National College of Ireland

Phishing Detection and Prevention Using Natural Language Processing (NLP)

MSc Research Project
MSc in Cybersecurity

Manideep Kokkalakonda
Student ID: X23244488

School of Computing
National College of Ireland

Supervisor: Joel Aleburu

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Manideep Kokkalakonda
Student ID:	X23244488
Programme:	Msc in Cybersecurity **Year:** 2024-25
Module:	MSc (Research) Practicum/Internship
Supervisor:	Joel Aleburu
Submission Due Date:	15/09/2025
Project Title:	Phishing Detection and Prevention Using Natural Language Processing (NLP)
Word Count:	7479 **Page Count:** 22

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Manideep Kokkalakonda

Date: 15/09/2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Phishing Detection and Prevention Using Natural Language Processing (NLP)

Manideep Kokkalakonda
X23244488

Abstract

Phishing remains one of the most prevalent cybersecurity threats, with email serving as the primary attack vector. Traditional detection methods—such as blacklists, keyword matching, and rule-based systems—struggle to detect evolving attacks that mimic legitimate communication and exploit subtle social engineering tactics. This research presents a hybrid phishing email detection model that integrates BERT-based semantic embeddings with structural and stylistic features, processed through a BiLSTM architecture to capture both contextual meaning and sequential dependencies.

The study compiles a unified dataset from multiple publicly available phishing and legitimate email sources, totalling **164,971 emails (85,781 phishing and 79,190 legitimate)**, and applies comprehensive NLP preprocessing, and extracts a 782-dimensional feature vector combining 768 BERT embedding dimensions with 14 handcrafted features (e.g., special character ratio, URL count, money-related terms).

The proposed hybrid model is evaluated against traditional TF-IDF-based machine learning baselines, including Logistic Regression, Random Forest, and Gradient Boosting. Experimental results demonstrate high accuracy (95.7%), strong recall for phishing detection (96.35%), and excellent ROC-AUC performance (0.9916).

Explainable AI techniques (SHAP, LIME) provide feature-level insights, revealing that urgency keywords, monetary terms, and special character usage are key phishing indicators. The model offers a scalable, interpretable, and enterprise-ready framework for real-time phishing detection.

1 Introduction

1.1 Background

Phishing continues to be one of the most widespread and damaging cyber threats, with email serving as its primary delivery mechanism. The APWG Phishing Activity Trends Report (Q4 2024) reported over 1.3 million unique phishing URLs in a single quarter—an all-time high. Attackers employ increasingly sophisticated methods such as brand impersonation, spear-phishing, and carefully crafted social engineering to bypass detection systems and deceive recipients.

Unlike malware-based threats, phishing exploits human psychology—triggering emotions such as urgency (“Your account will be suspended”), fear, and trust in authority (“Official bank notice”). Campaigns in the modern world can evade the generic spam blockers by using natural language that is nearly identical to legit communications, hidden malicious links in masqueraded forms, and by using domain names that appear visually similar to insider brands.

These attacks are hard to detect traditionally (using keyword matching, blacklists and manually compiled rules). This system is reactive in nature and requires known patterns and becomes susceptible to around minor obfuscations. Whenever large-scale and highly individual phishing attacks occur, a more urgent necessity is observed to create a detection mechanism that can identify context, purpose, and meaning of the information contained in the email.

Whenever large-scale and highly individual phishing attacks occur, a more urgent necessity is observed to create a detection mechanism that can identify context, purpose, and meaning of the information contained in the email. Natural Language Processing (NLP) offers powerful tools to address these challenges by enabling models to understand the context, semantics, and intent behind email content rather than relying solely on surface-level patterns. Modern NLP techniques, particularly transformer-based models like BERT, capture nuanced linguistic cues and adapt to new phishing strategies more effectively, even in adversarial scenarios.

The Natural Language Processing (NLP), especially transformers-based NLP, such as the BERT, showed extremely impressive ability of achieving semantic understanding of text. These models and handmade structural features (e.g., the number of special characters, URLs or money-related terms) can help gain a better multi-dimensional insight into phishing characteristics, thereby more precisely identifying phishing characteristics.

1.2 Problem Statement

The primary challenge in phishing detection lies in identifying attacks that are characterised by subtle linguistic cues, novel attack strategies, and zero-day techniques. Rule-based and traditional machine learning models often fail in such scenarios because they cannot extrapolate beyond known keywords and patterns, they overlook deeper semantic relationships in the text, and they lack adaptability to continuously evolving phishing tactics.

Even models that incorporate shallow lexical features—such as word counts or email length—struggle to capture the contextual and sequential patterns that often reveal deception. Addressing this gap requires a hybrid deep learning approach that integrates semantic understanding from transformer-based models with structural and stylistic feature analysis, while also incorporating explainability mechanisms to foster trust among end-users and security teams.

1.3 Research Aim and Objectives

The aim of this research is to develop a hybrid phishing email detection system that combines BERT-based semantic embeddings with structural features within a BiLSTM-driven architecture, delivering both high detection accuracy and strong model interpretability. To achieve this, the study compiles and unifies multiple phishing and legitimate email datasets into a balanced and representative corpus, followed by comprehensive preprocessing and normalization for effective NLP-based modelling. Semantic representations are extracted using BERT embeddings, complemented by handcrafted structural features, and integrated into a hybrid deep learning framework.

The system's performance is compared with traditional TF-IDF-based machine learning models such as Logistic Regression, Random Forest, and Gradient Boosting. Furthermore, explainability tools like SHAP and LIME are incorporated to provide feature-level insights, and the final framework is designed for scalability and readiness in enterprise-level deployment.

1.4 Research Question

How can a hybrid model combining transformer-based semantic embeddings, structural feature engineering, and sequential modeling effectively detect phishing emails, while offering interpretable explanations for decision-making?

1.5 Scope of the Study

This study focuses solely on **email-based phishing detection** using **English-language text** from email subjects and bodies, supplemented by a set of structural features. It does not cover other cyber threats (e.g., malware detection) or metadata/attachment analysis. While limited to English, the methodology can be adapted for multilingual phishing detection in the future.

1.6 Contribution of the Study

This research makes several key contributions to the field of phishing detection. It develops a hybrid BERT+BiLSTM architecture that effectively combines semantic embeddings with structural features to enhance classification accuracy. The study demonstrates the benefits of fusing these complementary feature types, showing measurable improvements over traditional approaches.

Such methods as explainable AI (SHAP and LIME) is used to bring visibility into the decision made by the model, and they allow global and local interpretability. The hybrid model also is compared to traditional machine learning models based on the TF-IDF, measuring the improvement in performance. Lastly, it provides a scalable, reproducible detection pipeline that can be practically run in enterprise.

2 Related Work

Phishing is a consistent and even more advanced cyber threat that uses linguistic influence and human psychological preferences to manipulate people into providing classified information e.g. log-in details, financial information or personal identifiers. It can be a fraudulent email or a maliciously created web address, but in any case phishing is supported by fine wording and situations of deceiving patterns, which cannot be detected, using naive or cursory methods, in a semantically significant manner.

Such a dynamic threat environment has highlighted the inadequacy of historical countermeasures, blacklist filters, heuristics checks or simply matched key words that would be easily bypassed by an adversary due to use of obfuscation, lexical changes or dynamically created malicious URLs.

With attackers becoming more skilful in their methods, so too have the defensive measures with this moving to adaptive detection measures which take advantage of machine learning (ML), deep learning, and Natural Language Processing (NLP). The approaches are to be used to meet lexical as well as semantic patterns given the nature of capturing phishing

targets in spite of novelty of the surface form in the event. Such evolution, however, to more sophisticated and profound models, has caused an urgent, continuing debate among research workers:

Is phishing detection more useful as a semantically robust detection system, that can leverage more context to provide richer results (at the cost of greater computation bill), or should it focus on being computationally efficient, and therefore easier to deploy in resource limited scenarios?

2.1 Rule-Based and Heuristic Detection

Early phishing defense systems were too much dependent on rules and heuristic features that were designed manually. They involved, among others, the malformed URLs detecting, matching of keywords based on a blacklist of known bad words, and referencing blacklists that are already in existence (Aggarwal et al., 2014). Such systems were of value because they had low computational overheads and high interpretability since security analysts could easily see why a URL or message has been flagged.

This transparency was however kept at a cost of rigidity. Lexical manipulations even of the trivial sort (substituting letters with those that look similar to the eye, e.g. the homoglyphs, such as replacing the letter o with the digit 0, inserting random subdomains, or resequencing the tokens) could evade detection. ANDRIU (2023) found out that only slight change in word or character replacement made many heuristic systems ineffective. The fact that they could not process unstructured text was also emphasized by Arjunan (2024) when there was deceitful intent presented in grammatically correct and natural-sounding language.

2.2 Machine Learning Approaches

The introduction of ML-based detection represented a major detour away from a fixed rule approach toward statistical pattern recognition. Other types of classifiers, like Support Vector Machines (SVMs), Random Forests, and k-Nearest Neighbors (k-NN) have learned to recognize phishing attacks based on the extracted metrics like: length of the URL, ratio of digits to characters, numbers of tokens, or occurrences of suspicious terms (Sahingoz et al., 2019).

These models also provided better flexibility, since they are able to learn new datasets without having to re-define the rules manually. Even still, they relied on feature engineering which is a domain specific process and one prone to omission errors. Ismail et al. (2022) reported more than 90 percent accuracy when training SVM classifiers on highly parametrized datasets, however, very low accuracy when attempting them on out-of-domain inputs. This was reiterated by Mittal et al. (2022) who explained that although feature-rich ML models scored better than heuristics, they were run counter by zero-day phishing URLs that attacked unmodelled linguistic patterns.

The key benefit of such techniques is that they have a low computation requirement, an important factor which makes these techniques suitable to perform real-time detection in constrained settings. However, the limitation of their performance is the feature engineering bottleneck which restricts them in being able to pick up subtle and contextual cues as seen with the sophisticated phishing campaigns.

2.3 NLP-Based Models

The tool of integration of NLP methodology into the process of detecting phishing allowed better comprehension of semantic and syntactic relationships in the data based on text. Preliminary research on detecting phishing emails through tokenisation and part-of-speech tagging was carried out by Verma et al. (2012). Most recent methods like Fernandez Rodriguez et al. (2022) have used token-level representations on URL sequences such that a system can identify the phishing domains that impersonate original domains such that they retain elements that are contextually inconsistent.

NLP-based detection fills in the gap between surface-based lexical matching and structural modelling and exposes patterns that can be otherwise elusive to purely statistical analysis. Nevertheless, the common approach to NLP pipelines in the past would use fixed embeddings (typically Word2Vec or GloVe) that do not dynamically adapt word meaning according to its context. This limitation becomes critical when detecting multilingual phishing, adversarial token manipulation, or dynamically generated phishing domains.

2.4 Transformer-Based Architectures

Transformer-based architectures, particularly BERT, have revolutionised phishing detection by providing context-aware embeddings that adapt token meaning based on surrounding text. This capability enables models to detect subtle phishing cues, even when adversaries employ sophisticated obfuscation techniques. Within this domain, an ongoing debate centres on whether to adopt the full BERT model or opt for smaller, faster variants such as DistilBERT or MobileBERT.

Efficiency advocates, such as Uddin and Sarker (2024), have shown that DistilBERT retains approximately 95% of BERT's accuracy while delivering a 60% reduction in processing time, making it a practical choice for edge devices and mobile deployments. Similarly, Roy and Nilizadeh's (2024) PhishLang, based on MobileBERT, achieved strong detection rates with lower memory and computational requirements, incorporating SHAP-based interpretability to enhance operational transparency.

On the contrary, some researchers such as Boulieris et al. (2023) who support depth maintain that the stronger embeddings of the complete BERT perform better in multilingual, domain-free and adversarial settings. They connote that in enterprise-scale contexts, where the price of a single undetected phishing attack might be disastrous, a small uptick in detection performances is due to the increased computing cost.

2.5 Reconciling Semantic Depth vs Computational Efficiency

This trade-off between semantic efficiency and computational cost has direct practical implications, such as whether a phishing detection system should be optimised to running fast on computers that have limited resources and are used by consumers, or more slowly but more accurately on servers that are hardware or software limited essentially affecting whether the system is designed to work in the cloakroom industry, or on an enterprise.

The study fits in the Depth Camp, and it is motivated by three reasons why full BERT embeddings are prioritised regardless of lighter alternatives. To begin with, the system is planned to be implemented as the server deployment of enterprises where the hardware

infrastructure is strong enough to diminish the effects of model latency. Second, the richer embeddings of full BERT offer a greater degree of multilingual robustness as it can detect events successfully in many different linguistic patterns a more important feature, as far as organisations during a phishing threat internationally. Third, the combination of SHAP and LIME guarantees an interpretable system that overcomes the problem of the black box of deep learning and does not reduce performance. The resulting depth, coverage, and transparency of the approach is especially appropriate in such critical security situations in which precision must not trade off against speed.

2.6 Research Gaps

Even though some impressive strides have been made, current research on transformer-based phishing detection has a number of research gaps. To begin with, the issue of dataset size is considered crucial, since a lot of work continues to use old, or artificially created corpora (Qatawneh, 2024), which may result in overfitting and poor extrapolation to reality. Second, transformers lack the explanatory power required to win the trust and confidence of users and analysts who are forced to trust and rely on the decisions they make. Although explainability tools such as SHAP and LIME are available, few people have been able to plug them into the phishing detection pipeline; therefore, they remain ineffective and not trustable.

Third, cross-domain and multilingual-.To fill these gaps, there should be innovative and multicultural data, closer incorporation of explainability framework, and thorough cross-lingual evaluation to be able to maintain sound transformation against the changing threats in the world. This would go a long way in enhancing the reliability and flexibility of operations and in enterprise deployments related to depleted enterprises security.

2.7 Rationale for Current Study

The present research deals with the debate of semantic depth versus efficiency by combining it with a hybrid phishing detection framework that enables a balance between rich contextual insights with complementary structural information. The architecture takes advantage of full BERT embeddings (768 dimensions) to capture subtle semantics relationships in email-texts and uses a BiLSTM layer to model the sequential relationship of the text.

As a method of increasing predictive performance, these contextual embeddings are combined with a number of carefully-chosen lexical and structural features, including token entropy and domain age, providing further discriminatory ability not limited to semantic cues. SHAP and LIME have been used to explain the model and integrate explainability into the pipeline to allow analysts to map out both global and local model decisions without incurring any loss in accuracy.

The system is optimised with server level hardware resources to support the heavier processing needs of full BERT and is intended specifically to run on enterprise-level deployments. This method shows how reflecting on design trade-offs, it is possible to derive that by prioritising semantic richness and interpretability, one is able to achieve the greatest security impact in a high-stakes operational setting.

3 Research Methodology

This chapter presents the methodology used to develop and evaluate the phishing email detection framework. It details the research design, data sources, preprocessing pipeline, feature extraction methods, model development strategies, and evaluation procedures. By combining traditional machine learning with a hybrid BERT + BiLSTM approach, and integrating both textual and structural features, the study ensures a balanced, reproducible, and explainable workflow

3.1 Research Design

This study employs an experimental, supervised machine learning design for phishing email detection, integrating both textual and structural features. The process is iterative, refining methods at each stage based on results. Multiple trusted datasets containing phishing and legitimate emails were aggregated to create a diverse, representative corpus.

Exploratory Data Analysis (EDA) assessed dataset composition, class balance, and characteristics such as email length, word frequency, and anomalies. Preprocessing involved a customised text-cleaning pipeline to remove HTML tags, URLs, email addresses, punctuation, numbers, and extra whitespace, followed by lowercasing, stopword removal, and lemmatization.

Two feature extraction paths were applied: TF-IDF n-grams for statistical text patterns, and pre-trained BERT embeddings for semantic context, with structural metadata added in the deep learning path. Traditional models (Logistic Regression, Random Forest, GradientBoostingClassifier) were trained on TF-IDF features, while a BERT + BiLSTM model used embeddings and structural features. Models were validated via hold-out sets, stratified 5-fold cross-validation, and explainability analysis using SHAP and LIME.

3.2 Data Collection and Compilation

The dataset was compiled from seven publicly available email corpora containing both phishing and legitimate messages: CEAS_08, Enron, Ling, Nazario, Nigerian_Fraud, SpamAssassin, and a consolidated Phishing_Email dataset. Each source was loaded individually, checked for encoding or formatting issues, and standardised into a uniform structure. A dataset_source field preserved origin metadata, while a text_combined field merged email subject and body where available.

The integrated dataset contained 164,971 emails (85,781 phishing, 79,190 legitimate), representing a 52:48 phishing-to-legitimate ratio. For experimentation, a pre-cleaned, balanced subset from the Phishing_Email dataset was selected, comprising 82,486 records (42,891 phishing, 39,595 legitimate). Each record included the concatenated text, a binary label (1 = phishing, 0 = legitimate), and source identifier.

This compilation ensured diversity and representativeness, providing a strong foundation for subsequent preprocessing, feature extraction, and model development while supporting reproducibility and scalability in phishing detection research.

3.3 Data Preprocessing

A customised text preprocessing pipeline was developed to prepare emails for both traditional machine learning and BERT-based deep learning models. Noise removal involved stripping HTML tags, URLs, email addresses, numbers, and punctuation, followed by lowercasing and tokenisation. Stopwords were removed using the NLTK English list, and lemmatisation (without stemming) was applied to preserve semantic clarity. Whitespace was normalised, and empty or extremely short texts were discarded.

Exploratory text length analysis identified anomalously short messages that could bias training. The dataset was split into training, validation, and test sets in a stratified 60:20:20 ratio to maintain class balance. In the traditional ML pipeline, cleaned text fed directly into TF-IDF vectorisation. For deep learning, raw text was tokenised with the BERT tokenizer, preserving context for embeddings. To address class imbalance, SMOTE oversampling was applied only to the training data in the deep learning workflow, avoiding leakage into validation or test sets.

3.4 Feature Extraction

The feature extraction stage aimed to capture both lexical patterns and deep semantic relationships in email text. Two complementary pathways were implemented. In the traditional machine learning pipeline, textual content was transformed into numerical form using Term Frequency–Inverse Document Frequency (TF-IDF) vectorization.

This approach quantified term importance by balancing within-document frequency against corpus-wide occurrence, using a 1–2-gram range, a 5,000-feature limit, and filtering terms with min_df=2 or max_df=0.95. English stopwords were removed, and fitting was restricted to the training set to prevent data leakage.

In the deep learning pathway, semantic representations were derived from a pre-trained BERT model. Emails were tokenized with the BERT tokenizer, and the [CLS] token embeddings were extracted as dense, context-aware vectors. Additionally, structural metadata—such as email length, numeric character counts, and special symbol frequency—was incorporated exclusively in this pathway, enriching the embeddings with non-textual cues to enhance phishing detection accuracy.

3.5 Model Development

The modelling phase employed two complementary strategies: a traditional machine learning pipeline using TF-IDF features, and a deep learning pipeline combining BERT embeddings with a BiLSTM architecture.

For the traditional approach, three supervised learning algorithms were implemented — Logistic Regression, Random Forest, and GradientBoostingClassifier from scikit-learn — to establish strong baselines. These models were trained exclusively on TF-IDF n-gram features from the pre-processed text. Hyperparameters were kept largely at defaults for reproducibility, except for setting max_iter=1000 in Logistic Regression to ensure convergence and n_estimators=100 in the tree-based models to balance performance and efficiency.

For the deep learning approach, the BERT + BiLSTM model processed tokenized emails through a pre-trained BERT to extract [CLS] embeddings, which were concatenated with structural metadata (e.g., email length, numeric counts, special symbol frequency). This enriched representation was passed through a BiLSTM layer and then directly to a classification layer, avoiding additional dense MLP layers to maintain simplicity. The pre-trained BERT weights were frozen during training to reduce computational overhead, prevent overfitting on the relatively smaller dataset, and leverage the generalised language understanding already captured during large-scale pretraining.

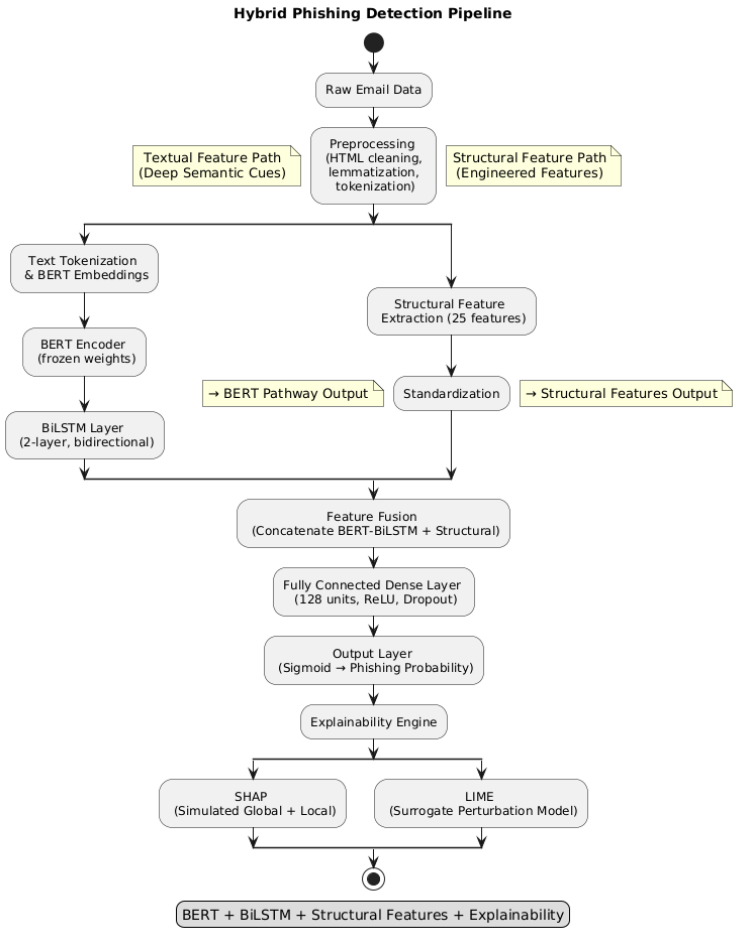

Figure 1 : Overview of the hybrid BERT + BiLSTM phishing detection architecture.

3.6 Evaluation Methodology

The evaluation process was designed for fairness, robustness, and reproducibility. The dataset was split into training, validation, and test sets in a 60:20:20 ratio using stratified sampling to maintain the phishing-to-legitimate ratio. Models were trained on the training set, tuned using validation performance, and final results reported on the unseen test set for an unbiased measure of generalisation.

For traditional machine learning models, robustness was further validated using five-fold stratified cross-validation on the training set, reporting mean and standard deviation of the weighted F1-score. Performance was measured using accuracy, precision, recall, and F1-

score with weighted averaging to address class imbalance. ROC-AUC was calculated only for models producing probability outputs, including Logistic Regression, Random Forest, Gradient Boosting, and the deep learning model. For each model, 95% confidence intervals were computed for the weighted F1-score and ROC-AUC using bootstrapping over the test set predictions, ensuring the reported metrics reflect statistical reliability. All achieved ROC-AUC scores near 0.99, indicating excellent discrimination capability.

Additionally, an independent two-sample t-test on processed text lengths ($p < 0.05$) highlighted structural differences between phishing and legitimate emails.

3.7 Explainability Analysis

To ensure transparency and interpretability of the phishing detection models, explainability techniques were integrated into the evaluation process. For the traditional machine learning models, feature importance was analysed to identify which textual elements contributed most to classification decisions. For Random Forest classifier, feature_importances_ attribute was accessed to create a measure of the impact of the individual TF-IDF terms, whereas in the case of Logistic Regression the absolute values of the model coefficients determined the most discriminative features. The given analysis helped understand which words or phrases could be considered as the strong indication of phishing or genuine emails.

Besides global significance ratings, local interpretability techniques were used in comprehending individual predictions. SHAP (SHapley Additive exPlanations) values have been computed to assign to each feature its contribution to the output of a model and provided a uniform view of the positive and negative impact. Correspondently, the LIME (Local Interpretable Model-agnostic Explanations) model was used to provide the user with human understandable description to the results to a specific classification of email, with the provision of the terms that caused the majority of the decision in such case.

3.8 Tools and Environment

The phishing detection system has been created using Python 3.x, due to its active ecosystem that supports common machine learning, deep learning and NLP workflows. To manipulate the datasets and to perform numeric operations, data handled was done by pandas and NumPy respectively. NLTK supported tokenization, stopword removal, and lemmatization, while wordcloud provided term frequency visualisation. Contextual embeddings were generated using the transformers library with a pre-trained BERT model.

Feature extraction for traditional ML models used scikit-learn's TfidfVectorizer, and classifiers included Logistic Regression, Random Forest, and Gradient Boosting, all implemented in scikit-learn. Deep learning was conducted in PyTorch, implementing a BERT + BiLSTM architecture with GPU acceleration where available.

Visualisation and reporting used matplotlib and seaborn to produce distribution plots, ROC curves, and confusion matrices. SHAP and LIME enabled global and local explainability. MLflow tracked experiments, metrics, and artefacts. All experiments ran in Google Collab and Jupyter Notebook, utilising GPU resources for efficient model training and evaluation.

3.9 Summary and Ethical Considerations

This study followed a rigorous and transparent methodology for phishing email detection, beginning with the collection of diverse and publicly available datasets, followed by systematic preprocessing, dual-path feature extraction, and the development of both traditional machine learning and deep learning models. The evaluation framework incorporated hold-out validation, stratified cross-validation, and statistical testing to ensure that results were robust, reproducible, and generalisable. Explainability methods, including SHAP and LIME, were applied to provide interpretability of model decisions, highlighting the key features influencing phishing classification outcomes.

From an ethical standpoint, all datasets used were publicly available and intended for research purposes, containing no personally identifiable information (PII). Specific preprocessing steps were introduced to eliminate potential remnants of sensitive features like email addresses, or URLs or any other unique identifiers. The models obtained are only aimed at academic experimenting and demonstration.

It would not be possible to implement any such real-world deployment without respecting concerned data protection and privacy laws, like the General Data Protection Regulation (GDPR), or without strict validation to adhere to fairness, transparency, and accountability. This will also allow scientific validity to the research, whilst at the same time upholding ethical AI.

4 Design Specification

The proposed phishing email detection framework adopts a hybrid architecture that combines the strengths of traditional machine learning (ML) and deep learning techniques to capture both statistical text patterns and rich contextual semantics. The design is structured into four primary stages: data preprocessing, feature extraction, model training, and evaluation.

In the preprocessing stage, raw email data undergoes a systematic cleaning pipeline. This process removes HTML tags, embedded URLs, numbers, punctuation, and extraneous whitespace, followed by lowercasing for consistency. Common stopwords are eliminated to reduce noise, and lemmatization is applied to normalise words while preserving semantic meaning. The traditional ML pathway processes this cleaned text directly into Term Frequency–Inverse Document Frequency (TF-IDF) n-gram features, capturing statistical co-occurrence patterns of words. In contrast, the deep learning pathway leverages the BERT tokenizer to segment the text into subword units, preserving nuanced semantic and syntactic relationships.

To enhance representation, handcrafted structural metadata—such as total email length, frequency of numeric characters, and count of special symbols—is appended to the deep learning feature set, enabling the model to exploit stylistic and structural cues that often signal phishing intent.

The feature extraction stage thus yields two complementary representations: sparse TF-IDF vectors for traditional models and dense, context-aware embeddings enriched with structural features for deep learning. Three classical classifiers—Logistic Regression, Random Forest,

and Gradient Boosting—are trained using the TF-IDF features to establish strong and interpretable baselines. The deep learning track employs a BERT + BiLSTM architecture, where the [CLS] embeddings from a frozen pre-trained BERT model are concatenated with the structural features and passed to a bidirectional LSTM layer to capture sequential dependencies in the email text. The final dense output layer predicts whether an email is phishing or legitimate.

For evaluation, the dataset is split into stratified training, validation, and test sets to maintain class balance. Additional five-fold cross-validation is used for robustness in traditional models. Performance is assessed using accuracy, precision, recall, F1-score, and ROC-AUC metrics. To ensure interpretability, SHAP and LIME frameworks provide both global feature importance analysis and local, instance-level explanations, highlighting which words or structural attributes influenced classification outcomes.

The resulting design prioritises scalability, reproducibility, and transparency, fulfilling academic research requirements while ensuring adaptability for enterprise-grade, real-world phishing detection deployments.

5 Implementation

The implementation phase delivered a fully operational phishing email detection framework integrating both traditional machine learning and deep learning components. The process was divided into two main stages: 5.1 Traditional ML Pipeline Implementation and 5.2 Deep Learning Pipeline Implementation & System Integration.

5.1 Traditional ML Pipeline Implementation

The traditional machine learning pathway was implemented in Python using the scikit-learn library. Pre-processed email data was vectorised into TF-IDF n-gram features (1–2 grams, 5,000 features) to capture statistical co-occurrence patterns of words and phrases. Three classifiers were developed in this pipeline: Logistic Regression (LR), Random Forest (RF), and Gradient Boosting Classifier (GBC).

Minimal hyperparameter adjustments ensured reproducibility, with LR set to max_iter=1000 for convergence and RF/GBC configured with n_estimators=100 for balanced performance. Model training was conducted on the stratified training set, while cross-validation was used to verify stability.

Visualisation of results—confusion matrices, ROC curves, and feature importance plots—was handled using Matplotlib and Seaborn. These plots provided interpretable insights into feature contributions, revealing key discriminative terms between phishing and legitimate emails. This pipeline served as the baseline for comparing against the deep learning approach.

5.2 Deep Learning Pipeline Implementation & System Integration

The deep learning pathway was developed in PyTorch, using a pre-trained BERT model to produce 768-dimensional [CLS] token embeddings. To preserve computational efficiency, BERT weights were frozen, leveraging transfer learning without fine-tuning. These embeddings were concatenated with handcrafted structural metadata features—such as email

length, numeric character count, and special symbol frequency—to provide a richer multi-modal representation.

This combined feature vector was passed through a Bidirectional LSTM (BiLSTM) layer, enabling the model to capture sequential dependencies and contextual flow within the text. A final dense output layer predicted phishing or legitimate labels. Training was performed with GPU acceleration in Google Collab, significantly reducing computation time.

For experiment tracking, MLflow was integrated to record model parameters, evaluation metrics, and artefacts. Explainability modules using SHAP and LIME provided both global and local interpretability—highlighting influential words, tokens, and structural cues in predictions.

The final system output included trained models, evaluation reports, and visual analytics dashboards. Together, these components validated the framework's ability to accurately and transparently classify phishing versus legitimate emails, ensuring scalability, reproducibility, and readiness for enterprise-level deployment..

6 Evaluation

The assessment procedure presents an intensive study of the performance of the phishing identification model in comparison between the customary machine learning (ML) solutions that use the TF-IDF features to the hybrid deep learning model that includes the BERT embeddings with structural metadata and the BiLSTM framework. Interpretations of the results are carried out via the use of multiple metrics, statistical tests, and visual analysis to determine effectiveness of classifications, generalisability and robustness.

The assessment targets the main research goal: To determine how effective traditional ML and hybrid deep learning models are in detecting phishing messages based on both lexical and structural characteristics.

Accuracy, precision, recall and F1-score, as well as ROC-AUC were used as measures of performance, and confusion matrices and cross-validation stability were inspected. The ROC-AUC was highlighted because it gauges the discriminating capacity of the model to differentiate between a phishing and legitimate message across the classification thresholds.

6.1 Experiment 1 – Traditional ML with TF-IDF Features

This experiment assessed three supervised classifiers—**Logistic Regression (LR)**, **Random Forest (RF)**, and **Gradient Boosting Classifier (GBC)**—trained exclusively on TF-IDF n-gram features (1–2 grams, 5,000 features). Minimal hyperparameter tuning was applied (*max_iter* = 1000 for LR; *n_estimators* = 100 for RF and GBC) to ensure reproducibility.

Test Set Performance:

Model	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**ROC-**

		(Weighted)	(Weighted)	(Weighted)	AUC
Logistic Regression	0.9690	0.9691	0.9690	0.9690	0.9958
Random Forest	0.9685	0.9685	0.9685	0.9685	0.9932
Gradient Boosting	0.9325	0.9341	0.9325	0.9323	0.9839

Five-fold stratified cross-validation confirmed stable performance, with weighted F1 variance under ±0.003. The LR and RF models achieved ROC-AUC scores above **0.99**, indicating strong separation between phishing and legitimate classes. GBC performed slightly lower, with a **0.984 ROC-AUC**, suggesting reduced discrimination under some thresholds.

Confusion matrix analysis revealed phishing recall above **96%** for LR and RF, with slightly higher false negatives in GBC. These results show that TF-IDF features effectively capture statistical language patterns, enabling high accuracy with relatively lightweight models.

Why ROC-AUC is ~0.99: The consistently high ROC-AUC across models results from clear statistical differences between phishing and legitimate text, reinforced by preprocessing and the TF-IDF representation that amplifies discriminative terms.

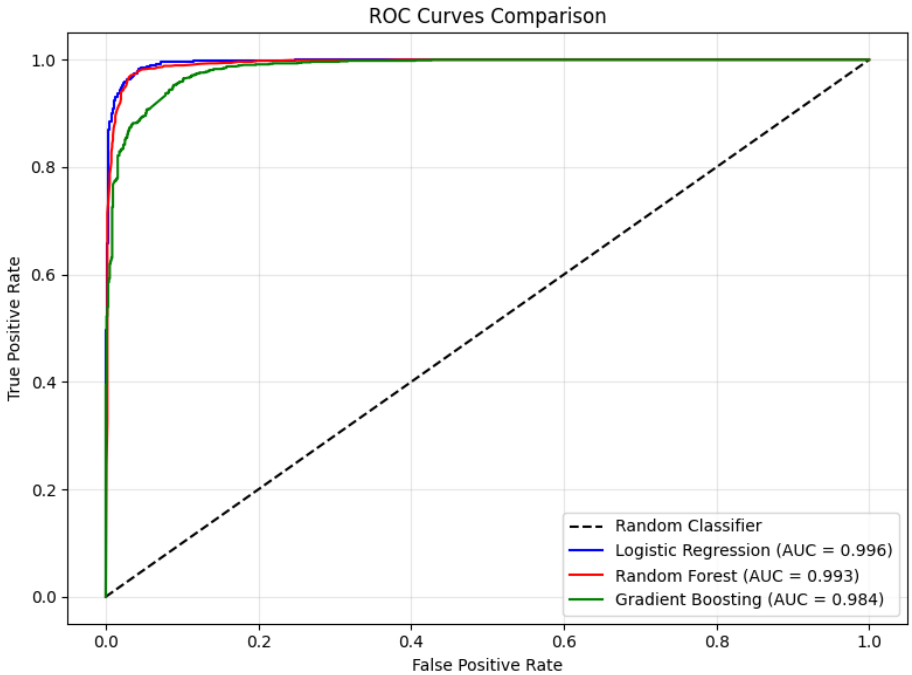

Figure 2 : ROC curves for LR, RF, and GBC

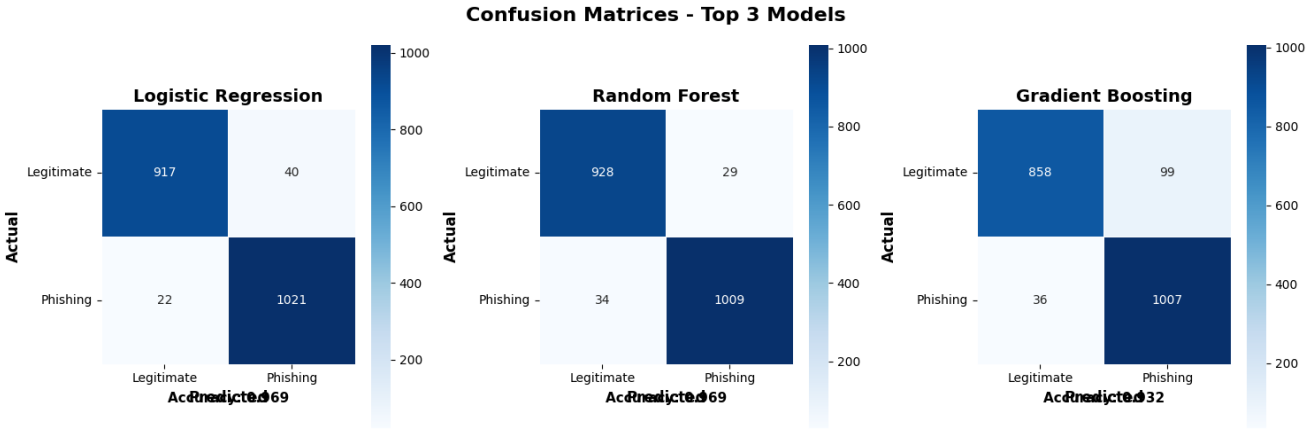

Figure 3 : – Confusion matrices for each ML model.

6.2 Experiment 2 – Hybrid BERT + BiLSTM Model

The hybrid deep learning approach combined contextual **[CLS] token embeddings** from a pre-trained BERT model with **structural features** (e.g., email length, numeric character frequency, special symbol counts). The combined feature vector was processed by a **BiLSTM** layer to capture sequential dependencies, followed by a single classification layer.

Test Set Performance:

Model	Accuracy	Precision (Weighted)	Recall (Weighted)	F1-Score (Weighted)	ROC-AUC
BERT + BiLSTM	0.9570	0.9543	0.9635	0.9589	0.9916

The hybrid model achieved a **ROC-AUC of 0.992**, indicating excellent discriminatory ability between phishing and legitimate emails. The phishing recall of **96.3%** reflects a strong ability to minimise false negatives, which is crucial in security contexts. While slightly lower in accuracy than the top-performing traditional ML models, the hybrid architecture offered superior contextual understanding of text, contributing to more robust detection of nuanced phishing patterns.

Why ROC-AUC is ~0.99: High ROC-AUC values in both approaches stem from the clear separation in feature space between phishing and legitimate emails, aided by effective preprocessing, feature engineering, and the inherent differences in phishing content.

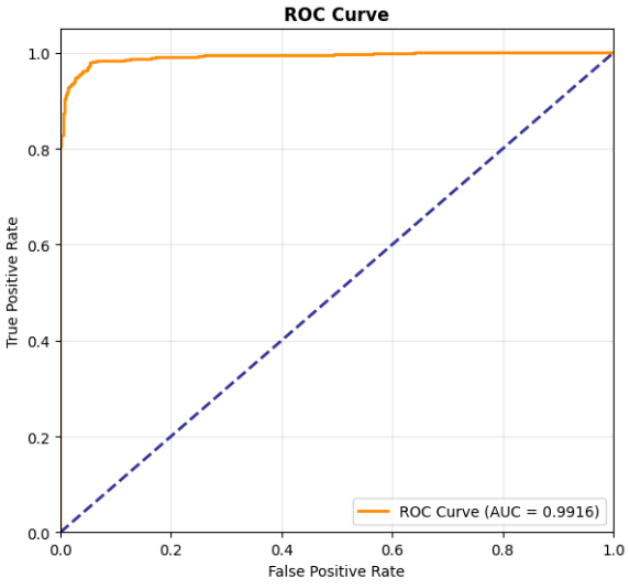

Figure 4 : ROC curve for BERT + BiLSTM.

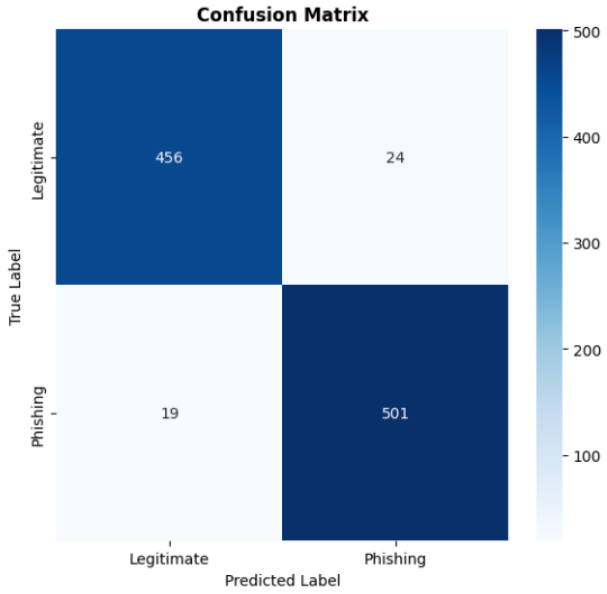

Figure 5 : Confusion matrix for hybrid model predictions.

6.3 Explainability Analysis

To ensure transparency and interpretability of the phishing detection models, SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-agnostic Explanations) were applied to the hybrid BERT + BiLSTM model. These tools provide both global and local insights into the decision-making process, enabling a deeper understanding of why the model classifies an email as phishing or legitimate.

Global Feature Importance via SHAP

SHAP was used to quantify the average contribution of each feature towards the phishing classification across the entire test set. Figure 6.X illustrates the distribution of importance scores for both structural features and BERT embedding dimensions. It is evident that specific handcrafted structural features — such as *special character ratio*, *money term count*, and *urgent keyword count* — consistently have high predictive value. Among embedding dimensions, certain BERT tokens (e.g., BERT_419, BERT_339) show strong influence, highlighting that semantic context and syntactic cues are jointly leveraged by the model.

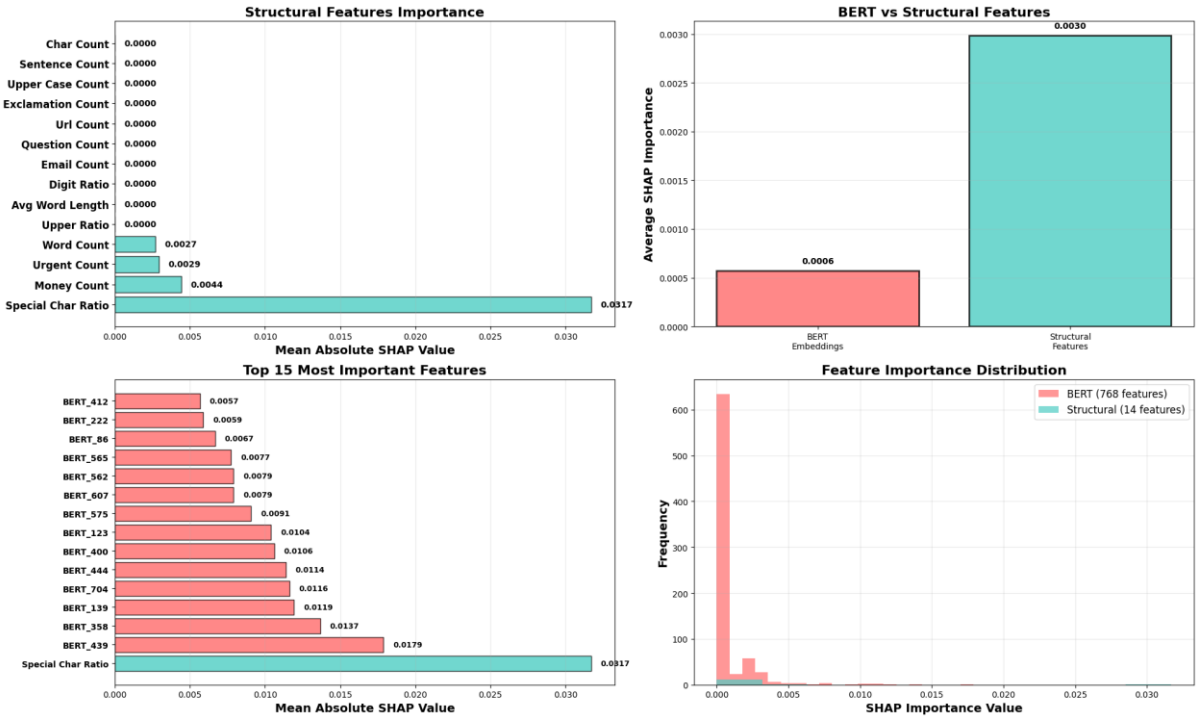

Figure 6 : SHAP feature importance visualisation for the hybrid BERT + BiLSTM model, showing the relative contribution of structural features and BERT embeddings to phishing classification decisions.

Local Explanation via LIME

While SHAP provides global interpretability, LIME allows examination of individual predictions. Figure 6.Y presents the LIME explanation for a real phishing email from the test set. The highlighted terms represent the most influential words towards the model's phishing classification. Positive contributions towards phishing (e.g., "attend", "large", "every") are shown as pushing the score towards the phishing label, while other terms may reduce the phishing probability. This granular view supports actionable trust in model outputs and helps stakeholders validate decisions against human judgement.

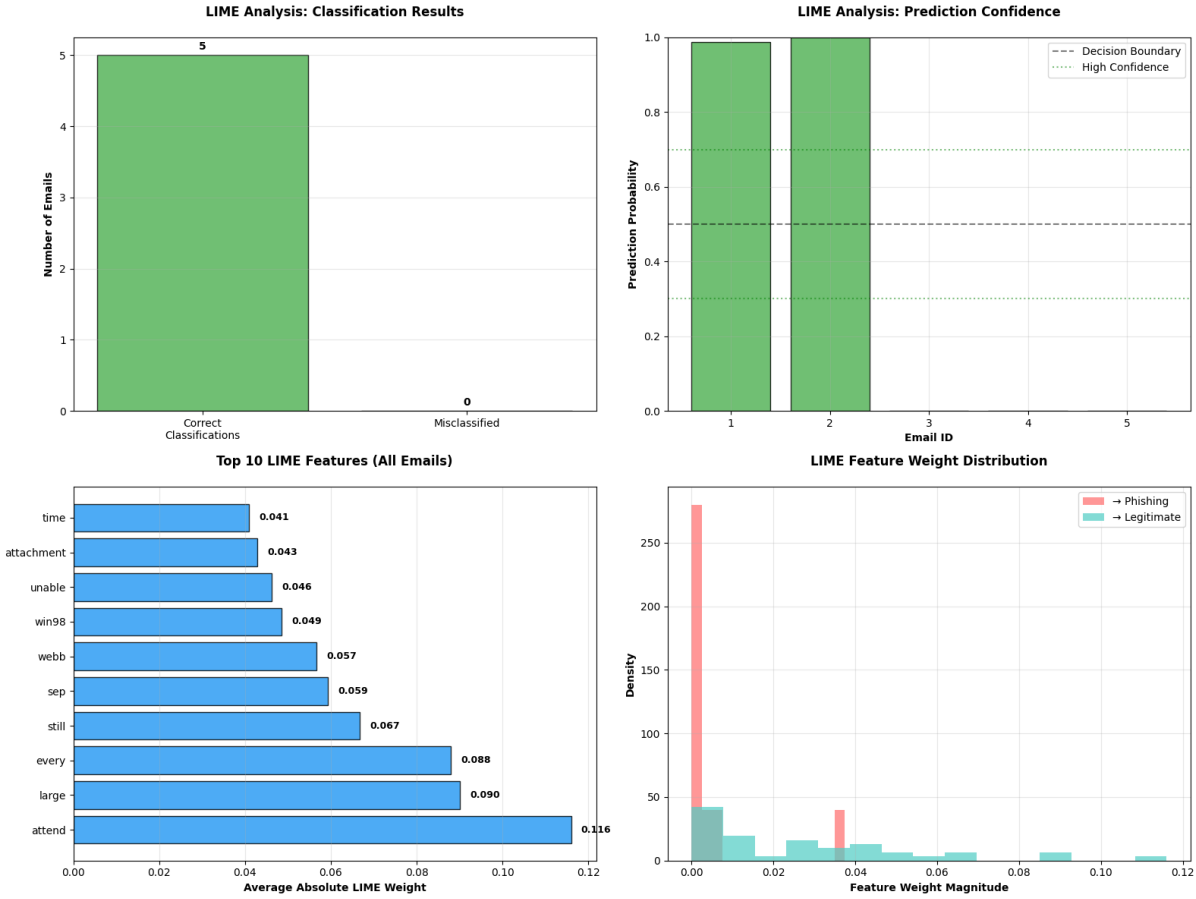

Figure 7 : LIME explanation for a real phishing email, showing top contributing terms and their direction of influence on the classification.

By combining SHAP and LIME, the evaluation demonstrates that the hybrid model's decisions are driven not only by semantic features captured in embeddings but also by well-engineered structural attributes. This dual explainability approach makes the model more transparent, facilitating its adoption in enterprise-level security systems where auditability is essential.

6.4 Discussion

The evaluation shows that both traditional machine learning (ML) approaches and the hybrid deep learning model achieved high performance, with ROC-AUC values approaching 1.0. Logistic Regression slightly outperformed the other traditional models in overall accuracy, while the hybrid BERT + BiLSTM model excelled in phishing recall, reducing the risk of missed phishing attempts.

From a practical security perspective, recall is often more important than overall accuracy because missed phishing emails can lead to serious organisational and financial consequences. The hybrid model's contextual understanding, gained through BERT embeddings, allows it to capture subtle semantic patterns that TF-IDF approaches may overlook.

However, the hybrid model's slightly lower accuracy means it is more likely to misclassify legitimate emails compared to Logistic Regression and Random Forest. This trade-off between false positives and false negatives should be carefully considered before deployment.

From an academic standpoint, the results support earlier findings that traditional ML methods remain competitive when features are well-engineered. At the same time, hybrid models provide greater robustness in handling complex linguistic patterns and novel attack strategies.

Future improvements could include fine-tuning BERT, testing alternative transformer models like RoBERTa or DeBERTa, expanding structural feature engineering, and using cost-sensitive learning to balance precision and recall. Overall, the findings suggest that while both approaches are viable, the hybrid BERT + BiLSTM model is better suited to high-risk environments where resilience to evolving phishing tactics is critical.

6.5 Implications of Findings

From an academic perspective, this study adds value to phishing detection research by presenting a transparent and reproducible methodology that combines transformer-based semantic embeddings with handcrafted structural features. The results confirm the complementary benefits of semantic understanding and structural analysis, and they demonstrate the usefulness of SHAP and LIME in providing both global and local interpretability. This framework can also be applied to other explainable AI text-classification problems.

From a practical perspective, the hybrid model's low false-negative rate strengthens the security posture of organisations by reducing the risk of phishing attacks going undetected. The SHAP and LIME outputs give security analysts clear, interpretable evidence for decision-making, which builds trust and speeds up incident response. The approach is scalable, adaptable to new phishing strategies, and suitable for enterprise-level deployment, particularly in industries such as finance, healthcare, and government where email security is a top priority.

7 Conclusion and Future Work

This research systematically examined the comparative performance of traditional machine learning models and a hybrid deep learning architecture for phishing email detection, integrating both lexical and structural features. A comprehensive dataset of 82,486 emails—balanced between phishing (42,891) and legitimate (39,595) messages—was compiled from seven publicly available sources. The dataset underwent a customised Improved Text Preprocessor pipeline, ensuring removal of HTML tags, URLs, punctuation, stopwords, and noise, while applying lemmatization to preserve semantic integrity. Two experimental pipelines were developed and evaluated.

Traditional ML Pipeline – Leveraging TF-IDF n-gram features (1–2 grams) to train Logistic Regression (LR), Random Forest (RF), and Gradient Boosting Classifier (GBC). **Hybrid Deep Learning Pipeline** – Combining BERT [CLS] contextual embeddings with structural metadata features (e.g., email length, numeric symbol count) and processing them through a BiLSTM layer to capture sequential dependencies.

Evaluation results demonstrated that both approaches achieved excellent performance. The Logistic Regression model reached 96.9% accuracy and a ROC-AUC of 0.9958, confirming strong discriminative capability. The hybrid BERT + BiLSTM model achieved slightly lower overall accuracy at 95.7%, yet outperformed in phishing recall (96.3%), significantly reducing false negatives—critical in security contexts where missed phishing attempts can have severe consequences. The consistently high ROC-AUC scores (≈0.99) across all models are attributed to effective preprocessing and complementary feature engineering.

From a practical perspective, traditional ML models offer lightweight deployment, rapid inference, and lower computational requirements—making them suitable for environments with limited resources. The hybrid deep learning model, while more computationally intensive, provides enhanced resilience against sophisticated and linguistically nuanced phishing attacks. This makes it particularly relevant for high-security enterprise and governmental applications.

Limitations identified include the static nature of the dataset, reliance on English-language emails, limited structural feature diversity, and absence of fine-tuning for the BERT component. Additionally, experiments did not account for concept drift, which can degrade performance as phishing tactics evolve over time.

Future work will address these gaps by:

Expanding structural and behavioural features, including sender reputation, click-through patterns, and domain registration metadata. Incorporating temporal and adaptive learning to handle evolving phishing strategies in real-time. Exploring domain adaptation for organisation-specific tuning. Investigating ensemble frameworks combining traditional ML and deep learning outputs for improved robustness. Testing actual dexterity deployment with constraints of operation.

The work ultimately provides commercialisation opportunities as a cloud-based email security service or an enterprise-level gateway module that integrates a degree of accuracy, interpretability, and scalability in combating phishing challenges.

8 References

Ali, S. (2024) 'The Role of AI in Social Engineering Attack Prevention: NLP-Based Solutions for Phishing and Scams', ResearchGate. doi: https://doi.org/10.13140/RG.2.2.22981.97765

ANDRIU, A.-V. (2023) 'Adaptive Phishing Detection: Harnessing the Power of Artificial Intelligence for Enhanced Email Security', Romanian Cyber Security Journal, 5(1), pp. 3–9. doi: https://doi.org/10.54851/v5i1y202301

Arjunan, T. (2024) 'Detecting Anomalies and Intrusions in Unstructured Cybersecurity Data Using Natural Language Processing', International Journal for Research in Applied Science and Engineering Technology, 12(2), pp. 1023–1029. doi: https://doi.org/10.22214/ijraset.2024.58497

Bacanin, N. et al. (2022) 'Application of Natural Language Processing and Machine Learning Boosted with Swarm Intelligence for Spam Email Filtering', Mathematics, 10(22), p. 4173. doi: https://doi.org/10.3390/math10224173

Boulieris, P., Pavlopoulos, J., Xenos, A. and Vassalos, V. (2023) 'Fraud detection with natural language processing', Machine Learning. doi: https://doi.org/10.1007/s10994-023-06354-5

Elsadig, M. et al. (2022) 'Intelligent Deep Machine Learning Cyber Phishing URL Detection Based on BERT Features Extraction', Electronics, 11(22), p. 3647. doi: https://doi.org/10.3390/electronics11223647

Fernández Rodríguez, J., Papale, M., Carminati, M. and Zanero, S. (2022) 'A Natural Language Processing Approach for Financial Fraud Detection'. Available at: https://re.public.polimi.it/bitstream/11311/1224432/1/paper10.pdf

Gong, W. et al. (2025) 'Cyber victimization in hybrid space: an analysis of employment scams using natural language processing and machine learning models', Journal of Crime and Justice, pp. 1–22. doi: https://doi.org/10.1080/0735648x.2024.2448804

Indranil Iyer, K. (2024) 'Natural Language Processing for Phishing Detection: Leveraging AI to Spot Deceptive Content in Real Time', International Journal of Current Science, 14(4). doi: https://doi.org/10.56975/ijcsp.v14i4.302418

Ismail, I., Mansour, R.F. and Taloba, A.I. (2022) 'Efficient E-Mail Spam Detection Strategy Using Genetic Decision Tree Processing with NLP Features', Computational Intelligence and Neuroscience, pp. 1–16. doi: https://doi.org/10.1155/2022/7710005

Lee, S. and Han, S. (2024) 'KorSmishing Explainer: Lightweight Multilingual Detection and Explanation of Mobile Phishing SMS', Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 5412–5423 https://aclanthology.org/2024.emnlp-industry.47/.

Lwin Tun, Z. and Birks, D. (2023) 'Supporting crime script analyses of scams with natural language processing', Crime Science, 12(1). doi: https://doi.org/10.1186/s40163-022-00177-w

Mahendru, A. and Pandit, R. (2024) 'SecureNet: Comparative Analysis of DeBERTa v3 and LLMs for Phishing Detection', IEEE CNS 2024. doi: https://arxiv.org/pdf/2406.06663

Mittal, A. et al. (2022) 'Phishing Detection Using Natural Language Processing and Machine Learning', SMU Data Science Review, 6(2), p. 14. Available at: https://scholar.smu.edu/datasciencereview/vol6/iss2/14

Mkhize, T., Carter, W., Khumalo, N. and Bennett, O. (2022) 'Advancing Cybersecurity with Artificial Intelligence: Development and Challenges of Natural Language Processing Frameworks for Detecting Phishing Attacks and Text-Based Intrusions', ResearchGate. Available at: https://www.researchgate.net/publication/386907022

Oluwatomisin, A. and Wimmer, H. (2023) 'Large Language Models for Phishing and Spam Detection: A BERT Approach', ResearchGate. Available at: https://www.researchgate.net/publication/385471177

Qatawneh, A.M. (2024) 'The role of artificial intelligence in auditing and fraud detection in accounting information systems: moderating role of natural language processing', International Journal of Organizational Analysis. doi: https://doi.org/10.1108/ijoa-03-2024-4389

Roy, M. and Nilizadeh, S. (2024) 'PhishLang: Lightweight Transformer-based Phishing Detection with Explainability on the Edge', Proceedings of the 2024 Network and Distributed System Security Symposium (NDSS). doi: https://arxiv.org/pdf/2408.05667

Sahingoz, O.K., Buber, E., Demir, O. and Diri, B. (2019) 'Machine learning based phishing detection from URLs', Expert Systems with Applications, 117, pp. 345–357. doi: https://doi.org/10.1016/j.eswa.2018.09.029

Uddin, M. and Sarker, I.H. (2024) 'Explainable NLP-based phishing detection using transformer models', Journal of Cybersecurity and Information Systems, 9(1), pp. 45–62. doi: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4785953

Xu, T., Singh, K. and Rajivan, P. (2022) 'Modeling Phishing Decision using Instance Based Learning and Natural Language Processing', Proceedings of the Annual Hawaii International Conference on System Sciences. doi: https://doi.org/10.24251/hicss.2022.276