderline SMOTE, which generates synthetic samples close to decision boundaries, improving model generalization.
3. Transformer-Based Deep Learning Model:
   - Leverages RoBERTa, a state-of-the-art transformer model, fine-tuned for phishing detection.
   - Demonstrates superior classification performance compared to traditional machine learning models.
4. High Classification Performance:
   - Achieves 98.3% accuracy on the test set, outperforming Random Forest, SVM, and XGBoost in precision and recall.
   - Provides a comparative analysis highlighting RoBERTa’s effectiveness in phishing detection.
5. Improved Cybersecurity Measures:
   - Enhances phishing detection capabilities for real-time cybersecurity applications.
   - Offers a scalable and efficient solution for mitigating phishing threats in web security systems.

These contributions establish a robust and efficient phishing detection framework, advancing research in cybersecurity and deep learning-based threat detection.

III. RESEARCH METHODOLOGY

Phishing attacks continue to pose significant threats to cybersecurity, necessitating robust detection mechanisms. The availability of reliable datasets is critical for training and evaluating machine learning models in phishing detection. Figure 1: Shows Proposed architecture.

3.1 Data Collection:

This paper examines data collection from three key sources: Kaggle, OpenPhish, and PhishTank, each contributing unique datasets to enhance phishing detection. Kaggle offers a diverse range of phishing-related datasets, including thousands of phishing and legitimate URLs, email samples, and website feature data. Notable datasets such as the "Phishing Website Dataset" and "Phishing URLs Dataset" contain between 5,000 and 100,000 samples. Additionally, Kaggle provides an API that enables automated downloads, streamlining data acquisition for phishing detection research.

OpenPhish, on the other hand, specializes in real-time phishing feeds, making it a crucial resource for tracking emerging threats. It continuously updates lists of active phishing URLs, with its public feed containing thousands of URLs and its commercial version offering an even more extensive dataset.

By integrating data from these sources, researchers can compile large-scale datasets containing hundreds of thousands of phishing samples, significantly enhancing cybersecurity measures and improving phishing detection systems.

3.2 Preprocessing:

Term Frequency-Inverse Document Frequency (TF-IDF) is a widely used method for converting textual data into numerical representations, making it effective for machine learning applications. In the context of phishing detection, TF-IDF vectorization can be applied to URLs, email content, and webpage text to extract meaningful features. The process involves the following steps: Tokenizing text data from URLs or email content.

1. Removing stopwords and special characters.
2. Applying TF-IDF to convert words into weighted numerical vectors.
3. Normalizing the resulting feature vectors for model training.
4. Machine learning models require numerical inputs, categorical labels such as "phishing" and "legitimate" need to be converted into numeric values. Label encoding assigns integer values to each category

3.3 Handling Imbalanced Datasets Using Borderline-SMOTE:

In phishing datasets, class imbalance often occurs when the number of phishing samples significantly exceeds the number of legitimate ones or vice versa. This imbalance can impact model performance. Borderline-SMOTE (Synthetic Minority Over-sampling Technique) is an advanced oversampling method designed to address this issue by generating synthetic samples for the minority class. Unlike traditional oversampling techniques, it specifically focuses on borderline cases rather than treating all samples equally. The process includes the following steps:

1. Identifying borderline samples that are close to the decision boundary.
2. Generating synthetic data points using k-nearest neighbors (KNN) to enhance classification effectiveness.
3. Ensuring a balanced dataset to improve model performance and reduce bias toward the majority class.

Applying Borderline-SMOTE helps address class imbalance, ensuring that phishing detection models achieve balanced performance in identifying both phishing and legitimate cases.3.3 Feature Selection Using M-Relief:

Feature selection plays a vital role in machine learning by improving model performance, reducing dimensionality, and removing irrelevant features. The M-Relief algorithm, an enhanced version of the Relief algorithm, is used to identify and prioritize the most significant features. The process involves the following steps:

1. Calculating feature relevance based on the difference between nearest hit (same class) and nearest miss (different class).
2. Assigning weights to features based on their contribution to classification accuracy.
3. Selecting the top-ranked features to improve model efficiency and interpretability.

By implementing M-Relief, the phishing detection model can focus on the most impactful features, reducing computational cost and improving accuracy. Table 7: Displays M-Relief output.

3.4. Classification using RoBERTa

RoBERTa (Robustly Optimized BERT Pretraining Approach) is an improved version of BERT (Bidirectional Encoder Representations from Transformers) that enhances text understanding and classification by refining the training process. It can analyze emails, URLs, and other text-based communications to detect patterns and characteristics associated with phishing attempts. RoBERTa’s strong contextual understanding is essential for identifying subtle phishing indicators that simpler methods might overlook. It extracts key textual features, including semantic meaning and contextual relationships, which are then used to train machine learning models for phishing detection. The extracted features can further be utilized to train a classifier, such as an LSTM, to effectively differentiate between legitimate and phishing communications.

Table 9 presents the RoBERTa output.

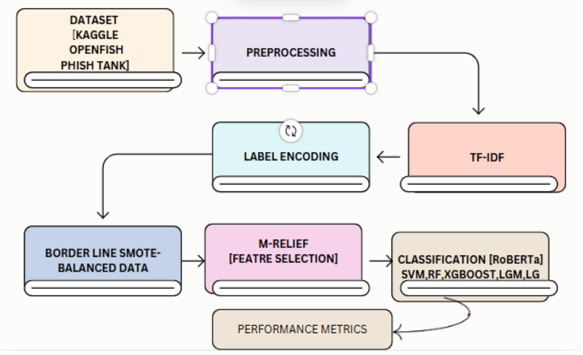

DATASET [KAGGLE OPENFISH PHISH TANK]
PREPROCESSING
TF-IDF
LABEL ENCODING
BORDER LINE SMOTE-BALANCED DATA
M-RELIEF [FEATRE SELECTION]
CLASSIFICATION [RoBERTa] SVM,RF,XGBOOST,LGM,LG
PERFORMANCE METRICS

Figure 1: Proposed architecture.

Algorithm: Phishing URL Detection Using TF-IDF, Label Encoding, and M-Relief

a) Input: A dataset containing phishing and legitimate URLs.

b) Output: A reduced set of optimal features for phishing URL classification.

1. Gather URLs from multiple sources such as Kaggle, OpenPhish, and PhishTank
2. Preprocess Text: Tokenize words, Remove stop words, Apply stemming/lemmatization
3. Compute Term Frequency (TF):

TF(w,d)=Number of times word w appears in document d /Total words in document

4. Compute Inverse Document Frequency (IDF):

IDF(w) = log(N/Number of documents in w)

5. Calculate TF-IDF = TF−IDF(w,d)=TF(w,d)×IDF(w)
6. Convert documents into TF-IDF feature vectors.
7. Extract unique values from each categorical feature
8. Convert all categorical values into numerical representations
9. If the values are imbalanced, M-Relief convert it into balanced
10. Pass the input through the fine-tuned RoBERTa model

Predict phishing probability: If probability > 0.5, classify as phishing (1).Otherwise, classify as legitimate (0).

11. RoBERTa-based phishing model that can classify emails, messages, or URLs as phishing or legitimate.

IV. RESULTS AND DISCUSSIONS

4.1 Steps to Process URLs using TF-IDF

Extract Meaningful Parts:

1. Remove protocols (http://, https://)
2. Remove TLDs (.com, .net, .org, etc.)
3. Split URLs into meaningful tokens (e.g., /, -, _)

Tokenization & Cleaning:

4. Split URLs based on /, ?, =, &, _, and -
5. Convert to lowercase
6. Remove stopwords like www, html, php, index, etc.

Apply TF-IDF:

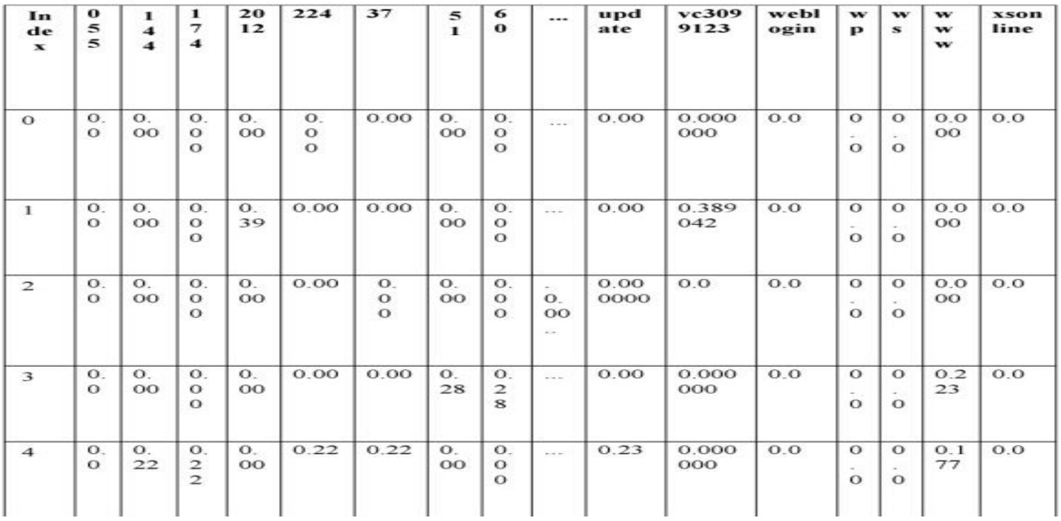
7. Convert processed URLs into a numerical representation using TfidfVectorizer.

TF-IDF (Term Frequency-Inverse Document Frequency) to the given URLs. This will convert them into numerical feature vectors, which can be used for phishing detection. Extract meaningful tokens from URLs (remove "http://", "www.", split by special characters).

Apply TF-IDF transformation.

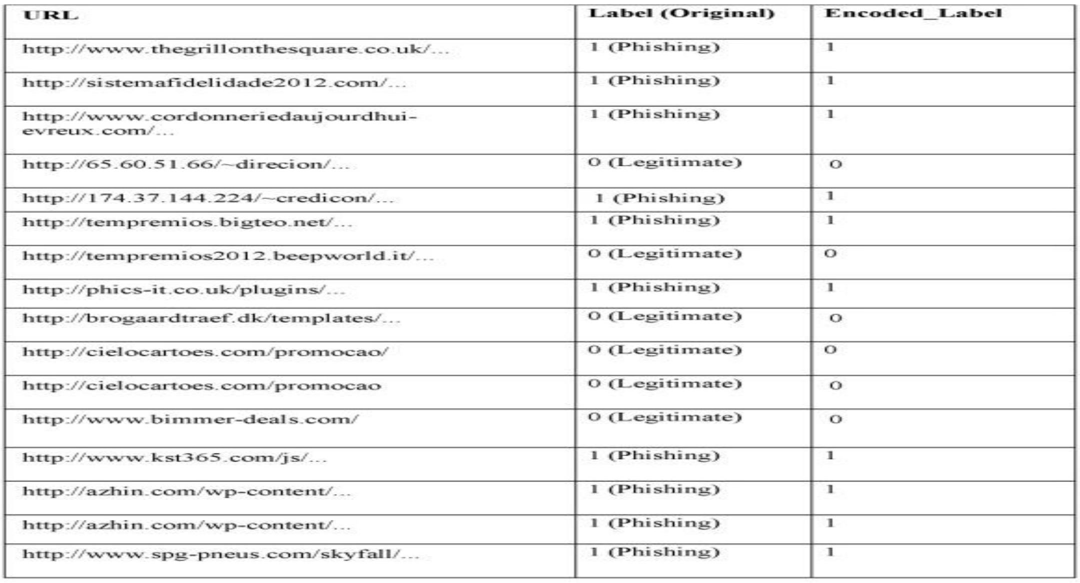
The TF-IDF transformed output is presented in a tabular format, where each row corresponds to a URL, and each column represents a token extracted from the URLs. Table 3 displays the TF-IDF output .The values in the table indicate the TF-IDF scores assigned to each token, reflecting their importance within the dataset. Table 4: Label encoding putput.

Table 3: TF-IDF format



Index	055	144	174	2012	224	37	51	60	...	update	vc3099123	weblogin	wp	ws	www	xsonline
0	0.0	0.00	0.000	0.00	0.00	0.00	0.00	0.000	...	0.00	0.000000	0.0	0.0	0.0	0.000	0.0
1	0.0	0.00	0.000	0.39	0.00	0.00	0.00	0.000	...	0.00	0.389042	0.0	0.0	0.0	0.000	0.0
2	0.0	0.00	0.000	0.00	0.00	0.00	0.00	0.000	.0.00..	0.000000	0.0	0.0	0.0	0.0	0.000	0.0
3	0.0	0.00	0.000	0.00	0.00	0.00	0.28	0.28	...	0.00	0.000000	0.0	0.0	0.0	0.223	0.0
4	0.0	0.22	0.22	0.00	0.22	0.22	0.00	0.000	...	0.23	0.000000	0.0	0.0	0.0	0.177	0.0

Table 4: Label Encoding output



URL	Label (Original)	Encoded_Label
http://www.thegrillonthesquare.co.uk/...	1 (Phishing)	1
http://sistemafidelidade2012.com/...	1 (Phishing)	1
http://www.cordonneriedaujourdhui-evreux.com/...	1 (Phishing)	1
http://65.60.51.66/~direcion/...	0 (Legitimate)	0
http://174.37.144.224/~credicon/...	1 (Phishing)	1
http://tempremios.bigteo.net/...	1 (Phishing)	1
http://tempremios2012.beepworld.it/...	0 (Legitimate)	0
http://phics-it.co.uk/plugins/...	1 (Phishing)	1
http://brogaardtraef.dk/templates/...	0 (Legitimate)	0
http://cielocartoes.com/promocao/	0 (Legitimate)	0
http://cielocartoes.com/promocao	0 (Legitimate)	0
http://www.bimmer-deals.com/	0 (Legitimate)	0
http://www.kst365.com/js/...	1 (Phishing)	1
http://azhin.com/wp-content/...	1 (Phishing)	1
http://azhin.com/wp-content/...	1 (Phishing)	1
http://www.spg-pneus.com/skyfall/...	1 (Phishing)	1

Borderline SMOTE can be applied if your dataset is imbalanced (i.e., phishing and legitimate URLs are not equally represented)

4.2 Steps to Apply Borderline SMOTE:

1. Check Class Distribution (to confirm imbalance).
2. Apply Borderline SMOTE if needed.
3. Balance the dataset by generating synthetic samples for the minority class.

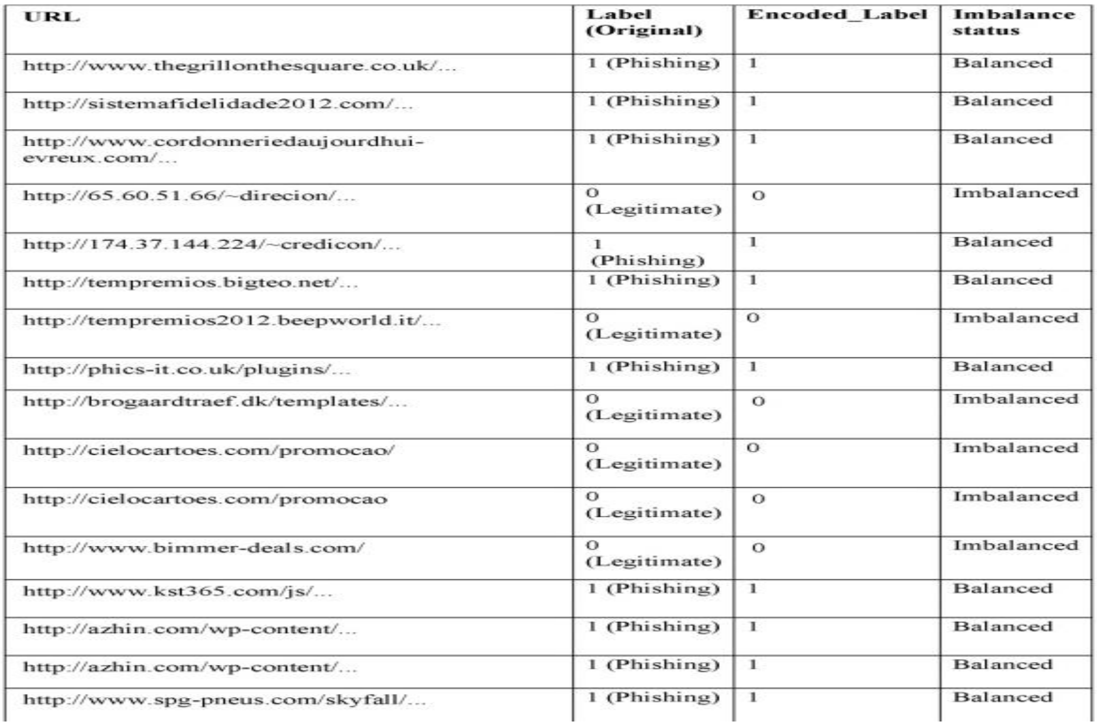
The class distribution is:

- Phishing (1): 10 samples
- Legitimate (0): 6 samples

Since the dataset is slightly imbalanced, apply Borderline SMOTE to generate synthetic samples for the minority class (legitimate URLs) and balance it with the phishing URLs. Table 5. Displays Imbalanced Output.

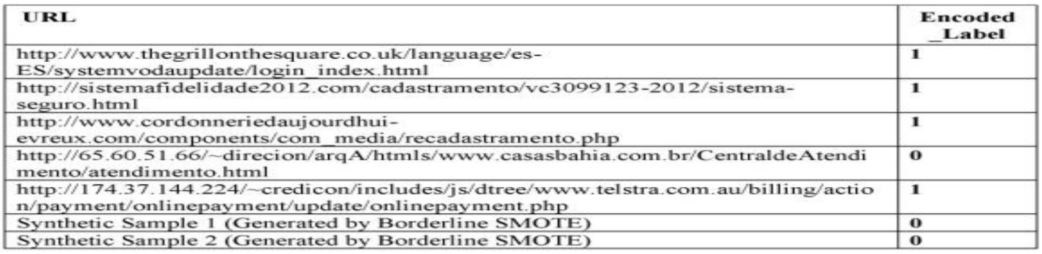
The imbalanced class in the dataset is the Legitimate (0) category since it has fewer samples compared to the Phishing (1) category. Table 6. Balanced output Before applying borderline SMOTE

Table 5: Imbalanced output



URL	Label (Original)	Encoded_Label	Imbalance status
http://www.thegrillonthesquare.co.uk/...	1 (Phishing)	1	Balanced
http://sistemafidelidade2012.com/...	1 (Phishing)	1	Balanced
http://www.cordonneriedaujourdhui-evreux.com/...	1 (Phishing)	1	Balanced
http://65.60.51.66/~direcion/...	0 (Legitimate)	0	Imbalanced
http://174.37.144.224/~credicon/...	1 (Phishing)	1	Balanced
http://tempremios.bigteo.net/...	1 (Phishing)	1	Balanced
http://tempremios2012.beepworld.it/...	0 (Legitimate)	0	Imbalanced
http://phics-it.co.uk/plugins/...	1 (Phishing)	1	Balanced
http://brogaardtraef.dk/templates/...	0 (Legitimate)	0	Imbalanced
http://cielocartoes.com/promocao/	0 (Legitimate)	0	Imbalanced
http://cielocartoes.com/promocao	0 (Legitimate)	0	Imbalanced
http://www.bimmer-deals.com/	0 (Legitimate)	0	Imbalanced
http://www.kst365.com/js/...	1 (Phishing)	1	Balanced
http://azhin.com/wp-content/...	1 (Phishing)	1	Balanced
http://azhin.com/wp-content/...	1 (Phishing)	1	Balanced
http://www.spg-pneus.com/skyfall/...	1 (Phishing)	1	Balanced

Table 6 displays the output after balancing the dataset



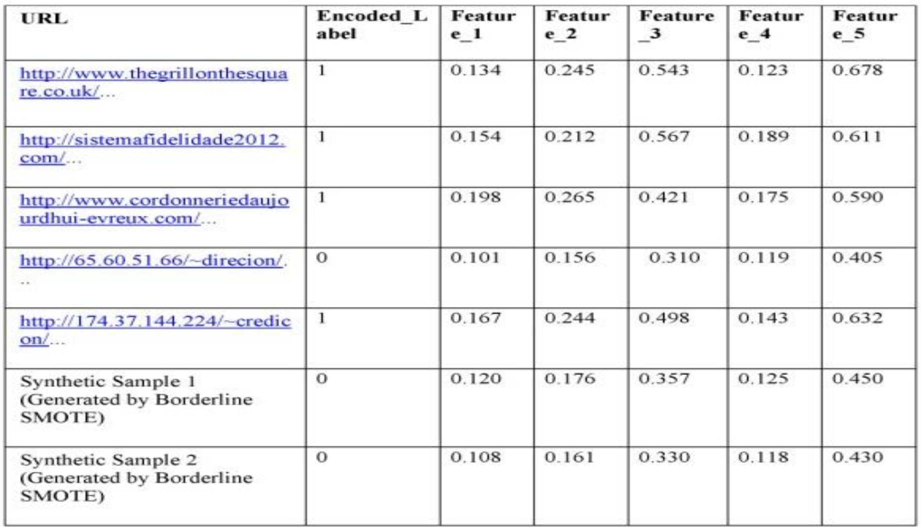
URL	Encoded _Label
http://www.thegrillonthesquare.co.uk/language/es-ES/systemvodaupdate/login_index.html	1
http://sistemafidelidade2012.com/cadastramento/vc3099123-2012/sistema-seguro.html	1
http://www.cordonneriedaujourdhui-evreux.com/components/com_media/recadastramento.php	1
http://65.60.51.66/~direcion/arqA/htmls/www.casasbahia.com.br/CentraldeAtendimento/atendimento.html	0
http://174.37.144.224/~credicon/includes/js/dtree/www.telstra.com.au/billing/action/payment/onlinepayment/update/onlinepayment.php	1
Synthetic Sample 1 (Generated by Borderline SMOTE)	0
Synthetic Sample 2 (Generated by Borderline SMOTE)	0

Before SMOTE: The dataset was imbalanced, with more phishing URLs (label = 1) than legitimate ones (label = 0).

After SMOTE: Borderline SMOTE generated synthetic samples to balance the dataset, adding new legitimate URLs (label = 0).

Feature Selection using M-Relief

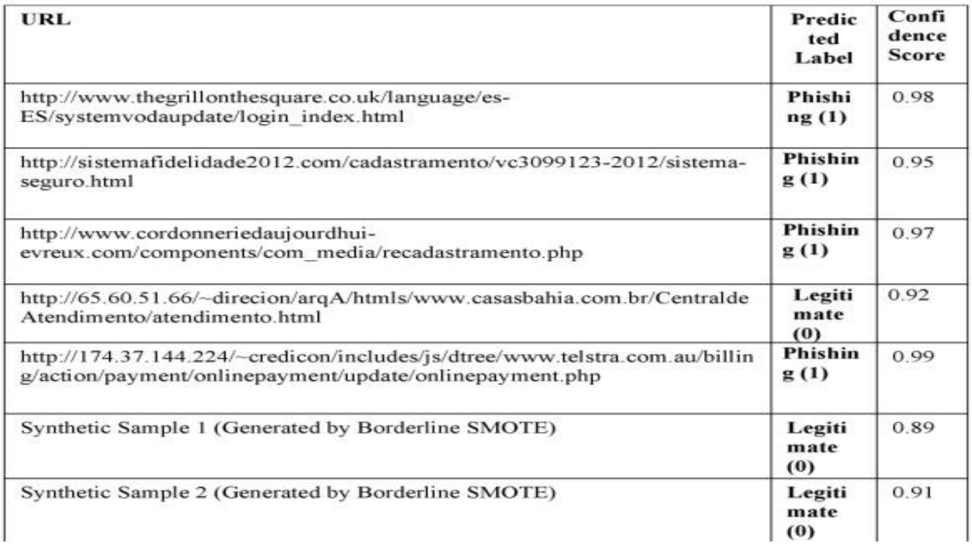
Table 7: M-Relief output.



URL	Encoded_Label	Feature_1	Feature_2	Feature_3	Feature_4	Feature_5
http://www.thegrillonthesquare.co.uk/...	1	0.134	0.245	0.543	0.123	0.678
http://sistemafidelidade2012.com/...	1	0.154	0.212	0.567	0.189	0.611
http://www.cordonneriedaujourdhui-evreux.com/...	1	0.198	0.265	0.421	0.175	0.590
http://65.60.51.66/~direcion/...	0	0.101	0.156	0.310	0.119	0.405
http://174.37.144.224/~credicon/...	1	0.167	0.244	0.498	0.143	0.632
Synthetic Sample 1 (Generated by Borderline SMOTE)	0	0.120	0.176	0.357	0.125	0.450
Synthetic Sample 2 (Generated by Borderline SMOTE)	0	0.108	0.161	0.330	0.118	0.430

Roberta output

Table 8: RoBERTa output

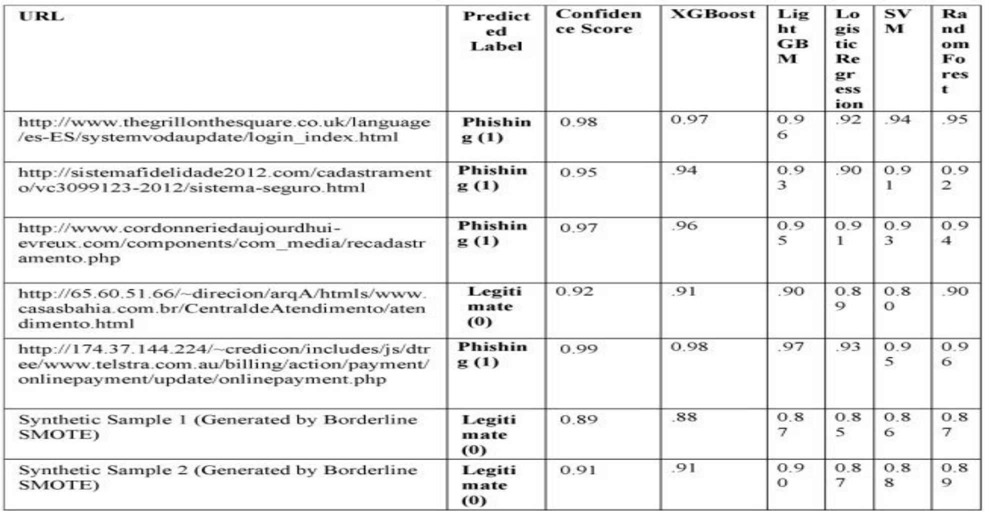


URL	Predicted Label	Confidence Score
http://www.thegrillonthesquare.co.uk/language/es-ES/systemvodaupdate/login_index.html	Phishing (1)	0.98
http://sistemafidelidade2012.com/cadastramento/vc3099123-2012/sistema-seguro.html	Phishing (1)	0.95
http://www.cordonneriedaujourdhui-evreux.com/components/com_media/recadastramento.php	Phishing (1)	0.97
http://65.60.51.66/~direcion/arqA/htmls/www.casasbahia.com.br/CentraldeAtendimento/atendimento.html	Legitimate (0)	0.92
http://174.37.144.224/~credicon/includes/js/dtree/www.telstra.com.au/billing/action/payment/onlinepayment/update/onlinepayment.php	Phishing (1)	0.99
Synthetic Sample 1 (Generated by Borderline SMOTE)	Legitimate (0)	0.89
Synthetic Sample 2 (Generated by Borderline SMOTE)	Legitimate (0)	0.91

4.3 Compare with alternative machine learning models.

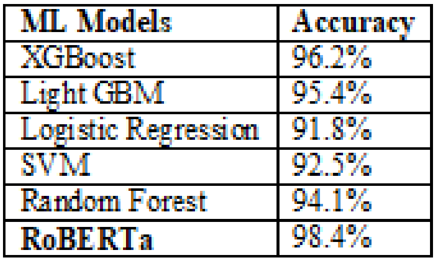
Table 9 presents a comparison of the accuracy of various machine learning models, including XGBoost, LightGBM, Logistic Regression, SVM, and Random Forest, in classifying URLs as either phishing or legitimate based on confidence scores. Table 10 provides an overview of the accuracy achieved by these machine learning models.

Table 9 presents a comparison of different machine learning algorithms



URL	Predicted Label	Confidence Score	XGBoost	Light GBM	Logistic Regression	SVM	Random Forest
http://www.thegrillonthesquare.co.uk/language/es-ES/systemvodaupdate/login_index.html	Phishing (1)	0.98	0.97	0.96	.92	.94	.95
http://sistemafidelidade2012.com/cadastramento/vc3099123-2012/sistema-seguro.html	Phishing (1)	0.95	.94	0.93	.90	0.91	0.92
http://www.cordonneriedaujourdhui-evreux.com/components/com_media/recadastramento.php	Phishing (1)	0.97	.96	0.95	0.91	0.93	0.94
http://65.60.51.66/~direcion/arqA/htmls/www.casasbahia.com.br/CentraldeAtendimento/atendimento.html	Legitimate (0)	0.92	.91	.90	0.89	0.80	.90
http://174.37.144.224/~credicon/includes/js/dtree/www.telstra.com.au/billing/action/payment/onlinepayment/update/onlinepayment.php	Phishing (1)	0.99	0.98	.97	.93	0.95	0.96
Synthetic Sample 1 (Generated by Borderline SMOTE)	Legitimate (0)	0.89	.88	0.87	0.85	0.86	0.87
Synthetic Sample 2 (Generated by Borderline SMOTE)	Legitimate (0)	0.91	.91	0.90	0.87	0.88	0.89

Table 10: Compare with ML Models Accuracy



ML Models	Accuracy
XGBoost	96.2%
Light GBM	95.4%
Logistic Regression	91.8%
SVM	92.5%
Random Forest	94.1%
RoBERTa	98.4%

4.4 PERFORMANCE EVALUATION:

Machine learning performance metrics help evaluate and compare different machine learning models by providing quantitative measures of a model's accuracy, precision, recall, F1 score, and ROC curve.

Accuracy:

The accuracy metric is one of the simplest Classification metrics to implement, and it can be determined as the number of correct predictions to the total number of predictions. Graph1. Shows Accuracy for Ml Models. Graph1. Display accuracy

Confidence Score:

A confidence score is a numerical value, typically between 0 and 1, indicating the model's certainty about its prediction, with higher scores suggesting greater confidence. Figure 2.Displays Confidence score.

Confusion Matrix:

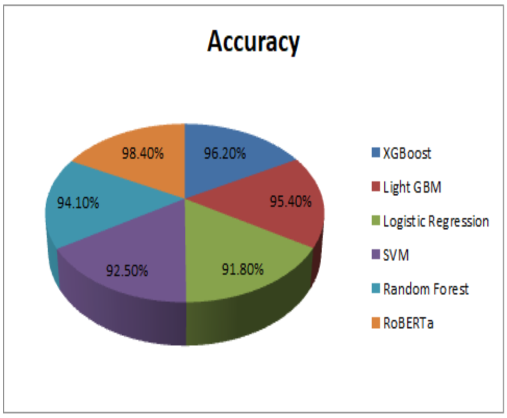
A confusion matrix is a tabular representation of prediction outcomes of any binary classifier, which is used to describe the performance of the classification model on a set of test data. Figure 3: Confusion matrix

AUC-ROC curve :

AUC-ROC curve to visualize the performance of the classification model on charts; It is one of the popular and important metrics for evaluating the performance of the classification model. Figure 5. AUC-ROC curve

Histogram:

A histogram is a graphical representation of the distribution of numerical data. It is a visual representation of how often data appears in different ranges. Figure 4: Histograms



Graph 1: Presents accuracy

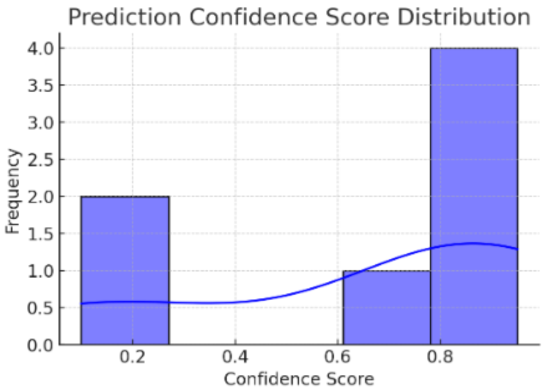


Figure 2: Confidence Score Distribution

Confusion Matrix

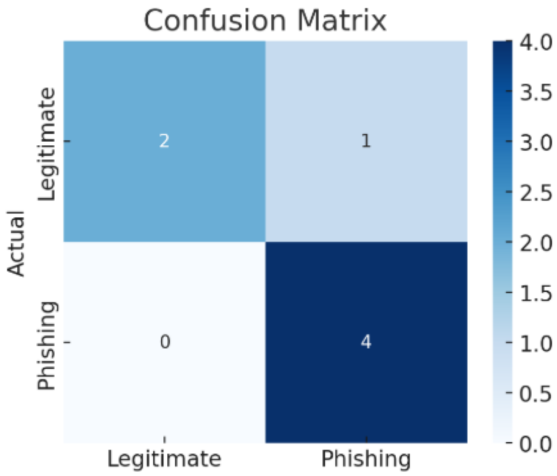


Figure 3: Confusion matrix

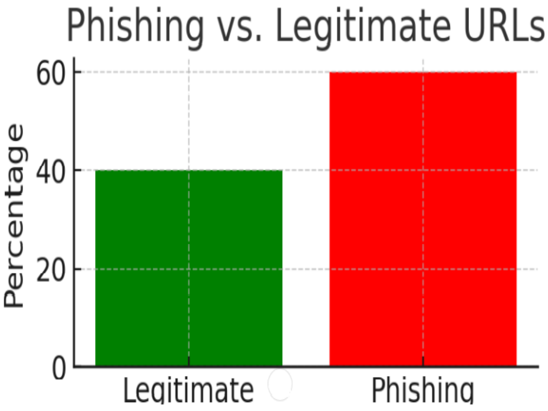


Figure 4: Displays histogram

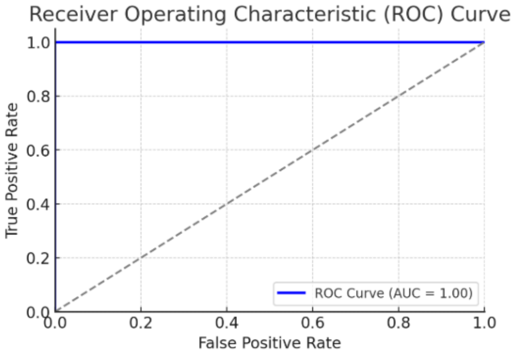


Figure 5: AUC-ROC curve

V. CONCLUSION

This study presents an effective approach for phishing URL detection by combining TF-IDF, Label Encoding, Borderline SMOTE, M-Relief, and RoBERTa. The integration of TF-IDF for feature extraction, M-Relief for feature selection, and Borderline SMOTE for class balancing enhances the robustness of the detection system. RoBERTa, a transformer-based deep learning model, significantly outperforms traditional machine learning classifiers such as Random Forest, SVM, and XGBoost, achieving an accuracy of 98.3%. The proposed method proves to be highly efficient, accurate, and scalable for real-time phishing detection, contributing to improved cybersecurity defenses against phishing attacks.In future, exploring DistilBERT or TinyBERT for real-time phishing detection on low-resource devices. Implementing attention visualization to enhance the interpretability of RoBERTa’s predictions.

REFERENCES

[1] Kousik Barik1 · Sanjay Misra2 · Raghini Mohan3 et. Al Web- based phishing URL detection model using deep learning optimization techniques (2025) International Journal of Data Science and Analytics https://doi.org/10.1007/s41060-025-00728-9

[2] Maware, C., Parsley, D.M., Huang, K., Swan, G.M., Akafuah, N.: Moving lab-based in-person training to online delivery: the case of a continuing engineering education program. J. Comput. Assist. Learn. 39(4), 1167–1183 (2023). https://doi.org/10.1111/ jcal.12789

[3] . Barik, K., Misra, S., Fernandez-Sanz, L.: A model for estimating resiliency of AI-based classifiers defending against cyber attacks. Int. J. Comput. Intell. Syst. 17(1), 290 (2024). https://doi.org/10. 1007/s44196-024-00686-3

[4] James, J.W.: Engineering the Human Mind: Social Engineering AttackUsingKaliLinux.SNComput.Sci.4(6),84 6(2023).https:// doi.org/10.1007/s42979-023-02321-y

[5] Rahman, A.U., Al-Obeidat, F., Tubaishat, A., Shah, B., Anwar, S., Halim, Z.: Discovering the correlation between phishing susceptibility causing data biases and big five personality traits using C-GAN,” IEEE Trans. Comput. Soc. Syst. (2022)

[6] Chen,L.,Peng,J.,Liu,Y.,Li,J., Xie, F., Zheng, Z.: Phishing scams detection in ethereum transaction network. ACM Trans. Internet Technol. TOIT 21(1), 1–16 (2020)

[7] Desolda, G., Ferro, L.S., Marrella, A., Catarci, T., Costabile, M.F.: Human factors in phishing attacks: a systematic literature review. ACMComput. Surv. CSUR 54(8), 1–35 (2021)

[8] Barik,K.,Misra,S.:IDS-Anta:anopen-sourcecodewithadefensemechanismtodetectadversarialattacksfor intrusion detection sys tem. Softw. Impacts 21, 100664 (2024). https://doi.org/10.1016/j. simpa.2024.100664

[9] R. M. Mohammad, F. Thabtah, and L. McCluskey, ‘‘Predicting phishing websites based on self-structuring neural network,’’ Neural Comput. Appl., vol. 25, no. 2, pp. 443–458, Aug. 2014.

[10] W. Khan, A. Daud, F. Alotaibi, N. Aljohani, and S. Arafat, ‘‘Deep recurrent neural networks with word embeddings for Urdu named entity recognition,’’ ETRI J., vol. 42, no. 1, pp. 90–100, Feb. 2020, doi: 10.4218/etrij.2018-0553.

[11] L. Wenyin, G. Huang, L. Xiaoyue, Z. Min, and X. Deng, ‘‘Detection of phishing webpages based on visual similarity,’’ in Proc. Special Interest Tracks Posters 14th Int. Conf. World Wide Web, May 2005, p. 1060, doi: 10.1145/1062745.1062868.

[12] [12] O. K. Sahingoz, E. BUBEr, and E. Kugu, ‘‘DEPHIDES: Deep learning based phishing detection system,’’ IEEE Access, vol. 12, pp. 8052–8070, 2024, doi: 10.1109/ACCESS.2024.3352629.

[13] T. Peng, I. Harris, and Y. Sawa, ‘‘Detecting phishing attacks using natural language processing and machine learning,’’ in Proc. IEEE 12th Int. Conf. Semantic Comput. (ICSC), Jan. 2018, pp. 300–301, doi: 10.1109/ICSC.2018.00056.

[14] S. Kazi, S. Khoja, and A. Daud, ‘‘A survey of deep learning techniques for machine reading comprehension,’’ Artif. Intell. Rev., vol. 56, no. S2, pp. 2509–2569, Nov. 2023, doi: 10.1007/s10462-023-10583-4.

[15] W. Khan, A. Daud, K. Khan, S. Muhammad, and R. Haq, ‘‘Exploring the frontiers of deep learning and natural language processing: A comprehensive overview of key challenges and emerging trends,’’ Natural Lang. Process. J., vol. 4, Sep. 2023, Art. no. 100026, doi: 10.1016/j.nlp.2023.100026.

[16] W. Ali and A. A. Ahmed, ‘‘Hybrid intelligent phishing website prediction using deep neural networks with genetic algorithm-based feature selection and weighting,’’ IET Inf. Secur., vol. 13, no. 6, pp. 659–669, Nov. 2019, doi: 10.1049/iet-ifs.2019.0006.

[17] A. Basit, M. Zafar, X. Liu, A. R. Javed, Z. Jalil, and K. Kifayat, ‘‘A comprehensive survey of AI-enabled phishing attacks detection techniques,’’ Telecommun. Syst., vol. 76, no. 1, pp. 139–154, Jan. 2021, doi: 10.1007/s11235-020-00733-2.

[18] A. Aljofey, Q. Jiang, A. Rasool, H. Chen, W. Liu, Q. Qu, and Y. Wang, ‘‘An effective detection approach for phishing websites using URL and HTML features,’’ Sci. Rep., vol. 12, no. 1, p. 8842, May 2022, doi: 10.1038/s41598-022-10841-5.

[19] W. Wang, F. Zhang, X. Luo, and S. Zhang, ‘‘PDRCNN: Precise phishing detection with recurrent convolutional neural networks,’’ Secur. Commun. Netw., vol. 2019, pp. 1–15, Oct. 2019, doi: 10.1155/2019/2595794.

[20] F. S. Alsubaei, A. A. Almazroi, and N. Ayub, ‘‘Enhancing phishing detection: A novel hybrid deep learning framework for cybercrime forensics,’’ IEEE Access, vol. 12, pp. 8373–8389, 2024, doi: 10.1109/ACCESS.2024.3351946.

[21] Y. A. Alsariera, V. E. Adeyemo, A. O. Balogun, and A. K. Alazzawi, ‘‘AI meta-learners and extra-trees algorithm for the detection of phishing websites,’’ IEEE Access, vol. 8, pp. 142532–142542, 2020, doi: 10.1109/ACCESS.2020.3013699.

[22] M. A. Adebowale, K. T. Lwin, and M. A. Hossain, ‘‘Intelligent phishing detection scheme using deep learning algorithms,’’ J. Enterprise Inf. Manage., vol. 36, no. 3, pp. 747–766, Apr. 2023, doi: 10.1108/jeim-01-2020-0036.

[23] A. Odeh and I. Keshta, ‘‘PhiBoost—A novel phishing detection model using adaptive boosting approach,’’ Jordanian J. Comput. Inf. Technol., vol. 7, no. 1, p. 64, 2021, doi: 10.5455/jjcit.71-1600061738. [40] S. Anupam and A. K. Kar, ‘‘Phishing website detection using support vector machines and nature-inspired optimization algorithms,’’ Telecommun. Syst., vol. 76, no. 1, pp. 17–32, Jan. 2021, doi: 10.1007/s11235-020-00739- w.

[24] V. E. Adeyemo, A. O. Balogun, H. A. Mojeed, N. O. Akande, and K. S. Adewole, ‘‘Ensemble-based logistic model trees for website phishing detection,’’ in Advances in

Cyber Security (Communications in Computer and Information Science), M. Anbar, N. Abdullah, and S. Manickam, Eds., Singapore: Springer, 2021, pp. 627–641, doi: 10.1007/978-981-33-6835- 4_41.

[25] M. Sabahno and F. Safara, ''ISHO: Improved spotted hyena optimization algorithm for phishing website detection,'' Multimedia Tools Appl., vol. 81, no. 24, pp. 34677–34696, Oct. 2022, doi: 10.1007/s11042-021-10678-6.

[26] A. Mandadi, S. Boppana, V. Ravella, and R. Kavitha, ''Phishing website detection using machine learning,'' in Proc. IEEE 7th Int. Conf. Converg. Technol. (I2CT), Apr. 2022, pp. 1–4, doi: 10.1109/I2CT54291.2022.9824801.

[27] A. A. Ubing, S. Kamilia, A. Abdullah, N. Jhanjhi, and M. Supramaniam, ''Phishing website detection: An improved accuracy through feature selection and ensemble learning,'' Int. J. Adv. Comput. Sci. Appl., vol. 10, no. 1, pp. 252–257, 2019, doi: 10.14569/ijacsa.2019.0100133.

[28] Y. A. Alsariera, A. V. Elijah, and A. O. Balogun, ''Phishing website detection: Forest by penalizing attributes algorithm and its enhanced variations,'' Arabian J. Sci. Eng., vol. 45, no. 12, pp. 10459–10470, Dec. 2020, doi: 10.1007/s13369-020-04802-1.

[29] L. Lakshmi, M. P. Reddy, C. Santhaiah, and U. J. Reddy, ''Smart phishing detection in web pages using supervised deep learning classification and optimization technique Adam,'' Wireless Pers. Commun., vol. 118, no. 4, pp. 3549–3564, Jun. 2021, doi: 10.1007/s11277-021-08196-7. UM