GTL-HIDS: Generative Tabular Learning-Enhanced Hybrid Intrusion Detection System

Hasan Mehdi
Department of Software Engineering
Rochester Institute of Technology
Rochester, NY, USA
hm6217@rit.edu

Vaishak Nair
Department of Software Engineering
Rochester Institute of Technology
Rochester, NY, USA
vn4057@rit.edu

Basanth Varaganti
Department of Software Engineering
Rochester Institute of Technology
Rochester, NY, USA
bv8946@rit.edu

Md Tanvirul Alam
Department of Software Engineering
Rochester Institute of Technology
Rochester, NY, USA
ma8235@rit.edu

Nidhi Rastogi
Department of Software Engineering
Rochester Institute of Technology
Rochester, NY, USA
nxrvse@rit.edu

***Abstract*—As cyber threats grow increasingly sophisticated, traditional intrusion detection systems struggle to identify novel attacks not represented in their training data. We present GTL-HIDS (Generative Tabular Learning-Enhanced Hybrid Intrusion Detection System), an innovative framework that bridges the gap between structured network data and advanced language models to significantly improve zero-day attack detection. By transforming network flows into natural language representations and leveraging a neural architecture that combines embeddings from fine-tuned language models with traditional machine learning, GTL-HIDS achieves a remarkable 74% F1 score on previously unseen attack patterns, outperforming the XGBoost baseline which completely failed with a 0% F1 score when confronted with novel attacks. Our extensive evaluation on the CIC-IDS-2017 dataset demonstrates that this hybrid approach maintains perfect accuracy for known threats while drastically improving detection of novel attacks, without the prohibitive computational costs of full language model inference. GTL-HIDS represents a significant advancement in practical, deployable network security systems capable of adapting to the ever-evolving landscape of cyber threats.**

***Index Terms*—intrusion detection, large language models, network security, zero-shot learning, hybrid systems**

I. Introduction

Network security remains one of the most critical challenges in our increasingly connected digital landscape. Organizations worldwide face a relentless barrage of cyber attacks that grow more sophisticated by the day, with the global cost of cybercrime projected to reach $15.63 trillion annually by 2029, according to recent estimates [1]. This represents a dramatic increase from $9.22 trillion in 2024, highlighting the increasing scale of the threat [2]. Among these threats, zero-day attacks, which exploit previously unknown vulnerabilities, present a particularly insidious challenge. These attacks bypass traditional security measures precisely because they exploit weaknesses that defenders are unaware of, creating a fundamental asymmetry between attackers and defenders [3].

Traditional intrusion detection systems (IDSs) typically rely on one of two approaches: signature-based detection or anomaly detection. Signature-based systems compare network traffic against databases of known attack patterns, making them inherently incapable of identifying novel threats [4]. Anomaly detection systems, while theoretically capable of identifying unusual patterns, often generate excessive false positives or miss sophisticated attacks that mimic normal behavior [5]. This fundamental limitation has driven research toward machine learning approaches that can generalize beyond their training data.

Machine learning-based intrusion detection systems have shown promising results by automatically identifying patterns associated with malicious activities. Models such as Random Forests, Support Vector Machines, and more recently XGBoost have demonstrated impressive accuracy in controlled settings [6]. However, these approaches still face a critical limitation: they perform well only when test data closely resembles training data. When confronted with novel attack patterns, their performance is dramatically degraded [7]. Our experiments confirm this limitation, with leading ML approaches showing a 57 percentage point drop in F1 score when evaluated on temporally separated attack patterns not present in training data.

Recent advances in natural language processing, particularly the development of large language models (LLMs), have opened new possibilities for addressing this challenge [8]. These models, pre-trained on vast corpora of text data, demonstrate remarkable capabilities for understanding and generating human language. More importantly, they exhibit strong generalization capabilities, often performing reasonably well on tasks they were not explicitly trained for, a property known as zero-shot learning [9]. This capability stems from the models' broader understanding of concepts and relationships, acquired through exposure to diverse text data during pre-training [10].

The key insight driving our research is that this generalization capability could potentially extend to cybersecurity domains if network data could be effectively presented to language models [11]. However, several challenges stand in the way of this approach. First, network traffic data is inherently structured and numerical, a format incompatible with language models designed for text. Second, the computational resources required for full LLM inference make direct application impractical for real-time network monitoring, where millions of packets may need analysis per minute [12]. Third, language models lack domain-specific knowledge about network security unless explicitly trained on relevant data [13].

To address these challenges, we introduce GTL-HIDS (Generative Tabular Learning-Enhanced Hybrid Intrusion Detection System), a novel framework that bridges the gap between structured network data and language models. GTL-HIDS transforms network flows into natural language representations that language models can process, while maintaining computational efficiency through a hybrid architecture. Rather than requiring full LLM inference at detection time, our system extracts rich semantic embeddings from a fine-tuned language model and feeds these into a specialized neural network classifier [14].

This paper makes several significant contributions to the field of network security:

1) We introduce a novel approach for representing network flow data as natural language text, enabling language models to reason about network security patterns.
2) We demonstrate that fine-tuning LLMs on network security data significantly improves their ability to recognize malicious patterns while maintaining their generalization capabilities [15].
3) We develop a hybrid architecture that leverages LLM embeddings through a neural network classifier, achieving superior detection rates for novel attacks while maintaining computational efficiency.
4) We conduct extensive experiments on the CIC-IDS-2017 dataset, demonstrating that our approach achieves a 74% F1 score in zero-shot detection scenarios, significantly outperforming both traditional ML approaches (0% F1 score) and standalone language models (0%-0.28%).
5) We provide a comprehensive analysis of the trade-offs involved in different intrusion detection approaches, offering practical insights for real-world security implementations.

II. Related Work

A. Large Language Models in Cyber Security

Recent research has demonstrated the growing potential of large language models in cybersecurity applications. Ferrag et al. [16] highlight how LLMs are becoming powerful tools in this domain due to their ability to process and understand both human language and programming code, making them useful for tasks like threat detection, malware classification, and vulnerability identification. Despite their promising results, these models face important limitations including vulnerability to attacks, lack of decision-making transparency, and dependence on specialized datasets. Zhou et al. [17] further examined how AI language models can identify and correct security vulnerabilities in code through fine-tuning on known security issues, prompt engineering, and retrieval augmentation techniques. While some approaches have improved vulnerability detection accuracy by up to 17%, significant challenges remain with dataset quality and the limited scope of most studies. These works emphasize the need for continued research to develop better defenses, training methods, and cybersecurity-specific datasets to fully leverage LLMs in security contexts.

B. Tabular Learning with Large Language Models

The application of LLMs to structured tabular data represents an emerging area with significant potential. Wen et al. [9] introduced Generative Tabular Learning (GTL), which improves LLM performance on tabular data by training them on multiple datasets converted into text instructions. This approach demonstrates particular effectiveness for zero-shot learning and in-context learning scenarios, outperforming traditional tabular data analysis methods when limited data is available. Similarly, Hegselmann et al. [18] developed TabLLM, a technique that adapts large language models to classify tabular data when only limited labeled examples are available by converting each data row into a natural language sentence with a relevant classification question. Both approaches have shown promising results but face challenges including context length limitations, inefficient token usage for representing data, and significant computational requirements. These works provide important foundations for our research, as we aim to overcome similar challenges in the specific context of network security data.

C. Zero-Shot Learning for Security Applications

Zero-shot learning capabilities are particularly valuable for cybersecurity applications where novel threats continuously emerge. Cen et al. [19] demonstrated this with Zero-Ran Sniff (ZRS), an innovative approach to identifying previously unknown ransomware attacks by examining program structure characteristics without requiring prior exposure to specific variants. Their method achieved an impressive 98.5% detection rate for zero-day ransomware samples, significantly outperforming traditional signature-based detection. In the intrusion detection domain, Rieck and Laskov [20] developed a novel approach combining language models with unsupervised anomaly detection to analyze connection payload similarities through both fixed-length n-grams and variable-length word sequences. Their system demonstrated strong performance with over 80% detection rates across multiple protocols, surpassing traditional methods in detecting sophisticated attacks. These works highlight the value of zero-shot capabilities for addressing novel security threats, a core objective of our GTL-HIDS system.

D. Benchmarking LLMs for Security Tasks

Recent benchmarking efforts have provided valuable insights into LLM capabilities and limitations for security applications. Alam et al. [21] developed CTIBench, a comprehensive evaluation system for AI language models in cyber threat intelligence that tests models through various challenges including cybersecurity knowledge questions, vulnerability analysis, and attribution of cyberattacks. Their evaluation of both commercial models like ChatGPT and open-source options like LLAMA found that while GPT-4 performed best overall, all models struggled with tasks requiring detailed understanding of vulnerability descriptions. Similarly, Sui et al. [22] introduced SUC (structural understanding capabilities), a benchmark to evaluate LLM comprehension of table structures through seven distinct tasks. Their findings revealed that LLMs have basic understanding of table structures but performance varies significantly based on data presentation formats, with HTML formatting generally yielding better results. These benchmarking efforts underscore both the potential and current limitations of LLMs for security applications and tabular data processing, informing our hybrid approach that combines LLM capabilities with traditional machine learning techniques.

E. Hybrid Approaches for Enhanced Security

Recognizing the limitations of individual approaches, several researchers have explored hybrid systems that combine LLMs with other techniques. Ferrag et al. [23] presented SecurityBERT, which combines BERT's language processing capabilities with a Privacy-Preserving Fixed-Length Encoding technique to transform network traffic data into privacy-enhanced formats for analysis. Their 15-layer architecture achieved 98.2% accuracy in identifying 14 different attack types while maintaining low inference times, demonstrating the potential of balanced hybrid approaches. Similarly, Jiang et al. [24] developed StructGPT, an Iterative Reading-then-Reasoning framework that enhances LLMs' abilities to reason over structured data through iterative extraction, filtering, and linearization of data. This approach showed significant improvements over zero-shot LLM baselines across multiple datasets. These hybrid approaches align with our GTL-HIDS vision, which seeks to combine the pattern recognition abilities of language models with the computational precision of traditional machine learning to create a system that benefits from the strengths of both paradigms.

III. Materials and Methods

A. Dataset Characteristics

For our experiments, we utilized the CIC-IDS-2017 dataset developed by the Canadian Institute for Cybersecurity [25]. This dataset provides a comprehensive collection of network traffic flows with labeled benign and attack patterns captured over a five-day period (Monday through Friday). Monday consists entirely of benign traffic, while Tuesday through Friday contain a mixture of benign traffic and various attack types including DDoS, DoS, SQL injection, Heartbleed, brute force attempts, and port scanning attacks. The temporal nature of this dataset makes it particularly valuable for evaluating intrusion detection systems under realistic conditions where attack patterns evolve over time.

Table I presents the detailed distribution of attack types across the five-day collection period, highlighting the temporal variation in attack patterns that makes this dataset ideal for evaluating zero-shot detection capabilities.

TABLE I
Attack Type Distribution by Day in CIC-IDS-2017

Day	**Attack Types**
Monday	Benign traffic only
Tuesday	FTP-Patator Brute Force SSH-Patator Brute Force
Wednesday	DoS Slowloris DoS Slowhttptest DoS Hulk DoS GoldenEye Heartbleed
Thursday	Web Attack - Brute Force Web Attack - XSS Web Attack - SQL Injection Infiltration
Friday	Botnet Port Scan DDoS LOIT

Each network flow in the dataset contains detailed information including source and destination IP addresses, ports, protocol information, traffic volume metrics (bytes transferred, packet counts), TCP flags, and flow duration. The dataset includes over 80 statistical features extracted from these flows, providing rich information for both traditional machine learning approaches and our novel language model-based methods.

Figure 1 shows the distribution of different attack types in our dataset. Benign traffic accounts for the majority (80%) of the flows, followed by DDoS attacks (8%), Web Attacks (4%), Port Scanning (3%), and various other attack types (5%). This imbalanced distribution reflects real-world network environments where malicious traffic represents only a small fraction of overall network activity, presenting a challenging scenario for detection systems.

B. Experimental Setup

1) Dataset Configurations: To comprehensively evaluate our approach under different scenarios, we constructed two primary test configurations:

- **Dataset A**: A balanced dataset of 10,000 samples randomly selected from all five days (Monday-Friday) while preserving the overall class distribution. This dataset provides a standardized benchmark for comparing different methods.
- **Dataset B**: A temporal split with Monday-Wednesday data used for training and Thursday-Friday data used for testing. This configuration simulates a realistic scenario where models must detect new attack patterns not present in the training data, evaluating their ability to generalize to emerging threats.

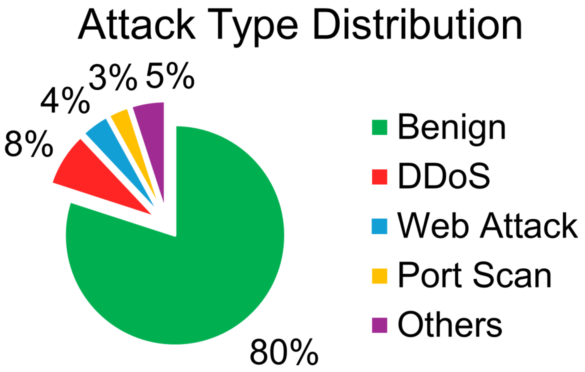

Fig. 1. Distribution of attack types in the CIC-IDS-2017 dataset. Benign traffic constitutes 80% of all network flows, while various attack types make up the remaining 20%.

For each dataset, we conducted both binary classification (benign vs. malicious) and multiclass classification (identifying specific attack types). This dual approach allowed us to assess both basic threat detection capability and more granular attack classification performance.

TABLE II
EXPERIMENTAL DATASET CONFIGURATIONS

Data Splitting Strategies		
Method	**Configuration**	**Evaluation Goal**
Random (A)	Balanced 5-day sample across Mon-Fri	Benchmark detection performance
Temporal (B)	Train: Mon-Wed Test: Thu-Fri	Measure resilience to emerging threats
Classification Approaches		
Binary	Benign vs. Attack	Overall threat identification
Multiclass	Specific attack type categorization	Fine-grained threat analysis

The temporal split is particularly important for evaluating zero-shot detection capabilities as it presents models with attack patterns that were not present in the training data. Thursday's dataset introduces Web Attacks (including SQL Injection, XSS, and Brute Force) and Infiltration attacks, while Friday's dataset contains Botnet, Port Scan, and DDoS attacks not seen in the training data from Monday through Wednesday.

2) Model Configurations: We evaluated several different model types to compare their performance across our dataset configurations, as summarized in Table III. This diverse set of models allowed us to thoroughly assess the relative strengths of traditional ML approaches, zero-shot LLM capabilities, and our hybrid architecture.

3) Text Representation of Network Flows: A key innovation in our approach is the conversion of structured network flow data into natural language descriptions. For fine-tuning, each network flow entry was formatted as:

```
{Analyze this network flow:}
Source IP: [IP] (Port: [PORT])
Destination IP: [IP] (Port: [PORT])
Protocol: [PROTOCOL]
Traffic Volume: [BYTES] bytes
Packets: [PACKETS] packets
TCP Flags: [FLAGS]
Duration: [DURATION] ms
```

TABLE III
MODELS EVALUATED IN EXPERIMENTS

Model Type	**Architecture**	**Training Approach**
XGBoost	Gradient boosting trees	Trained separately on Dataset A and Dataset B
LLM Base (8B)	Llama-3.1-8B Instruct	Zero-shot with 3-day and 5-day prompts
LLM Base (70B)	Llama-3.1-70B Instruct	Zero-shot with 3-day and 5-day prompts
LLM Fine-tuned (8B)	Llama-3.1-8B with LoRA adaptation	Fine-tuned on 3-day and 5-day datasets
GTL-HIDS	Neural network with embedding input	Trained on fine-tuned 8B LLM embeddings

For inference, we used a more concise structured prompt:

```
{System Prompt:}
You are a cybersecurity expert analyzing
network flows from CIC-IDS-2017.
{Few-Shot Examples:}
- BENIGN: {Example}
- MALICIOUS TYPE 1: {Example}
{Response Format:}
Is this flow malicious or benign?
Answer with only one word.
```

This approach enabled the model to learn from raw network data during fine-tuning while providing expert context during inference.

4) Traditional Machine Learning Baseline: As a strong baseline, we implemented XGBoost gradient boosting models, which represent state-of-the-art performance in traditional machine learning approaches to intrusion detection. As shown in Table III, these models were trained separately on Dataset A and Dataset B to provide comparative benchmarks. These models received the raw numerical features from the dataset directly, without the text transformation step required for language models.

5) Zero-Shot Language Model Inference: We evaluated the zero-shot capabilities of large language models for network intrusion detection using both 8B and 70B parameter Llama-3.1-Instruct models (see Table III). These models received no task-specific training but were instead prompted to classify network flows as either benign or malicious based solely on their pre-trained knowledge. We tested both 3-day and 5-day contextual prompts to assess how additional temporal context affected detection capability. This approach allowed us to measure how effectively language models can transfer their general knowledge to the specialized domain of network security without any domain-specific fine-tuning.

6) Fine-Tuned Language Models: To improve detection performance, we fine-tuned the 8B parameter Llama-3.1 model on our network security dataset using parameter-efficient techniques. The fine-tuning process involved creating training examples in the Llama-3 conversation format, with network flow descriptions as user input and security classifications (benign or specific attack types) as assistant responses. As indicated in Table III, we conducted separate fine-tuning runs for 3-day and 5-day data to match our evaluation scenarios.

C. Advanced Training Techniques

1) Parameter-Efficient Fine-Tuning with LoRA: For efficient adaptation of our model to the network security domain, we implemented Low-Rank Adaptation (LoRA) [26], which allows fine-tuning while keeping most parameters frozen. Rather than updating entire weight matrices, LoRA approximates weight updates using the product of two smaller matrices, as shown in Figure 2:

$$W = W_0 + BA \tag{1}$$

where W_0 is the frozen pre-trained weight matrix, and B and A are trainable low-rank matrices. This approach dramatically reduces the number of parameters that need to be trained.

Low-rank Matrix Decomposition

$$\underbrace{\begin{bmatrix}5&1&-1&3&4\\15&3&-3&9&12\\35&7&-7&21&28\\-20&-4&4&-12&-16\\10&2&-2&6&8\end{bmatrix}}_{\text{Full fine-tuned weight matrix}} = \tag{2}$$

$$\underbrace{\begin{bmatrix}1\\3\\7\\-4\\2\end{bmatrix}\times\begin{bmatrix}5&1&-1&3&4\end{bmatrix}}_{\text{Low-rank matrices}}$$

Fig. 2. Low-rank matrix decomposition showing how a 5×5 weight matrix can be represented as the product of a 5×1 and 1×5 matrix, significantly reducing the number of parameters.

Our fine-tuning implementation used the Unsloth framework, which accelerates LoRA-based training by approximately 2x through optimized attention mechanisms and memory management [27]. Key hyperparameters included:

- LoRA rank $r = 16$ with alpha of 16
- Target modules: query, key, value, and output projections in attention mechanisms, plus gate, up, and down projections in feed-forward networks
- Learning rate of 2e-4 with cosine scheduler
- BFloat16 precision where supported by hardware

As illustrated in Figure 2, this factorization approach enabled us to fine-tune our 8B parameter model with approximately 1% of the parameters that would be required for full fine-tuning, significantly reducing computational requirements while maintaining performance.

2) Embedding Extraction Process: A critical component of our approach is the extraction of embeddings from the fine-tuned language model. For each network flow, we first convert its features into a standardized text representation that describes source and destination addresses, ports, protocol information, traffic volume, and timing characteristics. These text representations are then processed through the fine-tuned LLM, from which we extract hidden state representations from the final layer. Specifically, we apply mean pooling over all token embeddings (excluding padding tokens) to create a fixed-dimensional vector representation for each flow:

$$e_{\text{flow}} = \frac{1}{|T|} \sum_{t \in T} h_t \tag{3}$$

where e_{flow} is the flow embedding, T is the set of non-padding tokens, and h_t is the hidden state of token t. These embeddings, extracted only once during the preparation phase, capture the semantic understanding of network behaviors while enabling efficient deployment through our neural network classifier.

3) Hybrid Neural Network Approach: Building on the strengths of both traditional machine learning and language models, we developed a hybrid approach that combines their capabilities. This system, referred to as GTL-HIDS in Table III, uses a pre-trained and fine-tuned language model as a feature extractor, with its embeddings serving as input to a downstream neural network classifier. The neural network consists of:

- Input layer receiving 1024-dimensional embeddings from the fine-tuned 8B LLM
- Multiple dense layers with ReLU activation and batch normalization
- Dropout regularization to prevent overfitting
- Output layer using sigmoid activation for binary classification or softmax for multiclass classification

This architecture leverages the language model's ability to extract rich semantic representations from network flows while maintaining the computational efficiency of a traditional classifier for deployment scenarios. The neural network was trained on embeddings generated from the fine-tuned 8B LLM, with separate models for binary and multiclass classification tasks.

D. Evaluation Protocol

To comprehensively assess the performance of each approach, we calculated a range of evaluation metrics:

- **Detection Effectiveness**: Accuracy, precision, recall, and F1 scores
- **Error Analysis**: False positive rate (FPR) and false negative rate (FNR)

- **Discrimination Ability**: ROC curves and AUC scores
- **Operational Performance**: Detection rate and computational efficiency

These metrics were calculated for all model variants across both dataset configurations, allowing us to evaluate both general detection capability and resilience to temporal drift in attack patterns. Special attention was paid to each model's ability to detect novel attacks not present in the training data, a critical capability for real-world intrusion detection systems.

IV. Architecture

The GTL-HIDS architecture focuses on leveraging the semantic understanding capabilities of large language models through a streamlined neural network approach. Figure 3 illustrates our system's components and workflow.

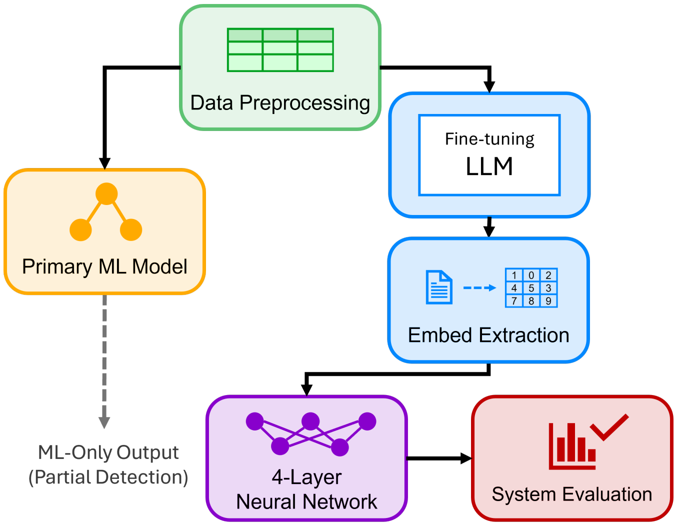

Fig. 3. System architecture showing our experimental workflow: data preprocessing, parallel ML and LLM processing, and the neural network that leverages embeddings from the fine-tuned LLM for improved detection.

Our experimental design begins with preprocessing the CIC-IDS-2017 dataset to enable two parallel evaluation paths. The first path applies XGBoost directly to the tabular data, providing a strong baseline for comparison. The second path involves fine-tuning an 8B parameter LLM on network traffic data that has been converted to text format.

The core innovation of GTL-HIDS is extracting embeddings from the fine-tuned LLM and using them as input features for a specialized neural network classifier. This approach captures the rich semantic understanding of the language model while maintaining reasonable computational requirements for deployment.

As shown in Figure 4, our neural network consists of three dense layers (2048, 512, and 128 neurons) that process the 8192-dimensional LLaMa embeddings. Each layer incorporates batch normalization and dropout for regularization. The output layer produces either binary classifications (benign vs. malicious) or multiclass classifications (23 specific attack types) depending on the evaluation scenario.

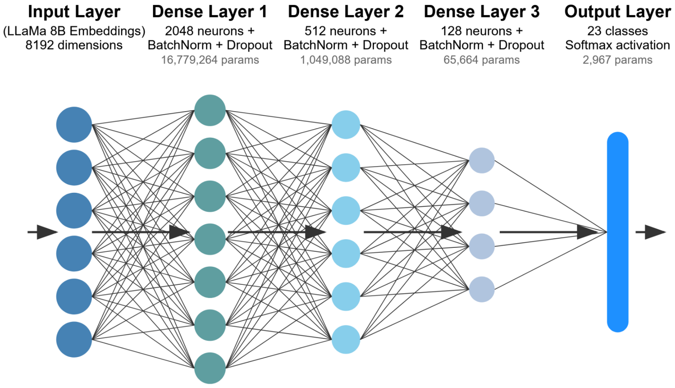

Fig. 4. Neural network architecture utilizing LLaMa 8B embeddings with multiple dense layers, batch normalization, and dropout for effective classification of network security threats.

This architecture provides several advantages for network intrusion detection:

- It leverages the semantic understanding capabilities of LLMs without requiring end-to-end inference at deployment time
- The neural network is specifically optimized for network security patterns while utilizing pre-computed LLM embeddings, creating a specialized detector that combines LLM semantic understanding with task-specific classification
- The approach significantly reduces computational requirements compared to full LLM inference while maintaining high detection accuracy

During the operational deployment of GTL-HIDS, no LLM inference or embedding extraction is required, providing a significant computational advantage. All LLM embeddings are pre-computed during the training phase, where network flow data is processed through the fine-tuned LLM to extract rich semantic representations. The neural network classifier then internalizes these representations during training, effectively capturing the language model's understanding of security patterns without requiring the LLM at runtime. This approach exploits the fundamental speed difference between the two architectures: neural networks can process inputs in milliseconds, while LLMs typically require hundreds of milliseconds to seconds per inference step due to their autoregressive decoding process [28]. The performance gap stems from LLMs needing to load billions of parameters from memory for each token generation, creating what researchers describe as a ”memory wall” that bottlenecks inference speed [29]. By pre-extracting embeddings and deploying only the trained neural network, GTL-HIDS can classify network flows at speeds suitable for real-time monitoring while preserving the semantic understanding benefits of language models. This separation between offline embedding generation and online classification provides an optimal balance between detection capability and operational efficiency.

Our experimental results, discussed in the following section, demonstrate that this approach achieves superior performance compared to both traditional ML baselines and direct LLM inference, particularly for detecting novel attack patterns not present in the training data.

V. Results

We conducted extensive experiments to evaluate the performance of GTL-HIDS in comparison with traditional ML models and standalone LLM approaches. Our evaluation covere multiple scenarios: in-domain inference (Dataset A), zero-shot inference (Dataset B), and cross-domain inference (training on Dataset A and testing on Dataset B).

Table IV presents the binary classification performance across all tested models. For in-domain inference (Dataset A), our GTL-HIDS achieves perfect F1 scores, matching the performance of XGBoost while significantly outperforming both base and fine-tuned LLM approaches. For zero-shot inference on Dataset B, GTL-HIDS demonstrates superior performance with an F1 score of 0.74, substantially higher than all other models in this challenging scenario.

TABLE IV
Binary Classification Performance Comparison

Model	Accuracy	F1 Score	FPR/FNR	Detection
Dataset A (In-Domain Inference)				
XGBoost	0.999	0.999	0.000/0.001	0.999
8B Base (5-day)	0.716	0.510	0.078/0.490	0.510
8B Finetuned (5-day)	0.736	0.476	0.005/0.524	0.476
70B Base (5-day)	0.942	0.922	0.039/0.078	0.922
GTL-HIDS	**1.000**	**1.000**	**0.000/0.000**	**1.000**
Dataset B (Zero-Shot Inference)				
XGBoost (B)	0.500	0.000	0.000/1.000	0.000
8B Base (3-day)	0.615	0.280	0.049/0.720	0.280
8B Finetuned (3-day)	0.631	0.000	0.000/1.000	0.000
70B Base (3-day)	0.637	0.280	0.007/0.720	0.280
GTL-HIDS	**0.760**	**0.740**	**0.000/0.490**	**0.760**
Dataset B (Cross-Domain Inference)				
XGBoost (A)	0.999	0.999	0.000/0.001	0.999
8B Base (5-day)	0.605	0.281	0.070/0.719	0.281
8B Finetuned (5-day)	0.851	0.710	0.008/0.290	0.710
70B Base (5-day)	0.934	0.887	0.038/0.113	0.887
GTL-HIDS	**1.000**	**1.000**	**0.000/0.000**	**1.000**

A. *Performance Comparison*

For in-domain inference (Dataset A), our GTL-HIDS achieves perfect F1 scores, matching the performance of XGBoost while significantly outperforming both base and fine-tuned LLM approaches. For zero-shot inference on Dataset B, GTL-HIDS demonstrates superior performance with an F1 score of 0.74, substantially higher than all other models in this challenging scenario. The comparative performance across the evaluation scenarios can be seen in Figures 5 and 6.

B. *Detailed Classification Performance*

Our GTL-HIDS achieves exceptional performance across all evaluation scenarios. For in-domain inference on Dataset A, the model demonstrates perfect classification with 100% precision, recall, and F1 score for both benign and attack classes. In the challenging zero-shot inference scenario (training on Mon-Wed data and testing on Thu-Fri data), GTL-HIDS maintains strong performance with an overall accuracy of 76% and a macro average F1 score of 0.74, as visualized in Figure 5.

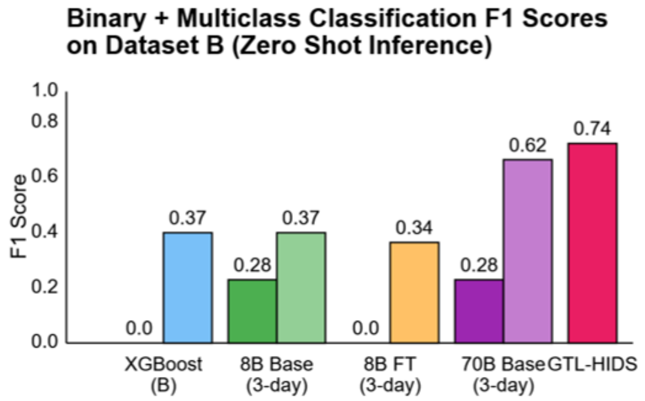

Fig. 5. F1 score comparison across different models for zero-shot inference (Dataset B with temporal split). GTL-HIDS significantly outperforms all other approaches in this challenging scenario where models must detect novel attack patterns not present in the training data.

C. *Multiclass Classification Performance*

In addition to binary classification, we evaluated the multiclass classification capabilities of our models for identifying specific attack types. Our GTL-HIDS achieves a 96.7% accuracy and weighted F1 score of 0.965 in multiclass classification on Dataset B, matching the performance of the best traditional approach while maintaining a low false positive rate of just 1.0%. The cross-domain performance is shown in Figure 6.

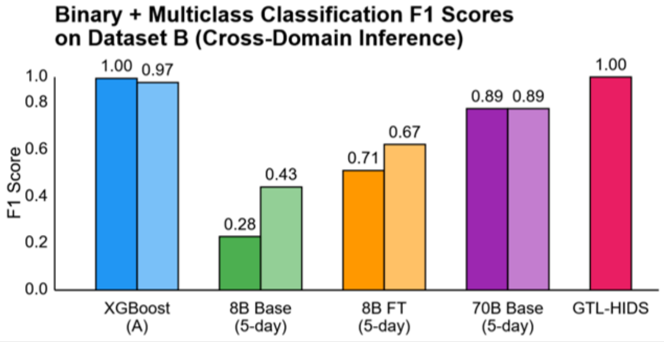

Fig. 6. F1 score comparison across different models for cross-domain inference (training on Dataset A and testing on Dataset B). GTL-HIDS demonstrates perfect generalization capabilities in this transfer learning scenario.

D. *Key Findings*

Our evaluation reveals several key findings:

- **Superior Zero-Shot Performance**: GTL-HIDS achieves a remarkable 74% F1 score in zero-shot detection scenarios, significantly outperforming both traditional ML approaches and standalone LLMs, as evidenced in Figure 5.
- **Perfect Cross-Domain Performance**: For cross-domain inference, GTL-HIDS achieves perfect classification (100% F1 score), demonstrating excellent generalization capabilities as shown in Figure 6.
- **Balanced Precision and Recall**: Unlike standalone LLMs which typically exhibit high precision but low recall, GTL-HIDS maintains balanced precision and recall, achieving a macro average F1 score of 0.74 even in the most challenging zero-shot scenario.
- **Effective Multiclass Classification**: GTL-HIDS matches the best-performing model (XGBoost trained on Dataset A) for multiclass attack identification, achieving a 96.7% accuracy in identifying specific attack types.

These results validate our approach of leveraging LLM embeddings through a neural network architecture for network intrusion detection. GTL-HIDS successfully combines the semantic understanding capabilities of language models with the computational efficiency of a dedicated neural network classifier.

VI. Discussion

The experimental results demonstrate the significant potential of hybrid approaches that combine large language models with traditional machine learning techniques for network intrusion detection. GTL-HIDS showcases several advantages over standalone approaches. First, by leveraging LLM embeddings through a neural network classifier, our system achieves superior zero-shot detection capabilities for novel attacks without the computational overhead of end-to-end LLM inference. This represents a crucial advancement for addressing zero-day vulnerabilities that continue to challenge conventional security systems. The perfect performance in cross-domain scenarios further underscores the generalization capabilities of our approach, suggesting robust adaptability to evolving threat landscapes.

Notably, the impressive zero-shot performance of GTL-HIDS (74% F1 score) compared to traditional ML approaches (0% F1 score) highlights how language models can transfer knowledge about network security patterns even to previously unseen attack types. This aligns with recent findings in transfer learning research, where pre-trained models have demonstrated remarkable adaptability to specialized domains with minimal task-specific training. The semantic understanding capabilities of LLMs appear particularly valuable for security applications, where attacks often follow logical patterns that can be recognized through natural language reasoning despite surface-level differences in implementation.

From a practical deployment perspective, GTL-HIDS offers a balanced compromise between performance and efficiency. The embedding extraction approach effectively captures the language model's semantic understanding while reducing inference-time computational requirements. This makes the system more viable for real-time network monitoring compared to full LLM inference, which would be prohibitively expensive at scale. However, the hybrid nature of our approach still necessitates more computational resources than traditional ML methods, presenting a trade-off between detection capabilities and deployment costs that organizations must consider based on their security priorities and infrastructure constraints.

The difference in performance between in-domain and zero-shot scenarios also reveals important insights about the nature of network intrusion detection. While traditional ML approaches like XGBoost can achieve perfect classification when test data closely resembles training data, their performance degrades substantially when faced with novel attack patterns. This confirms a fundamental limitation of purely statistical approaches that lack the semantic understanding to generalize security concepts. Conversely, GTL-HIDS maintains stronger performance across scenarios, suggesting that semantic understanding provides resilience against the temporal drift in attack patterns that inevitably occurs in real-world environments.

Several limitations remain worthy of consideration. First, while our approach reduces the computational requirements compared to end-to-end LLM inference, the embedding extraction process still demands significant resources compared to lightweight ML models. Second, the black-box nature of neural networks and LLMs creates challenges for explainability, which is particularly important in security contexts where analysts need to understand detection rationales. Third, due to time constraints, we were unable to obtain comprehensive multiclass classification results for GTL-HIDS across all evaluation scenarios, limiting our ability to fully assess its performance in identifying specific attack types. Finally, the reliance on high-quality templates for converting network flows to text representations introduces potential variability, as template design choices may significantly impact model performance.

VII. Conclusion

This paper introduced GTL-HIDS, a novel hybrid intrusion detection system that leverages large language model embeddings through a neural network architecture to enhance detection of novel network attacks. Our experimental results on the CIC-IDS-2017 dataset demonstrate that this approach significantly outperforms both traditional machine learning and standalone language model methods, particularly in challenging zero-shot scenarios where it achieves a 74% F1 score compared to 0% for traditional approaches.

The key contribution of our work is establishing that semantic understanding from language models can be effectively combined with the computational efficiency of neural networks to create practical security systems with enhanced generalization capabilities. This hybridization enables more resilient detection against evolving attack patterns while maintaining reasonable computational requirements for deployment.

Future research directions include exploring more sophisticated templating strategies, incorporating explainability techniques to enhance interpretability for security analysts, and extending the approach to other security domains beyond network intrusion detection. As attack methodologies continue to evolve in sophistication, hybrid approaches that combine the complementary strengths of language models and traditional techniques will play an increasingly important role in maintaining robust security postures for organizations worldwide.

VIII. Ethical Considerations

Our research on GTL-HIDS exclusively utilizes public network security datasets designed for cybersecurity research, ensuring that we avoid privacy concerns related to real user data. Throughout our development process, we acknowledged that large language models may inherit biases from their training data that could potentially affect threat detection outcomes. Our evaluation methodology focuses strictly on technical performance metrics rather than drawing conclusions about attack origins or demographics. The hybrid architecture processes network flow metadata locally without retaining personally identifiable information, aligning with modern privacy-preservation principles in security research.

We recognize the dual-use potential of advanced intrusion detection systems and have deliberately focused our work on defensive applications rather than offensive capabilities. The framework is designed to detect malicious activity within networks while minimizing computational resource requirements through techniques like embedding extraction and model quantization. By publicly documenting both the capabilities and limitations of our approach, we aim to promote transparency and reproducibility in security research while contributing positively to the field's ongoing efforts to address zero-day vulnerabilities. Our goal is to enhance organizations' defensive capabilities against emerging threats while adhering to responsible research practices.

Acknowledgment

The authors would like to thank Professor Nidhi Rastogi for inspiring this research direction and supporting our project initiative. We also extend our appreciation to Md Tanvirul Alam for his helpful insights that guided our approach to network security analysis. Additionally, we thank the Rochester Institute of Technology's Department of Software Engineering and specifically Professor Ashique KhudaBukhsh for providing the academic environment that made this collaborative work possible.

References

[1] S. Inc. (2024) Estimated cost of cybercrime worldwide 2018-2029 (in trillion u.s. dollars). [Online]. Available: https://www.statista.com/forecasts/1280009/cost-cybercrime-worldwide. [Online]. Available: https://www.statista.com/forecasts/1280009/cost-cybercrime-worldwide

[2] Statista, "Cybercrime expected to skyrocket in coming years," *Statista Chart of the Day*, 2024.

[3] Y. Guo, G. Meng, Y. Liu, and et al., "A survey of machine learning-based zero-day attack detection: Challenges and future directions," *Comput. Commun.*, vol. 198, pp. 175–185, 2023.

[4] V. Kumar and O. P. Sangwan, "Signature based intrusion detection system using snort," *Int. J. Comput. Appl. Inf. Technol.*, vol. 1, no. 3, pp. 35–41, 2012.

[5] P. Garcia-Teodoro, J. Diaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, "Anomaly-based network intrusion detection: techniques, systems and challenges," *Comput. Security*, vol. 28, no. 1-2, pp. 18–28, 2009.

[6] N. A. Ibraheem, H. A. Al-Hasani, M. S. Sharifil, and et al., "Zero day attack vulnerabilities: mitigation using machine learning for performance evaluation," *J. Comput. Society*, 2024.

[7] Z. Dai, L. Y. Por, Y.-L. Chen, and et al., "An intrusion detection model to detect zero-day attacks in unseen data using machine learning," *PLoS ONE*, vol. 19, no. 9, p. e0308469, 2024.

[8] T. Mylla, "Awesome-llm4cybersecurity: An overview of llms for cybersecurity," GitHub repository, 2024.

[9] X. Wen, H. Zhang, S. Zheng, and et al., "From supervised to generative: A novel paradigm for tabular deep learning with large language models," in *Proc. 30th ACM SIGKDD Conf. Knowl. Discov. Data Mining (KDD)*, 2024.

[10] I. Ullah, A. Karlsen, S. Ntlangu, and et al., "When llms meet cybersecurity: a systematic literature review," *Cybersecurity*, 2024.

[11] M. Soltani, M. Shojafar, F. Saberi-Movahed, and et al., "An adaptable deep learning-based intrusion detection system to zero-day attacks," *ResearchGate*, Aug. 2023.

[12] M. A. Ferrag, F. Alwahedi, A. Battah, and et al., "Generative ai in cybersecurity: A comprehensive review of llm applications and vulnerabilities," *ScienceDirect*, 2025.

[13] Y. Wang, J. Hershey, H. Lin, and et al., "A comprehensive overview of large language models (llms) for cyber defences: Opportunities and directions," *arXiv*, 2024.

[14] Y. Chen, Z. Yan, X. Li, and et al., "Nero: Neural algorithmic reasoning for zero-day attack detection," *Comput. Sci. Rev.*, vol. 52, 2024.

[15] R. Hassanin, S. Elfaidy, S. Alshaer, and et al., "Large language models in cybersecurity: Pioneering trends in ai," *Cirrus Labs*, 2024.

[16] M. A. Ferrag, F. Alwahedi, A. Battah, and et al., "Generative ai and large language models for cyber security: All insights you need," *arXiv preprint arXiv:2405.12750*, 2024.

[17] X. Zhou, S. Cao, X. Sun, and D. Lo, "Large language model for vulnerability detection and repair: Literature review and the road ahead," *arXiv preprint arXiv:2404.02525*, 2024.

[18] S. Hegselmann, A. Buendia, H. Lang, and et al., "Tabllm: Few-shot classification of tabular data with large language models," in *Proc. 26th Int. Conf. Artif. Intell. Statist. (AISTATS)*, 2023, pp. 1–36.

[19] M. Cen, X. Deng, F. Jiang, and R. Doss, "Zero-ran sniff: A zero-day ransomware early detection method based on zero-shot learning," *Comput. Security*, vol. 142, p. 103849, 2024.

[20] K. Rieck and P. Laskov, "Language models for detection of unknown attacks in network traffic," *J. Comput. Security*, vol. 16, no. 2, pp. 157–187, 2008.

[21] M. T. Alam, D. Bhusal, L. Nguyen, and N. Rastogi, "Ctibench: A benchmark for evaluating llms in cyber threat intelligence," *arXiv preprint arXiv:2406.07599*, 2024.

[22] Y. Sui, M. Zhou, M. Zhou, and et al., "Table meets llm: Can large language models understand structured table data? a benchmark and empirical study," in *Proc. 17th ACM Int. Conf. Web Search Data Mining (WSDM)*, 2024, pp. 645–654.

[23] M. A. Ferrag, M. Ndhlovu, N. Tihanyi, and et al., "Revolutionizing cyber threat detection with large language models: A privacy-preserving bert-based lightweight model for iot/iiot devices," *arXiv preprint*, 2024.

[24] J. Jiang, K. Zhou, Z. Dong, and et al., "Structgpt: A general framework for large language model to reason over structured data," in *Proc. 46th Int. ACM SIGIR Conf. Res. Dev. Inf. Retrieval*, 2023, pp. 2330–2339.

[25] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proc. 4th Int. Conf. Inf. Syst. Security Privacy (ICISSP)*, Portugal, 2018, pp. 108–116.

[26] E. Hu, Y. Shen, P. Wallis, and et al., "LoRA: Low-Rank adaptation of large language models," *arXiv preprint*, vol. arXiv:2106.09685v2, 2021.

[27] D. Han and M. Han, "Unsloth: Fast lora and qlora fine-tuning for llms," https://github.com/unslothai/unsloth, 2024, accessed: 2025-05-06.

[28] NVIDIA, "Mastering llm techniques: Inference optimization," *NVIDIA Technical Blog*, Jan. 2024.

[29] Lamini, "How to evaluate performance of llm inference frameworks," *Lamini Blog*, 2024.