AI-POWERED PHISHING DETECTION AND PREVENTION SYSTEM FOR LARGE-SCALE EMAIL DATA SECURITY

Anamika Gurung, Dr. Heena Kousar, Gurunath Reddy N A, Jitendra S S, Nandan G M

Undergraduate Student, Professor, Undergraduate Student, Undergraduate Student, Undergraduate Student
Dept. of Computer Science and Engineering,
East Point College of Engineering and Technology Bengaluru, India

Abstract : Phishing is one of the most common cybersecurity threats. It tricks users through fake emails and harmful URLs. Traditional filters and blacklist systems often miss new phishing links and advanced email content. This paper introduces an AI- based phishing detection and prevention system designed for large email environments. The framework has two parts: a URL classification model that uses Logistic Regression and TF-IDF for quick real-time detection and a transformer email content classifier powered by BERT for understanding meaning. The system runs on a FastAPI backend and includes a Chrome extension that works with Gmail. This extension provides users with instant visual risk indicators in their inbox. Tests on large public phishing datasets show strong results. The URL classification model achieves high accuracy, and the transformer model effectively spots semantic phishing signals. Real-world testing confirms the system's ability to prevent credential theft by identifying suspicious emails before users interact with them. The findings highlight the value of hybrid AI approaches in improving email security.

IndexTerms **- Phishing detection, Machine learning, Logistic Regression, BERT, Email security, URL analysis, Deep learning, Chrome extension, FastAPI, Transformer models.**

I. INTRODUCTION

Phishing is among the most prevalent threats in cyberspace nowadays. Perpetrators never cease to enhance their techniques through mimic sites, relying on trust, and evading typical security mechanisms. As reported by the Anti-Phishing Working Group (APWG), the rate at which phishing has grown, along with the dynamic nature of phishing, has been supported by the industry [10]. Phishing attacks worldwide have become more prominent, and millions of attacks have been recorded annually. That is why there is a great need to have efficient detection systems. The issue is further complicated by the rising use of trustworthy-indicators such as HTTPS and the padlock symbol, making it even more challenging for users to differentiate between legitimate and phishing websites [1]. Traditional phishing detection techniques such as blacklist filters, rule-based approaches, and fixed signatures are not effective when dealing with newly crafted malicious URLs and phishing through immediate emailing. Studies have discovered that approaches utilizing set lists tend to overlook novel phishing attacks. The attackers usually employ automated tools to form these domains, Domain Generation Algorithms (DGA), as described in [4]. It reflects how there is a gap that requires intelligent and adaptive learning technologies to fill. However, with recent advances in machine learning (ML)/deep learning (DL), phishing detection techniques have received a substantial boost. Various researchers have investigated ML classification techniques like Decision Tree, Logistic Regression, SVM, or KNN for URL classification. This variety of models displays different degrees of accuracy [2]. Later studies have incorporated the use of hybrid deep learning architectural models that integrate CNNs, LSTMs, GRUs, and ResNeXt neural networks [4][6]. These models can reach a level of accuracy of about 98-99% in a lab-controlled environment for the detection of phishing URLs. Further, transformer models like DistilBERT have given very successful results in assessing emails in real- time for phishing [8] with more than 95% accuracy in real- time testing. Nevertheless, in spite of such developments, large-scale email environments, such as business mail servers or cloud based solutions, are inherently challenging settings. High email volume, cloaked URLs, variant content, and com- pound threats demand solutions that are able to (i) handle live data streams, (ii) analyze email body, header, and URLs embedded in them, and (iii) predict in real-time while having a low rate of false positives. Additionally, recent studies have identified problems such as class imbalance, underfitting, overfitting, and a weak ability to generalize to novel attack patterns revealed by current research in this area as well [4]. For the above challenges, the proposed research work extends an AI-powered phishing protection and prevention system suitable for large-scale email services. The system integrates URL classification using deep learning algorithms, sequential analysis performed using LSTMs, and the intelligent alerting system. It has the ability to identify emails as phishing emails, suspicious emails, and trusted emails. This solution is set to increase enterprise security with real-time phishing detection capabilities to counter changing phishing attacks.

II. RELATED WORK

Detection of phishing has become a serious concern in the recent times due to the increased cybercrime and Social engineering attacks. Various approaches have been proposed using machine learning and deep learning to enhance the efficacy of phishing identification systems. Traditional machine learning URL classifiers based on features like lexical structure, domain information, and statistical patterns show good results. However, the challenge of generalizing still remains open when dealing with newly generated malicious URLs [3]. Recent research shows that attackers are increasing the use of domain masking, dynamic redirections, and HTTPS spoofing to bypass static and heuristic filters. This creates a greater need for more flexible detection models [5]. Most notably, deep learning methods have significantly enhanced the performance of phishing detection. Hybrid models that incorporate the power of CNN, RNN, LSTM, and GRU networks have also been effective in feature representation for URL and website analysis. These models boost the efficacy of detection because they capture sequences and semantic URL characteristics. However, they are complex; therefore, they cannot be applicable in a real-time setting. Some research has applied the transformer-based language model to the recognition of the email phishing phenomenon. Past research also focused on anomaly-based techniques and behavior-driven approaches for phishing email identification with the aim of finding anomalies against normal communication [11]. Distil-BERT & BERT Classification performs better than conventional NLP methods by understanding intent, context, & "language tricks" used in phishing emails [9]. These models are able to perform the classification on the email content accurately, even if the phishing material in the email is discreet and then professionally written in a social engineering manner. However, these individual email-text classifiers face difficulties when faced with threatening emails embedded in URLs, attachments, and hyperlinked images. Recent research has also emphasized the significance of large and balanced datasets in the construction of trustworthy phishing classifiers. Unbalanced datasets tend to generate biased classifiers that fail when confronted with the minority instances of phishing emails [1]. Hybrid detection systems that merge analysis of URLs, content analysis, and analysis of metadata have been proposed to enhance reliability. Even so, many of these ideas have not yet received sufficient testing on controlled datasets, other than in experimental settings [8]. Even with these studies, some areas are still not covered in creating a system able to identify phishing in a real-time manner in email services. The current state-of-the-art systems lack frameworks for deployment, browser support, and a real-time classification chain. The existence of systems like PhishStorm, a real-time phishing system, shows successful real-time analytics in a large setting; however, they lack user integration [12]. For overcoming these challenges, this research work presents a novel approach that uses a hybrid phishing detector algorithm which is a combination of URL classification using machine learning and transformers. It is incorporated into a Chrome Extension and an FastAPI Service for scalable and real- time email functionality. Large-scale studies on phishing page classification highlight the need for detection methods that can easily handle millions of samples in real-world situations [15].

III. SYSTEM ARCHITECTURE

The proposed phishing detection system has two parallel parts: (1) URL-based phishing detection using a Chrome extension and Logistic Regression running on a FastAPI backend, and (2) Email content-based phishing detection using DistilBERT deployed on HuggingFace Spaces. Both parts work in real-time and connect smoothly with the Gmail interface through browser-side automation scripts. Client-side phishing detection using machine learning has also been studied. It aims to give early warnings directly to users and reduces the need for centralized filtering systems [13].

3.1. URL-Based Phishing Detection Architecture

Fig. 1 shows the setup of the URL scanning module. A content script monitors the Gmail interface and automatically extracts all hyperlinks from incoming emails. These URLs are sent to the background script of the Chrome extension, which acts as the main communication hub.

The background script sends a POST request to the FastAPI backend endpoint /predict with the extracted URLs in JSON format. The backend pipeline has a component that converts URL strings into TF-IDF feature vectors. A Logistic Regression inference engine then classifies each URL as legitimate, suspicious, or phishing. The backend responds with a JSON object that includes prediction labels and their probabilities.

When the background script gets the model output, it updates the browser interface. It displays color-coded risk indicators on the extension badge and tooltip. Users can also start manual scans or check URLs whenever they want through the popup UI. This setup offers lightweight, real-time protection with minimal processing load.

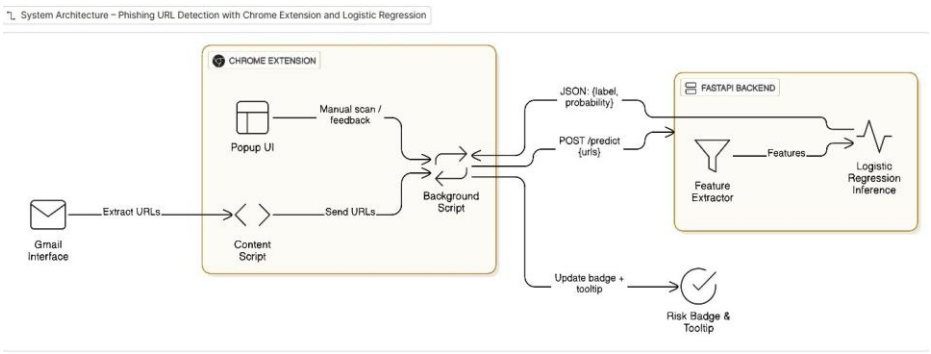

fig. 1. architecture of the url-based phishing detection module using chrome extension and fastapi logistic regression backend

3.2. Email Content-Based Phishing Detection Architecture

Fig. 2 shows the email scanning module powered by DistilBERT. A content script watches the Gmail interface and retrieves the visible email body text when a user opens an email or starts a manual scan.

The extracted text goes to the background script, which sends a POST request to the HuggingFace Space backend that hosts the fine-tuned DistilBERT model. The backend processes the text by breaking it into tokens, extracting embeddings, and using transformer inference to check the meanings and language patterns in the email content.

The backend sends back a JSON response with phishing probability scores. The extension quickly updates the risk badge and tooltip on the Gmail interface, providing users with a useful phishing alert system that is not intrusive. This setup allows effective detection of phishing emails, working alongside URL-based detection to improve overall security.

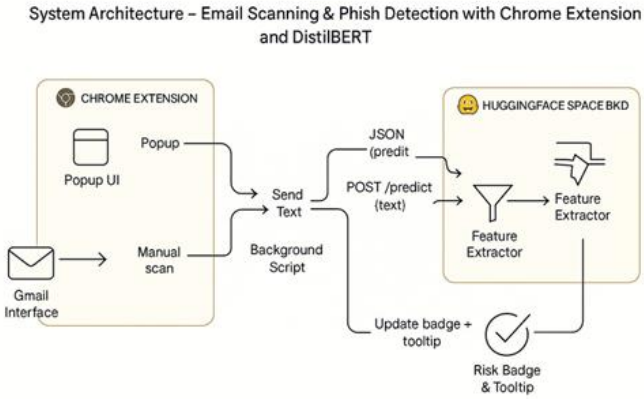

fig. 2. architecture of the email content-based phishing detection module using chrome extension and distilbert deployed on huggingface spaces.

IV. DATASET AND PREPROCESSING

The proposed system uses two separate datasets: (1) A URL dataset for detecting phishing websites and (2) An email text dataset for classifying semantic phishing. Each dataset goes through different preprocessing steps designed for the specific model architectures.

4.1 URL Dataset

The URL-based classifier is trained with a publicly available phishing URL dataset sourced from Kaggle and other open repositories. This dataset is labeled with URLs that fall under categories such as phishing URLs, suspicious URLs, and genuine URLs. We eliminate any duplicate URLs, invalid strings, blank URLs, and samples without labels to ensure proper training data. The size of the resulting data set after removing duplicates is data clean. We perform lexical processing in order to normalize the URL format. Lexical processing is also applied to preprocess the URL structure.

This includes:

- Making all URLs lowercase
- Removing control characters and trailing slashes
- Tokenizing URLs subdomains, paths, parameters, and query segments

• Character repetition and symbol normalization

We then convert the URLs to numerical vectors through Term Frequency-Inverse Document Frequency. TF-IDF This approach can recognize patterns, which are commonly found in phishing URLs, that include large obfuscation strings or suspicious path patterns. The sparse matrix obtained will now act as the input to the "Logistic Regression" classifier.

4.2 Email Text Dataset

For detecting semantic email phishing, we use a large text dataset made up of phishing and legitimate emails from publicly available sources and academic archives. Each email includes subject lines, body text, and social engineering cues when available. We tackle dataset imbalance with under sampling and oversampling techniques to prevent the model from favoring the legitimate classes.

Text preprocessing includes:

• Removing HTML tags, scripts, and inline metadata

• Lowercasing and cleaning up whitespace

• Removing email signatures and boilerplate templates

• Expanding contractions and normalizing punctuation

For transformer-based modeling, the DistilBERT tokenizer changes each email into token IDs, attention masks, and segment embeddings. Unlike traditional ML preprocessing, tokenization keeps grammar, context, and semantic relationships intact, allowing effective detection of socially engineered language.

4.3 Train-Test Split

Both of the models employ an 80/20 split for training and testing. The URL feature data is turned into a TF-IDF format, whereas the email data is converted into DistilBERT token embeddings. The stratified split method guarantees equal distribution of classes in both sets. These resulting datasets make sound inputs for the lightweight URL classifier as well as the deep semantic email classifier. classifier. This enables efficient and scalable phishing detection in various email contexts.

V. METHODOLOGY

The proposed system for phishing detection is a combination of traditional machine learning for URL classification using transformer-based deep learning for semantic email analysis. The approach utilizes two concurrent pipelines: (1) Logistic Regression trained on TF-IDF URL features, (2) A well-tuned DistilBERT model on email content classification. Both models work independently but provide complementary threat assessments.

5.1 TF-IDF Feature Engineering for URL Classification

To convert URLs into understandable numerical representations, we use the Term Frequency-Inverse Document Frequency (TF-IDF) technique. For a given token t in a URL document d, we compute the TF-IDF value as:

$$\text{TF-IDF}(t, d) = \text{TF}(t, d) \times \text{IDF}(t)$$

where

$$\text{TF}(t, d) = \frac{f_{t,d}}{\sum_k f_{k,d}}$$

is the normalized frequency of token t in document d, and

$$\text{IDF}(t) = \log \frac{N}{df_t + 1}$$

Represents the inverse frequency of token t across all N URL samples. TF-IDF effectively highlights uncommon, valuable tokens often found in phishing URLs, such as obfuscated paths, encoded parameters, and suspicious domain patterns. Lexical feature-based URL analysis is popular because it effectively identifies phishing traits without needing external website content or network-based features [14].

5.2 Logistic Regression for URL-Based Phishing Detection

After TF-IDF vectorization, we model the URL classification problem using Logistic Regression. This model estimates the probability that a URL is phishing. The model computes:

$$P(y = 1|x) = \frac{1}{1+e^{-(w^T x+b)}}$$

Where x is the TF-IDF feature vector, w is the weight vector learned during training, and b is the bias term. We optimize the model using the cross-entropy loss:

$$L = -[y\log(\hat{y}) + (1-y)\log(1-\hat{y})]$$

The trained classifier outputs three labels: legitimate, suspicious, and phishing, based on thresholded probability score.

5.3 DistilBERT for Semantic Email Phishing Detection

For email content analysis, we use a lightweight transformer model called DistilBERT. DistilBERT retains 97% of BERT's performance while being 40% smaller, which makes it suitable for real-time inference. The preprocessed email text is sent through the DistilBERT tokenizer, which produces token IDs and attention masks.

During inference, the transformer computes contextual embeddings as:

$$H = \text{Transformer}(X)$$

Where X represents the tokenized input sequence and H denotes the hidden state representations across layers. The denotes the hidden state representations across layers. The embedding for the [CLS] token goes to a classification head:

$$\hat{y} = \sigma(W.H_{CLS} + b)$$

where σ is a sigmoid activation that outputs phishing probability. This allows the model to capture meaning related to urgency, threats, deceptive patterns, and impersonation attempts often found in phishing emails.

5.4 Decision Fusion Mechanism

To reduce false negatives and ensure reliable detection, we combine predictions from both classifiers:

$$Final\_Label = \begin{cases} Phishing, & if\ P_{URL} > \tau 1\ or\ P_{Email} > \tau 2 \\ Suspicious, & if\ either\ score\ is\ moderate \\ Legitimate, & otherwise \end{cases}$$

Thresholds τ1 and τ2 are adjusted based on evaluation. This decision-level ensemble ensures that even if one model fails to identify a threat, the system overall remains protected.

5.5 Real-Time Prediction Pipeline

Both detection models operate in real time:

- The Chrome extension extracts email data.
- The background script sends REST API requests.
- The FastAPI server handles URL classification.
- HuggingFace Space carries out DistilBERT inference.
- The extension displays a risk badge and tooltip.

This setup allows for seamless integration with user work flows and provides instant phishing alerts directly within the Gmail interface.

VI. EXPERIMENTAL RESULTS

This section discusses the evaluation of the phishing detection system. The system has two independent classifiers: (1) A URL based Logistic Regression classifier using TF-IDF features, and (2) A DistilBERT-based email semantic classifier. Results for Module 1 are shown below, and Module 2 results will be added after evaluating the model.

6.1 Module 1, URL-Based Logistic Regression Classifier

The TF-IDF + Logistic Regression model was trained on a dataset of 30,000 labeled URLs, including both phishing and legitimate samples. An 80/20 stratified split was used for evaluation.

Table 1 summarizes the key performance metrics.

table i

performance of url-based logistic regression classifier

Metric	Score
Accuracy	0.9672
Precision (Class 0)	0.9703
Precision (Class 1)	0.9645
Recall (Class 0)	0.9596
Recall (Class 1)	0.9739
F1-Score (Class 0)	0.9649
F1-Score (Class 1)	0.9692
AUC (ROC)	0.9928
Cohen's Kappa	0.9341
MCC	0.9341
Cross-Validation Accuracy (5-Fold)	0.9671
Inference Time	0.003 sec (per 1000 URLs)

6.1.1 Confusion Matrix:

The confusion matrix in Fig. 3 demonstrates strong discrimination ability across both classes. The model shows excellent recall for phishing URLs, accurately identifying most malicious samples.

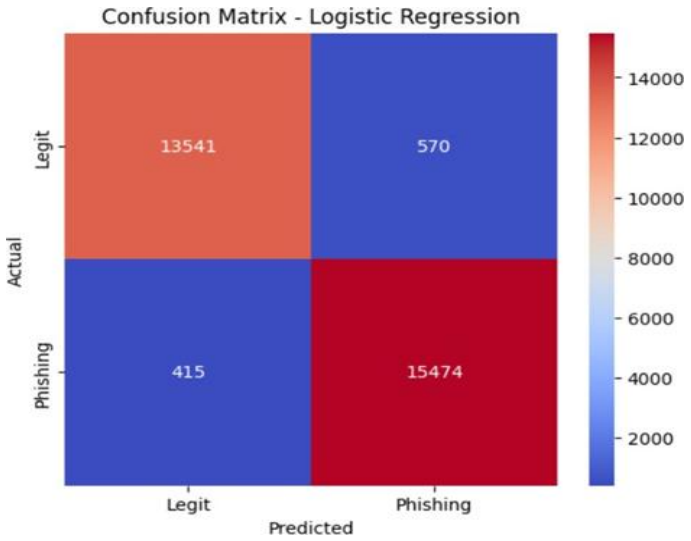

fig. 3. confusion matrix for url-based logistic regression model.

6.1.2 ROC Curve and Precision-Recall Curve:

Fig. 4 displays the ROC curve with an AUC of 0.9928, indicating almost perfect separability between phishing and legitimate URLs.

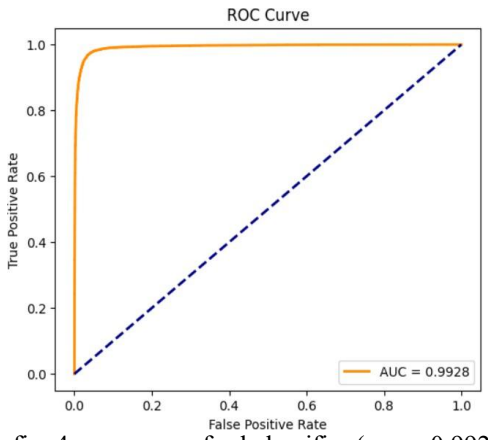

fig. 4. roc curve of url classifier (auc = 0.9928).

Likewise, the precision-recall curve in Fig. 5 maintains high precision and recall across different thresholds.

6.1.3 Feature Importance Analysis:

Table 2 shows the most important lexical indicators identified by the TF-IDF model. High-weight features like "/com/", "/om/", "/org/", and "/net/" often appear in phishing URLs. In contrast, legitimate indicators like ".co", "/wp", and ".php" lower the phishing score.

table ii

top contributing features for url classification

Phishing Indicators	Weight
.com/	25.275
com/	25.071
om/	25.023
.org/	9.534
org/	9.513
.net/	7.115
net/	6.661
.ca/	6.432
et/	6.278
.uk/	5.987

Fig. 6 visualizes these weights in a TF-IDF word cloud.

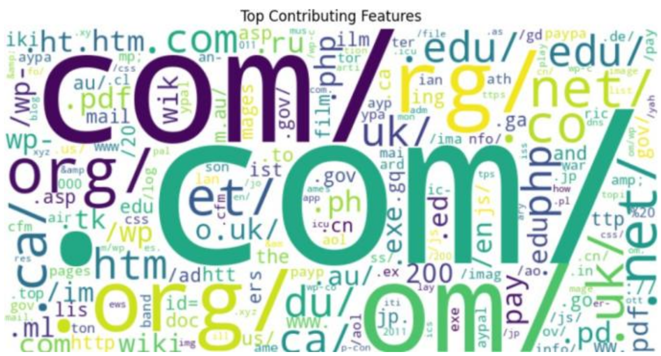

fig. 6. Word cloud of top contributing URL features

6.2 Module 2, DistilBERT Email Content Classifier

The email content-based phishing detection module uses a fine-tuned DistilBERT transformer model to capture semantic and contextual signs of phishing intent. Unlike URL analysis, this module aims to understand misleading language patterns, urgency cues, impersonation tactics, and social engineering methods commonly found in phishing emails.

6.2.1 Training Configuration:

The DistilBERT model is fine-tuned with a supervised learning method on a labeled email dataset containing phishing and legitimate samples. Each email is tokenized using the DistilBERT tokenizer, which generates input IDs and attention masks with a fixed maximum sequence length. The model trains with a binary cross-entropy loss function and utilizes the AdamW optimizer. Training occurs over several epochs, with early stopping to avoid overfitting.

6.2.2 Qualitative Performance Analysis:

During experimental evaluation, the DistilBERT model showed strong ability in spotting phishing emails that rely more on semantic manipulation than on clear malicious links. The model effectively identified patterns related to urgency, such as account suspension warnings, authority impersonation, like banking or institutional language, and misleading calls to action. A manual review of prediction outputs revealed that the model accurately flagged phishing emails even when URLs were hidden or when no hyperlinks were present. This emphasizes the benefit of transformer-based semantic understanding over traditional keyword filtering techniques.

6.2.3 Computational Characteristics:

DistilBERT strikes a good balance between performance and efficiency. When compared to full-scale BERT models, DistilBERT needs fewer parameters and has shorter inference time, making it suitable for

real-time use. The model is hosted on HuggingFace Spaces and accessed through REST API calls, allowing scalable inference without straining client-side resources.

6.2.4 Planned Quantitative Evaluation:

A thorough quantitative evaluation of the DistilBERT classifier will be part of future work. Planned metrics include accuracy, precision, recall, F1-score, confusion matrix, ROC curve, precision-recall curve, training loss, and inference latency. These results will allow for direct comparison with existing transformer-based phishing detection methods.

VII DISCUSSION

7.1 Discussion of Module 1: URL-Based Logistic Regression Classifier

The URL-based phishing detection model shows strong and reliable performance across all evaluation metrics. With an accuracy of 96.72 The confusion matrix further demonstrates the model's stability, revealing low misclassification rates. The high Cohen's Kappa score of 0.9341 and Matthews Correlation Coefficient of 0.9341 show strong agreement beyond chance and confirm the classifier's effectiveness even with slight class imbalance. Additionally, the 5-fold cross-validation accuracy of 0.9671 shows consistent performance across different data sets, rein- forcing the model's generalizability.

A feature-importance analysis uncovers clear lexical pat- terns linked to phishing URLs. Tokens like "/com/", "om/", ".org/", and ".net/" have high positive weights, indicating their frequent misuse in malicious domains and redirection schemes. On the contrary, definite indicators such as "co," "php," "/wp," and "exe" have negative weights. This indicates that they are characteristics of secure sites. This proves the point that TF- IDF can extract distinctive language patterns with the help of rules. A key advantage of this approach is that it is fast. With an inference time of only 0003 seconds per 1000 URLs, this classifier is very lightweight and suited to real-time browser extensions. Through this, users are able to get threat feedback instantly when browsing or reading their mails. Nevertheless, there are some limitations in this model. It could potentially have issues with very heavily obfuscated URLs, dynamic phishing sites and homograph attacks that utilize deceptive Unicode characters. As the model is dependent solely on lexical variables, it is not capable of identifying the context that is concealed in an attacker's message intent.

7.2 Discussion of Module 2:DistilBERT Email Content Classifier

The DistilBERT-based email content classifier works along- side the URL-based detection module.It focuses on the meaning and context of phishing emails. While URL analysis is based on the visible string pattern, the transformer model understands the deceptiveness of language, manipulative intent and social engineering techniques used by attackers. Qualitative analysis also shows that the DistilBERT model is especially good at finding phishing emails that do obvious malicious URLs, and using subtle persuasion methods. These tactics involve impersonation of trusted entities, urgency in requests, and the use of emotional appeals. manipulative language. These can be difficult for traditional rule-based or keyword-driven filters to catch. From a system design perspective, using DistilBERT finds a good balance between how expressive the model is and how well it runs. Compared to the larger transformer models, DistilBERT reduces inference time, along with memory usage, which makes it suitable for real-time email scanning in browser-based environments. Deploying through HuggingFace Spaces increases scalability by shifting computation away from client-side systems. However, the semantic classifier has its limitations. Trans- former models require a lot of labeled data in order to perform well, and their performance may drop with very specific or multilingual phishing content. Added to this, it results in a much longer inference time compared to the lightweight Logistic Regression model employed for URL classification, which again needs careful picking when to use, based on system-level decisions. Similar transformer-based methods for email phishing detection have given good results using DistilBERT for real-time classification tasks [7].

In general, the integration of DistilBERT will enhance the system's capability in detecting sophisticated phishing attempts that bypass Detection based on URL.

Remaining work would be completing a quantitative analysis, enhancing the performance of the inference, and carrying out End-to-end analysis between URL and email content classifiers.

REFERENCES

[1] D. Ferlin Deva Shahila, V. T., and N. Nalini, "AI Based Phishing Decrement for Immense E-Mail Data," IEEE, Sept. 2024. Available: https://ieeexplore.ieee.org/document/10673317

[2] F. S. Alsubaei, A. A. Almazroi, and N. Ayub, "Enhancing Phishing Detection: A Novel Hybrid Deep Learning Frameworks for Cybercrime Forensics," IEEE Access, Jan. 2024. Available: https://ieeexplore.ieee.org/document/10384876

[3] U. Zara, K. Ayyub, H. U. Khan, A. Daud, T. Alsahfi, and S. G. Ahmad, "Phishing Website Detection Using Deep Learning Models," IEEE Access, Nov. 2024.

[4] O. SENGEL, "Analysis of Learning Techniques for Phishing Website Detection," IEEE, Nov. 2024.

[5] Gurushankar H. B., Gururaj H. L., "Detection of Phishing Activities Using Deep Learning Approaches," IEEE, Feb. 2025.

[6] O. K. Sahingoz, E. Buber, E. Kugu, "DEPHIDES: Deep Learning Based Phishing Detection System," IEEE Access, Jan. 2024. Available: https://ieeexplore.ieee.org/document/10388305

[7] E. M. Damatie, A. Eleyan, and T. Bejaoui, "Real-Time Email Phishing Detection Using a Custom DistilBERT Model," IEEE, 2024.

[8] R. Zienietal., "Phishing or Not Phishing? A Survey on the Detection of Phishing Websites," IEEE Access, 2023.

[9] S. V. Boora and S. Singh, "A Comparative Analysis of Machine Learning Algorithms for Phishing Website Detection," IEEE, 2024.

[10] Phishing Trends Report, Anti-Phishing Working Group (APWG), 2023. Available: https://apwg.org

[11] A. A. E. Ahmed and I. Traore, "Anomaly Detection Based on Biometrics for Phishing Email Detection," IEEE Transactions on Information Forensics and Security, vol. 6, no. 4, pp. 1115–1125, 2011.

[12] S. Marchal, J. Francois, R. State, and T. Engel, "PhishStorm: Detecting Phishing With Streaming Analytics," IEEE Transactions on Network and Service Management, 2014.

[13] A. Jain and B. Gupta, "Towards Detection of Phishing Websites on Client-Side Using Machine Learning Based Approach," IEEE Communications Letters, 2017.

[14] A. S. Alsariera, I. A. M. Kamil, and N. Q. Mohammed, "Phishing Attacks Detection Using Lexical Features of URL," IEEE Access, 2020.

[15] C. Whittaker, B. Ryan, and M. Nazif, "Large-Scale Automatic Classification of Phishing Pages," NDSS Symposium, 2010.