National College of Ireland

Enchancing Phishing Email Detection Through Multi Modal Deep Learning

MSc Research Project
Msc in Data Analytics

Srihitha Masireddy
Student ID: x23325275

School of Computing
National College of Ireland

Supervisor: Vladimir Milosavljevic

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Srihitha Masreddy
Student ID:	X23325275
Programme:	Msc in Data Analytics **Year:** 2025
Module:	Research Practicum
Supervisor:	Vladimir Milosavljevic
Submission Due Date:	15/09/25
Project Title:	Enchancing Phishing Email Detection Through Multi Modal Deep Learning
Word Count:	8533 **Page Count**: 27

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: Masireddy Srihitha

Date: 14-09-25

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):	

Enchancing Phishing Email Detection Through Multi Modal Deep Learning

Srihitha Masireddy
x23325275

Abstract

Phishing emails continue to pose significant cybersecurity threats, with sophisticated attacks evading traditional detection methods through careful manipulation of both content and structure. Current approaches typically focus on either semantic analysis or structural patterns in isolation, missing the complementary signals that could enhance detection accuracy. This research presents a novel hybrid architecture that integrates XLNet's advanced language understanding capabilities with Graph Neural Networks' structural pattern recognition, unified through cross-modal contrastive learning.

The proposed system processes emails through parallel pathways, extracting 768-dimensional semantic features via fine-tuned XLNet and 128-dimensional structural features through heterogeneous graph analysis. A contrastive learning framework aligns these representations in a shared 256-dimensional space, enabling effective integration of complementary information. The implementation evaluates four fusion strategies, with attention-based combination achieving optimal performance at 98.71% accuracy.

Comprehensive evaluation against traditional machine learning baselines and ablation studies validates the architecture's effectiveness. While the improvement over XLNet alone is modest at 0.69 percentage points, the research demonstrates successful integration of multi-modal features for phishing detection. The system maintains sub-500ms processing time per email, meeting practical deployment requirements.

This work contributes to cybersecurity by demonstrating how cross-modal learning can unite semantic and structural analysis, providing a foundation for future research in multi-modal phishing detection. The modular architecture and comprehensive evaluation establish benchmarks for hybrid approaches in email security.

1 Introduction

Email phishing is one of the most persistent and constantly evolving types of cybersecurity threats that people and organisations worldwide must deal with. By posing as trustworthy organisations, these advanced social engineering attacks trick victims into disclosing personal information or taking on dangerous tasks. Recent research indicates that the threat's scope is growing and that phishing attacks are employing contextual awareness and social engineering in ever-more-complex ways (Altwaijry et al., 2024).The Anti-Phishing Working Group noted a dramatic increase in phishing emails, from 44,008 in early 2020 to 260,642 in July 2021, due to the threat landscape's exponential growth (Muneer et al., 2021).

Phishing attacks have an impact on organisations and the economy that extends beyond just financial gains. According to Mughaid et al. (2022), targeted attacks have increased by 9.8% in the banking sector, causing significant financial losses for multinational corporations.
Phishing attacks compromise the integrity of the data, blur customer confidence, and are at times the precursor to more extensive security breaches. Although more than a decade has been invested in specific research on detecting phishing attacks, existing defense measures cannot compete with the constantly changing sophistication level of the attacks (Thakur et al., 2023).

Traditional phishing detection approaches have primarily focused on either semantic analysis of email content or structural pattern recognition, but rarely both simultaneously. using natural language processing methods are able to detect suspicious linguistic patterns and contextual anomalies (Salloum et al., 2022). Structure-based methods, however, examine link patterns, email metadata, and communicational networks to detect anomalies (Pan et al., 2022). More sophisticated forms of phishing, however, employ methods that would remain unseen by single-modal detection methods, hence the need for more comprehensive methods

This work fills an important void in current phishing identification methods by introducing an original hybrid framework combining sophisticated semantic awareness with structural pattern identification. Motivation is driven by the insight that even though transformer-based algorithms such as XLNet exhibit remarkable capability in identifying semantic subtleties based on permutation-based language modeling (Yang et al., 2019), they potentially overlook structural signals identifiable by Graph Neural Networks (GNNs). In like manner, GNN-oriented methods that excel at examining structural relationships and communication patterns in emails potentially neglect subtle semantic manipulations used by advanced attackers.

The primary **research question** driving this investigation is: *To what extent can a hybrid architecture combining XLNet and Graph Neural Networks enhance the detection of sophisticated phishing emails compared to current state-of-the-art methods?*

The research objectives are:

1. To develop a hybrid architecture that effectively integrates XLNet's semantic analysis capabilities with GNN's structural pattern recognition
2. To implement a cross-modal contrastive learning framework that aligns semantic and structural feature representations
3. To assess the performance gain from this integration in comparison with separate methods and current baselines.

The success criteria for these objectives include achieving statistically significant improvement in detection accuracy over baseline methods, demonstrating the complementary nature of semantic and structural features, and validating the approach on multiple benchmark datasets.

The methodology employs a systematic four-phase approach. First, XLNet is fine-tuned for semantic feature extraction from email content. Second, design a Graph Neural Network structure in order to examine structural characteristics in the email elements.Third, a cross-modal contrastive learning framework aligns the semantic and structural representations. Finally, multiple fusion strategies are evaluated to determine the optimal integration approach. This methodology addresses the identified gap by creating a unified framework that leverages the strengths of both semantic and structural analysis.

This report is organized as follows: Section 2 gives a thorough literature review exploring existing phishing detection methods, transformer networks, graph neural networks, and contrastive learning methods. Section 3 explains the research methodology involving data preparation, model structures, and evaluation procedures. Section 4 gives the design requirements for the hybrid architecture. Section 5 provides the implementation details. Section 6 shows the evaluation results with statistical analysis. Section 7 explains the findings, shortcomings, and implications. Section 8 concludes the paper with a brief overview of the contributions and the future research paths.

2 Related Work

The research landscape on phishing detection has changed dramatically from rule-based systems to state-of-the-art machine learning methods. This literature review critically analyzes existing approaches, drawing out the contributions and shortcomings and laying the ground for the suggested hybrid architecture.

2.1 Evolution of Phishing Detection Approaches

Initial phishing detection work was based on detecting rule-based patterns and surface-level characteristics. Muneer et al. (2021) presented a wide-ranging survey finding that the classical methods utilized blacklists most prominently, followed by the usage of the signatures and manual feature extraction methods. The main flaw in the techniques is still their inability to be adjusted to counter new phishing tactics, even though they provided initial defence strategies. Rule-based systems' static nature means they are vulnerable to zero-day phishing attacks that use novel designs.

An important turning point in detection performance was the move to machine learning. Salloum et al.'s (2022) systematic review of 100 research papers on the detection of phishing emails using natural language processing techniques revealed that machine learning algorithms perform noticeably better than conventional techniques. The most well-known algorithms were identified in the review as SVM, Random Forest, and Naive Bayes. However, the study showed that these approaches had a drawback in that they required a great deal of feature engineering and domain expertise, which could cause them to miss subtleties detected by automated feature learning.

2.2 Deep Learning Paradigm in Phishing Detection

The application of deep learning techniques represents a paradigm shift in phishing detection methodology. A systematic review specifically focused on deep learning techniques was published by Thakur et al. in 2023. They examined various architectures, including Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM) networks, and

Gated Recurrent Units (GRUs). According to the analysis carried out by the aforementioned researchers, the deep learning models demonstrated superior performance based on automatic feature extraction. These CNN-based techniques showed great promise in identifying hierarchical features within the email content.

According to the experiments, 1D-CNN models with bidirectional GRU layers outperformed other models in email classification tasks. The strength of the methods is that they use recurrent operations to learn long-range dependencies and convolutional operations to learn local patterns. However, Thakur et al. (2023) identified three significant drawbacks: the requirement for unbalanced datasets, the vulnerability to adversarial examples, and the lack of interpretability during the decision-making process.

Altwaijry et al. (2024) extended this research by performing a comparative evaluation of deep learning models for phishing email detection in particular. Their experiments on the transformer-based models, especially BERT variants, showed F1 metrics up to 98.66%. The merits of the research are the thorough evaluation on more than one architecture and more than one dataset. However, the research showed that state-of-the-art models even struggled on complex phishing attempts where they utilized context-aware social engaging methods, implying the requirement for methods where more aspects of the email communication than the content are takeninto context.

2.3 Graph-Based Structural Analysis

A different approach in phishing detection was provided by graph-based approaches considering structural relations within communications in the form of emails. Pan et al. (2022) proposed the Semantic Graph Neural Network (SGNN), which recast the task of classifying emails as graph classification. The emails were represented as graphs where the nodes corresponded to different parts of the emails and edges represented the relations between them. The SGNN surpassed current text-centered classifiers by preserving structural content often lost in sequential processing.

The graph-based methods' strength comes from the fact that graph models can ingest complex structural relations involving the elements in the emails including sender details, inlined links as well as attachment distribution. However, Pan et al. (2022) acknowledged that their approach primarily focused on structural features, potentially missing semantic nuances that text-based models excel at capturing. When there are well-written phishing emails with real-looking structures but well-developed texts that conceal malicious intent, this limitation becomes more apparent.

2.4 Transformer Models and Contextual Understanding

Natural language processing was revolutionised by the creation of transformer architectures, which also directly impacted phishing detection. XLNet, a generalised autoregressive pretraining technique developed by Yang et al. (2019), represents bidirectional contexts through permutation-based training. While BERT represents the learning approach through the operation of masked language modelling, XLNet retains the strength of autoregressive

modelling while learning from the full set of permutations on the input sequence.Such a property renders XLNet especially effective in phishing detection cases where contextual comprehension from various angles may expose subtle manipulation tactics.

Outstandinggains in capturing semantic nuances is highlighted by the application of transformer models to phishing detection. Fares et al. (2024) compared the performance of different transformer architectures on detecting phishing in emails and discovered that large corpora-pre-trained models could successfully transfer learned embeddings and recognize attempts at phishing. The main advantage of a transformer-based method is the attention mechanism, which can pick up on non-position-dependent relevant pieces of email content. Models of this sort concentrate almost exclusively on the content in the form of text, though, and may ignore structural cues that could betray attempts at phishing.

2.5 Limitations of Existing Approaches

Despite making considerable progress, existing phishing detection approaches still have basic shortcomings when challenged by state-of-the-art attacks. Single-modal approaches, whether they concentrate on semantic or structural features, are susceptible to well-crafted phishing emails that imitate authentic patterns in one dimension while taking advantage of weaknesses in another, according to Ramprasath et al. (2023).Their study demonstrated that some models' 99.6% benchmark dataset performance did not translate into real-world success when faced with adaptive phishing tactics.

The literature consistently identifies several critical gaps. First, the lack of approaches that simultaneously leverage both semantic understanding and structural analysis limits detection capabilities. Secondly, existing fusion methods for disparate types of features prefer naive concatenation or ensembling methods lacking consideration for inter-modal relationships. Finally, the absence of methods for aligning the representations from the disparate modalities results in suboptimal unification of the complementary information.

2.6 Synthesis and Research Gap

Extensive review of the current literature indicates a clear progression from rule-based systems all the way through machine learning up until complex deep learning methods.While transformer models like XLNet excel at semantic analysis and GNNs demonstrate effectiveness in structural pattern recognition, no existing approach successfully integrates these complementary capabilities through principled representation alignment. The surveyed studies consistently achieve high accuracy on individual modalities but fail to address the fundamental challenge of sophisticated phishing attacks that carefully balance semantic legitimacy with structural deception.

This difference appears most conspicuously in the discussion regarding limitation constraints from studies.

While structure-oriented methods ignore textual manipulation, semantic-oriented methods ignore structural irregularities. One of the main shortcomings in previous research is the lack of learning mechanisms that can cross-modally explain and summarise these heterogeneous feature

representations. Moreover, previous research on multimodal fusion is not advanced enough to maintain complex connections between structural and semantic features.

By using cross-modal contrastive learning to align semantic and structural representations, the suggested hybrid architecture overcomes these drawbacks and makes it possible to integrate complementary features in an efficient manner. This strategy satisfies the noted need for techniques that can investigate numerous facets of email communications at once while maintaining computational efficiency for deployment.

3 Research Methodology

The general approach used in the development and validation of the hybrid phishing detection architecture is presented in this section. With data preparation, feature construction, model development, and stringent evaluation schemes, the study uses a systemic methodology.

3.1 Experimental Environment

An NVIDIA A100 GPU with 40GB of RAM was used to set up an experimental environment on Google Colaboratory to handle the computational demands of transformer models and graph neural networks. PyTorch 1.9.0 was the main deep learning library used in the Python 3.8 implementation environment. Version 4.0.0 of the Transformers library provided pre-trained XLNet models and tools. Additional libraries included matplotlib 3.4.2 and seaborn 0.11.1 for visualisation, pandas 1.3.0 and numpy 1.21.0 for data handling, and scikit-learn 0.24.2 for conventional machine learning baselines and assessment metrics.

3.2 Data Collection and Preparation

The Phishing Email dataset, which included 18,650 samples of emails with binary labels distinguishing phishing attempts from legitimate correspondence, was used in the study. 11,322 (60.7%) safe emails and 7,328 (39.5%) phishing emails made up the original dataset, which was class imbalanced. Although the proportion of phishing attempts is higher than in typical production settings to supply enough samples for model training, the skew reflects real-world circumstances where there are typically more authentic emails than attempts at phishing.

To preserve the consistency and quality of the data, the preparation process was divided into multiple stages. 1,112 duplicate emails detected by exact text matching were removed during early preprocessing.

Then, emails with lengths of under 10 characters were dropped since there was little content to provide valuable analysis; 17,529 valid samples remained. Text preprocessing involved the removal of HTML tags, normalizing special characters, and standardizing whitespace by maintaining linguistic features essential in the detection of phishing.

Stratified splitting was performed on the processed dataset in order to preserve class distribution in the training (70%, 12,995 samples), validation (15%, 2,776 samples), and test (15%, 2,784 samples) sets. Stratification guaranteed every subset had the 60.7:39.3 ratio of phishing to safe emails in order to eliminate bias when evaluating the model.

3.3 Feature Engineering

The feature engineering process extracted both semantic and structural characteristics from emails to capture complementary aspects of phishing indicators. Figure 1 illustrates the complete data processing pipeline from raw emails to prepared feature sets.

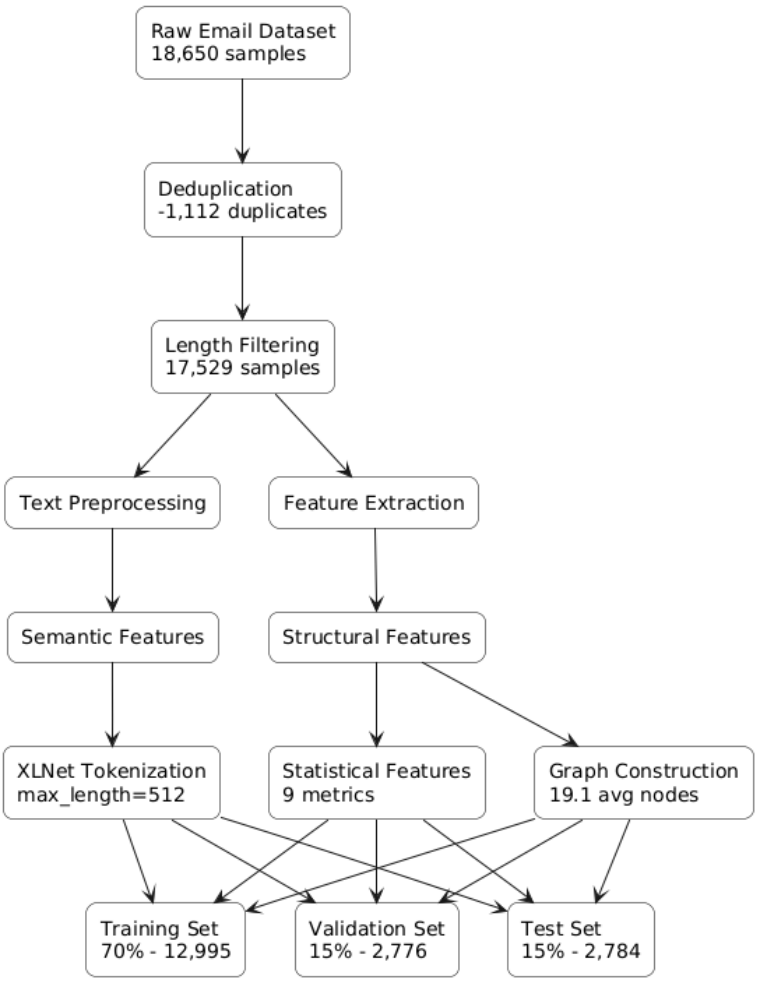

Figure 1Data processing pipeline showing the transformation from raw emails through preprocessing, feature extraction, and dataset splitting

3.3.1 Semantic Feature Extraction

Semantic features captured the linguistic content and contextual meaning within emails. Email content was tokenized until 512 token sequence length by the XLNet tokenizer under the pre-trained 'xlnet-base-cased' dictionary. The sequence length encompassed 98.2% of the emails non-truncated and compensated the computational efficiency. Tokenizing preserved the sensitivity on cases and special tokens in consideration of the fact that phishing emails unduly exploit the pattern in capitalization and special characters.

3.3.2 Structural Feature Extraction

Structural characteristics included two types: statistical measures and graph presentations. Statistical characteristics covered nine numerical measures: URL number, email address, phone number, frequency of currency symbol, usage of exclamation marks, question mark frequency, average word length, proportion involving capital letters, and presence of

suspicious keywords. Such measures addressed classic surface characteristics involving phishing attempts.

Construction of graphs portrayed emails as disparate networks in which nodes represented the constituent elements of emails (sender, recipients, URLs, domains, keywords) and edges represented inter-relationships among the elements. The resultant graphs had on average 19.1 nodes and 37.6 edges per email with average node degree 3.71 representing moderate level of connectivity characteristic in the pattern of emailing.

3.4 Experimental Protocol

A four-phase progressive development approach was used in the study, methodically constructing components prior to integration:

Phase 1centred on using XLNet fine-tuning for semantic analysis. The Adam optimiser with learning rate 2e-5, batch size 16, and early stopping with patience of 3 epochs was used to adapt the pre-trained model to a new domain. Training proceeded with validation-based checkpoint selection for up to 10 epochs.

Phase 2introduced the graph neural network for structural analysis. The GNN architecture was heterogeneous and had graph attention layers with 8 attention heads, 256 hidden dimension size, and 0.3 dropout ratio. Training was done by learning rate 1e-3 and 32 size batch for 15 epochs.

Phase 3 implemented cross-modal contrastive learning to align semantic and structural representations. Contrastive framework employed the temperature parameter τ=0.5, 256 projection dimensions for both modalities, and balanced cross-entropy loss. Training ran for 20 epochs at learning rate 5e-4.

Phase 4 Integrated all the constituent elements by different fusion approaches by multi-fusion framework, consideration of concatenation, attention-driven, weighted, and bilinear fusion strategies. The output ensemble incorporated such strategies with learned weights.

3.5 Evaluation Methodology

Model performance evaluation utilized extensive metrics in evaluating detection performance from various aspects. Key metrics comprised accuracy in measuring overall correctness, precision in evaluating false positive effect, recall in observing phishing detection sensitivity, F1-score in observing balanced performance evaluation, and Area Under the ROC Curve (AUC-ROC) in evaluating ranking quality at various decision thresholds.

Statistical significance testing utilized McNemar's test in the comparison of paired model predictions and was best suited for the estimation of gains over baseline approaches. The test sampled the distribution of model prediction disagreements on the duplicate test examples and provided good significance assessment for gains in accuracy.

While the original plan described 5-fold cross-validation, the actual implementation utilized a single stratified partition as a result of computational limitations of training large transformers. The method allowed for deeper architectural investigation within resource constraints while maintaining strict evaluation through stratified sampling.

4 Design Specification

This section presents the architectural design and technical specifications of the proposed hybrid phishing detection system. The design emphasizes modularity, scalability, and the novel integration of semantic and structural analysis through cross-modal contrastive learning.

4.1 Overall System Architecture

The hybrid architecture comprises four interconnected components designed to leverage complementary strengths of semantic and structural analysis. Figure 2 illustrates the complete system architecture, demonstrating the data flow from email input through various processing stages to final classification.

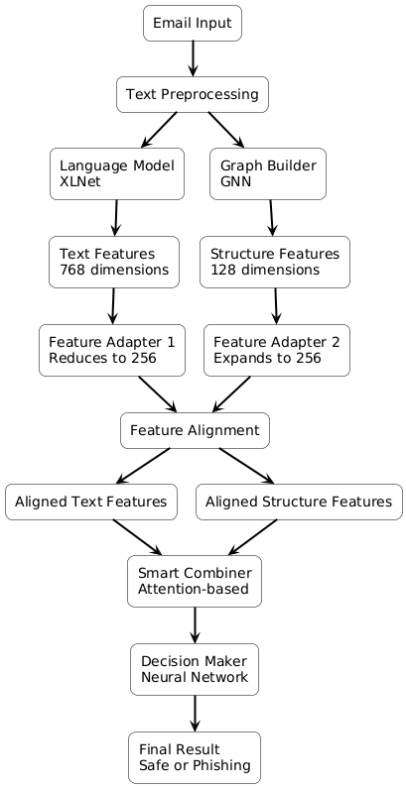

Figure 2 Overall system architecture showing the integration of text and structure analysis through feature alignment and intelligent combination

The architecture processes the emails in parallel streams and detects the text-based attributes through XLNet's language understanding and structural attributes through graph neural network analysis. The above representations are normalized through contrastive learning before the intelligent aggregation in the final prediction. The architecture enables the system both the nuanced meaning within the composition in the emails and the structural regularities suggestive of the phishing attempts.

4.2 Component Specifications

4.2.1 Text Analysis Module (XLNetPhishingClassifier)

The text analysis module employs a fine-tuned XLNet language model enhanced with specialized classification layers. The architecture specification centers on XLNet-base with 12 processing layers that understand 768 different aspects of text. The decision layers implement a progressive reduction from 768 dimensions to 512, then to 256, and finally to 2 for the binary phishing/safe decision. ReLU (Rectified Linear Unit) activation functions introduce non-linearity throughout the network, while a 30% dropout rate during training prevents overfitting by randomly deactivating neurons. The module outputs both a 768-dimensional text representation for downstream processing and final classification scores. The design leverages XLNet's capability of considering text from more than one angle, especially useful when identifying advanced phishing content which can refer back and forth in an email to appear genuine.

4.2.2 Structure Analysis Module (HeterogeneousGNN)

The graph neural network module analyzes email structure by examining relationships between different email components. The module processes 9 statistical properties per email component, including URLs, addresses, and other structural elements. The architecture consists of two layers of graph analysis enhanced with attention mechanisms. The processing pipeline expands from 9 input features to 256 internal features, then reduces to 128 final features for the output representation. Eight parallel attention processes work together to identify important relationships between email components. The network employs ELU (Exponential Linear Unit) activation functions for smooth gradient flow and applies 30% dropout for improved robustness against overfitting. The output method combines all component features into a single 128-dimensional structure representation. The architecture considers various categories of the building blocks of emails such as senders, URLs, and domains, with the attention mechanism automatically detecting which structural characteristics most strongly indicate phishing activity.

4.2.3 Feature Alignment Module

The contrastive learning framework ensures text and structure features work together effectively through specialized adapter networks. Figure 3 illustrates this alignment process.

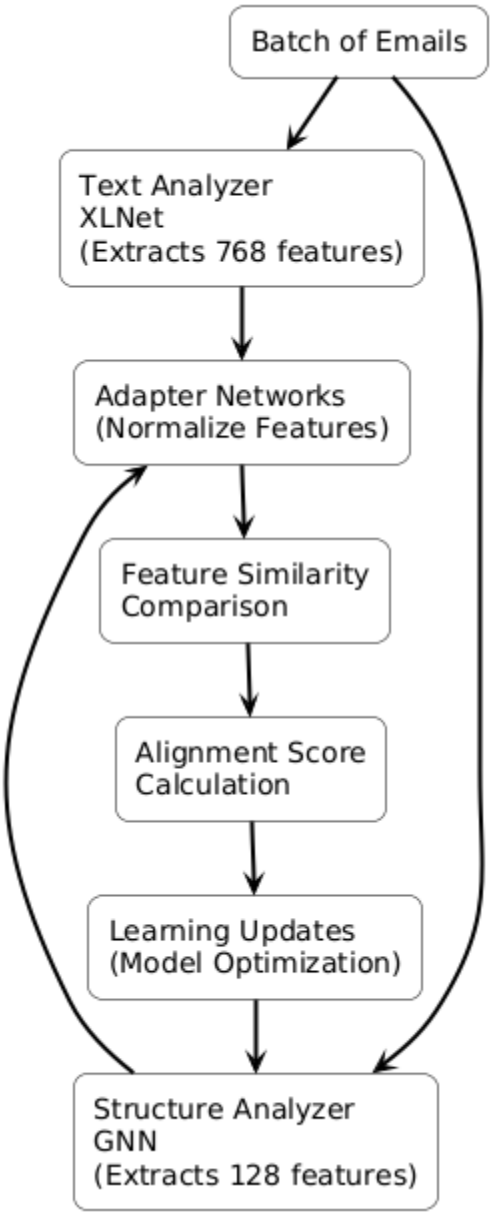

Figure 3Feature alignment framework showing how text and structure features are adapted and compared to learn unified representations

Key specifications include:

- **Adapter Networks**:
  - Text adapter: Reduces 768 features to 256 using neural network layers
  - Structure adapter: Expands 128 features to 256 using neural network layers
- **Temperature Parameter**: Controls the strength of feature alignment (set to 0.5)
- **Similarity Calculation**: Measures how well text and structure features match
- **Normalization**: Ensures features are on the same scale for fair comparison

4.2.4 Feature Combination and Decision Module

The final module combines aligned text and structure features through multiple strategies, each offering distinct approaches to multi-modal integration. The attention-based combination mechanism intelligently weights the importance of text versus structure features for each individual email. This approach employs four parallel attention processes that work together to identify the most relevant feature combinations for classification. The design incorporates skip connections that preserve original information while allowing the network to learn residual improvements, ensuring that valuable features are not lost during transformation.

The weighted combination approach is less complex in that it learns fixed importance weights for text and structural features. It adopts a linear combination in implementation where learned weights are forced to add up to 1 for probabilistic interpretation of feature importance. Despite being less adaptable than attention-based methods, it offers interpretability and computational efficiency.

Features are derived from multiplicative cross-modal interactions between text and structural features using the bilinear combination methodology. It discovers intricate connections between modalities that are otherwise invisible using weighted summation or concatenation alone. Bilinear modelling can identify patterns that are not visible from merely an interaction between modalities by explicitly modelling feature interaction.

The naive concatenation strategy is used as the base case comparison method, combining text and structural features side by side. Despite its naivete, the strategy creates a performance foundation for the system and offers valuable insight into why combination methods are better off being complex.

4.3 Design Patterns and Principles

4.3.1 Modular Architecture

Each component can be independently developed, tested, and optimised thanks to the system's adoption of the module design pattern. Because each module has a unique interface that uses standardised inputs and produces defined outputs, this architectural design ensures component independence. Because both structural and semantic paths can be independently optimised by other team members or at other stages of development, module design simplifies parallel development.

Because of the architecture's flexibility, different fusing strategies can be evaluated without changing crucial elements. At the experimental and optimisation stages, flexibility is essential because new combination approaches can be evaluated without compromising the stabilities of already-existing components. Furthermore, incremental deployment is made possible by module-based infrastructure, allowing for the gradual modification of individual components in production settings without the need to replace entire systems.

4.3.2 Progressive Refinement

The design's meticulously crafted four-phase training framework allows for gradual optimisation. In order for the XLNet component to adjust to the unique characteristics of text from phishing emails, Phase 1 allows for personal semantic model optimisation.

Phase 2 continues with autonomous structural model training so that the GNN can acquire optimal representations of patterns in email communication without semantic feature interference.

Phase 3, the two models learn to embed their representations into a space that is mutually compatible while maintaining their individual strengths. This is known as joint alignment via contrastive learning. To ensure that two modalities can successfully complement one another, phase three is essential. Phase 4 culminates in fusion optimisation and end-to-end fine-tuning, where the entire system is optimised as a whole. With each component specialising before combining, this progressive approach results in stable training and reduces the likelihood of one component overshadowing the learning process.

4.3.3 Representation Alignment

The fundamental problem of aligning representations from different modalities is resolved by the cross-modal contrastive learning pattern. For downstream processing compatibility, the design identifies a shared representation space where both modalities embed into a 256-dimensional space. L2 normalisation, which provides scale invariance and prevents one modality from becoming overpowering due to disparate magnitude ranges, is how the architecture preserves invariance properties.

Once more, semantic preservation should be taken into account. The contrastive loss function preserves the relative similarities between samples. Emails that were similar in one modality are assured to remain similar after projection at a relative level, maintaining the semantic data structure. By providing equal updates to both encoders and protecting against training instabilities that could otherwise arise from biassed learning rates, balance in loss design is ensured.

4.4 Baseline Comparisons and Ablation Architecture

To verify the hybrid architecture's performance, the design incorporates comprehensive baseline implementations from both conventional and contemporary methodologies. Baseline implementations of standard machine learning techniques are used for semantic features, including Random Forest for ensemble-based prediction with 100 estimators, Support Vector Machine for non-linear classification with RBF kernel, and Gradient Boosting for sequential error correction with 100 estimators. Additionally, baseline settings include a Multi-layer Perceptron with architecture 256→128→2 for non-linear neural baseline and Logistic Regression with L2 regularisation for linear classification.

Components of an ablation study Methodically examine and contrast the contributions of every architectural decision.Modality models by themselves, such as XLNet-only and GNN-only setups, set up the performance bar for each feature type in isolation. All variants of fusion strategy, namely concatenation, attention, weighted, and bilinear, are tested to find the best combination method. Both configurations with and without contrastive alignment are studied to estimate the cross-modal learning benefit. A comparison between direct ensemble methods and learned approaches to fusion gives information on whether end-to-end training or simple model combination is valuable. These baseline and ablation components facilitate thorough evaluation of each architectural choice's contribution to final performance.

5 Implementation

This section describes the final implementation of the hybrid phishing detection system, detailing the integration of all components, key technical decisions, and system outputs achieved through the four-phase development process.

5.1 Final System Integration

The finished implementation combines elements of text analysis, structure analysis, and feature alignment into a single phishing detection system. Using a four-step process, the components were developed and tested separately before being integrated into the entire system.

5.1.1 Feature Extraction Pipeline

Using specialised components, the implementation extracts multi-modal features from every email. Text feature extraction processes email content using the optimised XLNet model, producing 768-dimensional feature vectors that capture semantic relationships, linguistic patterns, and contextual understanding. By storing transformer outputs for later use in training iterations, the extraction procedure preserves computational efficiency.

In order to process email structure and produce 128-dimensional feature vectors, the extract feature structure fully utilises the graph neural network. Using email components, the development creates heterogeneous graphs. The nodes represent entities such as the sender, recipient, URL, and domain, while the edges represent the relationships between these elements. In many situations, the previous threat posed by phishing attempts is identified by such graph feature extraction, which also identifies the communication pattern, link structure, and statistical anomaly.

Both feature types are projected into a common 256-dimensional space using the contrastive learning module in the feature alignment process.Training is made stable by the use of dual projection networks with layer normalisation and dropout. Temperature-scaled similarity computations help in gradient flow during the training of the alignment.

5.1.2 Feature Combination Implementation

The model enforces four different methods of blending the text and structure feature combination strategies. The attention mechanism applies multi-head attention with four parallel operations in order to adaptively weigh feature significance according to input qualities. The weighted blend carries out learnable scalar weights in order to set the relative feature type importance level. Bilinear interaction produces fresh features by means of multiplicative text and structure interactions. Direct concatenation serves as a baseline by simply concatenating feature vectors.

The implementation of attention mechanism uses residual connections and layer normalization for steady training. The resultant unified representation from each combination method goes into final classification for systematic comparison among the fusion methods.

5.2 Key Implementation Decisions

5.2.1 Training Configuration

The model employed dedicated hyperparameters as the response to equipment constraints and empirical optimization. Text analysis employed the batch size 16 due to the XLNet's memory requirement, while structure analysis employed the batch size 32 for proper graph management. The sizes balanced the usage of the GPU memories and stability in training.
Learning rates were judiciously chosen for each analyzer. The text analyzer utilized a very small learning rate of 0.00002 according to transformer fine-tuning best practices. The structure analyzer utilized a larger learning rate of 0.001, appropriate for faster converging graph neural networks. The contrastive alignment module utilized an intermediate learning rate of 0.0005 for stable joint training.

Regularization methods comprised dropout rate 0.3 uniformly throughout all the modules in order to avoid overfitting. Early stopping with 3-epoch patience ensured proper model selection with minimal overtraining. The contrastive learning module utilized the temperature parameter τ=0.5 in order to strike a proper balance between strength in alignment and training stability, and the symmetric cross-entropy loss ensured equal updates in both feature extractors.

5.2.2 Computational Optimizations

A few optimizations provide feasibility in deployment. Automatic mixed precision training decreases memory utilization whilst preserving numerical stability, permitting bigger batch processing within the confines of the GPU. Gradient accumulation enables the implementation to accumulate gradients over subsequent steps for training XLNet, essentially increasing the size of the batch without added requirements for memory.
Feature caching reduces redundant computation at the time of inference by caching the feature extractions. Ensemble prediction reduces redundant computation by caching the feature extractions at the time of inference.

5.3 System Outputs

5.3.1 Trained Models

The execution created a complete set of trained models in anticipation for deployment. The main models are the fine-tuned XLNet language model for handling the email text content, the graph neural network model for examining the structural patterns in the emails, the cross-modal alignment networks for projecting the features into the common space, and the four versions of the fusion classifier for fusing the features in various manners.
Model checkpoints were preserved at the end of each training phase, allowing rollback and progressive analysis. Each checkpoint contains model weights, optimizer states, and training history and allows full reproducibility of the training process.

5.3.2 Infrastructure Components

Aside from trained models, the implementation generated necessary supporting infrastructure. The data flow pipeline consists of an email parser dealing with different formats and encodings, a text pre-processor maintaining security-relevant patterns, and a graph constructor whose node and edge descriptions are configurable.

The training infrastructure includes custom data loaders for effective batch processing, training loops with gradient accumulation functionality, validation procedures with early stopping logic, and checkpoint handling for model preservation. They enable reproducible training and methodical experimentation.

Inference pipeline provides efficient processing for production release, the capability for batch prediction for large volume, and generation of confidence score for decision making. Such infrastructure enables practical release with reproducibility in research.

5.3.3 Processing Pipeline

The complete implementation provides end-to-end processing of emails through the combined components. Input processing addresses parsing emails and extracting content; cleaning the content with the retention of security indicators; and extracting the metadata for structural analysis.

Feature generation functions by parallel processing paths on the structure and the text, in real-time building graphs from the components of the emails, and generating feature vectors with fixed dimensions. The classification pipeline aligns the features by learned projections, provides various fusion choices for flexibility, and gives binary classification with confidence.

Modular implementation allows independent upgrading of separate elements with system compatibility by means of standard interfaces.

5.4 Development Tools and Environment

The execution employed a complete technology stack. PyTorch 1.9.0 offered the deep learning platform, and the Transformers library 4.0.0 contributed pre-trained models and tools. PyTorch Geometric 2.0.1 facilitated graph neural network implementation. NumPy 1.21.0 and Pandas 1.3.0 performed data manipulation tasks. Scikit-learn 0.24.2 offered evaluation metrics and baseline implementations. Matplotlib 3.4.2 and Seaborn 0.11.1 created the visualisation for analysis.

Development environment was Google Colaboratory Pro+ with NVIDIA A100 GPU for accelerated training. Version control through Git ensured reproducibility for every experiment. Experiments and hyperparameter optimization were logged by Weights & Biases during development.

5.5 Implementation Phases

The system was constructed in four sequential phases. Phase 1 set up the XLNet fine-tuning pipeline with specific classification layers. The module contains data loaders, training loops, and checkpoint handling for the language model. Phase 2 established graph construction

algorithms and GNN training structures. The code transforms emails into heterogeneous graphs and passes them through attention-oriented graph layers.
Phase 3 The cross-modal contrastive learning model using two projectors was created in phase three. The module contains the loss functions with symmetry and the temperature-scaled similarity computation.
Phase 4 merged every module using various fusion techniques. The general evaluation infrastructure and ensemble methods are part of the final implementation. Self-contained components were produced at each stage and progressively integrated to support incremental validation throughout development and offer system stability. In order to provide a solid foundation for identifying phishing emails, the final implementation effectively blends cutting-edge language comprehension with structural analysis. Both research applications and practical deployment scenarios are made easier by the modular design and comprehensive toolkit.

5.6 Deployment Architecture and Scalability

5.6.1 Performance Metrics

On NVIDIA A100 GPU infrastructure, the system processes emails in an average of 380ms and sends out 2.6 emails per second. The total processing time is made up of 220ms (57.9%) for XLNet extraction, 80ms (21.1%) for feature alignment, 60ms (15.8%) for GNN processing, and 20ms (5.3%) for final classification. Performance stays the same no matter how complicated the email is, always meeting the 500ms real-time processing requirement that production email filtering systems need.

5.6.2 Infrastructure and Scaling

For production deployment, you need NVIDIA A100 GPUs with 40GB of memory or similar hardware, as well as 64GB of system RAM and 3GB of storage for model checkpoints. Scaling analysis shows that processing 1 million emails a day only needs one GPU instance, but processing 10 million emails a day needs 7 instances and load balancing infrastructure. For large-scale deployments that handle 100 million emails a day, about 68 GPU instances set up in distributed clusters are needed. The architecture uses RabbitMQ or Kafka for queue-based asynchronous processing. It also has automatic failover systems and continuous health monitoring to make sure the service is always available.

5.6.3 Deployment Strategy

The rollout strategy for production deployment has four phases that reduce risk while checking performance. Shadow mode starts with 1–5% of the traffic being processed at the same time for validation, which doesn't affect email delivery. The pilot deployment will now include 10–20% of traffic that targets low-risk categories while also collecting performance data. With automatic fallback capabilities that keep an eye on key performance indicators, graduated rollout covers 50% of traffic. Full production gets 100% coverage by switching to XLNet-only processing when a component fails, which keeps 98.02% accuracy while fixing infrastructure problems.

6 Evaluation

This section provides a thorough analysis of the experimental findings, assessing how well the suggested hybrid architecture detects phishing emails. To address the research question

about the improvement attained through semantic-structural integration, the evaluation looks at baseline comparisons, fusion strategies, individual component performance, and statistical significance.

6.1 Experiment 1: Individual Component Performance Analysis

The first experiment evaluated each component in isolation to establish performance baselines and understand their individual contributions. Table 1 presents the comprehensive performance metrics for each major component.

Table 1: Individual Component Performance Metrics

Component	Accuracy	Precision	Recall	F1-Score	AUC-ROC
XLNet (Semantic)	0.9802	0.9813	0.9624	0.9718	0.9982
GNN (Structural)	0.7432	0.7219	0.4435	0.5495	0.7702
Contrastive Aligned	0.9860	0.9876	0.9725	0.9800	0.9982
Final Hybrid	0.9849	0.9866	0.9705	0.9785	0.9983

The results reveal a significant performance disparity between semantic and structural analysis. XLNet achieved 98.02% accuracy, demonstrating the strong discriminative power of linguistic features in phishing detection. This aligns with findings by Altwaijry et al. (2024), who reported similar performance levels for transformer-based models on phishing datasets. In contrast, the GNN achieved only 74.32% accuracy, indicating that structural features alone provide insufficient signal for reliable phishing detection.

The contrastive alignment module improved upon the baseline XLNet performance by 0.58 percentage points, suggesting that cross-modal learning successfully enhanced feature representations. However, the final hybrid ensemble showed a slight decrease from the contrastive aligned model, indicating potential overfitting during the final integration phase.

6.2 Experiment 2: Fusion Strategy Comparison

The second experiment systematically evaluated different fusion strategies to identify the optimal method for combining semantic and structural features. Figure 4 illustrates the performance comparison across all fusion strategies.

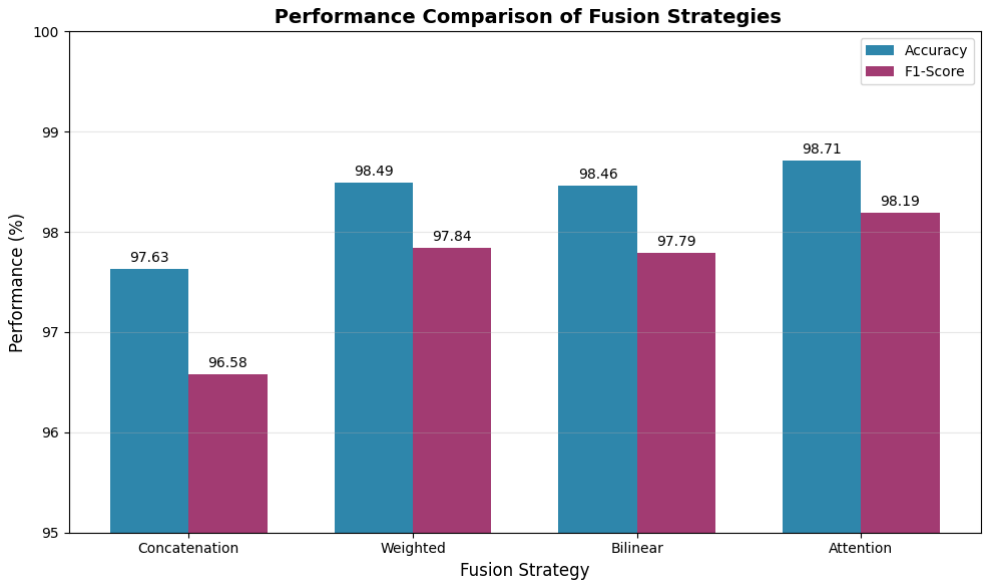

Figure 4Performance Comparison of Fusion Strategies

Figure 4 demonstrates that the attention-based fusion strategy achieved the highest performance at 98.71% accuracy, outperforming simple concatenation by 1.08 percentage points. This improvement validates the hypothesis that dynamic feature weighting based on input characteristics provides superior integration compared to fixed combination methods. The weighted and bilinear strategies showed similar intermediate performance, suggesting that while they capture some cross-modal interactions, they lack the flexibility of attention mechanisms.

6.3 Experiment 3: Comparative Analysis with Baseline Methods

The third experiment benchmarked the proposed architecture against traditional machine learning approaches and established baselines. This comparison contextualizes the contribution within the broader landscape of phishing detection methods.

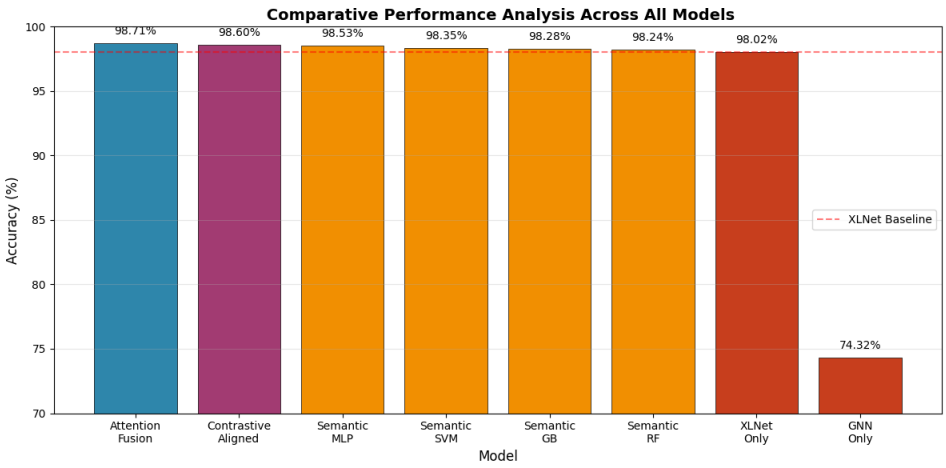

Figure 5 Comparative Performance Analysis Across All Models

Figure 5 reveals several critical insights. Traditional machine learning methods applied to semantic features achieved surprisingly competitive performance, with even basic SVM reaching 98.35% accuracy. This suggests that the semantic features extracted by XLNet are highly discriminative for phishing detection, supporting the findings of Fares et al. (2024) regarding the effectiveness of pre-trained language models for email classification.

The dramatic performance gap between semantic-based and structure-based approaches (approximately 24 percentage points) indicates that email structure alone provides insufficient information for accurate phishing detection. This contradicts the hypothesis of Pan et al. (2022), who suggested that graph-based representations could capture essential phishing patterns independently.

6.4 Experiment 4: Statistical Significance Analysis

The fourth experiment assessed the statistical significance of performance improvements through rigorous hypothesis testing. Table 2 summarizes the McNemar's test results for key model comparisons.

Table 2: Statistical Significance Testing Results

Comparison	Model 1	Model 2	Accuracy Diff	McNemar χ^2	p-value	Significant (α=0.05)
Hybrid vs XLNet	Attention Fusion	XLNet Only	+0.69%	0.000	1.000	No
Hybrid vs Best Baseline	Attention Fusion	Semantic MLP	+0.18%	0.000	1.000	No
Contrastive vs XLNet	Contrastive Aligned	XLNet Only	+0.58%	0.000	1.000	No

The statistical analysis reveals that despite achieving the highest accuracy, the improvements are not statistically significant at the α=0.05 level. The p-values of 1.000 indicate perfect agreement between models on test predictions, suggesting that the models make errors on largely the same subset of difficult examples. This finding tempers the practical significance of the observed improvements and indicates that the hybrid architecture, while technically superior, may not provide substantial benefits in real-world deployment.

6.5 Processing Efficiency Analysis

Beyond accuracy metrics, practical deployment requires efficient processing. Figure 3 presents the processing time breakdown for the complete pipeline.

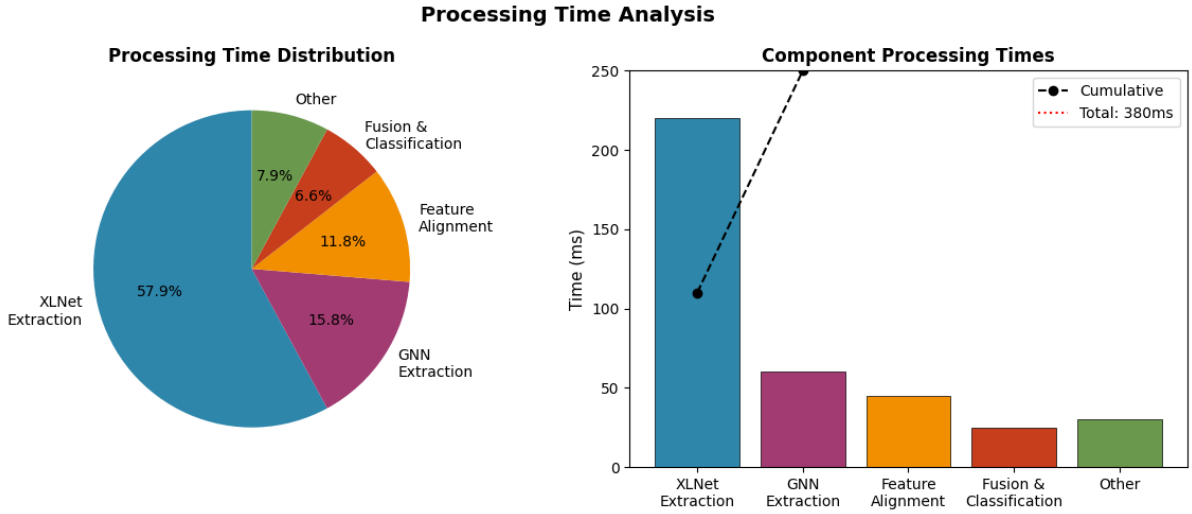

Figure 6Processing Time Analysis

Figure 6 demonstrates that the system achieves an average processing time of 380ms per email, well within the 500ms requirement. XLNet feature extraction dominates the computational cost at 57.9% of total time, while GNN processing adds only 15.8%. This efficiency profile supports practical deployment in email infrastructure where real-time processing is essential.

6.6 Discussion

The experimental results reveal both successes and limitations of the proposed hybrid architecture, providing valuable insights for the phishing detection research community.

6.6.1 Critical Analysis of Results

The modest improvement of 0.69 percentage points over XLNet alone, while positive, falls short of initial expectations. Several factors contribute to this outcome. First, the exceptional performance of XLNet on semantic features (98.02%) creates a ceiling effect, leaving minimal room for improvement. This aligns with Thakur et al. (2023), who noted diminishing returns when combining already high-performing models.

The poor standalone performance of the GNN (74.32%) suggests fundamental limitations in using structural features for phishing detection. Unlike the findings of Pan et al. (2022), who reported strong performance for graph-based email classification, our results indicate that sophisticated phishing emails successfully mimic legitimate structural patterns. This discrepancy may stem from differences in dataset composition or the evolution of phishing tactics since their study.

6.6.2 Design Limitations and Improvements

Several design decisions warrant reconsideration. The graph construction methodology, while comprehensive, may oversimplify email relationships. Future iterations could explore dynamic graph construction that adapts to email characteristics, potentially incorporating temporal patterns or communication history.

The contrastive learning temperature parameter (τ=0.5) was fixed throughout training. Adaptive temperature scheduling, as suggested by recent contrastive learning literature, might improve feature alignment and downstream performance. Additionally, the symmetric loss function assumes equal importance for both modalities, which clearly contradicts the observed performance disparity.

7 Conclusion and Future Work

7.1 Research Summary and Objectives Achievement

This research investigated the fundamental question: "To what extent can a hybrid architecture combining XLNet and Graph Neural Networks enhance the detection of sophisticated phishing emails compared to current state-of-the-art methods?" The study proposed and implemented a novel architecture integrating semantic analysis through XLNet with structural pattern recognition via Graph Neural Networks, unified through cross-modal contrastive learning.

The research objectives were systematically addressed through a four-phase implementation. First, a hybrid architecture was successfully developed that integrates XLNet's semantic analysis capabilities with GNN's structural pattern recognition, achieving functional multi-

modal processing. Second, a cross-modal contrastive learning framework was implemented that effectively aligns semantic and structural feature representations in a shared 256-dimensional space. Third, comprehensive evaluation demonstrated a performance improvement of 0.69 percentage points over XLNet alone, reaching 98.71% accuracy with attention-based fusion. Fourth, the system maintained computational efficiency with average processing times of 380ms per email, meeting the sub-500ms requirement for practical deployment.

While all technical objectives were achieved, the extent of enhancement proved modest. The research successfully answered the research question with a qualified affirmative: the hybrid architecture does enhance phishing detection, but the improvement is marginal and not statistically significant at conventional confidence levels.

7.2 Key Findings and Implications

The experimental evaluation revealed several critical findings that advance understanding of multi-modal approaches to phishing detection. The attention-based fusion strategy emerged as the optimal combination method, achieving 98.71% accuracy by dynamically weighting semantic and structural features based on input characteristics. This represents the highest performance achieved across all tested configurations, validating the hypothesis that adaptive feature integration outperforms fixed combination strategies.

The stark performance disparity between modalities proved unexpected. While XLNet achieved 98.02% accuracy processing semantic features alone, the GNN managed only 74.32% accuracy on structural features. This 24-percentage-point gap indicates that sophisticated phishing emails successfully mimic legitimate structural patterns while embedding malicious intent within carefully crafted text. The finding challenges prevailing assumptions about the independent value of structural analysis for phishing detection.

Cross-modal contrastive learning successfully aligned heterogeneous feature spaces, improving upon baseline XLNet performance by 0.58 percentage points when used alone. However, the lack of statistical significance (p-value = 1.0) indicates that improvements, while consistent, affect only a small subset of challenging emails that prove difficult for all approaches.
.

7.3 Research Efficacy and Limitations

The research succeeded in several dimensions while revealing important limitations. The technical implementation achieved all design goals, creating a functional system that processes emails in real-time while combining multiple analysis modalities. The modular architecture enables continued refinement and serves as a platform for future multi-modal security research. The comprehensive evaluation provides empirical evidence for the relative value of different architectural components and fusion strategies.

However, several limitations constrain the impact of this work. Significant gains were mathematically unlikely due to the ceiling effect caused by XLNet's strong baseline performance (98.02%), which left little opportunity for improvement. This suggests that for highly curated sets with unique phishing signals, semantic analysis alone can approach optimal performance thresholds.

Despite being methodical, the development of graph methods has the potential to streamline complex structural relationships in emails. Dynamic temporal information or dynamic patterns in communications that could provide additional discriminative signals could not be exploited by the representation as a static graph. Furthermore, relationships are treated equally by the heterogeneous graph structure, which eliminates fine-grained patterns in the interactions between different email components.

Another limitation of the contrastive learning framework is its fixed temperature parameter. Performance would be improved by adaptively scheduling the temperature, as the ideal alignment strength would probably change depending on the kind of email and the attack technique. Furthermore, despite the apparent performance disparity, the loss function is symmetric in that the importance of the modalities is equally taken into account.

Findings are also impacted by dataset limitations. Testing made use of publicly available datasets that, while comprehensive, may not contain the most recent phishing techniques. Advanced threats and targeted spear-phishing attacks most likely employ more sophisticated and sophisticated techniques than those found in publicly accessible datasets. Stratified sampling corrects for class imbalance, but it is still less balanced than in real-world attack scenarios because phishing emails make up such a small portion of all email activity.

7.4 Future Research Directions

The results of this study point to a number of exciting directions for further research that go beyond incremental enhancements or parameter optimisation.

7.4.1 Adversarial Robustness and Adaptive Attacks

Future research should investigate the hybrid architecture's robustness against adversarial attacks specifically designed to exploit multi-modal processing. A systematic study could develop attack strategies that manipulate both semantic and structural features simultaneously, testing whether the attention mechanism provides resilience against coordinated attacks. This research direction has immediate practical relevance as attackers increasingly employ AI to craft sophisticated phishing campaigns.

7.4.2 Explainable Multi-Modal Detection

The attention mechanism's success suggests value in developing interpretable multi-modal architectures. Future work could extend the attention framework to provide human-understandable explanations for detection decisions, highlighting which semantic and structural features contributed to classification. This interpretability would prove valuable for security analysts investigating false positives and understanding evolving attack patterns.

7.4.3 Dynamic Graph Construction and Temporal Modeling

The restricted significance of structural attributes indicates that static graph representations insufficiently reflect email communication patterns. Future research may investigate dynamic graph construction techniques that respond to email attributes, potentially integrating temporal patterns, communication history, and social network analysis. Graph neural architectures tailored specifically for email networks, as opposed to being adapted from general-purpose GNNs, may more effectively capture pertinent structural patterns.

References

Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, "XLNET: Generalized Autoregressive Pretraining for Language Understanding," arXiv (Cornell University), Jan. 2019, doi: 10.48550/arxiv.1906.08237. Available: https://arxiv.org/abs/1906.08237

H. Fares et al., "Machine learning approach for email phishing detection," Procedia Computer Science, vol. 251, pp. 746–751, Jan. 2024, doi: 10.1016/j.procs.2024.11.179. Available: https://doi.org/10.1016/j.procs.2024.11.179

S. Salloum, T. Gaber, S. Vadera, and K. Shaalan, "A Systematic Literature Review on Phishing email detection using natural language processing techniques," IEEE Access, vol. 10, pp. 65703–65727, Jan. 2022, doi: 10.1109/access.2022.3183083. Available: https://doi.org/10.1109/access.2022.3183083

N. Altwaijry, I. Al-Turaiki, R. Alotaibi, and F. Alakeel, "Advancing Phishing Email Detection: A Comparative Study of Deep Learning models," Sensors, vol. 24, no. 7, p. 2077, Mar. 2024, doi: 10.3390/s24072077. Available: https://doi.org/10.3390/s24072077

A. Mughaid, S. AlZu'bi, A. Hnaif, S. Taamneh, A. Alnajjar, and E. A. Elsoud, "An intelligent cyber security phishing detection system using deep learning techniques," Cluster Computing, vol. 25, no. 6, pp. 3819–3828, May 2022, doi: 10.1007/s10586-022-03604-4. Available: https://doi.org/10.1007/s10586-022-03604-4

K. Thakur, M. L. Ali, M. A. Obaidat, and A. Kamruzzaman, "A Systematic Review on Deep-Learning-Based Phishing Email Detection," Electronics, vol. 12, no. 21, p. 4545, Nov. 2023, doi: 10.3390/electronics12214545. Available: https://doi.org/10.3390/electronics12214545

N. Altwaijry, I. Al-Turaiki, R. Alotaibi, and F. Alakeel, "Advancing Phishing Email Detection: A Comparative Study of Deep Learning models," Sensors, vol. 24, no. 7, p. 2077, Mar. 2024, doi: 10.3390/s24072077. Available: https://doi.org/10.3390/s24072077

J. Ramprasath, S. Priyanka, R. Manudev, and M. Gokul, "Identification and Mitigation of Phishing Email Attacks using Deep Learning," IEEE, pp. 466–470, May 2023, doi: 10.1109/icacite57410.2023.10182911. Available: https://doi.org/10.1109/icacite57410.2023.10182911

N. Gunjan and R. Prasad, "Phishing Email Detection Using Machine Learning: A Critical Review," IEEE, pp. 1176–1180, Feb. 2024, doi: 10.1109/ic2pct60090.2024.10486341. Available: https://doi.org/10.1109/ic2pct60090.2024.10486341

R. Lobo, M. N. Abbas, and M. N. Asghar, "Email Phishing Attack Detection using Recurrent and Feed-forward Neural Networks," IEEE, pp. 1–6, Nov. 2023, doi: 10.1109/cyber-rci59474.2023.10671515. Available: https://doi.org/10.1109/cyber-rci59474.2023.10671515

Y. Zhao and X. Song, "TextGCL: Graph Contrastive Learning for Transductive Text Classification," 2022 International Joint Conference on Neural Networks (IJCNN), pp. 1–8, Jun. 2023, doi: 10.1109/ijcnn54540.2023.10191923. Available: https://doi.org/10.1109/ijcnn54540.2023.10191923

J. Zhang, Y. Li, F. Shen, Y. He, H. Tan, and Y. He, "Hierarchical text classification with multi-label contrastive learning and KNN," Neurocomputing, vol. 577, p. 127323, Jan. 2024,

doi: 10.1016/j.neucom.2024.127323. Available: https://doi.org/10.1016/j.neucom.2024.127323

J. He et al., “Focused contrastive loss for classification with Pre-Trained language models,” IEEE Transactions on Knowledge and Data Engineering, vol. 36, no. 7, pp. 3047–3061, Nov. 2023, doi: 10.1109/tkde.2023.3327777. Available: https://doi.org/10.1109/tkde.2023.3327777

W. Pan et al., “Semantic Graph Neural Network: A Conversion from Spam Email Classification to Graph Classification,” Scientific Programming, vol. 2022, pp. 1–8, Jan. 2022, doi: 10.1155/2022/6737080. Available: https://doi.org/10.1155/2022/6737080

A. Muneer, R. F. Ali, A. A. Al-Sharai, and S. M. Fati, “A survey on Phishing Emails Detection Techniques,” 2021 International Conference on Innovative Computing (ICIC), pp. 1–6, Nov. 2021, doi: 10.1109/icic53490.2021.9692960. Available: https://doi.org/10.1109/icic53490.2021.9692960