U.P.B. Sci. Bull., Series C, Vol. 88, Iss. 1, 2026 ISSN 2286 – 3540

RESEARCH AND APPLICATION OF LARGE LANGUAGE MODEL AGENT FOR OPERATION AND MAINTENANCE OF CONVERGED COMMUNICATION SYSTEM

Lidong WANG[1], Lijie WANG[1, *], Lin LONG[1], Hao FENG[1], Qi RAO[1], Luyao ZHOU[1], Yun MA[1], Zhaohui HUANG[2]

With the rapid development of information technology, the importance of converged communication systems in various fields is increasingly prominent. However, the complexity and diversity of converged communication systems bring huge challenges to operation and maintenance. This study proposes an intelligent operation and maintenance method for converged communication systems driven by Large Language Model (LLM). By utilizing the powerful language understanding and generation capabilities of LLM and combining the characteristics of converged communication systems, efficient and intelligent operation and maintenance of the system can be achieved. This method first applies to the machine learning isolation forest algorithm to perform fault diagnosis on the characteristic data of the converged communication system. Then, LLM is applied to conduct in-depth analysis of the fault diagnosis results and generate detailed fault diagnosis results and fault handling schemes according to the content of the knowledge base. In the fault handling stage, an LLM Agent is designed to automatically formulate the optimal repair strategy, which greatly improves the fault repair efficiency. Experimental results show that this method shows significant advantages in the operation and maintenance of converged communication systems. The accuracy rate of fault analysis reached 96.6% and the repair rate of manual fault handling assisted reached 93.3%. The proposed method can effectively reduce operation and maintenance costs, improve the reliability and stability of the system and provide strong technical support for the wide application of converged communication systems.

Keywords: Converged communication; Fault handling; LLM Agent; Isolation Forest

1. Introduction

Converged communication systems integrate multiple communication modalities and technologies to provide efficient, convenient and unified communication experiences [1]. Indigenous converged communication systems are constructed using ally developed, self-controllable hardware and software. Their complex architecture involves diverse autonomous components, primarily

[1] China Yangtze Power Co., Ltd., Wuhan, China
[2] Henan Communication Engineering Co., Ltd., Zhengzhou, China
* Corresponding author's e-mail: wang_lijie@ctg.com.cn

composed of indigenous infrastructure layers, software system layers and application layers [2]. Current converged communication systems rely on manual operations for maintenance, where personnel use accumulated experience and pre-defined rules to identify faults. For example, fault detection involves observing server indicator lights, specific error codes in system logs, or performance metrics exceeding thresholds in network devices. Due to the limitations of manual inspection, only pre-defined and rule-based checks can be performed, potentially missing complex or deeply hidden faults. When faults occur, maintenance personnel spend significant time troubleshooting component-by-component and reviewing configuration files/logs, resulting in low efficiency and high labor costs [3-4].

With advancements in artificial intelligence (AI), researchers have increasingly explored AI applications to address maintenance challenges in converged communication systems [5]. AI can analyze vast amounts of system data to uncover complex relationships between data points, identify hidden issues and rapidly diagnose fault causes using strong reasoning capabilities. This paper introduces an intelligent agent technology for indigenous converged communication systems, leveraging LLM and machine learning to enable rapid fault detection, localization and resolution, ensuring stable operations of self-controllable systems. The key contributions are:

(1) Development of an LLM-based intelligent agent for converged communication maintenance, improving operational efficiency and accuracy.

(2) Creation of a knowledge retrieval enhancement method for converged communication domains, combined with fine-tuning of text vectorization models to improve retrieval precision.

(3) Proposal of a hybrid model approach integrating traditional machine learning with LLMs for fault diagnosis, analysis and resolution.

2. Literature Review

The evolution of converged communication dates to the early 1990s, driven by rapid technological advancements and increasing demand for informatization [6]. Telecom operators pioneered converged systems: AT&T launched the first-generation AT&T OneNet for fixed-mobile-internet convergence [7]. BT introduced the seamless-switching BT Fusion [8] and NTT DoCoMo developed FOMA for multimedia transmission across mobile, internet and TV [9]. In China, indigenous converged systems have gained traction in recent years, enhancing security, reliability and national information security while promoting IT industrial development [10]. However, their complex architectures require manual monitoring, maintenance, optimization and fault handling, which depend on personnel experience and lead to delays and inefficiencies [11].

AI and machine learning (ML) have been increasingly applied to address these challenges. Qi et al. [12] used support vector machines (SVM) for network traffic prediction, achieving high accuracy for intelligent traffic scheduling. While effective for congestion reduction, this approach does not address fault diagnosis. Wang et al. [13] applied decision trees for fault classification, enabling automated detection but struggling with complex networks [14]. Neural network-based methods improved adaptability [15] yet lacked generalization. Recent advancements in large models, including retrieval-augmented LLMs and intelligent agents, show promise for fault diagnosis [16].

LLM have demonstrated breakthroughs in natural language processing, computer vision and speech recognition [17]. In communication, they have been applied to fault prediction [18], log parsing [19], automated maintenance [20] and visual monitoring [21]. However, generic LLMs trained on general domain data struggle in specialized scenarios like converged communication network maintenance [22-27]. This paper bridges this gap by integrating LLM retrieval augmentation, agent technology and ML for indigenous system maintenance.

3. Methodology for Indigenous Converged Communication Systems

3.1 Technical Framework for Intelligent Maintenance

To address maintenance challenges, this paper proposes an LLM-based intelligent agent framework as show in Fig. 1, consisting of three layers. Perception Layer: Collects real-time data from physical devices across access, bearer, core and data center networks. Control Layer: Comprises LLM agents and ML models, generating control instructions through prompt engineering and domain knowledge bases. Application Layer: Implements fault diagnosis, analysis and resolution modules for operational maintenance.

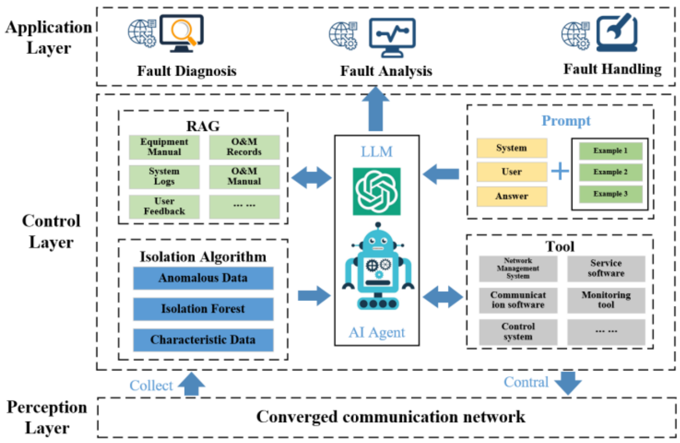

Fig. 1. Framework of the LLM agent for smart operation and maintenance of the integrated communication system

3.2 Fault Diagnosis of Converged Communication Systems Based on Isolation Forest

The perception layer of converged communication systems can collect a vast amount of time varying data of the converged communication system. When a fault occurs in the system, the fault characteristic data generated by the faulty device is included in this data. It is difficult for the traditional manual screening method to find abnormal items from the vast amount of data. This paper studies a fault diagnosis method for independently controllable converged communication systems based on the isolation forest. For the collected characteristic data, during the normal use of the system, the data is all normal data. Faults occur in rare cases and the data collected at this time will differ from the normal data. The characteristic data collected under such abnormal conditions will show an outlier phenomenon compared with the normal data. Therefore, the isolation forest method can be used to analyze the faults of converged communication systems. Compared with the data samples collected in the normal state of the system, the number of samples in the fault state is relatively small and the difference in characteristic values is relatively large. Therefore, abnormal samples are more likely to be isolated.

First, use the characteristic data collected by the converged communication system to train isolation trees and form an isolation forest. Then, bring each sample point into each isolation tree in the forest, calculate the average height and then calculate the anomaly score of each sample point. Let the matrix of all collected sample datasets be X, the vector of each sample point in the matrix be x_i and the vector x_i contain characteristic values of d dimensions. The expressions are as follows:

$$\begin{cases} X = \{x_1, x_2, x_3, \dots x_i \dots, x_n\} \\ x_i = \{x_{i1}, x_{i2}, x_{i3}, \dots, x_{id}\} \end{cases} \quad (1)$$

Randomly select m sample points from the matrix X as a subsample X' and use it as a single tree of the isolation forest and put it into its root node. Randomly select a dimension q from the d dimensions of the sample - point vector x_i. Within the range of all sample point data in the current dimension, randomly generate a split point p, where the split point p is generated between the maximum and minimum values of all sample point data in the current dimension:

$$min(x_{ij}, j = q, x_{ij} \epsilon X') < p < max(x_{ij}, j = q, x_{ij} \epsilon X') \quad (2)$$

The selection of this split point p generates a hyper-plane, which divides the data space of all sample points in the current dimension into two subspaces. Place the points less than p under the currently selected characteristic variable in the left branch of the current node and place the points greater than or equal to p in the right branch of the current node. Repeatedly split the left branch and right branch nodes of the node and continuously construct new leaf nodes until there is only one data point on the leaf node (no further splitting is possible) or the number of iterative splitting steps has reached the set number of generations.

Split the characteristic values corresponding to all other dimensions in the *d* dimensions contained in the vector x_i according to the above splitting method to complete the generation of the isolation tree for all dimensional characteristic data.

Finally, for each data point x_{ij}, traverse each isolation tree, calculate the average height $h(x_{ij})$ of the point x_{ij} in the isolation forest and simultaneously normalize the average height of all data points. The formula for calculating the anomaly score of the characteristic data collected by the converged communication system is as follows:

$$\boldsymbol{s}(x_{ij}, m) = 2^{-\frac{E(h(x_{ij}))}{c(m)}} \tag{3}$$

where $S(x_{ij}, m)$ is the anomaly index of the sample point x_{ij} in the isolation tree composed of the training data of m samples and the value range of $S(x_{ij}, m)$ is [0,1]. $h(x_{ij})$ is the height of $S(x_{ij}, m)$ in each isolation tree and c(m) is the average path length given the number of samples m, which is used to standardize the path length $h(x_{ij})$ of the sample point x_{ij}. The judgment of the abnormal situation is divided into the following cases: the closer it is to 1, the higher the possibility that the characteristic data is an abnormal point; the closer it is to 0, the higher the possibility that the characteristic data is a normal point; if the $S(x_{ij}, m)$ values of most training samples are close to 0.5, it indicates that there are no obvious abnormal values in the entire dataset. The pseudo code of the isolation forest for the converged communication system is shown in Fig. 2, which includes the training process and prediction process of the isolation forest model for fault diagnosis.

```
Isolation Forest Algorithm:
Input: Data set D, number of trees t, subsampling size s
Output: Anomaly score for a data point
1. Initialize an empty list of trees T = []
2. For i = 1 to t:
   a. Randomly samples data points from D to form a subset D_sub.
   b. If D_sub is empty or contains only one data point:
      i. Mark as a leaf node and store the data point or null if empty.
      ii. Add to T.
      iii. Continue to next iteration.
   c. Else:
      i. Randomly select a feature F.
      ii. Randomly select a split value V within the range of values of F.
      iii. Partition D_sub into two subsets D_left and D_right based on F and V.
      iv. Recursively call this algorithm on D_left and assign result to left node.
      v. Recursively call this algorithm on D_right and assign result to right node.
      vi. Create an internal node with F, V, left node and right node.
      vii. Add this node to T.
3. Given a data point x to be predicted:
   a. Initialize anomaly_score = 0
   b. For each tree T_i in T:
      i. Start at the root node N.
      ii. While N is not a leaf node:
         - If x's value for the selected feature at N is less than the split value at N:
            * Move to N's left child.
         - Else:
            * Move to N's right child.
      iii. If N is a leaf node and stores a data point y:
         - Calculate path_length = number of edges traversed to reach N.
         - If y is null (empty leaf node):
            * Set path_length = average path length of trees in T for an empty leaf node.
      iv. anomaly_score += path_length / average_path_length_over_all_trees
   c. Return anomaly_score as the anomaly score for x.
```

Fig. 2. Pseudo code of the Isolation Forest algorithm for fault diagnosis

In fault anomaly detection, common methods include isolation forest detection, anomaly detection based on K-means clustering and outlier detection based on histograms. This paper selects historical data containing normal and abnormal sample points to test the accuracy of the above three algorithms in fault detection of converged communication systems. 9000 sample characteristic data are selected for testing, among which 100 are abnormal point data. The test results are shown in Table 1. In Table 1, the abnormal sample detection rate refers to the percentage of correctly detected abnormal samples among the total actual abnormal samples, the accuracy of detected abnormal samples refers to the percentage of correctly detected abnormal samples among the total detected abnormal samples. The higher these two indicators are, the better the detection performance. It can be seen from Table 1 that the algorithm using the isolation forest for fault detection in converged communication systems has an abnormal point detection rate of 91% and the accuracy of detected abnormal samples is over 93%, which has a significant advantage compared with other abnormal data detection algorithms.

Table 1

Detection Results of Different Models

Methods	**Abnormal point detection rate (%)**	**Accuracy rate of detecting abnormal samples (%)**
Isolation Forest	91.0%	93.4%
K-means	62.0%	73.8%
Histograms	55.0%	42.4%

3.3 Construction of LLM Agents for Smart Operations and Maintenance of Converged Communication Systems

The isolation forest algorithm can diagnose abnormal characteristic data from the massive characteristic data collected by the converged communication system, thus completing the fault diagnosis of the system. After diagnosing the fault characteristic data, it is necessary to conduct fault analysis and operational maintenance based on these data. In the traditional operation and maintenance of communication systems, manual operations are still relied on. To achieve automated and intelligent operation and maintenance of converged communication systems, improve the operation and maintenance efficiency of converged communication systems and reduce the dependence of system operation and maintenance on manual labor, this paper proposes an intelligent operation and maintenance method for domestic software hardware converged communication systems driven by LLM.

1) Construction of the Knowledge Base for Intelligent Operations and Maintenance of Converged Communication Systems

To perform fault analysis and handling based on the diagnosed fault characteristic data, it is necessary to find the fault causes and corresponding solutions according to the fault characteristics. This paper adopts the Retrieval Augmented Generation (RAG) technology of large models to realize the retrieval of fault causes and solutions. First, a RAG knowledge base for intelligent operations and maintenance of converged communication systems is constructed. Then, based on the fault characteristic data diagnosed by the isolation forest, a search is carried out in the RAG knowledge base to find the fault causes and corresponding solutions for the fault characteristic data. Finally, the LLM Agent technology is applied to analyze the fault causes and corresponding solutions and relevant system tools are called to solve the fault problems.

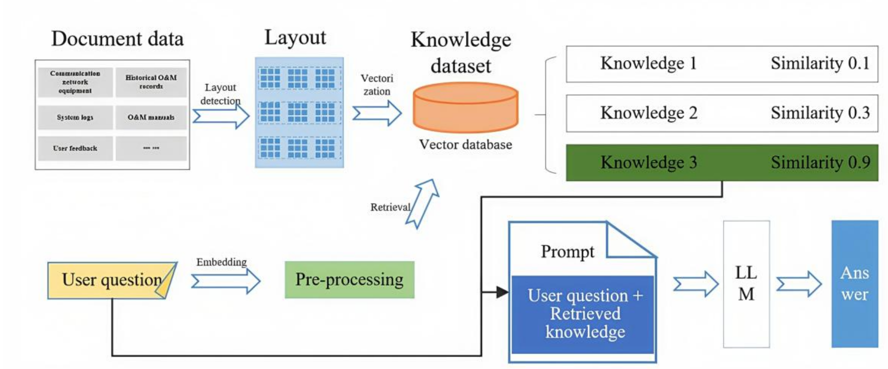

Fig. 3. Construction process of the RAG knowledge base for the domestic integrated communication system

The process of constructing the RAG knowledge base for intelligent operations and maintenance of converged communication systems is shown in Fig. 3. Firstly, data collection and preprocessing of domestic software and hardware systems are required. The data for intelligent operations and maintenance of converged communication systems mainly include various types of documents such as communication network equipment documents, system logs, user feedback, historical operation and maintenance records and operation and maintenance solution manuals. After obtaining the documents, preprocess them to remove duplicate, invalid and incorrect data. Then, divide the content within the documents into chunks and use an embedding model to convert the text into vectors and store them in a vector database. Finally, during the user question query stage, convert the user's question into a vector through the same embedding model and search for all text chunk contents related to the user's question in the vector database for subsequent result generation.

In the text chunking phase, a page layout detection model based on YOLO v7 serves as the core technical support, enabling intelligent parsing of operation and maintenance document layouts and accurate identification of text chunks; specifically, the model classifies regions with independent semantic attributes (such as different columns in a two-column layout and various paragraphs like body text, formula explanations, and figure/table annotations) into individual text chunks based on the document's inherent formatting, ensuring each chunk carries complete local semantic information. Regarding the chunks' dimensions, their character count varies dynamically with the original document's layout (e.g., paragraph length differences and column width settings, with no fixed standard), and strict measures prevent overlap between chunks during division—this not only maintains each chunk's semantic integrity but also avoids repeated information matching in subsequent retrieval, laying a foundation for the RAG system's retrieval accuracy and efficiency.

Text vectorization is the most critical stage in the RAG system and its accuracy affects the final answer result. In the text vectorization stage, the embedding model is used to convert the chunked text into vector data and store it in the vector database for retrieving the similarity of user questions. 135 device documents are used to create a test dataset, which contains 200 questions and their corresponding correct answers. The currently available vectorization models BGE, GTE, M3E and BCE are tested. The recall results of different models are calculated as shown in Table 2. It can be seen that the bce-embedding model achieves the best recall result in the question answering scenario of this paper. When constructing the RAG knowledge base in the intelligent agent of this paper, the bce-embedding model is selected as the basic model for vectorization.

Table 2

Test Results of Different Models

Model	Accuracy
BGE /bge-m3-large	72.5%
GTE /gte-large-zh	43.0%
M3E /m3e-large	76.5%
BCE /bce-embedding	88.0%

The bce-embedding model achieved an accuracy of 88.0% without undergoing specific training for this scenario and this result was obtained from training on a general-purpose dataset. When constructing the intelligent operation and maintenance intelligent agent, the long-term memory is derived from the RAG system and the accuracy of the RAG system will affect the accuracy of the intelligent operation and maintenance system. In order to further improve the accuracy of the bce-embedding model, this paper constructs a corresponding dataset for fine-tuning the bce-embedding model based on the data documents collected from the intelligent operation and maintenance scenarios of the converged communication system.

For the converged communication system, a fine-tuning training dataset is constructed. When training the bce-embedding model, each sample in the dataset consists of three lists: "query", "pos" and "neg". "Query" represents the question, "pos" represents the list of positive samples and "neg" represents the list of negative samples. Finally, the data is saved as a jsonl file. In this paper, 7,000 dataset samples are constructed for the converged communication scenario and used for the fine-tuning training of the model. Due to the limited amount of training data, in order to ensure that the bce-embedding model has good generalization ability, the number of epochs is set to 8 rounds during the fine-tuning process. Eventually, the accuracy of the fine-tuned bce-embedding advance model in the test question set reaches 96.7%, which is nearly a 10% improvement compared to the accuracy of the original bce-embedding model in retrieval.

2) Fault Handling of Converged Communication Systems Based on LLM Agents

Input the data of the fault status and phenomena into the RAG knowledge base and the RAG knowledge base can be used to obtain the data of the fault causes and solutions. In order to further achieve intelligent fault handling, it is also necessary to convert the solutions into specific fault handling actions. This paper designs an intelligent LLM Agent for fault handling of converged communication systems. By applying the LLM Agent technology, it integrates the isolation forest algorithm and the intelligent operation and maintenance RAG knowledge base to complete the intelligent processing of the whole process of fault detection, fault analysis and fault handling. The structure of the LLM Agent constructed in this paper is shown in Fig. 4. The Agent for fault handling of the converged communication system mainly includes planning, execution, tool and memory modules, which are mainly responsible for fault diagnosis, fault analysis and fault handling tasks. Different modules cooperate with each other to complete the fault handling of the converged communication system.

The Agent for fault handling of the converged communication system first conducts task planning. It designs prompt words to guide the LLM to decompose the task of "operation and maintenance of the converged communication system" into three steps: fault detection, fault analysis and fault handling.

The first step is fault detection. During fault detection, the isolation forest algorithm tool in the tools is first invoked. This tool detects the characteristic data from the converged communication system and inputs the characteristic data into the isolation forest model. Once the model detects isolated samples, the LLM formulates a further detection plan. Based on the characteristics of the detected samples, it determines which communication network data needs to be collected for in-depth analysis. For example, according to the fault phenomena, it determines the network traffic data, device status data, etc., that need to be collected.

The second step is fault analysis, which is executed if the first step determines that there is a fault. During fault analysis, the large model RAG knowledge base tool needs to be invoked to find the causes of problems and solutions from the knowledge base, assisting the large model in conducting fault analysis. In the fault analysis stage, the language understanding and generation capabilities of the LLM play a crucial role in the entire process. To improve the LLM's understanding ability of fault analysis in the converged communication system, this paper adopts the few-shot instruction learning method. As shown in Fig. 5, a small number of task examples are used to guide the model to complete specific tasks.

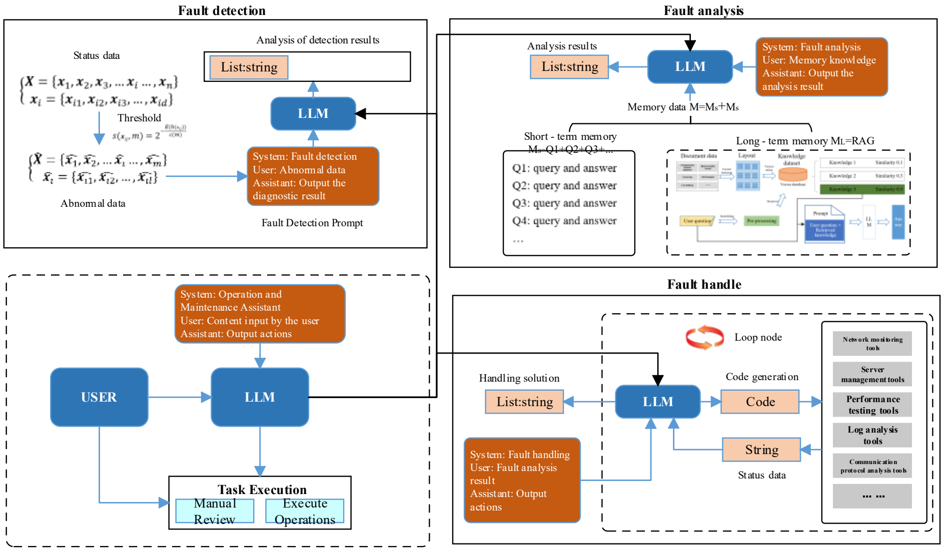

Fig. 4. The LLM Agent for fault handling of the converged communication system

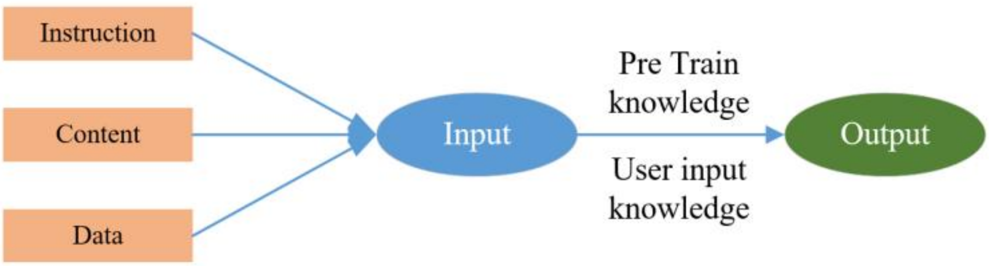

Fig. 5. Few-shot learning of LLM for fault handling of converged communication system

A small number of learning samples are written into the prompt words of the LLM. In addition, the prompt words of the LLM also consist of three parts: "system", "user" and "assistant". In the LLM for fault handling of the converged communication system, "system", "user" and "assistant" each play an important role. "System" provides the basic framework and background information for the entire interaction process, clarifies the functional scope, operating environment and task objectives of the converged communication system, etc. and sets the scene and rules for fault analysis. "User" represents the actual occurrence of a fault problem. Through the diagnosis of the isolation forest model and the retrieval of fault-related knowledge by the RAG, it provides specific problem inputs for the LLM. And "assistant" is like a professional technical expert. Based on the background and rules provided by "system", it accurately analyzes and diagnoses the fault phenomena proposed by "user", gives reasonable fault handling steps and suggestions and realizes the rapid solution of the fault problems of the converged communication system, thus ensuring the normal operation of the system. An example of the prompt words for fault handling by the LLM of the converged communication system constructed in this paper is shown in Table 3.

Table 3

Prompt Words for Fault Handling of The LLM of The Domestic Converged Communication System

Prompt for Agent Large Language Model
System: The converged communication system includes functions such as voice calls, video calls, instant messaging and file transfer. It is currently running normally, but various faults may occur. The task is to accurately diagnose and provide effective fault handling steps based on the input.
User: Fault phenomenon characteristic data output by the Isolation Forest model; Solution text content recalled by RAG.
Assistant: Fault handling steps.
Input Example 1: Fault Phenomenon: File transfer fails in the converged communication system and the progress bar remains static. Fault Diagnosis Result: It may be due to an overly large file or network issues. Fault Handling Steps: 1) Check the network connection to ensure a stable network with sufficient speed. 2) Try splitting the file and transfer smaller files in batches. 3) Check the file transfer settings of the converged communication system to confirm if there are any restrictions.
Input Example 2: Fault Phenomenon: The converged communication system cannot send instant messages and a sending failure prompt appears. Fault Diagnosis Result: It may be due to server connection issues or software faults. Fault Handling Steps: 1) Check the network connection to ensure normal access to the converged communication server. 2) Restart the converged communication software to see if it can return to normal. 3) Check the software settings to confirm whether settings such as message sending permissions are correct.

The third step is fault handling. Fault handling formulates a treatment plan based on the results of the second step fault analysis. This step also involves the invocation of the memory module and the execution module. By invoking the memory module, it is determined whether there is a historical record of the solution to this problem, further improving the accuracy of problem solving. If there are differences between the results of fault analysis for the same problem and the historical treatment results, corresponding difference information will also be generated. Finally, the execution module is invoked. To ensure the accuracy of fault diagnosis and handling by the LLM Agent, in the execution module, tools will also be invoked to send the fault phenomena, fault diagnosis results and fault handling steps found by the fault analysis module to manual confirmation. Only after manual confirmation and execution can the execution module finally call the tool module to perform live network fault handling.

4. Experimental and Analysis

4.1 Application Case

The LLM used in the experiments of this paper is Qwen2.5-72B, which is invoked in the form of the OpenAI API. The Agent development platform is the Dify platform, which is deployed in a Docker manner on a server with an NVIDIA 4090 graphics card and 128G of running memory. To test the performance of the LLM Agent for intelligent operation and maintenance fault handling, this paper simulates possible fault situations in the operation of the converged communication system from three aspects: hardware faults, software faults and network faults. A total of 30 fault problems are simulated in the experiments.

In terms of hardware faults, the situation of hardware failures is simulated by unplugging key hardware devices such as the hard disk and memory of the server. For example, unplugging the hard disk may cause problems in data storage and affect the normal operation of the server, unplugging the memory may lead to a decline in server performance or even prevent the server from starting.

In terms of software fault simulation, it is carried out by modifying the server's configuration files, shutting down key services, etc. For instance, modifying the configuration file may cause incorrect parameter settings of the server, affecting the communication function; shutting down key services may cause certain specific communication functions to be unable to be used normally.

In terms of network faults, methods such as disconnecting the server's network connection and setting network latency are adopted. Disconnecting the network connection will prevent the server from communicating with other devices and affect the operation of the entire converged communication system; setting network latency can simulate the situation of network congestion and test the performance of the system under poor network conditions.

In the experiments, the LLM Agent for converged communication fault handling proposed in this paper is compared with traditional fault handling methods. Traditional fault handling methods mainly rely on manual experience and professional knowledge. When a fault occurs, technicians diagnose the fault by checking the server's log files, network status, etc. Then, based on the diagnostic results, they determine the location and cause of the fault by relying on experience and professional knowledge and take corresponding repair measures. The LLM Agent, according to the fault handling plan, includes operations that can be automatically executed (such as restarting devices, adjusting network configurations, etc.) and operations that require manual intervention (such as replacing hardware components, fixing software vulnerabilities, etc.). Operations that require manual intervention are directly sent by the Agent to the operator with the treatment plan and steps. The fault handling results of different methods are shown in Table 4.

In Table 4, the fault handling time refers to the time from the occurrence of a fault to its complete resolution. The fault handling time reflects the efficiency of fault handling. The shorter the time, the higher the processing efficiency. The efficiency of the method using the Agent proposed in this paper can be increased by about 2.5 times. The fault analysis accuracy rate is the ratio of the number of times the fault location and the corresponding treatment plan are correctly located to the total number of faults. The location accuracy rate of the LLM Agent is also 26.6% higher than that of manual methods. The fault repair success rate is the ratio of the number of times the fault is successfully repaired to the total number of faults. The fault repair success rate of the LLM Agent is 10% higher than that of manual methods. It can be seen from Table 4 that the LLM Agent for converged communication fault handling proposed in this paper has certain advantages in terms of fault handling time, accuracy rate and repair success rate.

Table 4

Fault Handling Results of Different Methods

Performance indicator	**Traditional method**	**LLM Agent**
Average handling time	58 min	23 min
Analysis accuracy	70.0%	96.6%
Fault Repair Success Rate	83.3%	93.3%

To analyze the efficiency of the three core modules ("fault identification, fault analysis, and fault repair") and the impact of the absence of any single module on the overall performance of the operation and maintenance intelligent agent, a comparison was conducted between the complete LLM intelligent agent system (baseline performance: average handling time of 23 minutes, analysis accuracy of 96.6%, and fault repair success rate of 93.3%) and scenarios where one module was missing: When the fault identification module (responsible for initial fault detection and classification) was absent, redundant inference processes and misclassification led to a 78.3% increase in average handling time (rising to 41 minutes), a drop in analysis accuracy to 81.2%, and a decrease in fault repair success rate to 86.7%; When the fault analysis module was missing, performance degradation was the most severe, with average handling time increasing to 52 minutes, analysis accuracy becoming invalid, and fault repair success rate plummeting to 53.3%; When the fault repair module was absent, delays in manual execution caused the average handling time to increase to 45 minutes. The above results indicate that all three modules are indispensable: the fault identification module shortens the preliminary diagnosis cycle, the fault analysis module is key to formulating effective solutions, and the fault repair module ensures the implementation of repair strategies. The absence of any module will significantly reduce the efficiency and performance of the operation and maintenance intelligent agent.

4.2 Accuracy Analysis

As the core of the Agent studied in this paper, the accuracy of the LLM is crucial to the accuracy of fault handling. To test the effectiveness of the method proposed in this paper on different LLM, while only changing the LLM, this paper uses the OpenAI compatible interface to continue the fault analysis of the above mentioned 30 faults on several currently mainstream available LLM, such as MiniChat-2-3B, Llama3-8B, Qwen2.5-32B and Qwen2.5-72B and tests the fault analysis accuracy rates of different models. The test results are shown in Fig. 6. It can be seen from Fig. 6 that when other conditions remain unchanged, the basic capabilities of the LLM have a great impact on the accuracy of fault analysis. The fault analysis accuracy rates of models with relatively small parameter scales such as MiniChat-2-3B, Llama3-8B, Qwen2.5-32B are less than 50%, which are significantly lower than that of the Qwen2.5-72B model. When the number of model parameters is small, the system cannot accurately analyze the fault problems from the content retrieved by RAG. As the parameters of the LLM increase, the capabilities of the LLM gradually strengthen. Qwen2.5-72B can achieve the highest fault - analysis accuracy rate among the tested models, reaching 96.6%. In the future, with the improvement of the capabilities of LLM, the accuracy rate of fault handling based on Agents is expected to be further improved. From the experimental results, it can also be seen that the capabilities of currently available LLM increase with the increase of parameters. When applying in engineering, a parameter model of about 72B should be selected. When the parameters of the LLM reach about 72B, its capabilities emerge.

Apart from the basic capabilities of LLM, the RAG knowledge base and prompt words also have a significant impact on the processing results of intelligent agents for smart operations and maintenance. To further analyze the semantic similarity between the answers generated by the LLM and the expected outputs, which are influenced by the RAG knowledge base and prompt words constructed in this paper, Sentence Transformer is adopted as the evaluation model. The Sentence Transformer model can effectively conduct semantic similarity evaluation.

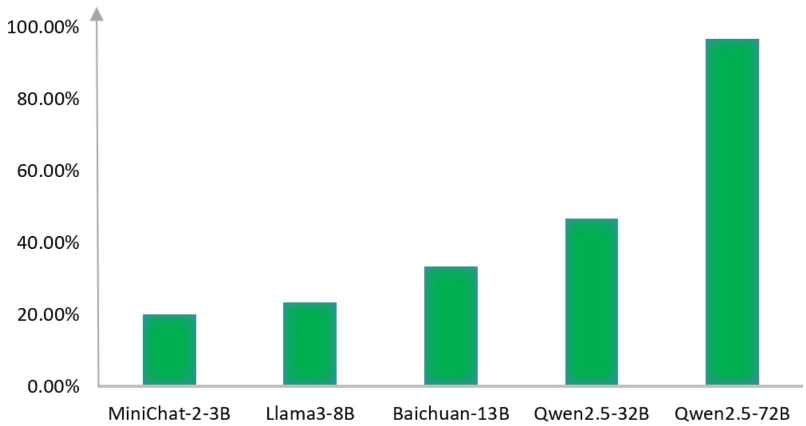

FIG. 6. The accuracy rates of fault analysis of different LLM

To study how the RAG knowledge base and prompts affect the similarity between the LLM-generated answers and the expected results, the Qwen2.5-72B model is kept unchanged in the operation and maintenance intelligent agent proposed in this paper. The semantic similarity between the response results to the same questions and the expected results is compared under five scenarios: using only Qwen2.5-72B, using the LLM with simple prompt words, using the LLM with advanced prompt words, using the LLM with the RAG knowledge base and using the LLM with both the RAG knowledge base and advanced prompt words, as shown in Fig. 7.

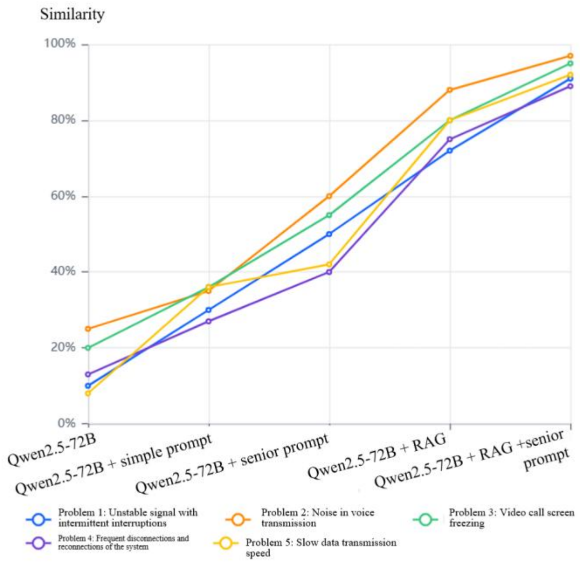

Fig. 7. Semantic similarity between the output of questions and the expected results under different conditions

As can be seen from Fig. 7, when neither prompt words nor the RAG knowledge base is used, the semantic similarity between the output answer and the expected result is relatively low, approximately 20%. When the RAG knowledge base is employed, there is a significant improvement in the similarity between the output answer and the expected result, an increase of about 60%. In the case of using the LLM along with the RAG knowledge base and advanced prompt words, the similarity between the output answer and the expected result is relatively high, ultimately reaching approximately 90%.

4.3 Limitations

In practical applications, the LLM-based Operation and Maintenance (O&M) Intelligent Agent has three obvious limitations: In scenarios with multiple overlapping faults, it is easily interfered by dominant fault characteristics (such as database service failure), ignores implicitly associated faults (such as switch port anomalies), and outputs one-sided solutions that require secondary manual intervention; When processing data in special formats, it cannot parse binary logs, and also struggles to identify professional information in text logs containing hexadecimal codes, thus requiring manual preprocessing of key content; In rare fault scenarios, due to the fact that the RAG knowledge base does not cover cases of faults in extreme environments (e.g., -15℃ low temperature) or new types of equipment, it fails to independently establish fault causality correlations and has to rely on external manuals, leading to a significant decline in fault handling efficiency.

5. Conclusion

This paper studies the intelligent agent technology for smart operation and maintenance of converged communication systems and proposes an operation and maintenance framework of a LLM intelligent Agent for converged communication systems. Firstly, the isolation forest algorithm is applied to diagnose faults in the characteristic data of the converged communication system and system fault information is diagnosed from massive amounts of data. Then, a RAG knowledge base of the LLM for converged communication systems is constructed, which is used by the LLM to conduct in-depth analysis of the results diagnosed by the isolation forest model. Detailed fault diagnosis results and fault handling plans are generated according to the content of the knowledge base. Finally, an LLM fault handling Agent is designed and the fault handling Agent is applied to formulate the optimal repair strategy, realizing the intelligent handling of faults in converged communication systems. This method has certain advantages in the operation and maintenance of converged communication systems. The fault analysis accuracy rate reaches 96.6% and the rate of assisting manual fault handling and repair reaches 93.3%, greatly improving the fault repair efficiency.

This study has achieved remarkable results in the intelligent agent technology for smart operation and maintenance of converged communication systems, but it still faces challenges such as insufficient generalization ability of LLM and the update and maintenance of the knowledge base. In the future, we will focus on the research of algorithm optimization and the enhancement of the generalization ability when applying LLM, so as to further improve operation and maintenance efficiency and promote the continuous development and widespread application of intelligent operation and maintenance technology.

REFERENCES

[1] He Lichong. Application of Integrated Communication in Emergency Dispatch and Command Communication System. Changjiang Information and Communication, 2023, 36(01): 227-230.

[2] Xue Weizhao, Zhang Huixin, Guo Ruiqing, et al. Domestic Design and Implementation of High-Reliability High-Speed Communication System. Electronic Measurement Technology, 2022, 45(24): 118-123.

[3] Zhu Yafei, Che Yu. Optimized Design of Integrated Operation and Maintenance Management System for Power Infrastructure in Communication Rooms. Telecom World, 2024, 31(03): 108-110.

[4] Chen Hongzhu. Research on Network Operation and Maintenance System Architecture of Telecom Operators. Communication and Information Technology, 2023, (S2): 23-25.

[5] Zhao Wei, Zhang Zhisen, Xiao Jiakang, et al. 5G Communication Network Operation and Maintenance Planning Method Based on Artificial Intelligence. Changjiang Information and Communication, 2022, 35(03): 219-222.

[6] Li Sen. Research on the Development Strategy of Service Network under the Background of Integrated Communication. China New Telecommunications, 2020, 22(22): 3-4.

[7] Trademarks; AT&T Services' Trademark Application for "AT&T SDN ONENET" Filed. elecommunications Business, 2015, 2491-2496.

[8] Signal Processing; Researchers at Chongqing University of Posts and Telecommunications Report New Data on Signal Processing (Robust Sensor Fusion With Heavy-tailed Noises). Electronics Newsweekly, 2020, 638-642.

[9] NTT Docomo Inc.; "Base Station And Communication Control Apparatus" in Patent Application Approval Process (USPTO 20200275497). Computer Technology Journal, 2020, 194-199.

[10] Hu Zhiqiang. Review of Domestic Substitution Based on Independent and Controllable Technologies. Cyberspace Security, 2018, 9(08): 90-97.

[11] Zhang Long, Dong Shutao. Research on the Operation and Maintenance Management System of Modern Communication Networks. Big Data Era, 2024, (07): 52-55.

[12] Qi Yongjun, Liu Xiaoshuo, Jia Zhengzheng, et al. Network Traffic Anomaly Detection Based on SVM. Journal of North China Institute of Aerospace Engineering, 2024, 34(02): 1-4.

[13] Wang Yanyan, Wang Yanning, Wang Min. Network Fault Diagnosis Method Based on Univariate Decision Tree. Computer Engineering and Design, 2007, (22): 5414-5416.

[14] Li Shengqi. Review of Network Fault Diagnosis and Performance Analysis. Technology Innovation and Application, 2015, (20): 99.

[15] Qi Fang, Yao Weiqing, Zhang Cheng, et al. Fault Diagnosis of Power Communication Network Based on Convolutional Neural Network. Information Technology, 2023, (06): 119-123 + 128.

[16] Shang Yu, Liu Kun, Han Xiaolong. Cross-Domain Stance Detection Based on LLM Agent. Information Security and Communications Privacy, 2024, (08): 62-71.
[17] Li X, Wang S, Zeng S, et al. A survey on LLM-based multi-agent systems: workflow, infrastructure and challenges. Vicinagearth, 2024, 1(1): 9-14.
[18] Hassanin M, Keshk M, Salim S, et al. PLLM-CS: Pre-trained LLM (LLM) for cyber threat detection in satellite networks. Ad Hoc Networks, 2025, 166103645-103645.
[19] Wen B. Research on the applications of natural language processing//Faculty of Medical and Health Sciences and Bioengineering Institute, University of Auckland, ITM Department, Illinois Institute of Technology, USA. Proceedings of the 5th International Conference on Computing and Data Science(part2). Tandon School of Engineering, New York University, 2023.108670.
[20] Ma Fangming, Han Longgang, Guo Bao, et al. Research on the Digital and Intelligent Transformation and Upgrading of Base Station Operation and Maintenance Based on Agent Technology. Telecommunications Engineering Technology and Standardization, 2024, 37(09): 48-52.
[21] Liao Jianxin, Qi Qi, Wang Jingyu, et al. 6G Intelligent Service Network: Vision, Architecture and Key Technologies. Scientia Sinica (Informationis), 2024, 54(05): 991-1024.
[22] Guo D, Zhu Q, Yang D, et al. DeepSeek-Coder: When the Large Language Model Meets Programming--The Rise of Code Intelligence. arXiv preprint arXiv: 2401. 14196, 2024.
[23] Ying Z, Zheng G, Huang Y, et al. Towards Understanding the Safety Boundaries of DeepSeek Models: Evaluation and Findings. arXiv preprint arXiv:2503.15092, 2025.
[24] Biswas S S. Potential use of chat gpt in global warming. Annals of biomedical engineering, 2023, 51(6): 1126-1127.
[25] Singh S K, Kumar S, Mehra P S. Chat gpt & google bard ai: A review//2023 International Conference on IoT, Communication and Automation Technology (ICICAT). IEEE, 2023: 1-6.
[26] Zamfirescu-Pereira J D, Wong R Y, Hartmann B, et al. Why Johnny can't prompt: how non-AI experts try (and fail) to design LLM prompts//Proceedings of the 2023 CHI conference on human factors in computing systems. 2023: 1-21.
[27] He J, Rungta M, Koleczek D, et al. Does Prompt Formatting Have Any Impact on LLM Performance. arXiv preprint arXiv: 2411. 10541, 2024.