

Cyber Threat Hunting by Finetuning LLM for DDoS Detection

Ronak Ajwani, Shreya Chhatwani, Abdul Hannan Chougale, Mani Khira, Dr. Mrs. Nupur Giri

Department of Computer Engineering, Vivekanand Education Society Institute of Technology, Chembur, Mumbai, India

Abstract—Cybersecurity threats, particularly Distributed Denial-of-Service (DDoS) attacks, pose significant risks to modern network infrastructures. Traditional detection mechanisms often struggle with scalability and adaptability, necessitating more advanced AI-driven solutions. This research initially explored zero-shot prompting with Large Language Models (LLMs) for cyber threat detection; however, the results highlighted the limitations of general-purpose models in handling domain-specific classification tasks.

To improve detection performance, we adopted a supervised fine-tuning approach, using the CICIDS 2019 dataset. Four state-of-the-art LLMs—Llama 3.1, Llama 3.2, Mistral—were fine-tuned using Low-Rank Adaptation (LoRA) to optimize classification performance while maintaining computational efficiency. The models were evaluated on key performance metrics, including accuracy, precision, recall, F1 score, and latency, with Llama 3.2 showing the highest accuracy and the most balanced trade-offs. Structured prompt engineering further enhanced the effectiveness of fine-tuned models in identifying network threats.

The findings underscore the potential of supervised fine-tuning for adapting LLMs to cybersecurity domains, offering a scalable and robust solution for real-time intrusion detection. Future work may focus on deployment strategies and integration with multimodal threat intelligence to enhance practical applicability.

I. INTRODUCTION

Distributed Denial-of-Service (DDoS) attacks have emerged as a critical threat to modern cybersecurity, targeting networks, servers, and applications by overwhelming them with malicious traffic. These attacks disrupt normal operations, leading to financial losses, service downtime, and data breaches. With the increasing complexity of attack patterns and the rise of automated botnets, traditional rule-based intrusion detection systems (IDS) and machine learning (ML) models struggle to adapt to evolving attack strategies. Consequently, cyberthreat hunting—the proactive identification and mitigation of security threats—has become essential to maintain network resilience.

Recent advances in Artificial Intelligence (AI), particularly in Large Language Models (LLMs), have introduced new possibilities for intelligent threat detection and analysis. LLMs, originally designed for natural language processing (NLP) tasks, possess the capability to analyze structured and unstructured network data, identifying hidden attack patterns with high precision. Their ability to process large-scale datasets, recognize anomalies, and adapt to evolving threats makes them a promising tool for DDoS detection. However, pre-trained LLMs lack domain-specific expertise in cybersecurity and require adaptation to effectively classify and detect network-based threats.

In this study, we investigate the effectiveness of fine-tuned LLMs in DDoS detection by leveraging LoRA (Low-Rank Adaptation) fine-tuning techniques on multiple state-of-the-art LLM architectures, including Llama 3.1, Llama 3.2 and Mistral. Our objective is to analyze how fine-tuning improves model accuracy and whether LLMs can outperform traditional ML-based approaches in detecting malicious network activity. The study evaluates model performance based on accuracy, precision, recall, F1 score, and inference latency to determine the most efficient architecture for real-time DDoS mitigation.

The remainder of this paper is organized as follows. Section II reviews the existing literature on LLMs in cybersecurity and previous approaches to DDoS detection. Section III outlines the novelty of our work, highlighting our fine-tuning strategy and its advantages over existing methodologies. Section IV describes the dataset preprocessing, feature extraction, and labeling techniques applied to the CICIDS 2019 dataset. Section V details the fine-tuning process, prompt engineering strategies, and hyperparameter selection used to optimize model performance. Section VI presents the evaluation metrics and experimental results, comparing different LLMs and their ability to classify network attacks. Finally, Section VII concludes our findings and discusses potential future directions for AI-driven cybersecurity solutions.

II. RELATED WORK

Reference [1] explores the application of Large Language Models (LLMs) like GPT-3.5, GPT-4, and Ada for DDoS attack detection in IoT networks, demonstrating 95%-96% accuracy over traditional machine learning models. However, challenges include false positives, hallucinations, and computational costs. Future work focuses on scalability. Reference [2] introduces DoLLM, which enhances Carpet Bombing DDoS detection by converting network flow data into LLM-compatible formats, improving accuracy by 33.3%. However, scalability remains a concern. Reference [3] presents a systematic literature review on LLMs for cybersecurity, covering applications in DDoS detection, malware analysis, and vulnerability repair. The study highlights dataset limitations, high computational costs, and explainability issues as key challenges, with future research emphasizing model transparency and proactive defenses. Reference [4] proposes DeepLog, a deep-learning-based anomaly detection system for system logs, achieving 91.29 precision on HDFS and significantly reducing retraining time. It excels in real-time anomaly detection but struggles on certain datasets like Hadoop. Reference

[5] utilizes correlation-aware neural networks for DDoS attack detection in IoT systems, improving F1-score to 81% against camouflaged attacks (compared to 35% for non-correlation models). The Pearson correlation technique enhances detection efficiency. Reference [6] applies flow entropy-based DDoS detection, revealing that entropy drops by 91.25% during an attack, making detection easier. The paper also highlights packet sampling as a technique to efficiently analyze large-scale networks for unusual traffic patterns.

The reviewed studies highlight the advancements in AI-driven cybersecurity, particularly in DDoS detection, anomaly identification, and network security. While LLM-based models and deep-learning approaches improve accuracy and efficiency, challenges such as false positives, scalability, and computational overhead remain. Future research must focus on enhancing model transparency, reducing retraining costs, and developing more adaptive threat detection mechanisms to ensure robust cybersecurity defenses.

III. NOVELTY OF THE WORK

The novelty of this research lies in the fine-tuning of Large Language Models (LLMs) for DDoS detection, an approach that has been largely unexplored in comparison to traditional machine learning and deep learning models. Unlike existing works that focus on pre-trained LLMs or conventional machine learning-based intrusion detection systems, our study implements LoRA fine-tuning on multiple LLM architectures—Llama 3.1, Llama 3.2 and Mistral—to enhance their adaptability for network threat classification. This method enables the models to retain their general knowledge while efficiently learning network log-specific patterns, significantly improving their detection accuracy. In contrast to Reference [1], which applies LLMs such as GPT-3.5 and GPT-4 for DDoS detection in IoT networks, our approach fine-tunes open-source models to improve classification accuracy while reducing false positives and computational overhead. Unlike DoLLM [2], which introduces LLM-compatible network flow representations, our method optimizes prompt engineering and adapts LLMs directly to raw network log analysis, ensuring a more seamless and interpretable approach to threat identification. Moreover, unlike DeepLog [4], which relies on deep-learning-based anomaly detection for system logs, our fine-tuned LLMs are capable of contextual threat identification, enhancing both accuracy and generalizability across different types of network traffic. Additionally, our research addresses key challenges highlighted in Reference [3], such as dataset limitations and high computational costs, by leveraging LoRA fine-tuning, which significantly reduces the number of trainable parameters, making model adaptation more feasible even on resource-constrained environments. Unlike entropy-based approaches [6] that rely on statistical traffic behavior, our methodology focuses on contextual understanding of network logs, allowing for more comprehensive and explainable threat detection. Furthermore, correlation-aware models [5] attempt to enhance DDoS detection by leveraging statistical dependencies, whereas our approach infuses language model capabilities

with domain-specific threat knowledge, outperforming pre-trained models in accuracy and efficiency. By demonstrating that fine-tuning LLMs with LoRA significantly improves DDoS detection accuracy, this study sets a foundation for AI-driven network security solutions that can be efficiently deployed in real-time cyber threat monitoring systems. The experimental results show that Llama 3.2 achieves superior accuracy compared to other models, proving that our approach enhances network security capabilities beyond conventional deep learning and LLM-based detection frameworks.

The reviewed studies highlight the advancements in AI-driven cybersecurity, particularly in DDoS detection, anomaly identification, and network security. While LLM-based models and deep-learning approaches improve accuracy and efficiency, challenges such as false positives, scalability, and computational overhead remain. Future research must focus on enhancing model transparency, reducing retraining costs, and developing more adaptive threat detection mechanisms to ensure robust cybersecurity defenses.

IV. DATA GATHERING, MAPPING, AND PREPROCESSING

A. Data Gathering

The dataset used in this study was sourced from Kaggle [?], containing network traffic records labeled with various attack types, including Distributed Denial of Service (DDoS) and benign traffic. It includes essential network-based features such as *Flow Duration*, *Total Forward Packets*, *Total Backward Packets*, and *Flow Bytes/s*, which are instrumental in cyber threat identification.

To facilitate model training and evaluation, the dataset was split into **80% training (800 samples)** and **20% testing (200 samples)**. The final dataset was curated to ensure a **balanced sample of 1000 rows**, with **800 samples of DDoS attacks (DrDoS DNS)** and **200 benign traffic samples**. This balanced distribution ensures a fair classification process, allowing the model to learn attack patterns while retaining a segment of benign network traffic as a baseline.

B. Data Mapping

To enhance data quality and relevance, feature selection and mapping were performed:

- **Feature Selection:** Key network traffic attributes were selected based on their significance in detecting cyber threats.
- **Label Standardization:** Attack labels were converted to lowercase, and redundant spaces were removed to ensure consistency.
- **Data Cleaning:** Duplicate records were eliminated to prevent redundancy and improve dataset integrity.

C. Data Preprocessing

Handling missing and infinite values was crucial for maintaining data consistency. Any missing or infinite numerical values were appropriately managed, and rows containing NaN values were removed after data transformation.

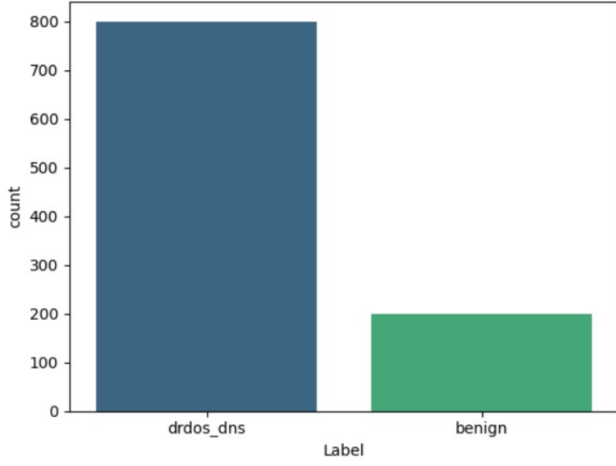


Fig. 1. Distribution of DrDoS DNS attack and benign samples in the dataset.

To standardize feature values, **StandardScaler** was applied, ensuring numerical features remained within a consistent scale. Additionally, categorical labels were preprocessed and formatted uniformly before being used in machine learning models.

The final dataset consists of two primary categories: **DrDoS DNS attack samples (80%)** and **benign traffic (20%)**, ensuring a structured and balanced dataset. This distribution was designed to facilitate the model’s ability to distinguish between attack and non-attack traffic effectively. The dataset was then split into training and testing subsets, with **80% of the data used for training and 20% for testing**, ensuring a comprehensive evaluation of the cyber threat detection model’s performance.

V. METHODOLOGY

1) LLMs Chosen: In this study, we employed state-of-the-art large language models (LLMs) to enhance the detection of Distributed Denial-of-Service (DDoS) attacks. The selected models include LLaMA 3.1, LLaMA 3.2, and Mistral—each recognized for their efficiency and adaptability in natural language processing tasks. LLaMA 3.1 and LLaMA 3.2, developed by Meta, are advanced iterations designed to improve inference speed, contextual understanding, and generalization capabilities. Mistral, an open-weight transformer model, is known for its optimized architecture that balances computational efficiency and predictive accuracy.

These models were chosen based on their ability to process structured network data, identify anomalies, and adapt to variations in cyber threat patterns. Their high adaptability to fine-tuning techniques ensures that they can be effectively trained on domain-specific datasets, such as CICIDS 2019, to enhance their classification accuracy. By leveraging these LLMs, we aimed to determine the most effective model for real-time DDoS detection, evaluating their respective strengths in terms of accuracy, precision, recall, and F1-score.

To enable domain-specific learning, we employed supervised fine-tuning on each of the selected models using the

CICIDS 2019 dataset. This approach allowed the models to learn directly from labeled examples of network attacks, thereby improving their performance in real-world intrusion detection scenarios. Details of the supervised fine-tuning strategy, including data transformation, training configuration, and evaluation methodology, are discussed in the following section. To this end, we adopted a **supervised fine-tuning** strategy, allowing each model to learn directly from labeled examples of normal and attack traffic. This approach improved their contextual understanding and classification capabilities in domain-specific tasks.

2) Fine-Tuning Process: To enhance the performance of large language models (LLMs) in detecting Distributed Denial-of-Service (DDoS) attacks, we conducted a structured fine-tuning process using the CICIDS 2019 dataset. Recognized for its extensive coverage of network intrusion scenarios, this dataset provided a comprehensive set of labeled network logs crucial for training our model. The fine-tuning process was executed on Google Colab, utilizing a T4 GPU to ensure efficient computation and seamless model training. For implementation, we leveraged the Hugging Face transformers library, which offers robust tools for adapting transformer-based architectures to domain-specific tasks. A **supervised fine-tuning strategy** was employed, allowing the models to learn directly from labeled examples of normal and malicious traffic.

The initial step in the fine-tuning pipeline involved importing pre-trained language models, including LLaMA 3.1, LLaMA 3.2, and Mistral into the Google Colab environment. These models were chosen for their distinct capabilities in processing structured textual data and their adaptability to cybersecurity applications. Before proceeding with fine-tuning, we performed an inference test using an initial prompt structure to evaluate how well the models classified network logs in their pre-trained state. The inference was conducted using a structured prompt template where the system was instructed to classify given network logs, taking raw network data as input and returning an expected classification label. This preliminary evaluation provided insights into the model’s baseline capabilities and helped establish a reference point for post-fine-tuning performance comparisons.

In addition to the standard React prompting approach, we also incorporated a few-shot prompting strategy. In this case, we provided the LLM with input examples along with the expected output to guide the model towards more deterministic responses, ensuring a focused direction for classification. This method helped in aligning the LLM’s output with our desired outcomes, improving its consistency in classification tasks.

Given the computational constraints associated with full fine-tuning of LLMs, we adopted the Low-Rank Adaptation (LoRA) technique, a parameter-efficient fine-tuning (PEFT) method that significantly reduces memory consumption while maintaining high adaptability. LoRA modifies the self-attention layers of transformer models by introducing trainable low-rank matrices, thereby allowing selective fine-tuning without the need to update all model parameters. This approach not only reduces GPU memory requirements but

also accelerates training by focusing updates on task-specific adaptations while preserving the pre-trained model’s language comprehension capabilities. The key advantage of LoRA is its ability to enhance model specialization without catastrophic forgetting, ensuring that the LLM retains its general linguistic knowledge while refining its ability to detect cybersecurity threats. Given its efficiency in resource-constrained environments, LoRA emerged as the ideal solution for fine-tuning our models.

Prior to training, we meticulously preprocessed the CICIDS 2019 dataset to eliminate inconsistencies and optimize it for model ingestion. Data cleaning involved handling missing values, standardizing categorical and numerical attributes, and balancing attack labels to prevent model bias. Following this, the dataset was partitioned into an 80

```
system_message = """
You are a cybersecurity expert trained to analyze network flow logs and determine whether the traffic is benign or part of a Distributed Denial of Service (DDoS) attack.

Use the ReAct (Reasoning + Acting) approach:
1. First, REASON through the meaning of the network features such as IPs, ports, protocols, flow duration, packet counts, and byte rates.
2. Then, ACT by concluding whether the flow is BENIGN or DDOSS based on patterns, anomalies, or suspicious behavior.

Be concise but clear in your reasoning, and always end with the final classification in the format: "Final Answer: BENIGN" or "Final Answer: DDOSS".
"""
```

Fig. 2. System Message

```
few_shot_example = """
Example:

Input:
Source IP : 172.16.0.5
Destination IP : 149.171.126.12
Source Port : 49855
Destination Port : 443
Protocol : 6
Flow Duration : 1147838
Total Fwd Packets : 12
Total Backward Packets : 15
Flow Bytes/s : 12.2

Analysis:
Let me analyze this network flow:
- Source IP 172.16.0.5 is a private IP address (likely internal network)
- Destination IP 149.171.126.12 is a public IP
- Source Port 49855 is a high ephemeral port (normal for client connections)
- Destination Port 443 indicates HTTPS traffic (normal web browsing)
- Protocol 6 is TCP (standard for web browsing)
- Flow Duration of 1147838 microseconds (~1.15 seconds) is reasonable for a web request
- Packet counts (12 forward, 15 backward) are balanced and typical for normal web traffic
- Flow Bytes/s at 12.2 is relatively low and consistent with regular web browsing

This appears to be normal HTTPS web traffic from an internal network to an external server
Final Answer: BENIGN
"""
```

Fig. 3. Prompt Template

```
Let me analyze this network flow:

- Source IP 172.16.0.5 is a private IP address (likely internal network)
- Destination IP 192.168.50.1 is a private IP address (likely internal network)
- Source Port 634 is a high ephemeral port (normal for client connections)
- Destination Port 8272 is an unusual port (not a standard service port)
- Protocol 17 is UDP (can be used for various services, but unusual for this destination port)
- Flow Duration of -0.1941285102705073 seconds is negative, indicating a flow that has not started
- Packet counts (0.8428394848929738 forward, -0.1451238114943991 backward) are not meaningful due to negative values
- Flow Bytes/s at -0.411765193712591 is negative, indicating no data transfer

This flow appears to be incomplete or invalid, with a negative flow duration and unusual packet counts.
Final Answer: BENIGN
```

Fig. 4. Response

With the dataset prepared, we proceeded with configuring LoRA hyperparameters to optimize fine-tuning performance. The LoRA-specific parameters included `lora_alpha`, which

determined the scaling factor for adaptation, and `lora_dropout`, which was applied to prevent overfitting during training. Additionally, we set the rank of adaptation matrices (`r`) to define the number of trainable low-rank matrices within the transformer layers. Bias parameters were kept disabled to focus exclusively on weight modifications relevant to threat detection. To ensure that fine-tuning was effectively targeted, we applied LoRA modifications specifically to the self-attention layers of the model, which play a crucial role in contextual pattern recognition.

Alongside LoRA configuration, we optimized key training hyperparameters to ensure stable and efficient learning. The number of training epochs was set between three and five, as longer training durations led to diminishing improvements. Gradient accumulation steps were configured to stabilize weight updates while handling larger batch sizes efficiently. Logging steps were incorporated to monitor loss fluctuations and prevent overfitting, while the AdamW optimizer was employed to enhance convergence stability. A learning rate in the range of $1e-5$ to $5e-5$ was chosen to balance the speed of weight adjustments, and weight decay regularization was applied to maintain controlled updates. The evaluation was conducted at periodic intervals to measure performance on the test dataset and refine training parameters accordingly.

Once all configurations were finalized, the training process was initiated, allowing the models to adapt to the cybersecurity domain using LoRA-enhanced learning. Throughout training, the LLMs underwent iterative refinement, adjusting their attention mechanisms to recognize intricate threat patterns embedded within network logs. Upon completion of fine-tuning, the models were evaluated on the test dataset to assess their classification accuracy. Standard classification metrics such as accuracy, precision, recall, and F1-score were used to quantify model performance, providing a comprehensive assessment of their ability to detect network threats. The results demonstrated a substantial improvement in classification accuracy post-fine-tuning, confirming the efficacy of LoRA in optimizing LLMs for cybersecurity applications. Through a combination of **supervised learning and LoRA optimization**, the models adapted effectively to the cybersecurity context. By leveraging LoRA fine-tuning, we successfully adapted LLaMA 3.1, LLaMA 3.2 and Mistral to detect network threats with improved precision and efficiency.

In addition to the fine-tuning techniques, we also employed a zero-shot classification approach, where the models were tasked with classifying network logs without prior task-specific examples. This allowed us to evaluate the models’ ability to generalize to unseen data, simulating real-world scenarios where labeled data may be scarce.

3) Evaluation Metrics: To assess the effectiveness of the fine-tuned large language models (LLMs) in classifying network intrusions, we employed four key evaluation metrics: accuracy, precision, recall, and F1-score. These metrics provide a comprehensive understanding of the model’s performance by evaluating its ability to correctly identify malicious activities within network logs while minimizing misclassifications.

Hyperparameters	LLaMA 3.1	LLaMA 3.2	Mistral
Learning Rate	0.00003	0.00002	0.00005
Batch Size	16	32	16
Epochs	5	1	5
LoRA Rank (r)	16	64	16
LoRA Alpha	32	16	32
LoRA Dropout	0.05	0	0.05
Bias	none	none	none
Task Type	CAUSAL LM	CAUSAL LM	CAUSAL LM

TABLE I
HYPERPARAMETERS AND LORA CONFIGURATIONS FOR DIFFERENT MODELS

Given the critical nature of intrusion detection, where false positives and false negatives can have significant security implications, a balanced evaluation approach was necessary to ensure that the models were optimized for both detection capability and reliability.

Accuracy serves as a fundamental measure of overall performance, representing the proportion of correctly classified instances out of the total samples. Mathematically, it is defined as the ratio of the sum of true positives (TP) and true negatives (TN) to the total number of predictions, including false positives (FP) and false negatives (FN). Although accuracy is a useful metric for balanced datasets, its reliability diminishes in scenarios where class distributions are skewed, as it may provide an inflated sense of performance when the model predominantly predicts the majority class. Given that network intrusion datasets often exhibit class imbalances, accuracy was considered alongside other performance metrics to obtain a more granular evaluation.

Precision was utilized to measure the model’s ability to correctly identify malicious network activities while minimizing false alarms. Defined as the ratio of true positives to the sum of true positives and false positives, precision quantifies how many of the instances classified as threats were actually threats. This metric is particularly crucial in cybersecurity applications, where a high false positive rate can lead to unnecessary alerts, potentially overwhelming security analysts and reducing trust in automated detection systems. A high precision score indicates that the model is highly selective in flagging network anomalies, ensuring that legitimate traffic is not frequently misclassified as malicious.

Complementary to precision, recall (also known as sensitivity) was used to assess the model’s ability to identify all actual threats within the dataset. It is calculated as the ratio of true positives to the sum of true positives and false negatives, reflecting the proportion of actual attacks that the model successfully detected. Recall is especially important in security contexts, as failing to detect an actual network intrusion can have severe consequences. A high recall score ensures that the model captures the majority of network threats, reducing the risk of undetected intrusions. However, an excessive focus on recall may lead to an increase in false positives, which is why it must be balanced with precision.

To achieve a trade-off between precision and recall, we

utilized the F1-score, a harmonic mean of the two metrics. The F1-score is particularly beneficial in cases where class distributions are imbalanced, as it prevents either precision or recall from disproportionately influencing the evaluation. It is defined as:

A high F1-score indicates that the model maintains a strong balance between identifying actual threats and minimizing false positives. This makes it a more informative metric than accuracy alone, especially when dealing with cybersecurity datasets where both false positives and false negatives have significant operational impacts.

To compute these evaluation metrics, we compared the predicted outputs of the fine-tuned LLMs against the ground truth labels from the CICIDS 2019 dataset. The test dataset, comprising 20% of the total samples, was used exclusively for evaluation to ensure an unbiased assessment of model performance. The classification outcomes were recorded, and a confusion matrix was generated to analyze the distribution of correct and incorrect predictions. This allowed for a more detailed examination of error patterns and facilitated adjustments in hyperparameter tuning where necessary.

Overall, the use of accuracy, precision, recall, and F1-score provided a well-rounded evaluation framework for assessing the fine-tuned models. While accuracy offered a general measure of correctness, precision and recall highlighted the model’s effectiveness in distinguishing between normal and malicious traffic. The F1-score served as a balancing metric, ensuring that the models maintained a practical trade-off between false positives and false negatives.

VI. EXPERIMENTAL RESULTS AND DISCUSSION

The fine-tuned large language models (LLMs) were evaluated based on key performance metrics, including accuracy, precision, recall, F1-score, and latency. These metrics provided a comprehensive assessment of each model’s ability to classify network threats effectively. Among the evaluated models—Llama 3.1, Llama 3.2, Mistral—the best overall performance was achieved by Llama 3.2, which demonstrated the highest accuracy and a well-balanced trade-off between precision and recall.

a) Performance Comparison of Fine-Tuned Models: To quantify the performance of each model, a detailed comparative analysis was conducted. The results of this evaluation

	Model	Approach	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
0	Llama 3.1	Zero-shot	60.0	78.57	68.75	73.33
1	Llama 3.1	Few-shot	65.0	84.62	68.75	75.86
2	Llama 3.1	Fine-tuned	80.0	80.0	100.0	88.89
3	Llama 3.2	Zero-shot	35.0	80.0	25.0	38.1
4	Llama 3.2	Few-shot	75.0	78.95	93.75	85.71
5	Llama 3.2	Fine-tuned	100.0	100.0	100.0	100.0
6	Mistral	Zero-shot	75.0	78.95	93.75	85.71
7	Mistral	Few-shot	80.0	80.0	100.0	88.89
8	Mistral	Fine-tuned	20.0	0.0	0.0	0.0

Table II: Performance Comparison of Language Models Across Different Training Approaches.
Note: All metrics are presented as percentages and color-coded for visual interpretation.

are summarized in **Table II**, which presents the accuracy, precision, recall, F1-score, and inference latency for each model.

From the table, it is evident that Llama 3.2 achieved the highest accuracy and overall performance across all evaluation metrics. Its precision-recall balance, as reflected in the F1-score, outperformed other models, indicating its superior ability to classify network threats effectively. Additionally, while latency varied across models, Llama 3.2 maintained a reasonable inference time, making it an optimal choice for real-time threat detection applications. **Zero-shot** evaluations were also conducted to assess the models' ability to generalize without task-specific fine-tuning examples. LLaMA 3.2 showed superior performance in this setting, effectively classifying unseen network logs. This demonstrates its robustness and adaptability in real-world threat detection scenarios.

decreasing and remaining stable with minor fluctuations. This stability suggests that the model generalizes well to unseen data, confirming that the chosen hyperparameters were well-optimized for fine-tuning.

c) Discussion: The experimental results highlight the effectiveness of fine-tuning LLMs for **DDoS detection** using CICIDS 2019 dataset. The superiority of **Llama 3.2** can be attributed to its enhanced parameter efficiency and adaptability to the structured prompt format used in this study. The use of **LoRA fine-tuning** played a crucial role in optimizing the models for the intrusion detection task, allowing for efficient adaptation without excessive computational overhead. Despite its strong performance, potential improvements could involve reducing inference latency, optimizing training for even better generalization, and evaluating additional LLM architectures to further enhance detection capabilities. Future work could also explore **real-time deployment** of the model in live network monitoring environments to validate its effectiveness in practical cybersecurity applications. In conclusion, the fine-tuned Llama 3.2 model demonstrated superior accuracy and reliability in detecting **DDoS attacks** from network logs, making it a promising candidate for AI-driven **cyber threat hunting** solutions.

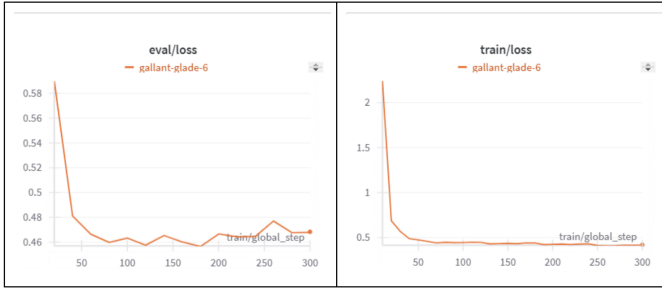


Fig. 5. Evaluation and Training Loss for Llama 3.2

b) Graphical Representation of Model Performance:

To further illustrate the performance of the best-performing model, Llama 3.2, we utilized Weights Biases to visualize the training and evaluation loss trends. Figure presents the loss curves recorded during fine-tuning, providing insights into the model's learning behavior and convergence across training steps.

From the graph, it can be observed that the training loss decreased sharply in the initial steps before stabilizing at a low value, indicating rapid adaptation and effective convergence. The evaluation loss followed a similar trend, gradually

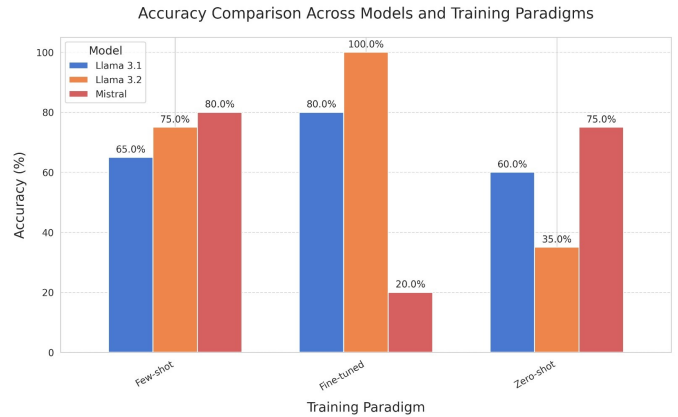


Figure 1: Accuracy comparison showing Llama 3.2's superior performance when fine-tuned (100%) but poor zero-shot performance (35%). Note the significant performance degradation in Mistral when fine-tuned (20%) despite strong few-shot performance (80%).

Fig. 6. Accuracy comparison of LLMs across training methods.

VII. CONCLUSION

This research examined the fine-tuning of large language models (LLMs) for cyber threat hunting, specifically DDoS detection using the CICIDS 2019 dataset. Four LLMs—Llama 3.1, Llama 3.2, and Mistral—were fine-tuned with LoRA to enhance adaptability while reducing computational costs. Among them, Llama 3.2 achieved the highest accuracy and a well-balanced precision-recall trade-off, making it the most suitable for intrusion detection. Structured prompt engineering further improved model generalization and classification performance. Additionally, the **zero-shot** classification approach demonstrated Llama 3.2's superior ability to classify network logs without requiring task-specific examples, further enhancing its real-world applicability.

The findings highlight the potential of fine-tuned LLMs in cybersecurity, particularly for automated threat detection. Future research can focus on real-time deployment, optimizing inference latency, and expanding detection to other network threats using multi-modal threat intelligence. This study demonstrates that targeted fine-tuning enables LLMs to enhance AI-driven cybersecurity solutions for mitigating complex threats.

REFERENCES

- [1] M. Guastalla, Y. Li, A. Hekmati, and B. Krishnamachari, "Application of Large Language Models to DDoS Attack Detection," University of Southern California, Los Angeles, CA, USA, 2023.
- [2] Q. Li et al., "DoLLM: How Large Language Models Understanding Network Flow Data to Detect Carpet Bombing DDoS," arXiv preprint arXiv:2405.07638, 2024.
- [3] Y. Chen et al., "Large Language Models for Cyber Security: A Systematic Literature Review," arXiv preprint arXiv:2405.04760, 2024.
- [4] M. Du, F. Li, G. Zheng, and V. Srikumar, "DeepLog: Anomaly Detection and Diagnosis from System Logs through Deep Learning," in Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS '17), Dallas, TX, USA, 2017, pp. 1285–1298.
- [5] A. Hekmati, N. Jethwa, E. Grippo, and B. Krishnamachari, "Correlation-Aware Neural Networks for DDoS Attack Detection In IoT Systems," arXiv preprint arXiv:2302.07982, 2023.
- [6] M. E. Ahmed, H. Kim, and M. Park, "DDoS Attack Detection Using Flow Entropy and Packet Sampling on Huge Networks," IEEE Transactions on Network and Service Management, vol. 15, no. 1, pp. 130–144, 2018.