ENHANCING DETECTION OF ANOMALIES ON IIOT NETWORKS WITH FINE-TUNED LARGE LANGUAGE MODELS

Huang Po-Hsun⋆

October 29, 2024

Abstract
This research plan proposes to enhance the detection of anomalies in Industrial Internet of Things (IIoT) networks by fine-tuning Large Language Models (LLMs). As the complexity and interconnectivity of IIoT networks increase, anomaly detection has become a critical challenge, especially in terms of potential security threats and operational failures. The plan explores the use of machine learning and deep learning algorithms to analyze the vast amounts of data generated by IoT devices, enabling real-time anomaly detection. The focus of the research is on combining traditional statistical methods with modern machine learning approaches to enhance detection capabilities and reduce false positives. Additionally, the study investigates how fine-tuning LLMs can improve their performance in specific tasks or domains, including the use of Reinforcement Learning with Human Feedback (RLHF) methods and Low-Rank Adaptation (LoRA) techniques to optimize model outputs and reduce computational overhead. The goal of this research is to build an adaptable anomaly detection model suitable for a variety of IIoT scenarios and devices, thereby enhancing the resilience and security of interconnected industrial systems. By leveraging the time series learning capabilities of large models to understand multiple patterns, distinguishing between normal fluctuations and abnormal situations, and reducing false alarm rates, we aim to achieve a 5-10% increase in accuracy through fine-tuning LLMs and enable the model to adapt more quickly to new or complex anomalies.

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

1 Introduction

1.1 Rearch Background

Blockchain technology and AI have emerged as two of the most transformative technologies of our time (Zuo et al. 2024). On the other hand, LLMs can understand and generate human-like text, based on extensive training on diverse datasets with billions of parameters (Kalita 2024). Task-oriented communications are an important element in future intelligent IIoT systems. Existing IoT systems, however, are limited in their capacity to handle complex tasks, particularly in their interactions with humans to accomplish these tasks (Cui et al. 2024). 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.

1.2 IIoT Networks With LLM

As IIoT networks become more complex and interconnected, the detection of anomalies has emerged as a critical challenge, with researchers exploring various techniques to identify unusual patterns in network traffic that may indicate potential security threats or operational failures(Reuer et al. 2022). These techniques often leverage machine learning and deep learning algorithms to analyze the vast amounts of data generated by IoT devices, enabling real-time detection of anomalies(Alharbi 2022).

The detection of anomalies in IIoT networks is particularly significant due to the potential consequences of undetected issues, such as production downtime, safety hazards, or data breaches. Recent studies have highlighted the effectiveness of combining traditional statistical methods with modern machine learning approaches to enhance detection capabilities (Hao et al. 2019). For instance, techniques like unsupervised learning and reinforcement learning have been applied to improve the accuracy of anomaly detection systems while minimizing false positives. Additionally, researchers have emphasized the importance of feature extraction and selection, as well as the integration of domain knowledge, to better identify relevant patterns in the data.

Overall, the landscape of anomaly detection in IIoT networks is evolving, with ongoing research aimed at developing more robust and adaptive systems. By addressing the unique challenges posed by IIoT environments, these efforts aim to enhance the resilience and security of interconnected industrial systems.

1.3 LLMs

LLMs have made significant strides in the field of natural language processing in recent years. These models are pretrained on vast amounts of text data, enabling them to generate fluent text and understand complex language patterns. As interest in LLMs has grown, researchers have begun exploring how to effectively fine-tune these models to better adapt to specific tasks or domains. The fine-tuning process typically involves further training the model on specialized datasets, allowing it to perform better in specific applications. For example, using supervised learning methods can help the model learn more precise task-related features(Brown et al. 2020).

1.4 Fine-tuning LLMs

Recent studies have shown that fine-tuning large models can not only enhance their performance on specific tasks but also improve

⋆ fh831.cp9gw@gmail.com

their understanding and generative capabilities in context(Hu et al. 2021a). Some researchers have proposed combining reinforcement learning with human feedback (RLHF) methods to enhance the quality of model outputs, especially in generative tasks(Stiennon et al. 2022). Additionally, emerging techniques like Low-Rank Adaptation (LoRA) have been introduced to reduce the computational overhead and resource requirements during the fine-tuning process. These research efforts are gradually enriching the application potential of LLMs, showing positive results across various domains, from chatbots to text summarization.

LoRA introduces low-rank matrices for fine-tuning the model, thereby reducing the number of parameters that need to be trained. The core idea is to decompose the weight matrix W into two low-rank matrices A and B, updating only these two matrices(Hu et al. 2021b). Its equation is 1.

$$W' = W + A \cdot B \tag{1}$$

Below is the algorithm for LoRA.

```
# Assume model is the pretrained model
for layer in model.layers:
    A = initialize_random_matrix(rank)
    B = initialize_random_matrix(rank)

    # Modify the forward propagation function
    def forward(input):
        W = layer.weight
        return W @ input + (A @ B) @ input
```

RLHF optimizes model outputs through human feedback. The training process typically consists of two steps: first, training the model using supervised learning, and then fine-tuning the model using reinforcement learning based on human feedback(Stiennon et al. 2020). Formula2:

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left(r + \gamma \max_{a'} Q(s',a') - Q(s,a) \right) \tag{2}$$

Below is the algorithm for RLHF.

```
# Assume model is the pretrained model
for episode in range(num_episodes):
    state = environment.reset()
    done = False

    while not done:
        action = model.predict(state)
        next_state, reward, done =
            environment.step(action)

        # Update Q values
        Q[state, action] +=
          alpha * (reward + gamma * max(Q[
              next_state]) - Q[state, action
              ])

        state = next_state
```

MoE MoE combines multiple expert models, dynamically selecting which experts to activate to improve model efficiency. By activating only a subset of experts during inference, it conserves computational resources(Shazeer et al. 2017). Formula3: The expression y is equal to the sum of the products of g_i and $E_i(x)$ for all i belonging to the set S.

$$y = \sum_{i \in S} g_i \cdot E_i(x) \tag{3}$$

Where S is the set of activated experts and g_i represents the gating output. Below is the algorithm for MoE.

```
# Assume experts is the collection of expert
    models
for input in inputs:
    gates = gate_model.predict(input)  #
        Calculate gate values
    output = 0

    for i, expert in enumerate(experts):
        if gates[i] > threshold:  # Select
            active experts
            output += gates[i] * expert(input
                )
```

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(Shirali et al. 2024). In Cui et al. (2024), they Design an LLM-based taskoriented AI agent framework that enables effective collaboration among IoT devices in Figure 1(Cui et al. 2024). However, although their system designed the conversion between natural language and code and executed it on IIoT devices, they did not further consider the problem of network anomalies in IIoT devices. This system is highly dependent on transmission between networks. If a network failure occurs, there is no response to troubleshoot the problem.

Additionally, LLMs have provided highly effective methodologies and solutions in various cybersecurity sectors (Mohamed et al. 2024). As the frequency and diversity of cybersecurity attacks continue to rise, the importance of incident detection has significantly increased (THANDI 2024). 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 IIoT scenarios and devices.

3 Questions or Hypotheses

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

3.1 Fine-tuning Model

Firstly, Can a large model, once fine-tuned, adapt to different IIoT devices? In which device environments does it perform best? I think that fine-tuning a large model significantly improves the accuracy

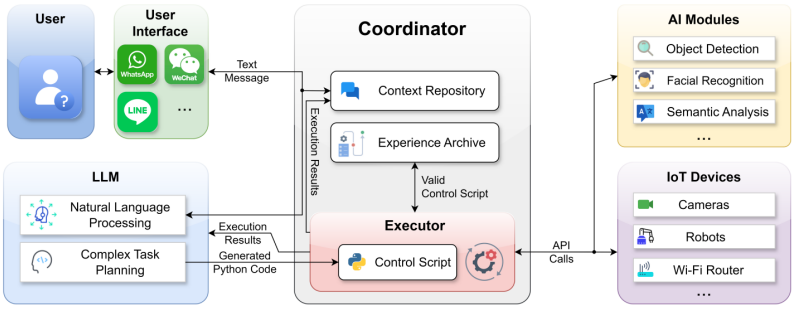

Figure 1. System diagram

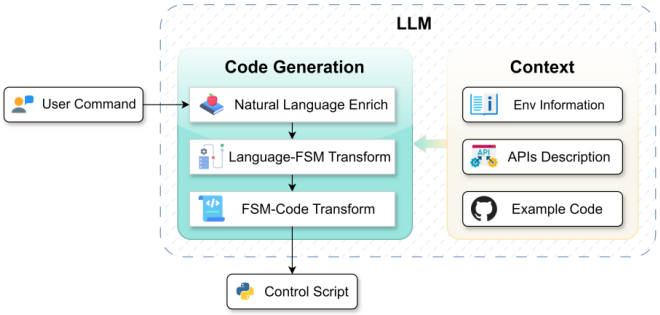

Figure 2. Language-Code transformation procedure

and recall rates of IIoT network anomaly detection, making it better suited for complex anomaly patterns than traditional detection methods.

On the other hand, the model flexibly adapt to emerging IIoT anomaly patterns, or does it require periodic fine-tuning or retraining? We can hypothesize that 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.

3.2 Data Processing

Some questions that arise when we collect IIoT data is: Does fine-tuning an IIoT anomaly detection model require a large amount of data? How do data scale and quality impact detection effectiveness? A moderate amount of IIoT anomaly data can greatly improve the model's generalization ability, enabling it to maintain high detection accuracy across various IIoT environments.

In Figure 2, if the RAG vector library and knowledge base are added, can the command problems of the system be better located? establishing this node requires additional resources. 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.

However, Data privacy is a serious and unavoidable issue. How can IIoT data privacy be effectively protected during data collection and model deployment? The fine-tuned model is suitable for real-time anomaly detection, with a response time of 1-2 seconds for most IIoT network anomalies.

3.3 Performance and Cost

About Performance, can the fine-tuned model meet real-time detection requirements in practical applications? If there is latency, how can it be optimized? I assume that 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.

Then, we need to seriously think about the cost issue. What are the computational, storage, and energy costs when deploying the fine-tuned model in IIoT environments? Can it effectively operate on resource-limited devices? Distributed fine-tuning can effectively reduce the computational burden on IIoT devices without significantly impacting detection performance.

What are the computational, storage, and energy costs when deploying the fine-tuned model in IIoT environments? Can it effectively operate on resource-limited devices? The model's accuracy in detecting new types of anomalies will degrade over time, but incremental fine-tuning can restore performance.

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

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. 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.

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)(Zhang et al. 2024). SLoRA(2024) observe that this paradigm presents significant opportunities for batched inference during serving(Sheng et al. 2024). So, we use LoRA for lightweight fine-tuning to save resources.

MoE 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(Gao et al. 2024).

Furthermore, the Hugging Face Transformers library and Datasets API can be utilized for managing datasets and model fine-

Table 1. Comparison of Different Methods

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)

tuning tasks, while DeepSpeed can be employed for distributed training and deployment. It serves as a versatile framework that offers flexible fine-tuning tools. However, techniques such as LoRA, RLHF, and MoE are more targeted approaches designed for specific fine-tuning needs and scenarios. Therefore, I don't choose the Transformers library as the tool for fine-tuning large models in this context.

4.2 Data Preparation

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.

Then, 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.

Finally, organize data into time series input formats, such as sliding window sequences, for the model to learn inter-event relationships.

4.3 Fine-Tuning Model

We will select an appropriate LLM and fine-tune it on the collected IIoT data. As mentioned in the work of Sarabi et al. (2023).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.

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

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. 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.

The performance of the fine-tuned LLM will be compared with traditional anomaly detection methods to evaluate its improvement in accuracy and efficiency. 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 anticipate establishing a new IIoT framework that enhances resilience against potential threats and anomalies in existing IIoT network security. This framework aims to improve the effectiveness of industrial IIoT security through validated evidence, contributing to the knowledge base of this emerging field.

With fine-tuned LLMs demonstrating their versatility beyond natural language tasks in industrial contexts, we expect the model to achieve higher accuracy and recall rates after fine-tuning, with an anticipated increase of 5-10% in accuracy. The model will adapt more swiftly to new or complex anomalies, thereby expanding its capability to detect various types of attacks or anomalies.

Finally, we will maintain the model's adaptability to emerging IIoT anomalies through regular fine-tuning or the incorporation of new data.

6 Conclusion

In this research plan, I analyze the current state of research on the integration of IIoT and LLMs in the fields of monitoring and control. Recent studies on IIoT device control have utilized OpenAI to generate control code for these devices.

I propose a framework for fine-tuning large models to enhance the detection of network anomalies in IoT devices that integrate LLM control systems. Frameworks such as LoRA, RLHF, and MoE, which are tailored for specific scenarios, can effectively address troubleshooting in network anomaly situations, thereby reducing time costs and improving efficiency.

Additionally, it is essential to consider data collection and significant cost factors. Ultimately, the goal is to achieve a 5-10% increase in efficiency, improving the control framework for industrial IoT research.

References

Alharbi Y., 2022, International Journal of Pervasive Computing and Communications

Brown T. B., et al., 2020, Language Models are Few-Shot Learners (arXiv:2005.14165), https://arxiv.org/abs/2005.14165

Cui H., Du Y., Yang Q., Shao Y., Liew S. C., 2024, IEEE Communications Magazine

Gao Y., Ye Z., Xiao M., Xiao Y., Kim D. I., 2024, arXiv preprint arXiv:2408.13071

Hao C., Hengyu L., Xuchen L., Defu W., Tie G., Wei W., 2019, in 2019 4th International Conference on Power and Renewable Energy (ICPRE). pp 65–69, doi:10.1109/ICPRE48497.2019.9034736

Hu E. J., Shen Y., Wallis P., Allen-Zhu Z., Li Y., Wang S., Wang L., Chen

W., 2021a, LoRA: Low-Rank Adaptation of Large Language Models (arXiv:2106.09685), https://arxiv.org/abs/2106.09685
Hu E. J., Shen Y., Wallis P., Allen-Zhu Z., Li Y., Wang S., Chen W., 2021b, CoRR, abs/2106.09685
Kalita A., 2024, arXiv preprint arXiv:2407.20970
Mohamed M., Elmor A., Smarandache F., Metwaly A. A., 2024, Neutrosophic Sets and Systems, 72, 1
Reuer K., Besse J.-C., Wernli L., Magnard P., Kurpiers P., Norris G. J., Wallraff A., Eichler C., 2022, Physical Review X, 12
Sarabi A., Yin T., Liu M., 2023, in Proceedings of the 2023 ACM on Internet Measurement Conference. pp 478–484
Shazeer N., Mirhoseini A., Maziarz K., Davis A., Le Q. V., Hinton G. E., Dean J., 2017, CoRR, abs/1701.06538
Sheng Y., et al., 2024, Proceedings of Machine Learning and Systems, 6, 296
Shirali M., Sani M. F., Ahmadi Z., Serral E., 2024, arXiv preprint arXiv:2409.03478
Stiennon N., et al., 2020, CoRR, abs/2009.01325
Stiennon N., et al., 2022, Learning to summarize from human feedback (arXiv:2009.01325), https://arxiv.org/abs/2009.01325
THANDI N. S., 2024, IEEE access
Zhang J., Wang X., Jin Y., Chen C., Zhang X., Liu K., 2024, arXiv preprint arXiv:2406.06606
Zuo X., Wang M., Zhu T., Zhang L., Ye D., Yu S., Zhou W., 2024, arXiv preprint arXiv:2406.04076