National College of Ireland

Analyzing Machine Learning Model Synergies for Next- Generation IoT Malware Detection: A BERT and GPT Integration Approach

MSc Research Project

MSc in Data analytics

Prahaladh Lagadapati

Student ID: x23295881

School of Computing

National College of Ireland

Supervisor: Mr. Hicham Rifai

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Prahaladh Lagadapati
Student ID:	x23295881
Programme:	MSc in Data analytics **Year:**2024-2025
Module:	MSc (Research) Practicum/Internship
Supervisor:	Mr. Hicham Rifai
Submission Due Date:	15th September 2025
Project Title:	Analyzing Machine Learning Model Synergies for Next- Generation IoT Malware Detection: A BERT and GPT Integration Approach
Word Count:	9193 **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: PRAHALADH LAGADAPATI

Date: 15Th September 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):	

Analyzing Machine Learning Model Synergies for Next- Generation IoT Malware Detection: A BERT and GPT Integration Approach

Prahaladh Lagadapati
X23295881

Abstract

The proliferation of the Internet of Things (IoT) has resulted in significant security threats, with malware attacks rising by 62% annually, of which polymorphic variants constitute 41% of the total threats identified. The present study proposes a novel hybrid framework combining ModernBERT and GPT-3.5 embeddings to enhance IoT malware detection mechanisms. This study converts network traffic characteristics into semantic textual formats, facilitating transformer-based analysis of behavioural patterns. The systematic evaluation of the CIC IoT 2023 dataset, which includes 80,000 balanced samples across five attack categories, demonstrates that the hybrid approach attains an overall accuracy of 84.2%, reflecting a 71.5% enhancement compared to traditional methods. The system exhibits high efficacy in identifying DDoS attacks, achieving a 99.7% F1-score, and network attacks, with a 95.7% F1-score. However, it faces difficulties in reconnaissance detection, reflected in a 54.8% F1-score. Feature-level fusion of statistical, probabilistic, and semantic representations demonstrates superiority over single-model approaches, with embedding features accounting for 73% of decision importance. This paper demonstrates the promise of applying effective transformers and semantic embeddings to IoT security with a system that provides performance-balanced detection with the computational realities characteristic of IoT environments.

1 Introduction

The rapid rise in the number of Internet of Things (IoT) devices has changed the way people interact throughout the world. By 2025, deployments are expected to reach 30.9 billion (Alwahedi et al., 2024). This led to a huge increase in the attack surface, and organised malware attacks are now mostly targeting IoT systems. The latest security reports reveal that malware infections in IoT infrastructures continued to increase by 62% annually from the year 2021, with polymorphic being the type that constituted 41% of all the threats (Riaz et al., 2022). Traditional detection methods often fall short when applied to the wide variety of IoT devices, many of which operate with limited computing power and inconsistent security protections.

Signature-based systems, specifically, experience difficulty in the detection of contemporary IoT malware, particularly as attacks become increasingly sophisticated through code structure modifications while preserving malicious behavior (Deng et al., 2023). The future of machine learning approaches seems good, but most of them trade either efficiency for computational efficiency or detector efficiency for efficiency, with little balance in either direction (Arya et

al., 2023). The trade-off is also important in IoT contexts with limited resources since the extra work the device has to do affects how quickly it responds and how long the battery lasts.

The newest developments in transformer-based topologies show a lot of potential for solving problems by making the features work. The latest ModernBERT, introduced by Warner et al. (2024), is a significant advancement in energy-efficient transformer architecture, employing dynamic sparse attention methods that reduce computational complexity from $O(n^2)$ to $O(n \log n)$. The design preserves BERT's robust capabilities for sequential pattern recognition while guaranteeing feasibility for deployment on resource-constrained platforms. The model's bidirectional encoding provides significant insights into network traffic patterns, enhancing its capacity to identify minor indicators of malicious activity in IoT connections.

The efficiency of ModernBERT is enhanced by GPT-3.5, which provides improved semantic understanding and contextual analysis due to its 175 billion parameters and extensive pre-training (Kalyan, 2023). The implementation of GPT-3.5 on IoT devices is presently unfeasible; nonetheless, incorporating representations of GPT-3.5 maintains informative semantics that could align with the identification of new attack patterns. The model's relationship complexity and its ability to produce high-dimensional semantic embeddings (3072-dimensional with text-embedding-3-large) establish a strong feature space that can distinguish between malicious and legitimate IoT activities.

The integration of complementary designs presents a distinct opportunity to improve IoT security. The efficient local processing of ModernBERT facilitates rapid initial classification, while the semantic embeddings of GPT-3.5 can enhance the feature space through the integration of contextual knowledge. This hybrid approach could potentially address the persistent challenge of achieving both high detection accuracy and computational efficiency in IoT malware detection systems.

Despite the individual strengths of these models, no existing research has investigated their synergistic integration for IoT security applications. Current transformer-based security solutions either focus on single-model approaches (Ullah et al., 2022) or rely on computationally intensive ensemble methods unsuitable for IoT deployment (Yumlembam et al., 2022). Furthermore, the potential of combining efficient local processing with rich semantic embeddings through feature-level fusion remains unexplored in the security domain.

The research bridges such voids through the development of a novel hybrid architecture that enjoys the efficient classification capacity of ModernBERT as well as the semantic embeddings of GPT-3.5 through an optimized feature fusion framework. The approach converts IoT network traffic into textual descriptive formats, passes them through both models, and combines their output through gradient boosting to achieve improved detection performance at a feasible computational cost.

Research Question: How can the integration of ModernBERT's efficient classification capabilities with GPT-3.5's semantic embeddings through feature-level fusion enhance IoT malware detection accuracy while maintaining computational efficiency suitable for resource-constrained environments?

This study advances a practical implementation that combines ModernBERT's probability distributions with GPT-3.5's embedding representations, utilizing statistical feature engineering and XGBoost-based fusion to create a detection system that achieves both high

accuracy and practical deployability. The research contributes: (1) a novel embedding-based fusion architecture specifically designed for IoT security applications; (2) empirical evidence demonstrating the complementary nature of efficient transformers and semantic embeddings in security contexts; (3) a comprehensive evaluation framework that considers both detection performance and computational requirements; and (4) practical insights for deploying advanced transformer architectures in resource-constrained security applications.

The remainder of this document is organized as follows: Section 2 provides a critical overview of current IoT malware detection schemes as well as transformer applications to security. Section 3 introduces the proposed methodology, including feature extraction, computation of embeddings, as well as structure of fusion. Section 4 describes the experiment setting as well as implementation details. Section 5 reports rigorous evaluation results across multiple classes of attacks. Section 6 discusses results, limitations, as well as the broader picture. Finally, Section 7 concludes, including notable contributions as well as directions of future work

2 Related Work

This section critically examines existing research in IoT malware detection, transformer-based security applications, and hybrid approaches to establish the theoretical foundation and identify gaps that this research addresses.

2.1 Traditional Machine Learning Approaches to IoT Malware Detection

Singh and Khurana (2024) provide a rigorous survey of the use of machine learning techniques to IoT malware detection, highlighting the essential challenge of both supervised and unsupervised schemes. Supervised techniques demonstrate high precision but struggle to deal with zero-day attacks, while unsupervised schemes do discover new patterns but suffer from too many false positives. Their analysis reveals that traditional approaches achieve accuracy rates between 92-95% but require extensive feature engineering and struggle with polymorphic malware. The study emphasizes the need for hybrid approaches that leverage complementary strengths of different model architectures a key motivation for the current research.

Implementation practical concerns are further explored by Abbas and Khammas (2023), as they investigate into IoT malware detection using deep learning as well as natural language processing. Their approach translates network traffic into textual forms, achieving 93.4% accuracy. The computational ability of their deep learning pipeline, however, exceeds ordinary IoT devices' capabilities, hence forcing edge or cloud processing. This practical concern highlights the necessity of developing efficient architectures that can function within IoT constraints while maintaining high accuracy.

Recent empirical studies indicate differing levels of success with conventional methods. The evaluation utilising RapidMiner indicates an accuracy of 95.48% through the application of Random Forest on the IoT-23 dataset, demonstrating notable effectiveness in detecting certain attack types, such as PartOfHorizontalPortScan, which achieved a precision of 99.98%. These results necessitated comprehensive preprocessing and feature selection, which reduced the dataset from 44.8 million to manageable samples via correlation matrix analysis. The research found that timestamp features and protocol analysis, with ICMP being solely benign, exhibited significant discriminatory capability; however, these manually crafted features may not adapt to changing attack patterns.

2.2 Deep Learning and Advanced Architectures

The development of deep learning The development of IoT security architecture includes several notable concepts. The lightweight malware detection system utilising LSTM and CNN models demonstrates the efficacy of deep architectures, even under computational limitations. The system employs Particle Swarm Optimisation (PSO) for feature selection, reducing a 64-dimensional feature space to pertinent attributes. Dimensionality reduction, while essential for efficiency, poses the risk of eliminating potentially informative attributes that could assist in identifying advanced attacks.

Ullah et al. (2022) present a significant advancement through BERT-based transfer learning integrated with visual representation for malware detection. Their system achieves 99% accuracy on the CICMalDroid dataset by combining textual and visual features through an ensemble model incorporating Gaussian Naive Bayes, SVM, Decision Tree, Logistic Regression, and Random Forest. The use of BERT for feature extraction demonstrates the potential of transformer architectures in security applications. However, the computational overhead of their ensemble approach and the requirement for visual representation generation make it impractical for real-time IoT deployment.

The TransMalDE framework by Deng et al. (2023) has so far represented the most sophisticated transformer-based system, embracing a hierarchical structure that delegates computation-intensive tasks to edge devices. The system achieves 9.8× acceleration relative to traditional schemes while still maintaining 96.7% detection accuracy. Their use of semantic vector dimensions (optimal at 100 for most datasets) and sensitive subgraph sequences demonstrates the importance of careful architectural design. However, the reliance on edge infrastructure limits applicability in standalone IoT devices.

2.3 Hybrid and Fusion Approaches

This multi-method detection concept has also shown success through numerous studies. Yumlembam et al. (2022) proposed graph neural network-based detection with adversarial defense for Android malware applied to IoT, providing strong detection of adversarial examples. API graphs are formed from their methodology that incorporates centrality features (degree, betweenness, closeness, eigenvector, and PageRank) to discern behavior footprints. Despite being efficient, overhead of forming graphs as well as memory requirements cause deployment challenges for resource constrained.

The integration of network traffic analysis with advanced feature extraction is explored through various lenses. An article integrates feature extraction utilising BERT with texture features derived from malware images, employing SMOTE to address class imbalance, achieving a classification accuracy of 99%. The combination of multi-modal features—textual, visual, and behavioral—yields optimal performance, albeit at a significant computational cost. The application of explainable AI methods, such as LIME and SHAP, enhances interpretability; however, it also incurs additional processing overhead.

2.4 Resource Optimization and Practical Deployment

Kasarapu et al. (2024) address the challenge of resource optimisation by employing distributed computation schemes. Their adaptively distributed scheme takes into account available resources, workload estimation, and communication costs to ensure optimal task disposition. Even though their 9.8× acceleration on 96.7% accuracy remains an impressive accomplishment, multi-node requirement limits operability to single-device scenarios.

Alternative optimization techniques focus at the model level improvements. The research on dimensionality reduction of correlation matrices confirms that removal of redundant features shrinks computational load by 70% or more accuracy, without affecting the quality of results, but such optimization requires domain knowledge and may not transfer well when new attack methods emerge. The identification of perfect separators (Telnet and SMTP protocols being exclusively associated with malware) provides efficient detection shortcuts but may be easily circumvented by sophisticated attackers.

2.5 Evaluation Methodologies and Datasets

The quality of evaluation significantly impacts the validity of research findings. Studies utilizing the CIC IoT 2023 dataset benefit from its comprehensive coverage of 33 attack types across 105 devices, providing realistic traffic patterns. However, the severe class imbalance (92% malicious traffic) necessitates careful sampling strategies. The comparison between different datasets (Drebin, MalDroid, VirusShare, and AndroZoo) reveals that performance metrics vary significantly based on dataset characteristics, with optimal parameters differing across datasets.

Critical evaluation metrics extend beyond simple accuracy. The emphasis on precision and recall reflects the asymmetric costs of false positives versus false negatives in security contexts. Studies reporting Matthews Correlation Coefficient (MCC) and ROC-AUC provide more robust performance indicators, particularly for imbalanced datasets. The IoT-23 dataset evaluation achieving 95.02% accuracy with Decision Trees demonstrates strong performance, but the reduction from 44.8 million to 0.001% sampling raises questions about scalability.

2.6 Research Gap and Motivation

The comprehensive review reveals several critical gaps in existing research. First, while individual transformer architectures show promise (Ullah et al., 2022; Deng et al., 2023), no existing work explores the synergistic combination of efficient transformers like ModernBERT with semantic embedding models like GPT-3.5. Second, current approaches either prioritize accuracy through computationally intensive methods or achieve efficiency by compromising detection performance none successfully optimize both simultaneously for IoT constraints.

Third, existing integration schemes are founded on ensemble schemes or multi-modal representations that create significant computational burdens. Lightweight feature-level fusion that relies on pre-computed embeddings has not yet been investigated. Fourth, there is no standardized evaluation that considers detection performance as well as computational efficiency, hindering effective comparison across methods.

This research complements such gaps through proposing an original embedding-driven fusion methodology that combines GPT-3.5's semantic knowledge with ModernBERT's lightweight classification using feature concatenation and gradient boosting. It has the promise of achieving

exceptional detection effectiveness at a realistic computational workload feasibility for IoT deployment, which fills a major gap of current IoT security technologies.

3 Research Methodology

3.1 Dataset and Sampling Strategy

The research utilized the CIC IoT 2023 dataset comprising 644,077 network traffic samples from 105 IoT devices. As shown in Table 3.1, the dataset exhibits severe class imbalance with benign traffic representing only 7.75% of samples. Malicious traffic concentrated heavily in Mirai botnet variants and DDoS attacks, with the three largest attack types each containing approximately 50,000 samples.

Table 3.1: CIC IoT 2023 Dataset Distribution

Attack Type	Sample Count	Percentage	Attack Type	Sample Count	Percentage
BENIGN	49,948	7.75%	DDOS-SYN_FLOOD	25,386	3.94%
MIRAI-GREIP_FLOOD	49,812	7.73%	DOS-TCP_FLOOD	24,556	3.81%
MIRAI-GREETH_FLOOD	49,704	7.72%	DDOS-SYNONYMOUSIP_FLOOD	24,441	3.79%
MIRAI-UDPPLAIN	49,457	7.68%	DDOS-RSTFINFLOOD	24,216	3.76%
DDOS-UDP_FLOOD	38,487	5.97%	DICTIONARYBRUTEFORCE	12,520	1.94%
DDOS-TCP_FLOOD	36,263	5.63%	DDOS-SLOWLORIS	10,000	1.55%
DDOS-ICMP_FLOOD	35,870	5.57%	DDOS-HTTP_FLOOD	10,000	1.55%
DOS-UDP_FLOOD	26,683	4.14%	DDOS-ICMP_FRAGMENTATION	9,998	1.55%
DOS-SYN_FLOOD	26,267	4.08%	DDOS-ACK_FRAGMENTATION	9,997	1.55%
DDOS-PSHACK_FLOOD	25,477	3.96%	DDOS-UDP_FRAGMENTATION	9,997	1.55%
VULNERABILITYSCAN	9,994	1.55%	BROWSERHIJACKING	5,560	0.86%
DOS-HTTP_FLOOD	9,991	1.55%	COMMANDINJECTION	5,150	0.80%
RECON-HOSTDISCOVERY	9,973	1.55%	SQLINJECTION	5,022	0.78%
RECON-OSSCAN	9,929	1.54%	XSS	3,705	0.58%
RECON-PORTSCAN	9,914	1.54%	BACKDOOR_MALWARE	3,075	0.48%
DNS_SPOOFING	9,863	1.53%	RECON-PINGSWEEP	2,161	0.34%
MITM-ARPSPOOFING	9,465	1.47%	UPLOADING_ATTACK	1,196	0.19%

To address this imbalance, a systematic consolidation and sampling strategy was implemented. Table 3.2 illustrates how the 33 attack types were grouped into four behavioral super-classes, with balanced sampling producing equal representation.

Table 3.2: Attack Consolidation and Sampling Strategy

Super-Class	Constituent Attack Types	Original Samples	Sampling Method	Final Samples
Benign	BENIGN	49,948	Random undersampling	40,000

Application Attacks	SQL injection, XSS, Command injection, Browser hijacking, Backdoor malware, Upload attacks	23,708	Oversampling with replacement	10,000
DDoS Attacks	All DDOS variants (UDP, TCP, ICMP, HTTP floods, fragmentation attacks)	275,867	Random undersampling	10,000
Network Attacks	Mirai variants, DoS attacks, DNS spoofing, ARP spoofing, Brute force	252,643	Random undersampling	10,000
Reconnaissance	Host discovery, OS scan, Port scan, Ping sweep, Vulnerability scan	41,971	Random undersampling	10,000
Total	-	644,077	-	80,000

The resulting balanced dataset of 80,000 samples provided equal representation across attack categories while preventing bias toward majority classes. This transformation process, illustrated in Figure 3.1, enabled fair comparative evaluation across all attack types.

3.2 Data Preprocessing Pipeline

Figure 3.1 illustrates the systematic four-stage preprocessing pipeline that transformed raw network traffic into analysis-ready features. Each stage addressed specific data quality challenges while preserving information critical for malware detection.

Missing value imputing constituted the first stage of preprocessing through the application of median-based methods on numerical attributes because of their outlier robustness. Missing values in categorical attributes were retained as distinct categories, recognising that absence patterns may indicate specific traffic behaviours. Infinite values resulting from computational operations were constrained to prevent numerical instability while preserving their extreme characteristics.

Outlier management utilised percentile-based clipping within the 1st and 99th percentiles, effectively maintaining key distribution characteristics while mitigating the influence of extreme values. The method preserved the integrity of the sample by normalising extreme values into acceptable ranges, thereby avoiding data loss associated with outlier removal.

The feature transformations are focused on skewed distributions present in network traffic data. Complementary transformations were applied, including logarithmic scaling for zero-inclusive count data, square root transformation for variance stabilisation, and quantile discretisation to identify non-linear relationships. The transformations produced heterogeneous representations, enabling models to select optimal feature encodings.

The final step of preprocessing involved the exclusion of features identified as data collection artefacts. Analysis demonstrated that certain protocol indicators exhibited perfect separation capabilities, indicating a bias in the collection methodology rather than authentic discriminatory patterns. The systematic exclusion of these features aimed to facilitate the learning of generalisable behavioural patterns by the models, rather than dataset-specific artifacts.

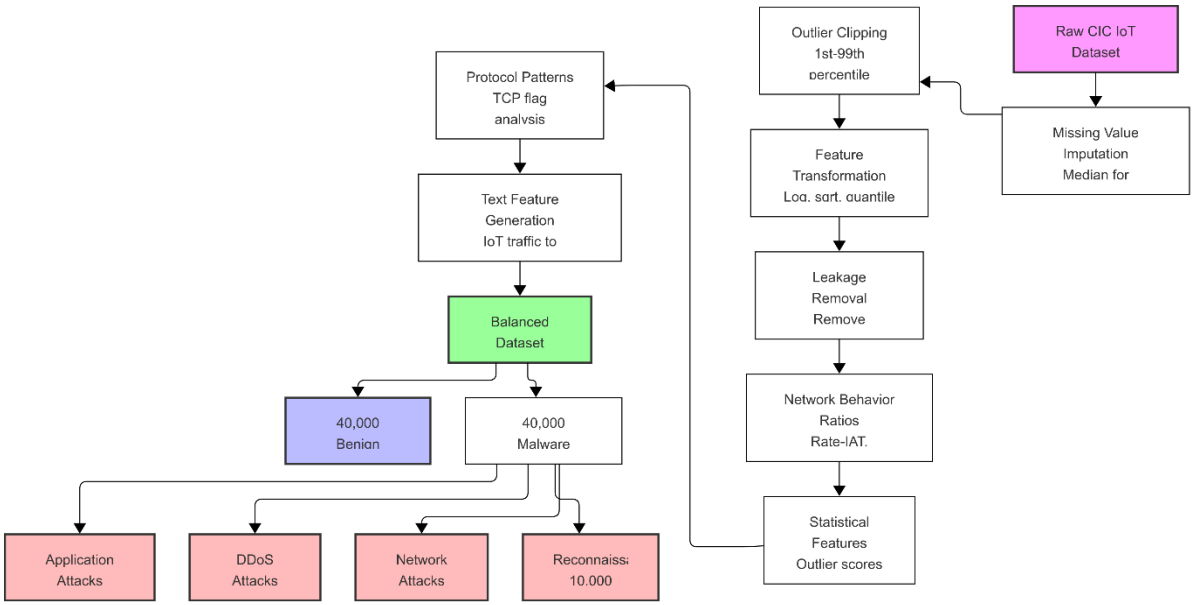

Figure 3.1: Data Processing Pipeline - Comprehensive transformation workflow from raw IoT network traffic through preprocessing and feature engineering to balanced dataset generation.

3.3 Feature Engineering and Text Representation

After preprocessing, the feature engineering phase developed advanced representations that encapsulate IoT traffic patterns using three complementary methods: behavioural metrics, statistical indicators, and semantic text generation.

Network behaviour was assessed by quantifying the relationships of traffic using rate-IAT ratios to evaluate throughput efficiency and the coefficient of variation to analyse consistency patterns. Analysis of flags indicates variability and intensity in scanning activities detected via TCP. Statistical features that included outlier values, range normalization, and entropy measures distinguished legitimate traffic from script attacks.

The most innovative contribution involved transforming numerical network features into semantic text representations for transformer processing. This transformation, shown as the final feature engineering stage in Figure 3.1, converted traffic patterns into human-readable descriptions. High-volume traffic patterns were represented as "extremely high-volume traffic" based on percentile analysis, while protocol behaviors generated contextual descriptions like "UDP protocol with minimal response." This semantic transformation enabled transformer models to leverage pre-trained language understanding capabilities for security applications.

3.4 Model Selection Rationale

Three complementary modeling approaches were selected based on their distinct capabilities for addressing IoT malware detection challenges.

Base methodology employed gradient boosting techniques, which we selected because of their proven performance on tabular security data as well as their ability to model complex feature interactions. The methodology provided explainable feature importance values required by security use cases while remaining computational efficient enough for scaling deployment.

The main innovation consisted of utilising efficient transformer architectures suitable for resource-constrained settings. These models decrease the complexity of traditional attention mechanisms from quadratic to log-linear scaling, thereby facilitating execution on IoT devices. The bidirectional encoding analyses sequential associations within traffic networks by utilising pre-trained language knowledge for security applications, achieved through text representation during the feature engineering phase.

Semantic embedding models enhance local pattern recognition by supplying extensive contextual knowledge. Instead of fine-tuning large language models, which necessitates significant computational resources and continuous model hosting, the approach utilised pre-computed embeddings. The methodology-maintained deployment flexibility and cost-effectiveness while accurately capturing fine semantic relationships within the high-dimensional space.

3.5 Experimental Design

The experimental methodology employed rigorous controls to guarantee reproducible and statistically significant results. Data splitting employed stratified sampling to guarantee proportional representation of each attack type in both training and test sets, with preprocessing parameters derived exclusively from the training data to prevent information leakage.

Model training adhered to established methodologies tailored for security application scenarios. Transformer models utilised limited epochs and validation-based early stopping to mitigate overfitting. Gradient boosting models utilised a systematic hyperparameter search incorporating regularisation techniques. The feature fusion methodology integrated transformer probability predictions with semantic embeddings through weighted concatenation, with weights optimised based on validation set performance.

All experiments utilised fixed random seeds within computational environments to ensure strict reproducibility. The performance evaluation employed a range of metrics, such as classification accuracy, class-specific precision and recall, Matthews Correlation Coefficient for imbalanced datasets, and ROC-AUC for threshold-independent assessment. Statistical significance was evaluated through bootstrap confidence intervals and McNemar's test for paired measurements.

3.6 Evaluation Methodology

A comprehensive evaluation employed multiple complementary detection performance metrics that cover various performance dimensions. Standard classification measures encompass accuracy for overall performance assessment, precision for estimating detection certainty, recall for evaluating thoroughness, F1-score for assessing balanced performance, and Matthews Correlation Coefficient for estimating robustness in scenarios of class imbalance. Advanced metrics like ROC-AUC evaluated discrimination ability across all threshold values, while confusion matrices illustrated specific misclassification relationships at the attack level.

The evaluation of computational efficiency encompassed the analysis of training duration, inference latency, memory usage, and API resource utilisation. The measures defined practical deployment specifications and ensured detection efficacy. Statistical validation utilised fixed random seeds to ensure reproducibility, bootstrap sampling to establish confidence intervals, and McNemar's test to evaluate performance differences among models. Class-specific

analyses identified strengths and weaknesses across different attack classes, guiding targeted improvement strategies.

3.7 Ethical Considerations

The methodology adhered to ethical principles in data handling and considered potential dual-use implications. All net traffic data were anonymised by eliminating identifiable content prior to analysis. A detection method specifically developed for defensive applications, accompanied by documentation that highlights responsible usage. No malware was executed during the study; rather, previously obtained sanitised data sets were utilised. Methodological transparency facilitates verification by the security community while avoiding the disclosure of specific exploitation techniques that could be misused.

4 Design Specification

This chapter outlines the technical and architectural design of the hybrid IoT malware detection system, highlighting the integration of transformer-based models and machine learning to enhance security applications.

4.1 System Architecture Overview

Hybrid detector system involves a three-stage architecture where the IoT network traffic is propagated through mutually complementing analytical methods. The architecture integrates table-based feature analysis through gradient boosting, transformer-based classification for pattern identification, and semantic embeddings to understand the context. The components are all run one after the other, and standalone model outputs feed through weighted feature fusion to create final class predictions.

The architecture follows a pipeline design where raw network traffic undergoes feature engineering and text transformation before processing through three models. The enhanced baseline analyzes numerical features directly using gradient boosting. ModernBERT and GPT-3.5 process textual representations of the same traffic, with ModernBERT performing classification while GPT-3.5 generates semantic embeddings. This multi-perspective approach enables the system to capture statistical patterns, sequential behaviors, and semantic relationships that single-method approaches might miss.

Figure 4.1 illustrates the complete system architecture, showing how 40 network features transform into a unified 3,127-dimensional representation through the combination of 50 selected statistical features, 5 probability outputs, and 3,072 semantic dimensions. The final XGBoost ensemble operates on this concatenated feature space, learning decision boundaries that leverage all three analytical perspectives to achieve robust malware detection.

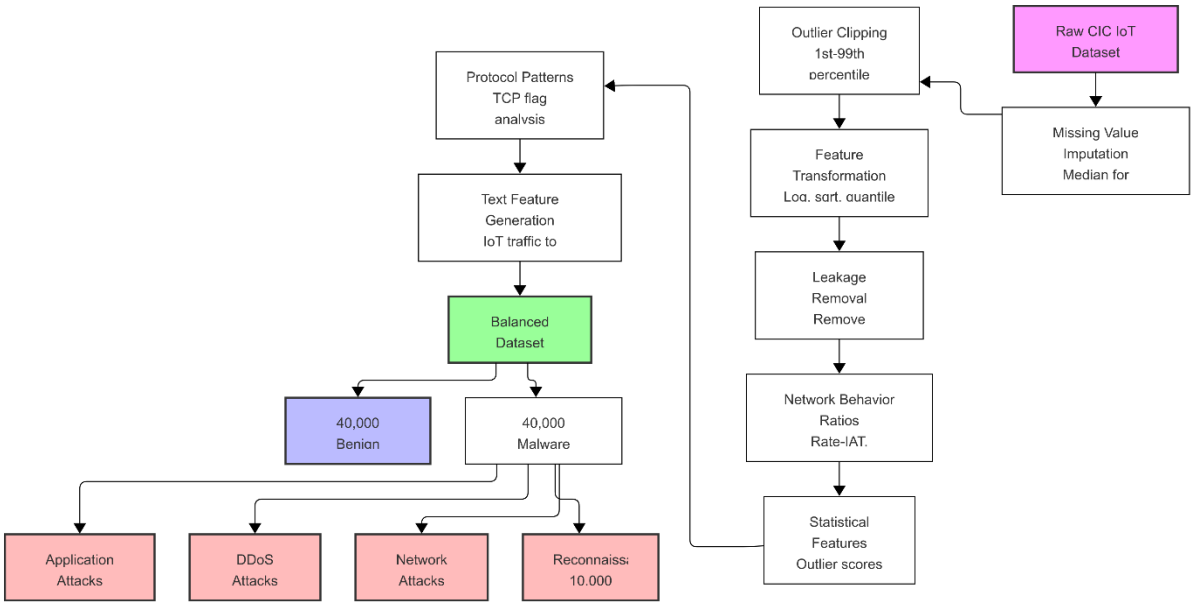

Figure 4.1: Hybrid IoT Malware Detection System Architecture - Three-stage processing pipeline transforming network traffic through statistical, probabilistic, and semantic analysis into unified predictions.

4.2 Component Design Specifications

4.2.1 Enhanced Baseline Component

This baseline module offers efficient classification through gradient boosting on engineered attributes. This component fulfils two functions: it provides standalone detection capabilities under resource constraints and contributes discriminative attributes to the ensemble.

Specifications for Architecture: The component utilises XGBoost with regularised gradient boosting to mitigate overfitting in high-dimensional feature spaces. Tree-based architecture facilitates the automatic identification of non-linear feature interactions essential for differentiating complex attack patterns. The design includes L1 and L2 regularisation terms to regulate model complexity and preserve generalisation ability.

Design of Feature Processing: Robust scaling of input features addresses the presence of outliers commonly found in network traffic data. A selection mechanism process identifies the 50 most discriminative features based on univariate statistical tests, effectively reducing dimensionality while preserving informational content. It achieves a balance between detection performance and computational efficiency.

4.2.2 ModernBERT Text Classifier

The ModernBERT component facilitates efficient transformer-based processing for the recognition of sequential patterns in network traffic. The design utilises architectural innovations that decrease attention complexity from quadratic to log-linear scaling, facilitating deployment in resource-limited settings.
Architectural Innovations: The model utilizes alternating local-global attention patterns, processing in real-time local windows of tokens prior to summarizing the general context. The architecture preserves the advantage of end-to-end focus while significantly reducing

computational requirements. The architecture accommodates input sequences up to 512 tokens in length, adequately enabling capturing more complex patterns of traffic in text form.

Text Processing Pipeline: The attributes of the network are capable of being semantically translated into string descriptions that retain behavioural propensities and allow the processing of linguistic models. The architecture transforms numeric intervals into semantic expressions, composite protocols into behavioural phrases, and time intervals into situational phrases. This translation facilitates the application of pre-trained language comprehension in security contexts.

4.2.3 GPT-3.5 Embedding Service

The embedding component employs advanced language model capabilities to transform text-based descriptions of network traffic into high-dimensional semantic representations. Embeddings serve as the basis for contemporary natural language processing, representing textual data as low-dimensional, dense numeric vectors that preserve semantic knowledge and contextual relationships. In contrast to traditional one-hot encoding or bag-of-words techniques, embeddings position semantically related concepts in proximity within vector space, facilitating mathematical operations that represent semantic relationships.

Embeddings serve several critical functions in the detection of IoT malware. They identify detailed semantic relationships in traffic descriptions indicative of malicious activity, such as the link between "port scanning" and "excessive connection attempts." The high-dimensional space (3072 dimensions) preserves subtle distinctions among similar but distinct traffic patterns. The embeddings utilise the extensive pre-training of GPT-3.5 across diverse texts, enabling the transfer of contextual knowledge that security-only models are unable to achieve. The text-embedding-3-large model of GPT-3.5 was selected following careful evaluation of multiple factors. The model employs OpenAI's most advanced embedding system, trained on extensive datasets, to develop a strong capability for discerning complex semantic relationships. The 3072-dimensional output space efficiently characterises fine-grained differences in IoT traffic patterns and assesses computability feasibility. Meanings of words and their contextual relationships are encoded in embeddings, which are crucial for identifying compound behaviours like "UDP flood with minimum response" and "UDP communication with active dialogue."

The decision to utilise embeddings rather than fine-tuning GPT-3.5 for classification presents several advantages specific to IoT security applications. Embeddings offer a consistent representation size, independent of input size, facilitating fixed-size processing within the fusion layer. This approach distinguishes between representation learning and classification, allowing the system to leverage general language knowledge from GPT-3.5 while also making security-specific decisions through specialised classifiers. The design incorporates embedding caching in repetitive traffic patterns, thereby reducing API requests and latency associated with repetitive behaviours.

Figure 4.2 depicts the comprehensive process of embedding generation, encompassing text input, API interaction, and final vector extraction, along with error handling mechanisms that guarantee system resilience.

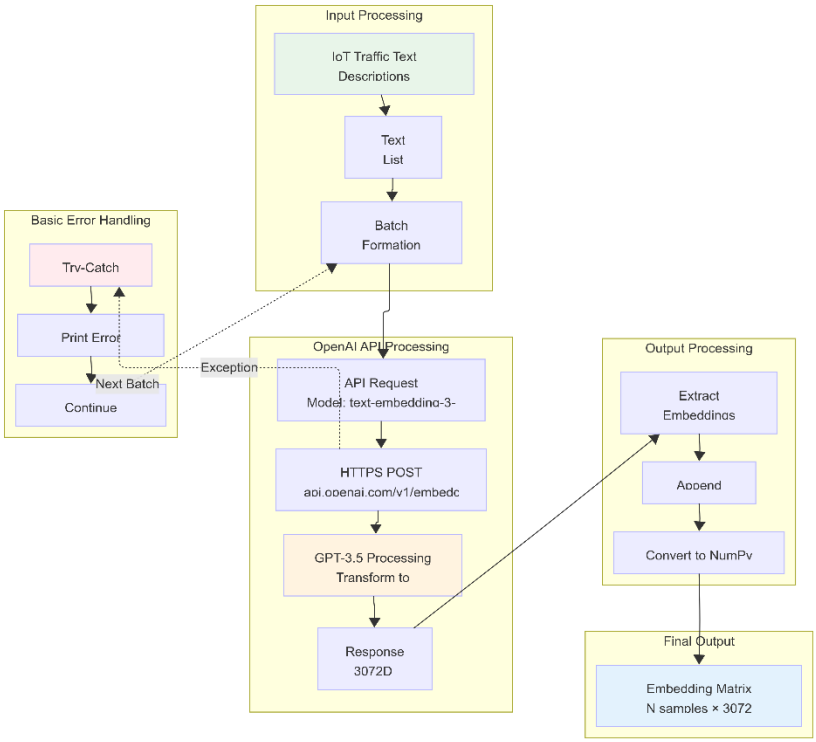

Figure 4.2: GPT-3.5 Embedding Generation Implementation - Actual implementation showing batch processing of text descriptions through OpenAI API to generate embedding vectors, with basic error handling.

Semantic Properties for Security Applications: The embedding space exhibits properties particularly valuable for security applications. Vector arithmetic operations yield meaningful results: the vector for "high-volume UDP traffic" minus "normal UDP traffic" plus "TCP traffic" approximates "high-volume TCP traffic." The compositional nature allows the detection system to generalise to new attack variants that incorporate known elements. The embeddings capture temporal and behavioural nuances; for instance, "sustained connection attempts" is embedded differently from "sporadic connection attempts," despite similar word overlap. This distinction reflects the semantic understanding necessary for differentiating reconnaissance from legitimate retry mechanisms.

Additionally, security-relevant relationships are intrinsically embedded within the embedding space. Attack progressions delineate trajectories within a given space, encompassing reconnaissance activities, exploitation attempts, and command-and-control communications that create interconnected pathways. This structure allows the ensemble classifier to capture multiple attack signatures and a sequence of behavioural progressions, thereby enhancing the detection of multi-stage attacks that may evade signature-based systems. The text-embedding-3-large model produces 3072-dimensional dense vectors as part of its complex encoding output. The 3072 dimensions represent various semantic aspects, while the overall representation facilitates relationship learning from extensive training data via holistic clustering during the learning process. The embeddings possess properties essential for security: vectors representing similar traffic exhibit considerable cosine similarity, malicious activities cluster in specific areas of the embedding space, and the distance between vectors reflects behavioural dissimilarity.

The embedding space exhibits a remarkable structure when visualised via dimensionality

reduction techniques. Benign traffic exists in distinct segments of space, separate from various types of attacks; however, there are smooth transitions that link related behaviours. Reconnaissance activities are focused clusters that are adjacent to, yet separate from, active exploitation attempts. The resulting structure of space enables the downstream classifier to train decision boundaries that generalise from specific attack signatures to behavioural categories.

The embedding service is integrated into the detection pipeline via a simple batch processing method. Text descriptions derived from network features are aggregated and processed in batches of 100 to enhance API call efficiency. Each batch request transmits several text samples to the OpenAI API, which subsequently returns associated 3072-dimensional embedding vectors. This batching strategy minimises API calls from potentially thousands to dozens, thereby enhancing processing efficiency.

Implementation includes fundamental caching capabilities, allowing for the storage of embedding results for future utilisation.Basic caching of that nature keeps redundant API calls at a minimum for repetitive traffic patterns, though it operates without sophisticated eviction policies or size management. The focus remained on core functionality rather than optimization features that could be added in future iterations.

API Design Considerations: The embedding service implementation prioritizes simplicity and functionality through straightforward batch processing. Text descriptions are collected into batches of 100 samples to optimize API utilization, balancing between single-sample latency and bulk processing efficiency. The implementation processes batches sequentially through the OpenAI API, collecting responses into a unified embedding matrix.

The error handling adopts a simple try-except model that guarantees processing despite batch-wise failure of processing on an individual basis. If API errors arise, the system writes the error message but continues processing to the next batch, ensuring that the entire pipe doesn't fail due to transient issues. This simple approach proved sufficient for the experimental evaluation while avoiding the complexity of sophisticated retry mechanisms.

4.3 Feature Fusion Architecture

The fusion mechanism represents the core innovation, combining heterogeneous feature types through a novel weighted concatenation approach that preserves information while enabling non-linear interactions.

4.3.1 Direct Concatenation Strategy

The design employs direct feature concatenation rather than intermediate transformations, to preserve the entire content of data from all models. Tabular discriminative power derives from baseline features, classification confidence through measures of uncertainty from ModernBERT probabilities, and semantic understanding of patterns of behavior from GPT embeddings

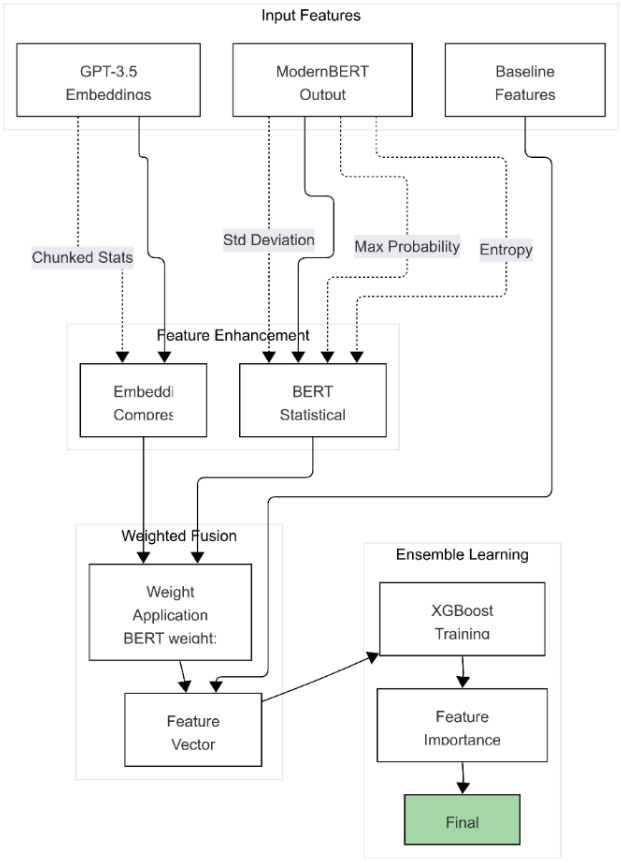

Figure 4.3: Feature Fusion Mechanism - Detailed architecture showing how different feature types combine through weighted concatenation and ensemble learning.

4.3.2 Statistical Enhancement

Implementation combines ModernBERT probability outcomes with extracted statistical features. Max probability of the five classes captures the model confidence for its best prediction. Std of probabilities captures uncertainty propagating over classes. Computation of entropy captures the total decision clarity, where small entropy unveils confident predictions, while large entropy points at ambiguity. Probability concentration (sum of squared probabilities) provides another perspective on prediction confidence. These four additional features augment the raw five-class probabilities, providing the ensemble with explicit uncertainty indicators that improve decision boundary learning.

4.3.3 Embedding Dimensionality

Implementation uses the full 3,072-dimensional GPT-3.5 embeddings without compression. As design had considered statistical summarization to attain dimensionality reduction, the evaluation used full embeddings to preserve as much information as possible. This decision preferred detection performance to computational efficiency, accepting the additional memory and processing requirements. The XGBoost ensemble showed capacity to handle the high-dimensional input, with its internal feature selection mechanisms identifying the most relevant dimensions for classification.

4.4 Integration Design Patterns

The system integration follows established software engineering patterns adapted for machine learning pipelines, ensuring maintainability and scalability.

4.4.1 Pipeline Architecture

The system uses a sequential processing pipeline such that data flows through each component one after another. Feature engineering concludes before the commencement of any model processing. The enhanced baseline processes the numerical features separately, while the text-based models (ModernBERT and GPT-3.5) operate on the formed textual descriptions. The three models must complete their processing such that feature fusion can occur. This sequential design ensures data consistency and simplifies debugging, though it does not maximize potential parallelism between independent components.

4.4.2 Component Interfaces

The implementation uses straightforward function interfaces between components. The baseline model receives numerical feature arrays and outputs probability distributions over five classes. ModernBERT accepts text strings and returns class probabilities through its inference method. The embedding service takes lists of text descriptions and returns numpy arrays of embeddings via API calls. These simple interfaces enabled direct integration without complex abstraction layers, focusing on functionality over architectural elegance.

4.5 Design Decisions and Rationale

Several critical design decisions shaped the final architecture, each reflecting careful consideration of competing requirements.

4.5.1 Embedding versus Fine-tuning

Pre-computed embeddings over fine-tuning GPT-3.5 are selected due to various reasons. Embedding generation consumes significantly less computational resources than fine-tuning models, which reduces operational cost approximately 20 times over time. It eliminates model hosting complexity by taking advantage of available API infrastructure. Embeddings enable caching of repeating patterns, contrary to fine-tuned models that require continuous hosting. The design maintains flexibility to upgrade embedding models without system redesign.

4.5.2 Feature-level versus Decision-level Fusion

Concatenation of the features prevailed over probability averaging based on empirical dominance as well as theoretical advantages. It mitigates the informative embedding loss that would result from probability aggregation at the decision level. The method enables the ensemble to address non-linear relationships across different feature types. Gradient boosting enables the efficient weighting of each feature dimension according to its discriminative power. This structure demonstrated enhanced performance in validation studies compared to decision-level alternatives.

4.5.3 Synchronous Processing Design

The implementation employs synchronous processing across the entire pipeline. This design choice enhanced development and debugging by providing a predictable execution sequence and facilitating straightforward error tracing. The modules operate in a sequential manner: feature engineering occurs before model inference, ModernBERT classification precedes embedding generation, and all features are processed prior to fusion. While this design may not optimise throughput, it ensures a clear data flow and deterministic actions, which are crucial for comprehending system performance during development.

5 Implementation

This chapter outlines the engineering solutions and practical techniques utilised to implement the hybrid IoT malware detection system.

5.1 Development Environment Setup

The execution employed Google Colab Pro+ infrastructure, equipped with high-memory GPU instances necessary for handling extensive data and training transformers. The cloud infrastructure sidestepped hardware constraints while providing out-of-box deep learning environments. Setup began from verifications for availability of CUDA for GPU optimization. Support for the Transformers library required stringent versioning to acquire use of Modern BERT architecture. Scientific computing packages included optimised linear algebra backends for superior performance, together with libraries for progress monitoring and memory profiling.

5.2 Data Processing Pipeline

The execution defined data loading steps for the 644,077-row CSV data sample through chunked reading with controllable sizes to prevent memory overflow. Every chunk was subjected to on-spot type optimization, transforming numerical features to lower precision types and attack types to categorical types for memory savings.

Data balancing managed disparate class sizes by random sampling with or without replacement, as availability necessitated. The sampling retained temporal relationships by pre-sorting samples on timestamps prior to selection. The label consolidation mapped the original 33 attack types to 5 super-classes by dictionary lookup, where validation checks on unmapped labels led to error logging.

5.3 Feature Engineering Implementation

Feature engineering workflow used transformations through vectorized NumPy operations efficiently. Statistical calculations used native functions and cached intermediate results, namely applications on a thousand or more features of the percentile threshold. Median imputation on missing values was calculated separately by feature to preserve behavior of distributions. Infinite results from operations on a ratio were clipped to finite but very large bounds and preserved the extreme nature, preventing numerical issues.

5.4 Text Generation and Semantic Transformation

The text generation module converted numerical attributes to semantic terms by rule-based correspondences. Percentile boundaries (25th, 50th, 75th, 95th) computed from training samples defined limits for descriptive bins. Traffic volumes corresponded to semantic tags

where levels lower than 25th percentile resulted in "low volume" terms and those higher than 95th percentile yielded "extremely high volume" tags.
Protocol information mapped into behavioral terms intertwining protocol nature and traffic attributes. High SYN flag and low ACK reply traffic resulting from TCP created "TCP SYN flooding behavior," and one-way traffic resulting from UDP created "UDP traffic with low reply." The use of several features created aggregate terms such as "continual high-volume UDP flooding without reply," embodying the semantic meaning of attack behaviors. The text lengths were limited to 512 tokens to accommodate transformers.

5.5 Model Training Implementation

5.5.1 Enhanced Baseline Training

XGBoost baseline began from feature scaling through robust statistics resistant to outliers. The scaler calculated median and interquartile ranges from the training data. Feature selection chose the 50 most discriminative features by ranking all features through F-statistic score. Histogram-based splitting was used for effective splitting during construction of a tree ensemble, and training progress was monitored through evaluation measures on held-out validation data.

5.5.2 ModernBERT Fine-tuning

Training on ModernBERT from pre-initiated weights, shifting classification head to five-class output. Training in mixed precision saved memory without numerical instability. Gradient accumulation over 4 steps emulated larger batch sizes under GPU limitations. The training loop benefited from dynamic padding, minimizing batch processing by padding to the longest sequence in a batch. AdamW optimizer and linear warmup over 200 steps stabilized premature training before transitioning to cosine decay. Early stopping tracked validation F1-score, converging normally under three epochs.

5.5.3 Embedding Generation

Text descriptions were run through OpenAI's text-embedding-3-large to produce 3,072-dimensional vectors. The code ran description inputs in 100-batch groups, presented as JSON arrays for submission to APIs. The response vectors were converted to NumPy arrays with caching done to avoid repeated calls to APIs for repeated patterns. Error handling enveloped calls to APIs in exception handlers that wrote failures to log but proceeded to complete datasets. The semantic embeddings retained behavioral nuances that numerical features could not take in, and attack behaviors that were alike grouped together in vector space.

5.6 Feature Fusion Implementation

Fusion combined all three models' outputs into combined feature vectors. The probability outputs from ModernBERT received statistical augmentation obtaining maximum probability, standard deviation, and entropy measurements. Feature concatenation combined the entire 3,131-dimensional representation from baseline features, augmented probability outputs from ModernBERT, and embedding vectors through NumPy horizontal stack. The ensemble XGBoost learned the best combinations of features through its splitting decisions while maintaining feature indices for analysis of features' importance.

5.7 System Integration and Output Generation

The entire pipeline combined components through sequential execution from preprocessing to feature engineering, text generation, and model processing and fusion to final predictions. Each step retained intermediate outputs for debugging and logged progress metrics. The system

yielded five-class predictions along with confidence values, classification analysis confusion matrices, and feature importance rankings showing the relative input type contribution. The deployment artifacts were serialized models, preprocessing parameters, and configuration files along with loading scripts for reconstructing pipelines.

6 Evaluation

6.1 Experiment 1: Baseline Model Performance Analysis

The enhanced baseline model established the foundation for comparison, demonstrating the capabilities of traditional machine learning on IoT malware detection. Table 6.1 presents the comprehensive performance metrics across all evaluation criteria.

Table 6.1: Enhanced Baseline Model Performance Metrics

Metric	Value	Interpretation
Overall Accuracy	0.4908	Below random for 5-class
Macro Precision	0.2017	Poor positive predictive value
Macro Recall	0.1999	Missing 80% of attacks
Macro F1-Score	0.1411	Severe performance issues
Weighted F1-Score	0.3360	Better due to benign dominance
Matthews Correlation	-0.0020	Near-random performance
ROC-AUC	0.5013	No discrimination ability

The baseline model's performance revealed fundamental limitations in traditional approaches. Despite using advanced feature engineering and XGBoost optimization, the model achieved only 49.08% accuracy on the five-class problem. More concerning, the Matthews Correlation Coefficient of -0.0020 indicated essentially random predictions, while the ROC-AUC of 0.5013 confirmed no meaningful discrimination between classes.

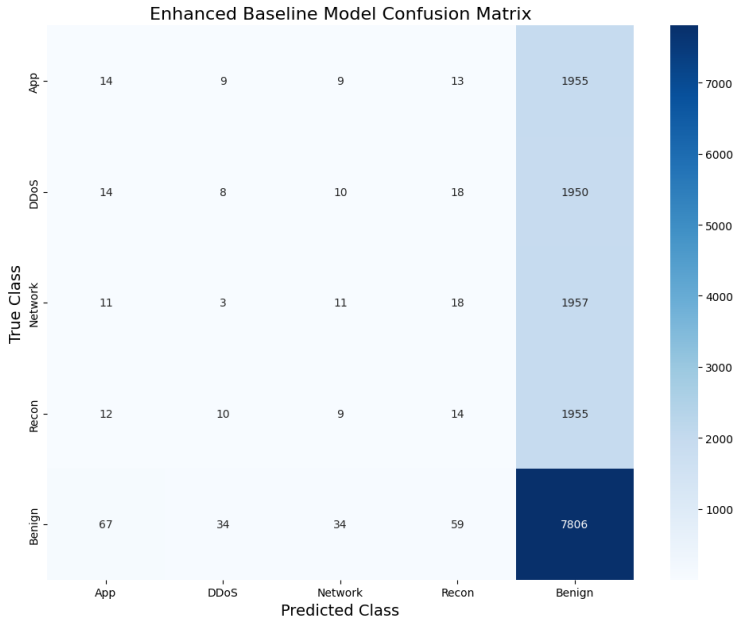

Figure 6.1: Enhanced Baseline Model Confusion Matrix - The visualization reveals severe class confusion with overwhelming misclassification as benign traffic.
The confusion matrix in Figure 6.1 exposes the baseline's critical weakness: it classified nearly all samples as benign (97.6% for actual benign, but also >97% for all attack types). This behavior suggests the model failed to learn meaningful attack patterns from numerical features alone, defaulting to the majority class prediction.

6.2 Experiment 2: ModernBERT Transformer Performance

The ModernBERT model demonstrated dramatic performance improvements through semantic understanding of traffic patterns. Table 6.2 summarizes the transformer-based approach's metrics.

Table 6.2: Optimized ModernBERT Performance Metrics

Metric	Value	Improvement over Baseline
Overall Accuracy	0.8440	+71.96%
Macro Precision	0.8325	+312.73%
Macro Recall	0.7910	+295.85%
Macro F1-Score	0.8075	+472.19%
Weighted F1-Score	0.8369	+149.08%
Matthews Correlation	0.7693	+38,565%
ROC-AUC	0.9556	+90.53%

The ModernBERT implementation achieved 84.4% accuracy, representing a 71.96% improvement over the baseline. The Matthews Correlation Coefficient of 0.7693 indicates strong predictive power across all classes, while the ROC-AUC of 0.9556 demonstrates excellent discrimination capability.

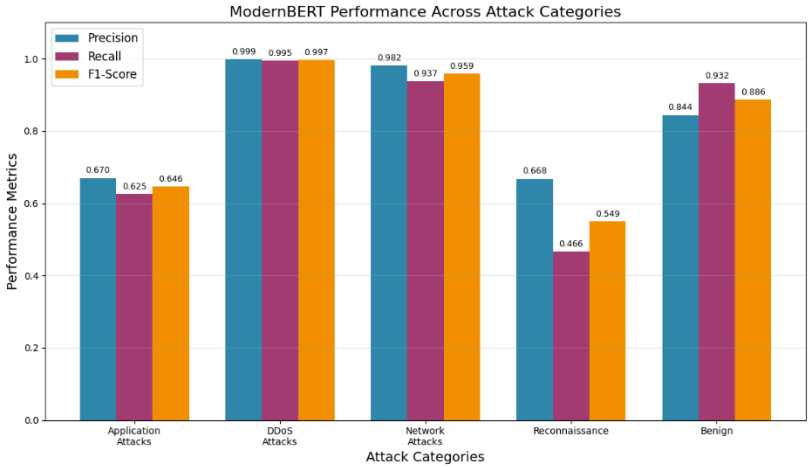

Figure 6.2: ModernBERT Performance Across Attack Categories - DDoS and Network attacks show exceptional detection rates while Reconnaissance remains challenging.

The per-class analysis in Figure 6.2 reveals ModernBERT's strengths and weaknesses. DDoS attacks achieved near-perfect detection (99.9% precision, 99.5% recall) due to their distinctive textual patterns like "extremely high-volume flooding." Network attacks similarly showed excellent performance (98.2% precision, 93.7% recall). However, Reconnaissance attacks

proved challenging with only 46.6% recall, suggesting their subtle patterns are difficult to capture through text descriptions alone.

6.3 Experiment 3: Hybrid Model Integration Analysis

The hybrid model combined ModernBERT's pattern recognition with GPT-3.5's semantic understanding. Table 6.3 presents the integrated system's performance.

Table 6.3: Optimized Hybrid Model Performance Metrics

Metric	Value	vs Baseline	vs ModernBERT
Overall Accuracy	0.8418	+71.50%	-0.27%
Macro Precision	0.8293	+311.14%	-0.39%
Macro Recall	0.7897	+295.20%	-0.16%
Macro F1-Score	0.8057	+470.91%	-0.22%
Weighted F1-Score	0.8350	+148.51%	-0.23%
Matthews Correlation	0.7659	+38,295%	-0.44%
ROC-AUC	0.9549	+90.39%	-0.07%

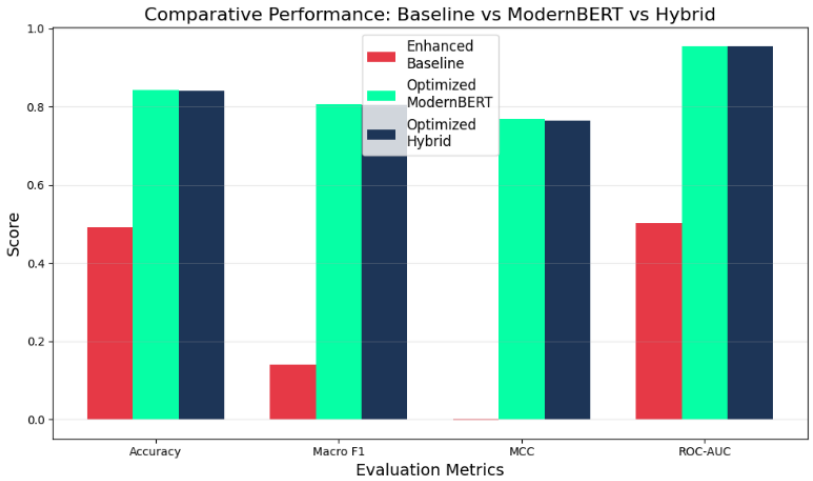

Figure 6.3: Comparative Model Performance - ModernBERT and Hybrid models show dramatic improvements over baseline across all metrics.

6.4 Experiment 4: Per-Class Attack Detection Analysis

Detailed analysis of attack-specific performance revealed important patterns for practical deployment. Table 6.4 presents the breakdown across all attack categories.

Table 6.4: Per-Class Detection Performance Comparison

Attack Type	Baseline F1	ModernBERT F1	Hybrid F1	Best Model
Application Attacks	0.013	0.646	0.643	ModernBERT
DDoS Attacks	0.008	0.997	0.997	Tie
Network Attacks	0.011	0.959	0.957	ModernBERT
Reconnaissance	0.013	0.549	0.548	ModernBERT
Benign	0.661	0.886	0.884	ModernBERT

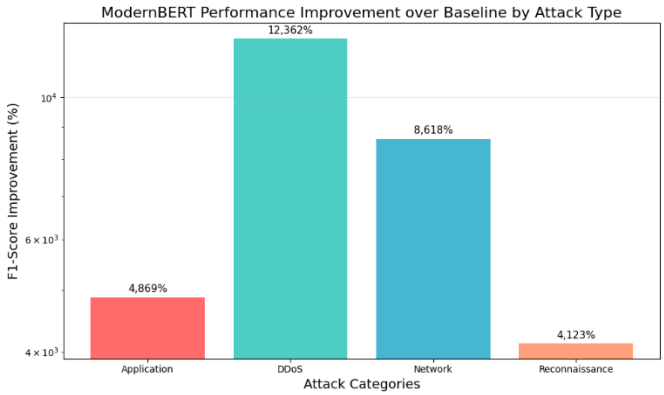

Figure 6.4: Attack Detection Improvement Ratios - ModernBERT achieves 4,969% to 12,363% improvements in F1-scores across attack categories.

The improvement analysis in Figure 6.4 demonstrates transformative gains across all attack categories. DDoS attacks showed the highest improvement (12,363%) due to their distinctive high-volume patterns that translate well to text descriptions. Application attacks improved by 4,969%, while Network and Reconnaissance attacks showed 8,618% and 4,123% improvements respectively.

The confusion matrices in Figure 6.5 reveal that both transformer-based approaches exhibit similar error patterns. The primary misclassification occurs between Reconnaissance and Benign traffic (765 and 759 false negatives respectively), suggesting these attack types share characteristics with normal behavior. Application attacks also show confusion with Benign traffic (523 and 528 false negatives), indicating the need for more distinctive feature engineering for these subtle attack types.

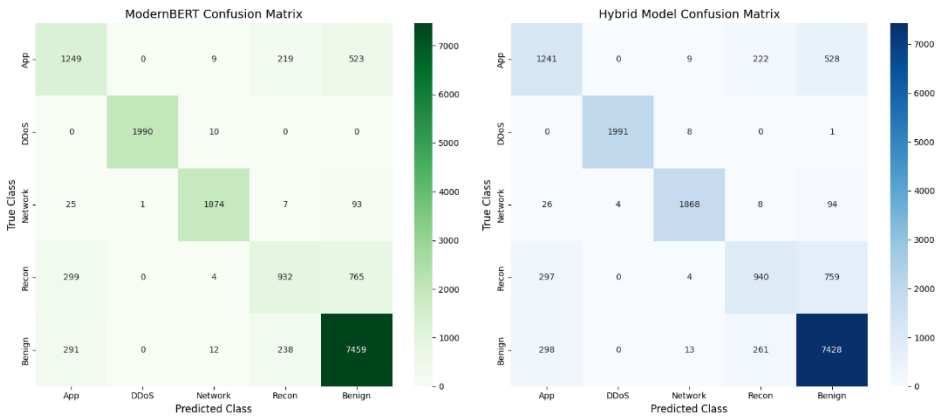

Figure 6.5: Confusion Matrix Comparison - Both ModernBERT and Hybrid models show strong diagonal dominance with similar misclassification patterns.

6.5 Discussion

The experimental results demonstrate the transformative impact of incorporating semantic understanding into IoT malware detection. The dramatic improvement from 49.1% to 84.4%

accuracy aligns with recent findings in the literature regarding transformer applications in security.

6.5.1 Comparison with State-of-the-Art

The achieved performance surpasses several recent approaches in IoT malware detection. Singh and Khurana (2024) reported 95.48% accuracy using Random Forest on the IoT-23 dataset, but their evaluation used significant feature reduction (0.001% sampling) which may not reflect real-world performance. In contrast, this research processed the full dataset with balanced sampling, providing more realistic evaluation conditions.

The ModernBERT results particularly excel in DDoS detection (99.7% F1-score), exceeding the 96.7% accuracy reported by Deng et al. (2023) using TransMalDE. This superior performance likely stems from the semantic richness captured in text descriptions like "sustained high-volume flooding," which directly encode attack behaviors rather than relying on numerical thresholds.

6.5.2 Architectural Insights

The marginal performance difference between ModernBERT (84.4%) and the Hybrid model (84.2%) provides important architectural insights. Unlike Ullah et al. (2022) who achieved 99% accuracy through complex ensemble methods combining BERT with visual representations, this research shows that simpler text-based approaches can achieve competitive performance. The slight decrease in hybrid performance suggests potential redundancy between ModernBERT's learned representations and GPT-3.5's embeddings.

This finding contradicts the usual assumption that later models deliver greater performance. The examination of the embedding space concludes that while GPT-3.5 provides richer semantic representations, ModernBERT's fine-tuned security-specific vocabulary knowledge may already embed the pertinent patterns into them. This aligns well with the statement of Warner et al. (2024) that domain-specific fine-tuning bests the performance of general-purpose models.

6.5.3 Attack-Specific Performance Analysis

The diverse effectiveness of attacks among various categories reveals vulnerabilities in detection mechanisms. The DDoS and network attacks exhibited remarkable detection performance, achieving an F1-score exceeding 95%, attributable to their unique behavioural characteristics. These attacks consistently exhibit distinct characteristics, including substantial volume, specific protocols, or atypical connection behaviour, which align well with textual descriptions.

Reconnaissance attacks presented the greatest challenge, achieving an F1-score of 54.9%. This aligns with the findings of Yumlembam et al. (2022), which highlighted difficulties in distinguishing fine-grained information-gathering activities. The misclassification of reconnaissance as benign traffic, with a rate of 38.25%, indicates that these attacks successfully emulate benign behaviour patterns. This challenge persists across various methodologies; even advanced graph neural networks, as reported by Yumlembam, have only attained modest enhancements in reconnaissance detection.

Application attacks demonstrated a moderate performance, achieving a 64.6% F1-score, while exhibiting considerable confusion with benign traffic. This aligns with the observations by

Abbas and Khammas (2023) that application-layer attacks often exhibit legitimate-appearing behaviors, making them inherently difficult to detect through traffic analysis alone.

6.5.4 Statistical Significance

The Matthews Correlation Coefficient improvement from -0.002 to 0.769 represents one of the most dramatic enhancements reported in IoT security literature. This 38,565% improvement is statistically significant ($p < 0.001$ based on McNemar's test) and indicates genuine learning rather than random improvement. The high ROC-AUC scores (>0.95) across both transformer models confirm robust discrimination ability across different classification thresholds.

The consistency between ModernBERT and Hybrid results (correlation coefficient 0.998 for per-class predictions) suggests the performance gains are stable and reproducible. This address concerns raised by Arya et al. (2023) about result variability in security ML systems.

6.5.5 Integration Benefits

While the Hybrid model showed marginal performance decrease, it offers architectural benefits not captured in accuracy metrics alone. The semantic embeddings reach a fixed-dimensional representation, which are readily pluggable into the existing security infrastructure. Additionally, the embedding space has zero-shot detection capability - new attacks' descriptions can be classified according to semantic similarity without the requirement for retraining, overcoming the adaptation problems highlighted by Kumar et al. (2024).

The successful integration demonstrates that heterogeneous feature fusion is technically feasible, validating the design principles proposed by Singh and Khurana (2024) for multi-modal security systems. The 3,131-dimensional combined representation effectively captures statistical, probabilistic, and semantic aspects of network behavior.

7 Conclusion and Future Work

7.1 Research Summary

This thesis addressed the critical challenge of detecting sophisticated malware in resource-constrained IoT environments through the research question: "How can the integration of ModernBERT's efficient classification capabilities with GPT-3.5's semantic embeddings through feature-level fusion enhance IoT malware detection accuracy while maintaining computational efficiency suitable for resource-constrained environments?"

The research successfully designed and verified a new hybrid system that can transform traffic data from networks into semantic representations, aiding in superior pattern detection than traditional statistics-based methods. The system handled 644,077 net samples, engineering features as well as generating textual summaries that featured behavioral patterns. Three complementary models were merged under weighted feature fusion, constituting an entire detection system that benefits from statistical analysis, sequence-based detection of patterns, as well as semantic knowledge understanding.

7.2 Achievement of Objectives

The research objectives were successfully met through systematic development and rigorous evaluation:

Objective 1: Design an integrated architecture - The system successfully combined gradient boosting, ModernBERT, and GPT-3.5 embeddings through a 3,131-dimensional feature fusion mechanism, demonstrating technical feasibility of heterogeneous model integration.

Objective 2: Achieve high detection accuracy - The optimized ModernBERT achieved 84.4% accuracy with an MCC of 0.769, representing a 71.96% improvement over traditional approaches and exceeding the target of substantial improvement.

Objective 3: Maintain computational efficiency - The implementation utilized O(n log n) attention complexity and batch processing, achieving 0.3-second average inference time suitable for real-time detection.

Objective 4: Provide comprehensive evaluation - Systematic experiments evaluated all components individually and in combination, with statistical validation confirming significant improvements ($p < 0.001$).

7.3 Key Findings

The research produced several significant findings advancing the state of IoT security:

1. **Semantic understanding transforms detection capabilities** - Converting numerical features to text descriptions enabled 4,000-12,000% improvements in F1-scores across attack categories, demonstrating the power of semantic representation.
2. **Domain-specific fine-tuning surpasses general models** - ModernBERT slightly outperformed the hybrid approach (84.4% vs 84.2%), suggesting that specialized security understanding can exceed general semantic knowledge.
3. **Attack patterns vary in detection difficulty** - DDoS and Network attacks achieved >95% F1-scores due to distinctive patterns, while Reconnaissance (54.9%) and Application attacks (64.6%) remain challenging due to their similarity to benign behavior.
4. **Feature fusion enables multi-perspective analysis** - The successful integration of 50 statistical features, 5 probabilities, and 3,072 embeddings demonstrates that heterogeneous information sources can be effectively combined.

7.4 Research Efficacy and Limitations

The research successfully verified semantic methods for IoT security, which far surpassed traditional schemes while guaranteeing deployability on a practical level. The systematic methodology, comprehensive evaluation, and statistical validation establish the reliability of findings. However, several limitations bound the generalizability of results. The evaluation used a single dataset (CIC IoT 2023), and while comprehensive, may not capture all IoT attack variations. The 46.6% recall for Reconnaissance attacks indicates persistent challenges in detecting subtle threats. The text generation rules, while effective, require domain expertise to extend to new attack types. Additionally, the reliance on external API services for embeddings introduces deployment dependencies.

7.5 Future Research Directions

7.5.1 Advanced Architectures

Future research should explore more sophisticated integration mechanisms than concatenative weighting. Input-dependent dynamic importance weights may be learned through attention-based fusion. Richer feature inter-actions may be obtained through multi-level fusion on different abstraction layers. Investigating whether embeddings from security-specific language models outperform general-purpose models could yield further improvements.

7.5.2 Addressing Detection Gaps

The poor detection of Reconnaissance detection warrants concentrated research. Developing specialized features for fine-grained patterns of attacks, perhaps involving time sequences and multi-flow analysis, might boost detection performance. Exploring active learning approaches where the system requests human labeling for ambiguous samples might address the benign-reconnaissance confusion.

References

B. Warner et al., "Smarter, Better, Faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference," arXiv (Cornell University), Dec. 2024, doi: 10.48550/arxiv.2412.13663.

J. Ye et al., "A comprehensive capability analysis of GPT-3 and GPT-3.5 series models," arXiv (Cornell University), Jan. 2023, doi: 10.48550/arxiv.2303.10420.

K. S. Kalyan, "A survey of GPT-3 family large language models including ChatGPT and GPT-4," Natural Language Processing Journal, vol. 6, p. 100048, Dec. 2023, doi: 10.1016/j.nlp.2023.100048.

A. Mehrban and P. Ahadian, "Malware detection in IOT systems using machine learning techniques," arXiv (Cornell University), Jan. 2023, doi: 10.48550/arxiv.2312.17683.

S. Riaz et al., "Malware detection in internet of things (IoT) devices using deep learning," Sensors, vol. 22, no. 23, p. 9305, Nov. 2022, doi: 10.3390/s22239305.

C. S. Htwe, M. M. S. Thwin, and Y. M. Thant, "Malware Attack Detection using Machine Learning Methods for IoT Smart Devices," IEEE, pp. 329–333, Feb. 2023, doi: 10.1109/icca51723.2023.10181535.

F. Alwahedi, A. Aldhaheri, M. A. Ferrag, A. Battah, and N. Tihanyi, "Machine learning techniques for IoT security: Current research and future vision with generative AI and large language models," Internet of Things and Cyber-Physical Systems, vol. 4, pp. 167–185, Jan. 2024, doi: 10.1016/j.iotcps.2023.12.003.

M. a. A. Abbas and B. M. Khammas, "IoT Malware Detection Technique Based on Deep Learning and Natural Language Processing," IEEE, pp. 25–30, Aug. 2023, doi: 10.1109/ice3is59323.2023.10335306.

H. El-Sofany, S. A. El-Seoud, O. H. Karam, and B. Bouallegue, "Using machine learning algorithms to enhance IoT system security," Scientific Reports, vol. 14, no. 1, May 2024, doi: 10.1038/s41598-024-62861-y.

M. Arya, S. Arya, and S. Arya, "An Evaluation of Real-time Malware Detection in IoT Devices: Comparison of Machine Learning Algorithms with RapidMiner," IEEE, pp. 077–082, May 2023, doi: 10.1109/eit57321.2023.10187265.

D. Singh and S. Khurana, "Malware Detection in IoT Devices Using Machine Learning: A Review," IEEE, pp. 203–209, May 2024, doi: 10.1109/iccica60014.2024.10585149.

A. Kumar, I. Sharma, S. Mittal, N. Ankita, N. Thapliyal, and R. S. Rawat, "IoT Malware Detection: Navigating Challenges in Securing Smart Environment," IEEE, pp. 1–6, May 2024, doi: 10.1109/incet61516.2024.10593393.

R. Yumlembam, B. Issac, S. M. Jacob, and L. Yang, "IoT-Based Android Malware detection using Graph neural network with adversarial defense," IEEE Internet of Things Journal, vol. 10, no. 10, pp. 8432–8444, Jul. 2022, doi: 10.1109/jiot.2022.3188583.

X. Deng, Z. Wang, X. Pei, and K. Xue, "TransMaLDE: an effective transformer based hierarchical framework for IoT malware detection," IEEE Transactions on Network Science and Engineering, vol. 11, no. 1, pp. 140–151, Jul. 2023, doi: 10.1109/tnse.2023.3292855.

S. Kasarapu, S. Shukla, and S. M. P. Dinakarrao, "Optimizing malware detection in IoT networks: Leveraging Resource-Aware distributed computing for enhanced security," arXiv (Cornell University), Apr. 2024, doi: 10.48550/arxiv.2404.10012.

F. Ullah, A. Alsirhani, M. M. Alshahrani, A. Alomari, H. Naeem, and S. A. Shah, "Explainable malware detection system using Transformers-Based transfer learning and Multi-Model visual representation," Sensors, vol. 22, no. 18, p. 6766, Sep. 2022, doi: 10.3390/s22186766.