Enhancing Detection of Anomalies on IIoT Networks with Fine-tuned Large Language Models

Huang Po-Hsun

October 28, 2024

Abstract

This research plan aims to enhance anomaly detection in Industrial Internet of Things (IIoT) networks by fine-tuning Large Language Models (LLMs). Addressing the limitations of current IoT systems in handling complex tasks and human interactions, as well as the shortcomings of the LLM-based task-oriented AI agent framework designed by Cui et al. (2024), which did not adequately consider network anomalies in IoT devices, this study proposes a new approach. Our goal is to leverage the time series learning capabilities of deep models to more accurately detect anomalous patterns, reduce false alarm rates, and construct an adaptable anomaly detection model suitable for various IoT scenarios and devices. By comparing with traditional detection methods, we expect the fine-tuned LLM to significantly improve the accuracy and efficiency of anomaly detection in IIoT networks, solving the problem of inadequate response to network anomalies in existing systems.

Keywords: IoT Device Control, Large Language Models, Intelligent Agents, IoT Networks

1 Background

Blockchain technology and artificial intelligence (AI) have emerged as two of the most transformative technologies of our time [1]. On the other hand, Large Language Models (LLMs) can understand and generate human-like text, based on extensive training on diverse datasets with billions of parameters [2]. Task-oriented communications are an important element in future intelligent Internet of Things (IoT) systems. Existing IoT systems, however, are limited in their capacity to handle complex tasks, particularly in their interactions with humans to accomplish these tasks [3]. The industry need to be good at making use of the LLM, which has been popular in recent years, to improve the shortcomings of the IoT in device networks.

2 Objectives and Motivation

The continuous flow of data collected by IoT devices, has revolutionised our ability to understand and interact with the world across various applications[4]. In Cui(2024), they Design an LLM-based taskoriented AI agent framework that enables effective collaboration among IoT devices in Figure 1[3]. However, although their system designed the conversion between natural language and code and executed it on IoT devices, they did not further consider the problem of network anomalies in IoT devices. This system is highly dependent on transmission between networks. If a network failure occurs, there is no response to troubleshoot the problem.

And LLMs have provided highly effective methodologies and solutions in various cybersecurity sectors [5]. As the frequency and diversity of cybersecurity attacks continue to rise, the importance of incident detection has significantly increased [6]. Our goal is to more accurately detect abnormal patterns through the time series learning capabilities of deep models. Use large models to understand multiple patterns to distinguish normal fluctuations from abnormal situations and reduce false alarm rates. Finally, an adaptable anomaly detection model is built that is suitable for a variety of IoT scenarios and devices.

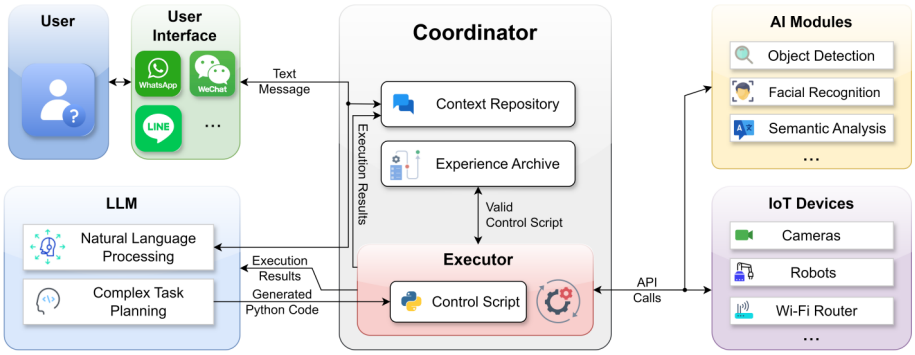

Figure 1: System diagram

3 Questions or Hypotheses

Below are posed to optimize network detection based on their designed IoT task-oriented framework, which may help identify potential challenges, define research directions, and evaluate the actual performance of the model:

3.1 Questions

1. **Model Adaptability:** Can a large model, once fine-tuned, adapt to different IoT devices? In which device environments does it perform best?

2. **Data Requirements:** Does fine-tuning an IoT anomaly detection model require a large amount of data? How do data scale and quality impact detection effectiveness?

3. **Knowledge Base & RAG Vector Database:** In Figure 2, if the RAG vector library and knowledge base are added, can the command problems of the system be better located? However, establishing this node requires additional resources.

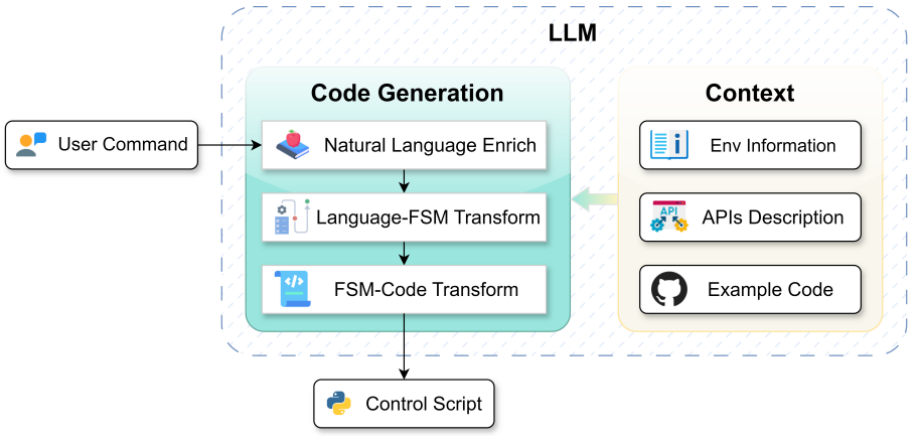

Figure 2: Language-Code transformation procedure

4. **Real-time Detection:** Can the fine-tuned model meet real-time detection requirements in practical applications? If there is latency, how can it be optimized?

5. **False Positive and False Negative Rates:** What are the model's false positive and false negative rates? How does detection accuracy vary across different types of anomalies, and are there specific anomaly types that are challenging to detect?

6. **Resource Consumption:** What are the computational, storage, and energy costs when deploying the fine-tuned model in IoT environments? Can it effectively operate on resource-limited devices?

7. **Model Scalability:** Can the model flexibly adapt to emerging IoT anomaly patterns, or does it require periodic fine-tuning or retraining?

8. **Data Privacy:** How can IoT data privacy be effectively protected during data collection and model deployment?

3.2 Hypotheses

1. **H1:** Fine-tuning a large model significantly improves the accuracy and recall rates of IoT network anomaly detection, making it better suited for complex anomaly patterns than traditional detection methods.

2. **H2:** A moderate amount of IoT anomaly data can greatly improve the model's generalization ability, enabling it to maintain high detection accuracy across various IoT environments.

3. **H3:** Because in the industrial Internet of Things, the environment that isolates the external network needs to be considered, it is necessary to establish its own LLM independently. Compared with the resources consumed by running LLM, a vector library can be added to speed up the troubleshooting process.

4. **H4:** The false positive rate of a fine-tuned large model will be substantially lower than traditional statistical or rule-based methods, thereby reducing interference in anomaly detection.

5. **H5:** Distributed fine-tuning can effectively reduce the computational burden on IoT devices without significantly impacting detection performance.

6. **H6:** The model's accuracy in detecting new types of anomalies will degrade over time, but incremental fine-tuning can restore performance.

7. **H7:** Multi-task fine-tuning (e.g., anomaly detection and classification in parallel) can further improve the model's ability to identify specific types of anomalies.

8. **H8:** The fine-tuned model is suitable for real-time anomaly detection, with a response time of 1-2 seconds for most IoT network anomalies.

4 Methodology

The research will follow a systematic approach to develop and evaluate the proposed anomaly detection framework. The methodology includes the following steps:

4.1 Technical Selection

1. **Base Model:** Choose large language models or sequence modeling models that support network anomaly detection, such as OpenAI's GPT series or Hugging Face's BLOOM.

    - **Specific Model Selection:** Models suitable for handling time series, such as T5, GPT, or transformer-based time series models, can be selected and fine-tuned for IoT data.

2. **Fine-Tuning Methods:**

    - **Supervised Learning Fine-Tuning:** As shown in Table 1, Fine-tune the model using labeled normal/abnormal IoT network data to enable the model to recognize common anomaly patterns.
    - **Reinforcement Learning from Human Feedback(RLHF) Fine-Tuning:** Further fine-tune the model using expert feedback after initial fine-tuning to optimize sensitivity to anomaly detection. The reward model for RLHF has proven effective in fine-tuning Large Language Models (LLMs)[7].
    - **LoRA or Adapter:** SLoRA(2024) observe that this paradigm presents significant opportunities for batched inference during serving[8]. We use LoRA for lightweight fine-tuning to save resources.
    - **Mixture of Experts(Moe):** MoE is a model architecture that combines different experts (i.e., sub-models or networks) with the aim of activating only a subset of these "experts" based on the specific needs of the input task. LLM-HAS innovative framework leverages a Mixture of Experts (MoE) approach, augmented with LLMs, to analyze users' personalized preferences and potential health risks from additional textual job descriptions[9].

Feature	LoRA	RLHF	MoE
Method Type	Parameter-efficient fine-tuning	Training optimization	Model architecture
Core Goal	Efficiently fine-tune the model	Enhance the quality of generated content through human feedback	Improve model performance through selective activation
Suitable Scenarios	High task adaptability, minimal parameter adjustments needed	User interaction, dialogue systems	Multi-task learning, handling complex inputs
Advantages	Resource-saving, lightweight fine-tuning	Aligns with human preferences, high-quality output	Large parameter count but controlled computational load
Implementation	Low-rank matrix insertion	Reward model + reinforcement learning	Mixture of experts network + sparse activation
Cost	Relatively low	Relatively high	High (but less computation during inference)
Complexity	Low	High (involves human feedback and reinforcement learning)	High (requires expert selection and sparse activation)

Table 1: Comparison of Different Methods

3. **Supporting Tools:** The Hugging Face Transformers library and Datasets API can be used to handle datasets and model fine-tuning tasks; DeepSpeed can also be employed for distributed training and deployment.

4.2 Data Preparation

1. **Collecting IoT Data:** Gather traffic, logs, and sensor data from IoT devices over a period. Focus on collecting and labeling anomalies and normal events occurring in actual network environments.

2. **Feature Extraction and Data Cleaning:** Extract features (e.g., traffic peaks, connection frequencies, packet sizes) from the data and remove noise. Data augmentation techniques can be used to expand anomaly samples.

3. **Training Data Construction:** Organize data into time series input formats, such as sliding window sequences, for the model to learn inter-event relationships.

4.3 Model Fine-Tuning

1. **Phased Fine-Tuning:** We will select an appropriate LLM and fine-tune it on the collected IIoT data. As mentioned in the work of Touvron et al. (2023)[10].First fine-tune the model using general network data to learn basic network patterns, then use IoT-specific data for further fine-tuning to learn specific patterns.

2. **Multi-Task Training:** Implement multi-task learning to enable the model to perform anomaly detection while also classifying types of anomalies (e.g., identifying DDoS attacks, packet loss).

4.4 Evaluation and Validation

1. **Test Set Evaluation:** The fine-tuned LLM will be trained on the IIoT dataset, and its performance will be validated using a separate validation set to ensure the model's effectiveness in detecting anomalies.

2. **Experimental Control:** Compare with traditional detection models (e.g., rule-based or statistical analysis methods) to analyze the advantages and shortcomings of the fine-tuned large model.

3. **Comparison with Traditional Methods:** The performance of the fine-tuned LLM will be compared with traditional anomaly detection methods to evaluate its improvement in accuracy and efficiency.

4. **Real-world Implementation and Testing:** The model will be deployed in a real-world IIoT network setting to test its practical applicability and robustness in various operational scenarios.

5 Expected Outcomes and Contributions

We expect the fine-tuned LLM to significantly enhance the accuracy and efficiency of anomaly detection in IIoT networks. The successful implementation of this research will contribute to the following:

1. **A Novel Anomaly Detection Framework:** A new framework that leverages the power of LLMs for IIoT security, providing a more proactive approach to threat detection.

2. **Improved Security and Resilience:** Enhanced security measures for IIoT networks, leading to increased resilience against potential threats and anomalies.

3. **Advancing LLM Applications:** Expanding the application domain of LLMs in industrial settings, demonstrating their versatility beyond natural language tasks.

4. **Empirical Evidence:** Providing empirical evidence on the effectiveness of fine-tuned LLMs in improving IIoT network security, contributing to the body of knowledge in this emerging field.

5. **Improved Detection Accuracy:** The model should achieve high accuracy and recall rates post fine-tuning, with expected accuracy improvements of 5-10%.

6. **Enhanced Real-Time Responsiveness:** Achieve real-time response in anomaly detection through distributed deployment.

7. **Improved Anomaly Pattern Recognition Ability:** The model will adapt more quickly to new or complex anomalies, expanding its ability to detect different types of attacks or anomalies.

8. **Scalability:** Maintain the model's adaptability to emerging IoT anomalies through periodic fine-tuning or the use of new data.

Example Application Scenario: In a home monitoring network, the fine-tuned model can identify anomalies such as sudden bandwidth spikes or frequent packet losses, alerting users to potential network intrusions or device anomalies in a timely manner. This helps in proactively preventing and reducing IoT security risks.

References

[1] Xuhan Zuo, Minghao Wang, Tianqing Zhu, Lefeng Zhang, Dayong Ye, Shui Yu, and Wanlei Zhou. Federated trustchain: Blockchain-enhanced llm training and unlearning. *arXiv preprint arXiv:2406.04076*, 2024.

[2] Alakesh Kalita. Large language models (llms) for semantic communication in edge-based iot networks. *arXiv preprint arXiv:2407.20970*, 2024.

[3] Hongwei Cui, Yuyang Du, Qun Yang, Yulin Shao, and Soung Chang Liew. Llmind: Orchestrating ai and iot with llm for complex task execution. *IEEE Communications Magazine*, 2024.

[4] Mohsen Shirali, Mohammadreza Fani Sani, Zahra Ahmadi, and Estefania Serral. Llm-based event abstraction and integration for iot-sourced logs. *arXiv preprint arXiv:2409.03478*, 2024.

[5] Mona Mohamed, Alaa Elmor, Florentin Smarandache, and Ahmed A Metwaly. An efficient superhypersoft framework for evaluating llms-based secure blockchain platforms. *Neutrosophic Sets and Systems*, 72:1–21, 2024.

[6] NARINDERJIT SINGH THANDI. Revolutionizing cyber threat detection with large language models: A privacy-preserving bert-based lightweight model for iot/iiot devices. 2024.

[7] Jinghan Zhang, Xiting Wang, Yiqiao Jin, Changyu Chen, Xinhao Zhang, and Kunpeng Liu. Prototypical reward network for data-efficient rlhf. *arXiv preprint arXiv:2406.06606*, 2024.

[8] Ying Sheng, Shiyi Cao, Dacheng Li, Coleman Hooper, Nicholas Lee, Shuo Yang, Christopher Chou, Banghua Zhu, Lianmin Zheng, Kurt Keutzer, et al. Slora: Scalable serving of thousands of lora adapters. *Proceedings of Machine Learning and Systems*, 6:296–311, 2024.

[9] Yulan Gao, Ziqiang Ye, Ming Xiao, Yue Xiao, and Dong In Kim. Guiding iot-based healthcare alert systems with large language models. *arXiv preprint arXiv:2408.13071*, 2024.

[10] Armin Sarabi, Tongxin Yin, and Mingyan Liu. An llm-based framework for fingerprinting internet-connected devices. In *Proceedings of the 2023 ACM on Internet Measurement Conference*, pages 478–484, 2023.