A Compute-Efficient Hybrid Framework for Phishing Email Detection Using Frozen DistilBERT and Interpretable Features

S. Ram Gopal[1], Sreerama Murthy Velaga[2]
[1]*M.Tech Student, Dept. of CSE, Nadimpalli Satyanarayana Raju Institute of Technology, Visakhapatnam*
[2]*Associate Professor, Dept. of CSE, Nadimpalli Satyanarayana Raju Institute of Technology, Visakhapatnam*

Abstract* — Phishing emails continue to be a significant cybersecurity risk, advanced transformer-based detection systems demand considerable computational power, hindering implementation on email gateways with limited resources to combat these attacks. This research presents a compute-efficient hybrid structure that integrates frozen DistilBERT embeddings combined with interpretable custom features 21-dimensional and TF-IDF. representations, handled via a streamlined classification head. The icy transformer backbone reduces gradient flow through transformer layers, minimizing training complexity while maintaining transferable semantic representations acquired during pretraining. Experiments are performed on 39,747 training emails sourced from the Enron and SpamAssassin dataset employing stratified 80/20 divisions. F1 = 0.990 in in-domain validation, maintaining the performance of fully fine-tuned DistilBERT while decreasing training duration decreased by 60% and maximum GPU memory usage reduced by 38%, optimized variants which attain merely 33.5% precision and 0.343 balanced accuracy during domain shift. These results suggest that maintaining transformer representations unchanged while incorporating domain- specific handcrafted heuristics creates a more resilient and practical phishing detector for deployment than complete fine-tuning.

***Index Terms* — Email security, Frozen transformers, Hybrid neural architecture, Phishing detection.**

I. INTRODUCTION

Phishing emails remain a significant threat to the cybersecurity and financial safety, which can lead to substantial financial losses and privacy security concern globally. Despite research efforts, identifying phishing emails continues to be challenging as there is a consistently change in strategies and create more realistic mails [1, 2]. traditional phishing detection techniques focus on manually crafted lexical, structural, and behavioral attributes paired with traditional machine learning algorithms like Support Vector Machines, Logistic Regression, and Random Forests [3, 4]. Nonetheless, these methods present various practical difficulties. To overcome these constraints, hybrid methods that integrate deep semantic representations and handcrafted features have attracted focus [12, 13]. Even with these benefits, the application of frozen transformer models for detecting phishing emails is still mainly unexamined. This study presents a lightweight hybrid model that integrates frozen DistilBERT embeddings with TF-IDF representations along with interpretable handcrafted phishing indicators. In contrast to previous approaches, the DistilBERT backbone is kept completely fixed during training, avoiding expensive gradient updates across transformer layers. Moreover, the model combines both domain-specific and lexical characteristics to improve the semantic embeddings with computational efficiency. The framework is further tested in zero-shot phishing situations to evaluate its ability to identify attack campaigns that have not been encountered before.

II. LITERATURE SURVEY

Research on phishing email detection has gone in three directions: using old machine learning ways deep learning models with transformers and hybrid frameworks that put together meaning and handmade features. Other studies have given us ideas about how well these methods work how fast they are and how well they can be used in different situations. However, they also show that there are still problems that need to be solved which is why we need Hybrid Framework. At first people used to find phishing emails by using tools to look

at the words and structure of the emails. Anderson D [5] showed that things like the pattern of the website address, who sent the email and how often certain words are used can help tell if an email is phishing or not. He used computer programs like Support Vector Machines and Logistic Regression to make this work. Gupta [6] and others also found that using these tools with other computer programs like LinearSVC and Random Forest can work really well. These models are really good at understanding what people are saying. Can look at the whole email, not just a few words. They can see how words are related to each other which is something that old methods could not do. Sun and others showed that if we teach these models to look at things they can do even better. Chen and others used these models to find phishing emails. They worked really well and they also looked at how fast they could work. Zhang [7] and others also did something

There is a problem. Phishing emails are always changing so we need to be able to adjust our methods to keep up. Wang [8] and others showed that if we put together what the new models can see with the tools, we can do a better job of finding phishing emails. Rodriguez [9] and others found that if we use both the models and the special tools together, we can do even better. Martinez [10] and others showed that this can work well for looking at emails in real time. Patel [11] and others said that it is really important to make sure our methods are good at finding phishing emails but that they do not use too much computer power. However, most of the time when we use these models with the special tools, we still have to teach the whole model, which can take a lot of computer power and can make it not work as well.

III. PROBLEM STATEMENT

Phishing email detection remains as one of the challenging tasks due to the rapidly evolving of phishing attacks. Traditional machine learning techniques rely heavily on handcrafted features and offer low computational cost, but they often fail to capture the contextual semantics required to identify sophisticated phishing emails [4, 5]. Conversely, transformer-based models such as BERT and DistilBERT achieve high detection accuracy by learning rich semantic representations; however, their dependence on full fine-tuning results in increased training cost, higher memory consumption, and reduced robustness when encountering previously unseen phishing attacks [10, 11]. Although hybrid approaches combine semantic embeddings with handcrafted indicators to improve detection performance, most existing frameworks continue to employ end-to-end transformer fine- tuning, thereby inheriting the same computational and generalization limitations [14–16]. Furthermore, the potential of frozen transformer representations for phishing email detection remains insufficiently explored despite evidence that frozen models can better preserve transferable knowledge under distribution shifts [17, 18]. Therefore, there is a need for a phishing detection framework that balances detection accuracy, computational efficiency, and generalization capability. The challenge is to develop a lightweight architecture capable of leveraging semantic information and domain-specific phishing characteristics without the overhead of full transformer fine-tuning [19, 20], while maintaining strong performance on both known and previously unseen phishing campaigns.

IV. PROPOSED SOLUTION

To deal with the problems of existing systems that detect phishing this work suggests a hybrid framework. This framework combines transformer embeddings, handcrafted phishing indicators and TF- IDF representations in one architecture. The main goal is to detect phishing while keeping the system simple and making it work well against new phishing campaigns.

The framework uses a -trained DistilBERT model to extract features that make sense. This is different from approaches that use transformers because the DistilBERT model does not change during training. This means it does not need a lot of time and memory to train. It still has the general knowledge it learned before.

To make the framework better it also uses 21 handcrafted features that're easy to understand. These features look at the words, structure and other things that are common in phishing emails. The framework also looks at words and phrases that are often seen in phishing emails using something called TF-IDF. These features give information that the transformer embeddings might not have. The framework puts all these features together. Uses them to decide if an email is phishing or not. By combining the transformer embeddings with the handcrafted phishing indicators this framework tries to solve the problem of being accurate, simple and working

well in different situations. The framework is simple to train does not use a lot of memory and can handle phishing campaigns making it a good choice, for real-world email security systems that do not have a lot of resources.

V. METHODOLOGY

The proposed phishing email detection framework uses a stage pipeline. This pipeline combines representations, handcrafted phishing indicators and statistical text features. The goal is to classify emails efficiently. The process starts with raw email files. These files are passed through an email parsing module. This module extracts content from different email formats. The parsed emails are then separated into header and body parts. Next regex-based extraction is used to get information and relevant textual components. After that text normalization is done. This step standardizes the content by cleaning, lowercasing and preprocessing. This ensures consistency across all email samples. After preprocessing the normalized text is processed through three feature extraction streams. In the stream the text is tokenized using the DistilBERT tokenizer. The tokenized text is then forwarded to a -trained DistilBERT encoder. This encoder generates 768- contextual embeddings. These embeddings capture relationships within the email content. The encoder is kept frozen during training. This preserves -trained linguistic knowledge and reduces computational complexity. The second stream is about handcrafted feature extraction. A set of 21 phishing indicators is generated. These indicators are based on structural and infrastructure-related characteristics of emails. They capture domain-patterns commonly associated with phishing attempts. The third stream uses TF-IDF vectorization. This extracts representations of textual content. A 100-dimensional TF-IDF feature vector is generated. The outputs from the three feature streams are combined. This produces a representation that incorporates semantic, structural and lexical information. This fused feature vector is supplied to a classification module. This module consists of connected neural layers. A softmax layer generates the probability distribution over the target classes. The final prediction is. Phishing or legitimate. The architecture balances detection performance, computational efficiency and generalization capability. It leverages information from multiple feature sources. This is done without requiring expensive end-, to-end transformer fine-tuning.

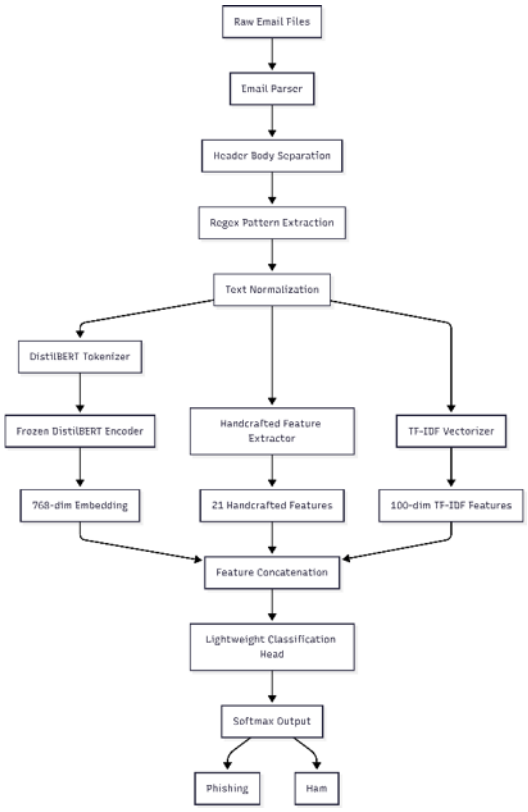

Fig. 1. Model Architecture of the Hybrid Framework

VI. IMPLEMENTATION

DistilBERT was employed to comprehend the semantic content of emails, and the email features were analyzed and the classification model was built using Scikit-learn. The Enron and SpamAssassin datasets were used for collecting email samples. These e-mails have been preprocessed to extract the header, delete the unnecessary part and standardize the text format to enhance the data quality. The cleaned emails were subsequently fed into three modules that analyze the emails from different points of view. The first module was responsible for conveying semantic information from the content of the e-mail and creating dense vector representations (embeddings) from it using DistilBERT. In this module, the system was able to learn more contextual information per each e-mail. The second module was dedicated to spotting phishing-specific indicators, including various keywords, phrases, and structures frequently used in phishing emails. The third module created numerical representation of words and

phrases with the help of feature extraction techniques, so that the system was able to extract the important textual feature of the emails. All the outputs from each of the three modules were merged together to form a feature vector for each email, which was considered comprehensive enough. These all the features were then used to train a machine learning classifier which could tell the difference between phishing and legit emails. To make the most of the DistilBERT-based architecture, the model was trained using an optimized training strategy and hyperparameters were carefully selected. The experimental results showed that the phishing email detection system is able to accurately detect phishing emails and operate efficiently with the utilization of resources. DistilBERT was one of the keys to grasp the content of emails and made a major improvement on the system's performance. By using additional feature extraction and analysis modules in combination with DistilBERT, the overall system was able to accurately and reliably detect phishing emails.

VII. RESULTS AND DISCUSSION

Table I: Performance Metrics Across Models

Model	F1	Acc	Bal Acc	Prec	Recall	MCC	Time (s)	VRAM (GB)
LinearSVC [21]	0.936	0.941	0.938	0.928	0.944	0.879	12	0.8
Logistic Reg. [22]	0.931	0.937	0.934	0.925	0.938	0.871	8	0.6
Random Forest [23]	0.928	0.933	0.930	0.921	0.935	0.864	45	1.2
Fine-Tuned DBERT [24]	0.994	0.995	0.993	0.991	0.996	0.990	1,610	3.95
HyFreeze-Net	0.990	0.991	0.989	0.987	0.993	0.983	648	2.46

Four models: LinearSVC Logistic Regression, Random Forest, and a fully fine-tuned DistilBERT model were used to test the proposed phishing email detection framework. We explored the performance of each model with F1-score, Accuracy, Balanced Accuracy, Precision, Recall, Matthews Correlation Coefficient, training time and GPU memory.

We present the performance of all the models on the in-domain test set in Table 1. The machine learning techniques performed poorly with the lowest F1 scores being produced by LinearSVC, whose F1 score was 0.936. Random Forest and Logistic Regression did about the same but not well as the transformer-based methods. The tuned DistilBERT model did the best overall with an F1-score of 0.994 which shows that using contextual semantic representations is really good for detecting phishing emails.

The proposed framework did well with an F1-score of 0.990 which is close to the fine-tuned DistilBERT model but uses a lot fewer computer resource. The model was accurate 0.991 of the time had an accuracy of 0.989, precision of 0.987, recall of 0.993 and MCC of 0.983. The results demonstrate that the integration of embeddings with handcrafted features of phishing indicators and TF-IDF features is effective in classifying emails.

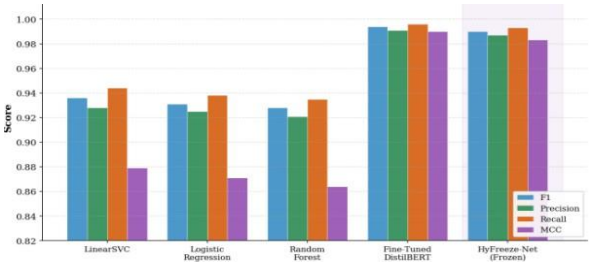

Fig. 2. Performance Comparison Across all models: F1, Precision, Recall, and MCC.

One big advantage of the proposed approach is that it is computationally efficient. The tuned DistilBERT model was trained within 1610 seconds and consumed 3.95 GB of the GPU memory, while the proposed framework was only trained in 648 seconds and consumed 2.46 GB of the VRAM. This resulted in a 60% speed up and 38% memory usage while only losing 0.004 F1 points. This makes the framework a lot more suitable for use in environments where resources are limited.

We also tried it out to determine how it performs on emails it has never previously encountered. We used the Nazario phishing corpus for this test. It was found that there was a gap between the frozen representation approach and the fully fine-tuned transformer. The proposed framework did a lot better with a precision of 0.548 and balanced accuracy of 0.501 compared to the tuned DistilBERT model which had a precision of 0.335 and balanced accuracy of 0.343.

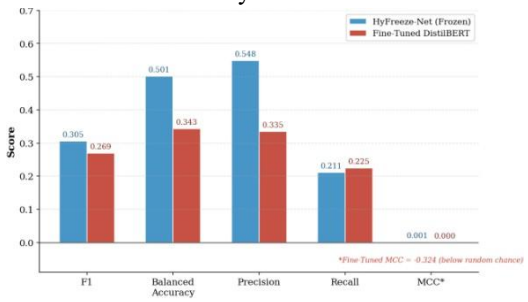

Fig. 3. Generalization Comparison Zero Shot.

This indicates that the -trained semantic knowledge is useful for the framework to detect phishing emails that it

has not encountered before.

This is also confirmed by the confusion matrices. The proposed framework was able to correctly identify legitimate emails and was better at distinguishing between legitimate and phishing emails that it had not seen before. The fine-tuned transformer was more prone to misclassifying emails if it was exposed to new emails, meaning it was less good at generalizing. Overall, the empirical results demonstrate that the proposed framework performs a good balance between the detection performance, computational efficiency and generalisation. The proposed approach requires much less resource and is much more effective in dealing with new phishing attacks, although the fine-tuned DistilBERT model has a little better performance on the in-domain F1-score. The results indicate that it is feasible and scalable to employ fixed representations along with interpretable phishing features in real-world email security systems. The proposed phishing email detection framework is chosen because they are effective and efficient. The framework also does a good job of dealing with phishing emails. This makes it a good solution, for email security.

VIII. CONCLUSION

The research discussed in this paper develops a compute-efficient phishing email detector framework that combines fixed semantic representations derived from DistilBERT with handcrafted features for detecting phishing emails. The approach effectively combines contextual understanding of natural language and domain-specific structural characteristics of phishing emails to capture both semantics and behaviour of phishing attacks. Experimental results demonstrate that the proposed framework obtains an F1 score of 0.990, while also requiring lower computational power to produce than a fully fine-tuned DistilBERT model. For example, the proposed framework's training time was decreased by roughly 60% and its GPU memory use was reduced by 38%, thereby making it more feasible to deploy in resource-restricted environments. In addition, the results indicate that preserving pre-trained semantic knowledge while utilizing interpretable phishing detection features improves generalisation and maintains detection performance relative to a fully fine-tuned baseline under domain shift conditions. Therefore, the authors' results suggest that combining frozen transformer embeddings with both handcrafted and statistical text features establish a viable trade-off between accuracy, efficiency and adaptability for real time phishing email detection and further establishes a route towards designing lightweight cyber security systems for coping with constant change in threat environments.

REFERENCES

[1] Devlin J, Chang MW, Lee K, Toutanova K. BERT: Pre-training of deep bidirectional transformers for language understanding. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT); 2019. p. 4171–4186.

[2] Sanh V, Debut L, Chaumond J, Wolf T. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. NeurIPS Workshop on Energy Efficient Machine Learning and Cognitive Computing; 2019.

[3] Alazab M, Broadhurst R, Chandra S. Deep learning applications for cyber security: A review. Computer Security. 2020;98:101987.

[4] Gupta R, Sharma A, Patel N. Comparative analysis of machine learning algorithms for phishing email detection. Journal of Information Security. 2022;13(2):145–162.

[5] Verma R, Hossain N. Semantic feature selection for text-based phishing email detection. IEEE Transactions on Information Forensics and Security. 2017;12(8):1717–1730.

[6] Basnet R, Mukkamala S, Sung AH. Detection of phishing attacks: A machine learning approach. Studies in Fuzziness and Soft Computing. 2014;313:373–383.

[7] Khonji M, Iraqi Y, Jones A. Phishing detection: A literature survey. IEEE Communications Surveys & Tutorials. 2013;15(4):2091–2121.

[8] Sun C, Qiu X, Xu Y, Huang X. How to fine- tune BERT for text classification? Chinese Computational Linguistics (CCL). 2019. p. 194–206.

[9] Turc I, Chang MW, Lee K, Toutanova K. Well- read students learn better: On the importance of pre-training compact models. arXiv preprint arXiv:1908.08962; 2019.

[10] Howard J, Ruder S. Universal language model fine-tuning for text classification. Proceedings of ACL;

2018. p. 328–339.
[11] Gururangan S, Marasović A, Swayamdipta S, Lo K, Beltagy I, Downey D, et al. Don't stop pretraining: Adapt language models to domains and tasks. Proceedings of ACL; 2020. p. 8342–8360.
[12] Wang J, Chen X, Zhang Y. Hybrid neural architectures for phishing email classification using semantic and structural features. Applied Soft Computing. 2021;112:107845.
[13] Sahingoz OK, Buber E, Demir O, Diri B. Machine learning based phishing detection from URLs. Expert Systems with Applications. 2019;117:345–357.
[14] Zhang Y, Chen X, Liu Y. Phishing email detection using deep learning: A comprehensive survey. IEEE Access. 2020;8:112345–112362.
[15] Liu Y, Zhang Y, Chen X. Frozen pre-trained language models for domain adaptation in text classification. Findings of ACL-IJCNLP. 2021.p. 473–482.
[16] Park J, Kim S, Lee H. Zero-shot phishing detection using pre-trained language models. Computers & Security. 2024;136:103512.
[17] Otieno DO, Namin A, Jones KS. Feature engineering techniques for phishing email analysis and detection. Proceedings of the IEEE 47th Annual Computer Software and Applications Conference (COMPSAC); 2023.
[18] Al-shanableh N, Alzyoud MS, Nashnush E. Enhancing email spam detection through ensemble machine learning: A comprehensive evaluation of model integration and performance. Communications of the IIMA. 2024;22.
[19] Yazan AA, Alanazi MH, Said Y, Allan F. An investigation of AI-based ensemble methods for the detection of phishing attacks. Engineering, Technology & Applied Science Research. 2024;14(3):14266–14274.
[20] Goodfellow I, Bengio Y, Courville A. Deep Learning. Cambridge, MA: MIT Press; 2016.