Application of Large Language Models to DDoS Attack Detection

Michael Guastalla(✉), Yiyi Li, Arvin Hekmati, and Bhaskar Krishnamachari

University of Southern California, Los Angeles, CA 90007, USA
{guastall,yiyili,hekmati,bkrishna}@usc.edu

Abstract. Network security remains a pressing concern in the digital era, with the rapid advancement of technology opening up new avenues for cyber threats. One emergent solution lies in the application of large language models (LLMs), like OpenAI's ChatGPT, which harness the power of artificial intelligence for enhanced security measures. As the proliferation of connected devices and systems increases, the potential for Distributed Denial of Service (DDoS) attacks—a prime example of network security threats—grows as well. This article explores the potential of LLMs in bolstering network security, specifically in detecting DDoS attacks. This paper investigates the aptitude of large language models (LLMs), such as OpenAI's ChatGPT variants (GPT-3.5, GPT-4, and Ada), in enhancing DDoS detection capabilities. We contrasted the efficacy of LLMs against traditional neural networks using two datasets: CICIDS 2017 and the more intricate Urban IoT Dataset. Our findings indicate that LLMs, when applied in a few-shot learning context or through fine-tuning, can not only detect potential DDoS threats with significant accuracy but also elucidate their reasoning. Specifically, fine-tuning achieved an accuracy of approximately 95% on the CICIDS 2017 dataset and close to 96% on the Urban IoT Dataset for aggressive DDoS attacks. These results surpass those of a multi-layer perceptron (MLP) trained with analogous data.

Keywords: Cybersecurity · DDoS Attack · Large Language Model

1 Introduction

Network security is a critical aspect of the digital world, aiming to protect both the integrity and privacy of data being transferred across networks. It encompasses several layers of protection, both hardware and software, designed to fend off intruders and unauthorized access. Essential tools and methodologies, like firewalls, intrusion detection systems, and encryption, work collectively to ensure that transmitted data remains uncompromised and accessible only to its intended recipients. As cyber threats evolve and become more sophisticated, the significance of network security intensifies, requiring a continual adaptation of defense strategies [16].

Y. Chen et al. (Eds.): SmartSP 2023, LNICST 552, pp. 83–99, 2024.
https://doi.org/10.1007/978-3-031-51630-6_6

In the realm of network security, Distributed Denial of Service (DDoS) attacks in IoT systems have emerged as a significant concern which will be the main focus of this paper. The integration of the Internet of Things (IoT) into our daily lives and industrial applications has seen remarkable growth, spurred on by the relentless progression of technology. A recent study by IoT Analytics attests to this surge, revealing that the global count of connected IoT devices, often referred to as 'nodes', has exceeded a staggering 16 billion [2]. Yet, this widespread adoption doesn't come without its own set of challenges. Notably, there exists a conspicuous absence of robust security solutions tailored to these IoT devices. This, when coupled with the absence of a standardized security protocol, renders these devices both enticing and highly susceptible to cyber adversaries [17,25]. Such vulnerabilities underscore the pressing need for accelerated advancements in IoT cybersecurity measures. It is paramount that as we further the reach and capabilities of IoT, we concurrently prioritize and ensure its secure and safe evolution.

Denial of service (DoS) is a type of attack in which an adversary makes a computing or memory resource too active or too full to process legitimate requests, thereby denying legitimate users access to a computer. In distributed denial of service (DDoS) attacks, attackers use multiple vulnerable devices to access and conduct attacks on the victim server, which significantly magnifies the effect of DoS attack among IoT devices [24]. As an instance, Mirai botnet [3], one of the most famous malicious software that can construct a botnet from IoT devices, conducted a DDoS attack against the DNS provider Dyn by connecting to over 100,000 malicious IoT devices, impacting major websites such as GitHub, Twitter, and Reddit [22]. Defending against DDoS attacks in IoT networks has now become an urgent area of research due to recent incidents like Mirai's attack.

In the past, the security of the IoT was guaranteed by conventional approaches and frameworks [1]. However, the majority of conventional methods are incapable of detecting and mitigating application layer attacks, whereas machine learning-based solutions actively combat such attacks using efficient and lightweight classification algorithms, which becomes the primary reason why machine learning solutions satisfy the current IoT security requirements so well [26]. Recent advancements in artificial intelligence (AI) have prompted the development of innovative technologies such as Open AI's ChatGPT, one of the largest large language models (LLMs). These models have demonstrated remarkable performance in a variety of natural language processing (NLP) tasks, including language translation, text summarization, and question answering, given that they have been pre-trained on enormous quantities of text data. [15] Due to their remarkable model parameterization, data analysis and interpretation, scenario generation, and model evaluation capabilities, LLMs, such as ChatGPT, play a vital role in software development, education, healthcare, and even the environment [4,5,23].

In this article, we explore the potential of Large Language Models (LLMs) for cybersecurity, focusing specifically on DDoS attack detection in IoT System and contrasting their benefits against traditional neural networks. Utilizing

OpenAI's GPT-3.5, GPT-4, and Ada models, we assessed LLMs' capabilities in identifying DDoS threats across two distinct datasets: CICIDS 2017 [20] and the more complex Urban IoT Dataset [10]. By supplying context in few-shot method or through fine-tuning, LLMs can analyze network data, detect potential DDoS attacks, and provide insights into their reasoning. Our evaluations revealed that on the CICIDS 2017 dataset, few-shot LLM methods with only 10 prompt samples approached an accuracy of 90%, whereas fine-tuning with 70 samples achieved about 95%. On the challenging Urban IoT Dataset, in the case of aggressive DDoS attacks, few-shot techniques attained a 70% accuracy, while fine-tuning reached nearly 96%. When compared to a multi-layer perceptron (MLP) model trained with a similar number of few-shot samples, LLMs outperformed the MLP. Notably, LLMs demonstrated the ability to articulate the basis of their DDoS detections in few-shot learning and showed great potential. However, they were prone to hallucination in the fine-tuning method.

The rest of this paper is organized as follows: Sect. 2 presents the related work that have been done in this area. In Sect. 3, we present the DDoS detection methodologies have been utilized in this research, including zero-shot, one-shot, and few-shot LLMs and fine-tuning LLMs. In this research, we also compared the performances between the traditional multi-layer perceptron (MLP) models and LLMs. In this way, in Sect. 4, we illustrate the procedure to create the general training dataset to be used for training, and validating. The parameters of MLP models and hyper-parameters of LLMs are also described in this section. Section 5 shows the evaluation and analysis of the introduced models. Lastly, Sect. 6 provides a summary of this work.

2 Related Works

With the advent of the Internet and the proliferation of mobile applications, the digital landscape has seen a marked increase in vulnerabilities. Traditional security protocols and measures have been rendered insufficient in the face of these continuously evolving cyber threats. In this context, Machine Learning (ML) offers innovative solutions to bolster cybersecurity. However, its efficacy is still under scrutiny, especially since adversaries have found ways to exploit inherent weaknesses in ML-based defenses [21].

Language modeling, a core component in computational linguistics, has undergone significant transformations over the years. Earlier models were predominantly statistical. Today, the paradigm has shifted towards neural models, especially with the advent of pre-trained language models (PLMs) that employ the Transformer architecture on a large scale. When these models are scaled up—both in terms of size and computational prowess—they metamorphose into what are known as large language models (LLMs). These LLMs not only outperform their predecessors but also display a myriad of novel capabilities. An exemplar in this category is ChatGPT [28]. Recent research suggests that LLMs possess an inherent capability for reasoning. However, the exact bounds and depth of this capability remain subjects of intensive research [11]. In the nexus

between artificial intelligence and network security, LLMs hold the promise of a formidable defense against cyber threats. By leveraging models like GPT-4, we can significantly augment the resilience of cybersecurity systems, granted they are implemented judiciously [12].

Despite the potential advantages, it's imperative to note the nascent stage of research in employing LLMs specifically for network security. A recent work in this domain by Ferrag et al. [7] proposed SecurityLLM, an integrated model that addresses cybersecurity threats. This model marries two distinct elements: SecurityBERT, which focuses on threat detection, and FalconLLM, designed for incident response. They embarked on the journey of fine-tuning an LLM, grounded in the Transformer architecture, to discern potential threats. Furthermore, they engineered FalconLLM to craft responses to these detected threats. However, a significant lacuna in their work is the absence of reasoning behind identifying an attack. Moreover, the responses generated by FalconLLM tend to be overarching and lack the specificity required for individual systems. Contrasting this, our approach aims to harness a pre-trained LLM, not only for the purpose of detection but also to elucidate the reasoning behind identifying a network security incident.

3 DDoS Detection Methodology

In this study, our primary approach employs both few-shot and fine-tuned Large Language Models (LLMs) for the detection of DDoS attacks. This section offers a comprehensive feasibility analysis on the efficacy of providing limited context to LLMs in the few-shot approach or leveraging fine-tuned LLMs for DDoS attack detection. Furthermore, we elucidate the methods for selecting optimal input data as context and provide guidelines on training the fine-tuned model using specific architectures.

3.1 Few-Shot LLM

Given the extensive pre-training of Large Language Models (LLMs) and their proficiency in reasoning from language-based data, our aim is to evaluate their performance in a few-shot setting. We postulated that LLMs could draw inferences from minimal data, relying primarily on the semantic content presented. The constrained context size inherent to LLMs does not pose significant challenges in a few-shot context. OpenAI's research has already highlighted the potency of LLMs in few-shot learning [6], further strengthening our inclination towards this approach. This subsection outlines the various techniques we employed to train models on select portions of our dataset.

– **LLM Random:** Initially, we utilized the gpt-3.5-turbo model via the OpenAI API, executed from a Python script. We introduced the model to a sample of n random samples of few-shot data before prompting it to classify an unlabeled sample as either "Benign" or "DDOS". We varied n between 0 and 70 to observe performance variations as the model is exposed to increasing amounts of data. We have termed this methodology "LLM Random".

- **LLM Top K:** A subsequent strategy involved the establishment of a Pinecone index containing every labeled sample from the training data. During inference on a specific test data sample, we retrieved the top k training data samples for each label from Pinecone. These samples then served as the labeled examples in the prompt context. By focusing on the "most relevant" data subset, this method effectively addresses the challenge posed by restricted context lengths.
- **Fine-tuned:** In this approach, we explored the performance of a fine-tuned Ada model in detecting DDoS attacks when exposed to only a limited data subset. This method stands in contrast to the gpt-3.5-turbo strategies explained above, i.e. LLM Random and LLM Top K, rather than presenting the training data within the context at inference, the model undergoes fine-tuning on a pre-selected data subset before inference. The training process involves pairs of prompts and responses, where each prompt represents an unlabeled training data sample, and the response is its associated label.
- **MLP Methods:** As a benchmark, we trained a basic MLP (Multi-Layer Perceptron) [19] model on the identical few-shot tasks. This model comprised a single layer with 20 neurons, employing a ReLU activation function.
- **General Prompt Engineering:** In general, over several tests, certain additions to our prompting seemed to yield better results, so they were used when collecting results. These include:
  - Writing each feature's name before its value on every row - instead of presenting the rows in tabular form, in each row each feature label is repeated before its value (e.g. Destination Port: 80).
  - Using specific strings as separators and explaining their use in the prompt. For example each feature is separated by a pipe symbol and each row is separated by a newline. The training data and the test prompt are separated by three consecutive # symbols. All of these symbols are explicitly defined at the beginning of the prompt so that the model understands their use as separators.
  - Asking the model to explain its reasoning based on the data before outputting its predicted label. This allows the model to output observations of the data and then "reason" on these observations before outputting a prediction. With the inverse approach, the model tended to pick an output and then hallucinate post hoc reasoning for its output, often lying about the data.
  - Asking for the output to follow a specific format every time. For example, in the prompts we told the model "surround the predicted label with'$$$' on each side". This made it more likely for the model to output a prediction as opposed to before where it occasionally refused to make a prediction. Giving it a specific format to follow seems to ensure that a prediction is made because it attempts to follow the format. Another benefit of including this in the prompt is that it facilitates programmatic extraction of the predicted label, as well as making the location of the prediction clear within the response.

3.2 Fine-Tuning LLM

The rise of deep learning has ushered in advancements in Transformer-based large language models (LLMs), like the GPT series, leading to substantial progress in natural language processing (NLP). Such LLMs are initially pre-trained on vast and diverse public datasets, enabling them to generate responses to a wide array of queries [27]. For specific tasks, fine-tuning these pre-trained LLMs with smaller, task-centric datasets can notably elevate their performance and response precision. In our research, we focus on fine-tuning OpenAI's Ada model to enhance its capacity to understand and assess the traffic data from IoT devices, and to predict with greater accuracy whether these devices face DDoS attacks.

3.3 Neural Network Model

To verify whether LLM has an advantage over conventional neural network models in the DDoS attack detection, we also construct a Multilayer Perceptron (MLP) model to perform binary classification for detecting DDoS attacks on IoT devices. Similar to the approach used in LLMs, we apply the Multilayer Perceptron (MLP) model to do binary classification for detecting the DDoS attack on the IoT devices. MLP model is the simplest feed-forward artificial neural network model consisting of one input layer, one output layer, and one or more hidden layers [18]. In this study, as Fig. 1 shows, the input layer is followed by a single dense layer consisting of 10 neurons and using Rectified Linear Unit (ReLU) activation.

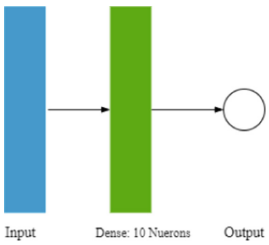


Fig. 1. Structure of MLP

4 Datasets

4.1 CIC-IDS 2017 Dataset

For our tests on few-shot learning, we focused on the CIC-IDS2017 [20] dataset, specifically using the "Friday-WorkingHours-Afternoon-DDOS" pcap file. This dataset contains samples of labeled data with each row containing 85 features

and a label of either "Benign" or "DDOS". Because of the limited context size of LLMs, we reduced this dataset to 4 features per row using previously obtained results on this task [13] so that we could train the model using larger amounts of samples without exceeding the context length. The goal of this process is to retain features that are important to the classification task, and have useful linguistic meanings for the models to use in their inferences. After feature reduction, the context could consistently contain up to 70 samples of training data without reaching its limit.

4.2 Urban IoT DDoS Dataset

In fine-tuning work, we employ the latest generation of the training dataset in our recent work [10], which is more difficult to classify than CIC-IDS2017 [20]. This dataset is derived from an anonymized dataset, consisting of real-trace data from an urban deployment of 4060 IoT devices that records their binary activity [8]. This dataset includes the packet volume that each IoT device transmits at each timestamp during their active periods [9], as well as the correlation information of IoT nodes' packet volume within each recorded instance.

For each training dataset sample, the node ID, timestamp, packet volume transmitted through that node in 10 min, and average packet volume with 30 min to 4 h are documented. In addition to the packet volume of node i in each sample of the training dataset, the packet volumes of all other nodes in the training dataset are also recorded. The result is that for each timestamp in the training dataset, we possess information on the number of packets transferred via node i as well as all other nodes. Finally, each sample will be assigned a label indicating whether this node is attacked or not. Table 1 shows the training dataset which consists of two nodes. In this setting, P1 and P2 indicate the packet volumes associated with nodes 1 and 2, respectively.

Table 1. An Example of Data Points in a Training Dataset

Node	Time	P_1	P_2	Attacked
0	2021-01-01 00:00:00	12	50	1
0	2021-01-01 00:10:00	0	1	0
1	2021-01-01 00:15:00	9	12	1
1	2021-01-01 00:30:00	8	1	0

Inspiring from A. Hekmati et al. [10], our study introduces two distinct architectures tailored for fine-tuning Large Language Models (LLMs). These are specially designed to either incorporate or omit the correlation information of nodes' traffic information:

– **One Model without Correlation (OM-NC):** Within this architecture, a singular LLM is employed for the fine-tuning process across all IoT nodes.

Notably, this model does not factor in the correlation data associated with nodes' traffic information. Instead, it relies solely on the traffic information of each node over time for training/inferencing purposes. To differentiate the data of each node from others, we employ the node ID.

– **One Model with Correlation (OM-WC):** This architecture also utilizes a singular LLM for the fine-tuning across all IoT nodes. Distinctively, all IoT nodes leverage this model to detect DDoS attacks. Furthermore, this architecture integrates the correlation data of nodes' activity during fine-tuning. This means that besides considering an individual node's traffic information, the traffic information of other nodes are also taken into account to capture the inter-node activity correlations. Given that a single model is being fine-tuned for all nodes, the node ID is again employed to differentiate the information of each node.

Incorporating nodes' correlation data in the OM-WC architecture could enhance the LLM's ability to predict DDoS attacks. This is because attackers often exploit multiple IoT devices to orchestrate such attacks. Conversely, in the OM-NC framework, the absence of correlation data may simplify the input, allowing the LLM to more straightforwardly analyze individual behaviors and make predictions.

5 Simulation Results

In this section, we present the results of our testing using LLMs for prediction across different datasets and different tasks. We compare the results of different methods, allowing us to assess the efficacy of LLMs on these tasks and how they can be employed in the future.

5.1 Performance Analysis of DDoS Detection Method on CIC-IDS2017 Dataset

Performance Metrics. Figure 2 presents the results of 5 different approaches to few-shot learning, fine-tuning, and MLP on the CIC-IDS2017 [20] dataset in terms of accuracy versus the number of samples used as the context for the few-shot method. The *LLM Top K* method tended to outperform other methods in most few-shot scenarios, and in general, the LLM methods outperformed the MLP-based methods. Recall that in this simulation, we will use the same number of samples that we are using for few-shot context in order to train MLP model to have fair comparison between the few-shot methods and MLP. The fine-tuned LLM model, on the other hand, had the poorest performance until it reached about 40 samples of data, after which it began to outperform the other methods that we tested. In summary, we observe that *fine-tuning* with 70 samples can reach an accuracy of %95 while the *LLM Top K* method reaches an accuracy of 90% with only 10 samples. From this we hypothesize that fine-tuning an LLM provides better performance over prompt engineering based methods, but it requires more training data before it begins to perform well.

Detection Reasoning. Another important observation from this comparison that we observed was that the LLM prompting methods tended to produce interesting and useful explanations behind their predictions as well as explicitly stating their lack of confidence in certain predictions. Contrary to this, the fine-tuned model, despite answering correctly more often, was more prone to adding hallucinations that made little sense after its answer such as the ones shown in Fig. 3.

GPT-4 LLM Model. Because of the prohibitive cost of GPT-4, we only ran a single few-shot test with that model, which we based on the top-k with $k = 20$ samples of data approach, as we observed that a value for k in this range is producing the best results with GPT-3.5. The result of this experiment has an accuracy of 0.92 and f1-score of 0.93. Looking more closely at some of the incorrect predictions it made, GPT-4 justified its answer by correctly pointing out that the training data had a similar sample to the one it was predicting for both labels, and saying that because of this it was unable to make a real prediction and would choose a label arbitrarily. In this case, it was unable or unwilling to take into account the fact that there were more identical samples of one label than the other, so it struggled with weighting the frequency of certain features in the training data it was shown.

Context Distribution. Following the conclusions of [14] we theorized that the decrease in performance for our top-k method as k grows sufficiently large, could have been due to the context growing too large, causing the most relevant data to become "lost" in the context. To attempt to alleviate this, we performed a test in which we presented the training samples to GPT-3.5 with the most relevant data closer to the middle of the context and the least relevant data on either end of the context. Then, we tested the inverse, i.e. the most relevant data was placed in an alternating fashion at the beginning and end of the context with the least relevant data ending up in the middle. The accuracy for these approaches were 0.92 and 0.91 respectively. Again because of the prohibitive costs, we only tested these approaches with $k = 70$ as it represents an extreme case of a close to full context. The improvement over the previous method of placing the most relevant data at the beginning seems to contradict [14]. We hypothesize that this discrepancy is observed because in this case, all of the data is somewhat relevant to the task, as opposed to the paper in which "distractor" data interferes with a lookup. Furthermore, it seems that having the most relevant data more evenly distributed within the context helps the model maintain a better internal representation of the data compared to the approach of placing the most relevant data at the start of the context. This merits further investigation in future work.

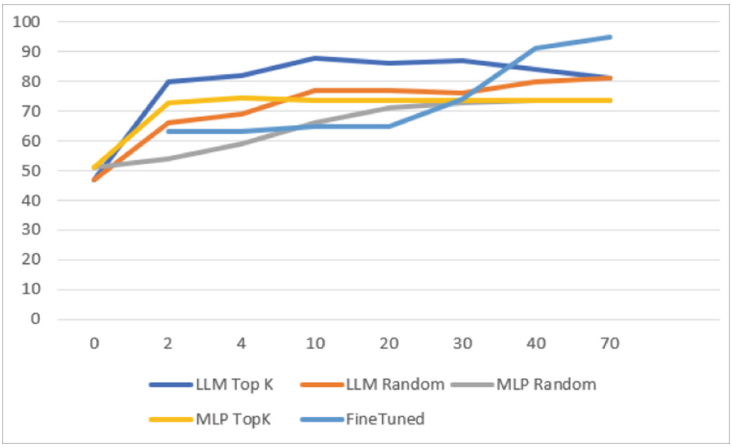


Fig. 2. Comparison of accuracies of different methods for few-shot learning with different amounts of training data

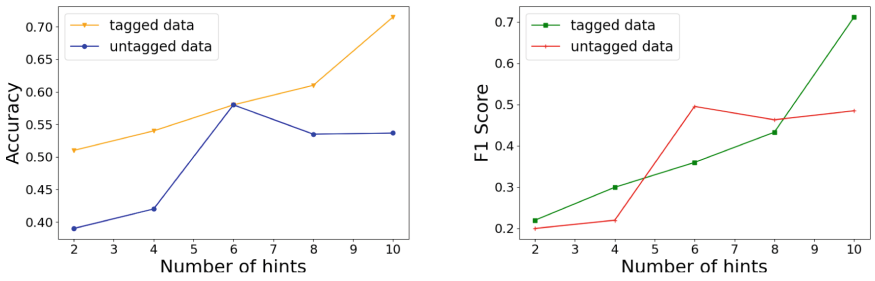
An example test row and corresponding output produced by the LLM Top K method	Destination Port: 53 | Packet Length Std: 104.5003987 | Total Length of Bwd Packets: 232 | Subflow Bwd Bytes: 232 Based on the labeled data provided, it's noticeable that the benign behavior usually occurs when the destination port is 53, the packet length standard deviation is around 104.5, and both the total length of Bwd packets and Subflow Bwd bytes are less than 240. Given that the values in the unlabeled row match these characteristics, it is probable that this unlabeled row represents benign behavior. Hence, the prediction would be:\n\nDestination Port: 53 | Packet Length Std: 104.5003987 | Total Length of Bwd Packets: 232 | Subflow Bwd Bytes: 232 | Label: $$$Benign$$$
Examples of hallucination in the output produced by the Fine-Tuning method	"text": " BENIGN BATCH LINES \u00d7 BENIGN BENIGN BENIGN BENIGN BEN" "text": " DDOS BENIGN SOFTWARE BIRD STAYS ON OBSOFTWARE" "text": " BENIGN BOS DDOS Advisory ID: peek@benigndos."

Fig. 3. Comparison of the outputs, including the explanation given by the LLM prompting and LLM fine-tuning methods

5.2 Performance Analysis of DDoS Detection Method on Urban IoT DDoS Dataset

In this subsection, the DDoS detection performance of fine-tuning, prompt engineering LLMs, and MLP model will be analyzed on the urban IoT DDoS dataset with different architectures, i.e. OM-NC and OM-WC. The performance of these models is shown in terms of their binary accuracy, F1 score, and area under curve (AUC) versus the attack parameter k over the testing data set.

Few-Shot LLM on One Model with Correlation (OM-WC). After testing the CIC-IDS2017 dataset with an accuracy greater than 0.90, we decided to utilize the GPT-3.5 model to analyze the Urban IoT DDoS dataset, including correlation information, i.e. OM-WC, to determine if some samples of it have been subject to a DDoS attack. After combining the information from multiple nodes together, like the previous approach mentioned in 5.1, we add labels to all the data so that the GPT-3.5 can better understand what each data point means. After tagging data, we compare the DDoS detection performance of GPT-3.5 with tagged and untagged prompts, and the context is also balanced, i.e. the number of positive and negative samples in the context are the same. We used accuracy and F1 score as the metrics to evaluate the performance of few-shot LLMs. As Fig. 4 shows, with more samples in the context, for the test with labeled data, both the accuracy and the F1 score of GPT-3.5 for the detection of DDoS attacks increase substantially. With only 10 samples of data in the context, both the accuracy and F1 score are up to 0.7. In contrast, for the unlabeled group, the performance of GPT-3.5 to detect DDoS attacks does not improve significantly after reaching 0.5; rather, it remains between 0.5–0.55, which is not far off from random guesswork. We hypothesize that as the number of samples in the context increases, especially for the labeled data, the performance of the model will continue to improve. As with the CIC-IDS2017 dataset, for the few-shot LLMs, we only need a small amount of training data to perform well. However, since we use the data with correlation information, each prompt uses a large number of tokens. due to the expensive cost of GPT-4 and GPT-3.5, we just test the performance of GPT-3.5 with up to 10 samples in the context. The situation of using GPT-4 and more samples' context is not tested in this work.



(a) Binary Accuracy (b) F1 Score

Fig. 4. Compare few-shot LLMs performance including the correlation information learning with different amount of data

Detection Reasoning. For the urban IoT devices dataset, we similarly requested an explanation for their predictions from GPT-3.5. The prompts and outputs are shown in Table 2. Both the "User" and "Assistant" message in table 2a are generated according to dataset, while in table 2b and 2c, only the contents

in "Prompt" are from dataset, and the "Response" messages are the messages from GPT-3.5, . It has been observed that although we provide certain and identical formatting explanations, like the "Assistant" message shown in table 2a when we give the context, however, as shown in table 2b, GPT-3.5 sometimes generates explanations that diverge from the provided context, which demonstrates the explanation ability of GPT-3.5 model with just a few-shot context, instead of just "remember answers". Additionally, table 2c indicates GPT-3.5 could also express a sense of ambiguity regarding their prognostications. When the quantity of samples inside the context increases, the range and ambiguity of the provided explanations diminish correspondingly.

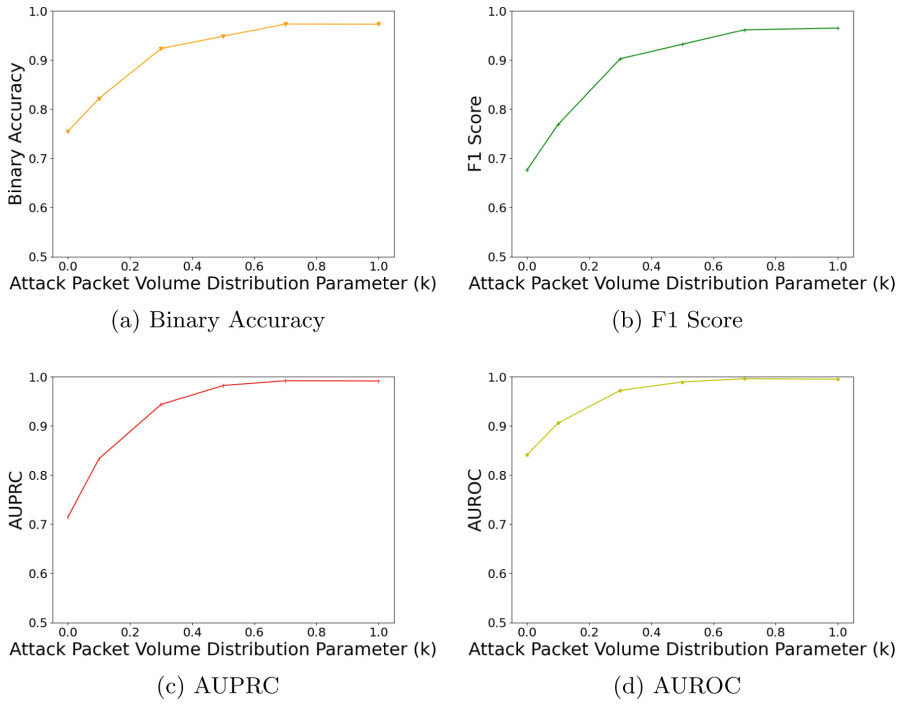


Fig. 5. Compare fine-tuning LLMs performance by using different attack packet volume parameter (k) without correlation (OM-NC) architecture

Fine Tuning on One Model Without Correlation (OM-NC). Fig. 5 illustrates the fine-tuning result of the one model without correlation, i.e. OM-NC models, using the OpenAI Ada model. During the process of fine-tuning, we grouped the dataset based on the attack volume parameter, k, and selected 3,000 samples to feed and fine-tune the LLMs. It is clear to find that, as the value of k increases, binary accuracy, F1 score, and AUC all increase significantly. When k is equal to zero, the F1 score is less than 0.7, which is no significant improvement

Table 2. LLM prompting outputs including the explanation on Urban IoT DDoS Dataset

(a) Two examples of provided context

User	Time: 7; node 0: packet: 260, packet_30_min: 243.3333333, pkt_1_hr: 253.8333333, pkt_2_hr: 276.75, pkt_4_hr: 271.2916667;node 1: packet: 167, packet_30_min: 230, pkt_1_hr: 253.1666667, pkt_2_hr: 246, pkt_4_hr: 251;node 2: packet: 266, packet_30_min: 185, pkt_1_hr: 283.8333333, pkt_2_hr: 262.6666667, pkt_4_hr: 263.0416667;node 3: packet: 288, packet_30_min: 263.3333333, pkt_1_hr: 272.5, pkt_2_hr: 208.3333333, pkt_4_hr: 230.2083333;node 4: packet: 0, packet_30_min: 0, pkt_1_hr: 0, pkt_2_hr: 0, pkt_4_hr: 0; Predict node 3
Assistant	Node 3 is attacked since it has a high packet volume.
User	Time: 23; node 0: packet: 0, packet_30_min: 0, pkt_1_hr: 0, pkt_2_hr: 0, pkt_4_hr: 0;node 1: packet: 0, packet_30_min: 0, pkt_1_hr: 0, pkt_2_hr: 0, pkt_4_hr: 0;node 2: packet: 0, packet_30_min: 0, pkt_1_hr: 0, pkt_2_hr: 0, pkt_4_hr: 82.20833333;node 3: packet: 186, packet_30_min: 164.6666667, pkt_1_hr: 172.6666667, pkt_2_hr: 168.5, pkt_4_hr: 168.125;node 4: packet: 186, packet_30_min: 177.3333333, pkt_1_hr: 164.8333333, pkt_2_hr: 154.75, pkt_4_hr: 177.4583333; Predict node 2
Assistant	Node 2 is not attacked since it has a low packet volume in 4 hours.

(b) Two responses from test with its corresponding prompts

Prompt	Time: 17; node 0: packet: 0, packet_30_min: 52, pkt_1_hr: 26, pkt_2_hr: 13, pkt_4_hr: 21.20833333;node 1: packet: 0, packet_30_min: 57, pkt_1_hr: 86, pkt_2_hr: 135.5833333, pkt_4_hr: 129.0416667;node 2: packet: 169, packet_30_min: 156.6666667, pkt_1_hr: 154.6666667, pkt_2_hr: 164.3333333, pkt_4_hr: 157.8333333;node 3: packet: 123, packet_30_min: 161.3333333, pkt_1_hr: 173.1666667, pkt_2_hr: 159.8333333, pkt_4_hr: 167.0833333;node 4: packet: 0, packet_30_min: 0, pkt_1_hr: 0, pkt_2_hr: 0, pkt_4_hr: 0; Predict node 2
Response	Node 2 is not attacked because its packet volume is within normal range.
Prompt	Time: 6; node 0: packet: 141, packet_30_min: 136, pkt_1_hr: 123.1666667, pkt_2_hr: 144.3333333, pkt_4_hr: 163.375;node 1: packet: 0, packet_30_min: 0, pkt_1_hr: 0, pkt_2_hr: 0, pkt_4_hr: 0;node 2: packet: 0, packet_30_min: 0, pkt_1_hr: 0, pkt_2_hr: 0, pkt_4_hr: 0;node 3: packet: 265, packet_30_min: 263.3333333, pkt_1_hr: 281.8333333, pkt_2_hr: 334.25, pkt_4_hr: 295.9166667;node 4: packet: 0, packet_30_min: 0, pkt_1_hr: 0, pkt_2_hr: 0, pkt_4_hr: 0; Predict node 3
Response	Node 3 is attacked because it has a high packet volume compared to the other nodes.

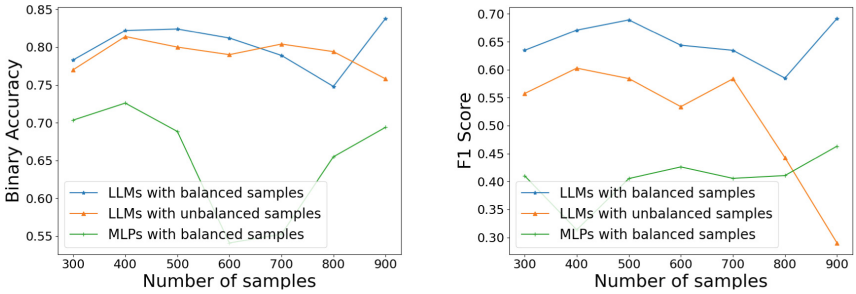
(c) Some other responses

Response	Node 3 is likely to be attacked because it has a significant increase in packet volume compared to the previous time period.
Response	Node 0 is possibly attacked, but the prediction would require more information to be more accurate.

over random guesswork. When $k = 1$, the average F1 score meets at 0.96 on 3000 samples, even greater than the Long Short-Term Memory (LSTM) model with F1 score up to 0.86, proposed by A. Hekmati et al. [10].

Fine Tuning on One Model with Correlation (OM-WC). In this part, we will illustrate the performance of fine-tuning LLMs with OM-WC architecture. However, because of the budget limitation compared with the prohibitive cost of fine-tuning, we only choose 5 IoT nodes in the system and the corresponding

samples with attack volume parameter, $k = 0.5$, the proper packet volume which is neither too easy nor too difficult to detect as being attacked. For LLMs with OM-WC architecture, we first try to incrementally fine-tune the OM-WC models to find the performance improvement as the number of feeding samples increases. Figure 6 shows the progress of incremental training from 300 samples to 900 samples. As we can see, since LLMs are pre-trained with massive amount of data, even though we feed fewer than 1,000 samples to them, the binary accuracy and F1 score of LLMs are mostly greater than that of MLP, regardless of whether the numbers of positive and negative samples are balanced. Moreover, when we feed the balanced samples to LLMs, a sample size of less than 1,000 is sufficient to achieve a binary accuracy of 0.84, and an F1 score of 0.69. This performance is close to that of MLP trained with whole dataset, which is 0.76 [10]. After determining that fine-tuning LLMs perform better for detecting DDoS attacks than conventional machine learning approaches such as MLP, we feed the entire training dataset to the Ada model for fine-tuning in order to verify how powerful fine-tuning LLMs are for detecting DDoS attacks with OM-WC architecture. The result of feeding all samples in the dataset seems promising, which has a binary accuracy up to 0.98, as well as an F1 score greater than 0.95. The above results indicate that, by using the same training dataset, i.e. the data including correlation information for all nodes, fine-tuning LLMs perform better than any neural network model proposed by A. Hekmati et al. [10].



(a) Binary Accuracy (b) F1 Score

Fig. 6. Compare incrementally fine-tuned LLMs performance as the number of samples increases

6 Conclusion

In this exploration into the realm of network security and the potential applications of large language models (LLMs) for DDoS attack detection, our study sheds light on the growing complexity of threats that organizations face.

Diving into the nuances of DDoS detection, we detailed methodologies encompassing zero-shot, one-shot, and few-shot LLM approaches, along with insights into the fine-tuning techniques of LLMs. A comparative analysis was drawn

between traditional multi-layer perceptron (MLP) models and the advanced capabilities of LLMs, leveraging platforms such as OpenAI's GPT-3.5, GPT-4, and Ada models.

By employing two distinct datasets, namely CICIDS 2017 and the Urban IoT Dataset, our evaluations showed that LLMs, with the right context and training, could achieve impressive accuracies in DDoS detection. Specifically, using few-shot methods on the CICIDS 2017 dataset, LLMs approached a 90% accuracy with merely 10 prompt samples. This surged to around 95% when fine-tuned with 70 samples. The more challenging Urban IoT Dataset showcased a similar trend, where aggressive DDoS attacks saw LLMs achieving a 70% accuracy with few-shot techniques and nearly 96% upon fine-tuning. Compared to traditional MLP models trained on similar few-shot samples, LLMs consistently showcased superior performance.

One of the most notable contributions of our study was the capability of LLMs to articulate the basis behind their DDoS detections, especially in few-shot learning scenarios. However, it is essential to note their tendency for hallucination in the case of fine-tuning, indicating that while LLMs promise significant advances, careful application and ongoing scrutiny are paramount.

Acknowledgments. This material is based upon work partially supported by Defense Advanced Research Projects Agency (DARPA) under Contract Number HR001120C0160 for the Open, Programmable, Secure 5G (OPS-5G) program. Any views, opinions, and/or findings expressed are those of the author(s) and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government. This document has been edited with the assistance of ChatGPT. We certify that ChatGPT was not utilized to produce any technical content and we accept full responsibility for the contents of the paper.

References

1. Abdullahi, M., et al.: Detecting cybersecurity attacks in internet of things using artificial intelligence methods: a systematic literature review. Electronics **11**(2), 198 (2022)
2. ANALYTICS, I.: State of IoT 2023: Number of connected IoT devices growing 16
3. Antonakakis, M., et al.: Understanding the Mirai botnet. In: 26th USENIX security symposium (USENIX Security 17), pp. 1093–1110 (2017)
4. Biswas, S.S.: Potential use of chat GPT in global warming. Ann. Biomed. Eng. **51**(6), 1126–1127 (2023)
5. Biswas, S.S.: Role of chat GPT in public health. Ann. Biomed. Eng. **51**(5), 868–869 (2023)
6. Brown, T.B., et al.: Language models are few-shot learners (2020)
7. Ferrag, M.A., Ndhlovu, M., Tihanyi, N., Cordeiro, L.C., Debbah, M., Lestable, T.: Revolutionizing cyber threat detection with large language models. arXiv preprint arXiv:2306.14263 (2023)
8. Hekmati, A., Grippo, E., Krishnamachari, B.: Dataset: Large-scale urban IoT activity data for DDOS attack emulation. arXiv preprint arXiv:2110.01842 (2021)

9. Hekmati, A., Grippo, E., Krishnamachari, B.: Neural networks for DDOS attack detection using an enhanced urban IoT dataset. In: 2022 International Conference on Computer Communications and Networks (ICCCN), pp. 1–8. IEEE (2022)
10. Hekmati, A., Jethwa, N., Grippo, E., Krishnamachari, B.: Correlation-aware neural networks for DDOS attack detection in IoT systems. arXiv preprint arXiv:2302.07982 (2023)
11. Huang, J., Chang, K.C.C.: Towards reasoning in large language models: a survey. arXiv preprint arXiv:2212.10403 (2022)
12. Johnson, A.: Leveraging large language models for network security, https://medium.com/@andrew_johnson_4/leveraging-large-language-models-for-network-security-b2027f03d522. Accessed 08 July 2023
13. Kurniabudi, Stiawan, D., Darmawijoyo, Bin Idris, M.Y., Bamhdi, A.M., Budiarto, R.: Cicids-2017 dataset feature analysis with information gain for anomaly detection. IEEE Access **8**, 132911–132921 (2020). https://doi.org/10.1109/ACCESS.2020.3009843
14. Liu, N.F., et al: Lost in the middle: How language models use long contexts (2023)
15. Liu, Y., et al.: Summary of ChatGPT/GPT-4 research and perspective towards the future of large language models. arXiv preprint arXiv:2304.01852 (2023)
16. Marin, G.: Network security basics. IEEE Secur. Priv. **3**(6), 68–72 (2005). https://doi.org/10.1109/MSP.2005.153
17. Mubarakali, A., Srinivasan, K., Mukhalid, R., Jaganathan, S.C.B., Marina, N.: Security challenges in internet of things: Distributed denial of service attack detection using support vector machine-based expert systems. Comput. Intell. **36**(4), 1580–1592 (2020)
18. Pal, S.K., Mitra, S.: Multilayer perceptron, fuzzy sets, classifiaction (1992)
19. Pal, S., Mitra, S.: Multilayer perceptron, fuzzy sets, and classification. IEEE Trans. Neural Netw. **3**(5), 683–697 (1992). https://doi.org/10.1109/72.159058
20. Sharafaldin, I., Lashkari, A.H., Ghorbani, A.A.: Toward generating a new intrusion detection dataset and intrusion traffic characterization. ICISSp **1**, 108–116 (2018)
21. Shaukat, K., Luo, S., Varadharajan, V., Hameed, I.A., Xu, M.: A survey on machine learning techniques for cyber security in the last decade. IEEE Access **8**, 222310–222354 (2020). https://doi.org/10.1109/ACCESS.2020.3041951
22. Sinanović, H., Mrdovic, S.: Analysis of Mirai malicious software. In: 2017 25th International Conference on Software, Telecommunications and Computer Networks (SoftCOM), pp. 1–5 (2017). https://doi.org/10.23919/SOFTCOM.2017.8115504
23. Surameery, N.M.S., Shakor, M.Y.: Use chat gpt to solve programming bugs. International Journal of Information Technology & Computer Engineering (IJITC) ISSN: 2455–5290 3(01), 17–22 (2023)
24. Suresh, M., Anitha, R.: Evaluating machine learning algorithms for detecting DDoS attacks. In: Wyld, D.C., Wozniak, M., Chaki, N., Meghanathan, N., Nagamalai, D. (eds.) CNSA 2011. CCIS, vol. 196, pp. 441–452. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22540-6_42
25. Tariq, U., Ahmed, I., Ali, K.B., Shaukat, K.: A critical cybersecurity analysis and future research directions for the internet of things: a comprehensive review. Sensors **23**(8), 4117 (2023)
26. Vishwakarma, R., Jain, A.K.: A survey of DDOS attacking techniques and defence mechanisms in the IoT network. Telecommun. Syst. **73**(1), 3–25 (2020)

27. Yu, D., et al.: Differentially private fine-tuning of language models. arXiv preprint arXiv:2110.06500 (2021)
28. Zhao, W.X., et al.: A survey of large language models. arXiv preprint arXiv:2303.18223 (2023)