

## Article

# Multimodal Large Language Model-Based Fault Detection and Diagnosis in Context of Industry 4.0

Khalid M. Alsaif \* , Aiiad A. Albeshri , Maher A. Khemakhem  and Fathy E. Eassa

Department of Computer Science, King Abdulaziz University, Jeddah 21589, Saudi Arabia;  
aalbeshri@kau.edu.sa (A.A.A.); makhemakhem@kau.edu.sa (M.A.K.); feassa@kau.edu.sa (F.E.E.)

\* Correspondence: kibrahimalsaif@stu.kau.edu.sa

**Abstract:** In this paper, a novel multimodal large language model-based fault detection and diagnosis framework that addresses the limitations of traditional fault detection and diagnosis approaches is proposed. The proposed framework leverages the Generative Pre-trained Transformer-4-Preview model to improve its scalability, generalizability, and efficiency in handling complex systems and various fault scenarios. Moreover, synthetic datasets generated via large language models augment the knowledge base and enhance the accuracy of fault detection and diagnosis of imbalanced scenarios. In the framework, a hybrid architecture that integrates online and offline processing, combining real-time data streams with fine-tuned large language models for dynamic, accurate, and context-aware fault detection suited to industrial settings, particularly focusing on security concerns, is introduced. This comprehensive approach aims to address traditional fault detection and diagnosis challenges and advance the field toward more adaptive and efficient fault diagnosis systems. This paper presents a detailed literature review, including a detailed taxonomy of fault detection and diagnosis methods and their applications across various industrial domains. This study discusses case study results and model comparisons, exploring the implications for future developments in industrial fault detection and diagnosis systems within Industry 4.0 technologies.



**Citation:** Alsaif, K.M.; Albeshri, A.A.; Khemakhem, M.A.; Eassa, F.E.

Multimodal Large Language Model-Based Fault Detection and Diagnosis in Context of Industry 4.0.  
*Electronics* **2024**, *13*, 4912. <https://doi.org/10.3390/electronics13244912>

Academic Editors: Wei Ji, Hao Fei and Fei Li

Received: 11 November 2024

Revised: 3 December 2024

Accepted: 9 December 2024

Published: 12 December 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

In modern technology, developing specialized fault detection and diagnosis (FDD) systems for industrial applications has emerged as a vital endeavor [1]. Manufacturing processes and industrial operations have been revolutionized by the advent of Industry 4.0 (I4.0), characterized by the integration of advanced technologies, including artificial intelligence (AI), the Internet of Things (IoT), and big data analytics. Digital transformation establishes opportunities and challenges for maintenance. Additionally, it optimizes complex industrial systems.

Large learning models (LLMs) represent a remarkable breakthrough in AI [2], transforming organizational decision making, problem resolution strategies, and operational efficiency. Trained on vast amounts of textual data, these models have exhibited noteworthy capabilities in understanding context, generating human-like text and carrying out a wide range of language-based tasks. Their impact extends to network systems, which address challenges of complex configurations and heterogeneous infrastructure management [3]. Although LLMs have been successfully applied in various domains, their potential for industrial FDD systems is still largely unexplored.

Existing FDD approaches are mainly dependent on traditional machine learning (ML) techniques [4], expert systems [5], or statistical methods [6]. The current methods often face difficulties in handling the intricacy and various fault scenarios in modern industrial systems. Additionally, they tend to concentrate on particular data types or predetermined

fault patterns, which constrains their ability to adapt to new or unanticipated faults [7]. The crucial obstacles in FDD research are the incorporation of multimodal data and the capability to analyze different data types.

In our proposed work, these gaps are addressed via a novel multimodal FDD framework that uses LLMs, which offers several key contributions. First, an LLM-based FDD method is proposed using GPT-4-Preview, which enhances the scalability, generalizability, and efficiency of complex systems and various fault scenarios. Second, by generating synthetic datasets using LLMs, we enrich the knowledge base and enhance the accuracy of FDD in imbalanced scenarios. Third, the approach optimizes the diagnosis accuracy and overall performance of the framework. Fourth, we introduce a hybrid architecture that integrates online and offline processing, which combines real-time data streams with fine-tuned LLMs for dynamic, accurate, and context-aware fault detection suitable to I4.0 environments. This comprehensive approach aims to overcome traditional FDD challenges and advance the field toward more adaptive and efficient fault diagnosis systems. In the following sections, a comprehensive literature review is presented, our methodology and system design are detailed, the results of our case study and model comparisons are discussed, and the implications of our findings for future developments in industrial FDD systems and I4.0 technologies are explored.

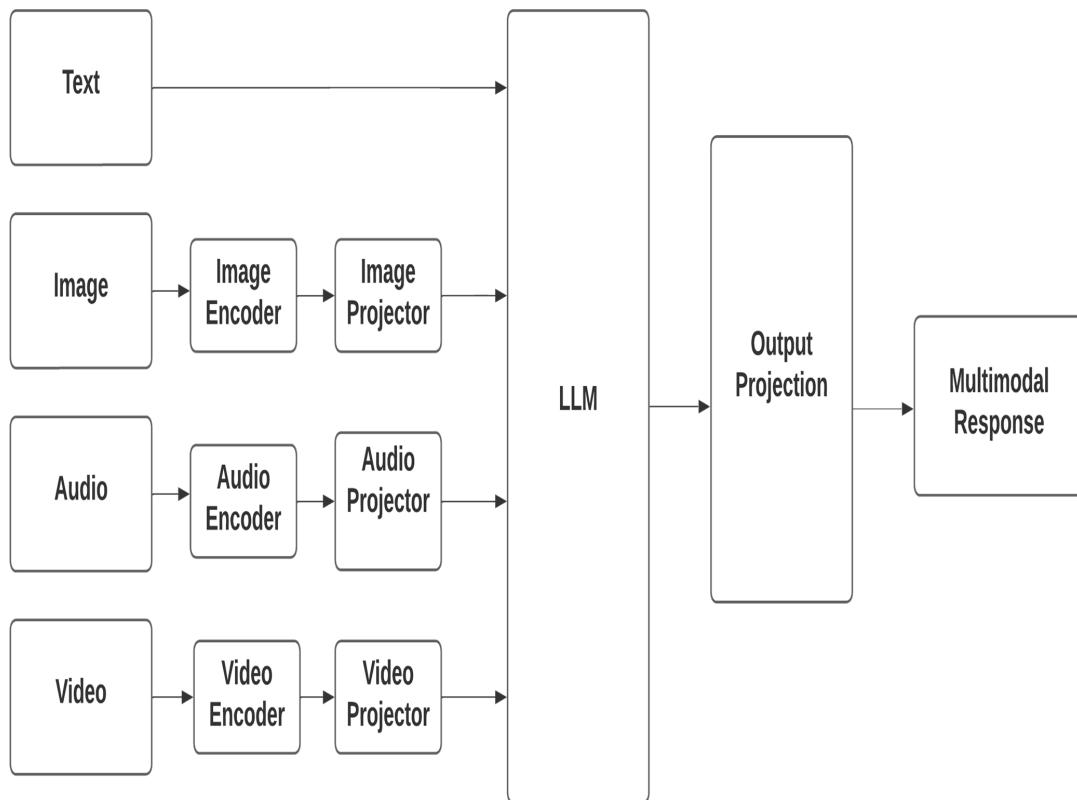
## 2. Background, Taxonomy, and Related Research

In recent years, FDD has seen remarkable advancements, particularly with multimodal LLMs (MM-LLMs) emerging as cutting-edge technology. These advanced AI-based systems were created to integrate and process various data types, including text, images, audio, and video, which makes them more applicable across multiple industrial contexts. The rapid evolution of LLMs, exemplified by ChatGPT's ability to handle diverse tasks and its unprecedented adoption rate [8], further demonstrates their potential for industrial applications.

Several studies have explored the use of LLMs and MM-LLMs in FDD. For instance, combining fuzzy logic with ChatGLM2-6B has potential in electrical equipment fault diagnosis, enhancing the adaptability and accuracy of fault diagnosis mechanisms [9]. Likewise, integrating knowledge graphs with LLMs has been applied to compressor fault diagnosis, which demonstrates significant improvements in fault detection accuracy [10]. Prompt engineering approaches have also been investigated for spacecraft fault diagnosis, which highlights the potential of LLMs in managing complex multi-level fault scenarios [11]. Moreover, the use of LLMs in aviation assembly and autonomous driving in mines emphasizes the versatility and effectiveness of these models in diverse industrial applications [12–14].

Figure 1 illustrates the architecture of MM-LLMs, which consists of four key components: a modality encoder, which encodes inputs from various modalities to obtain corresponding features; an input projector, which aligns these features with the text feature space via methods such as linear projectors, a multilayer perceptron, and cross-attention mechanisms; the LLM backbone, which processes the representations to perform semantic understanding, reasoning, and decision making; and an output projector, which maps signal token representations into features that are comprehensible to the modality generator, which produces outputs in distinct modalities via latent diffusion models. Examples of MM-LLMs include Flamingo, which integrates NFNetF6 and Chinchilla for visual and textual processing; BLIP2, which combines CLIP and Flan-T5 for image-text tasks; and LLaVA, which employs CLIP and Vicuna for visual question answering. The training pipeline is divided into multimodal pre-training, which focuses on aligning modalities using X-Text datasets, and multimodal instruction tuning, which fine-tunes pre-trained MM-LLMs with instruction-formatted datasets to enhance their zero-shot performance and generalization [15].

This comprehensive architecture enables MM-LLMs to effectively integrate multiple modalities, leveraging pre-trained models to optimize computational efficiency and improve performance across various multimodal tasks.

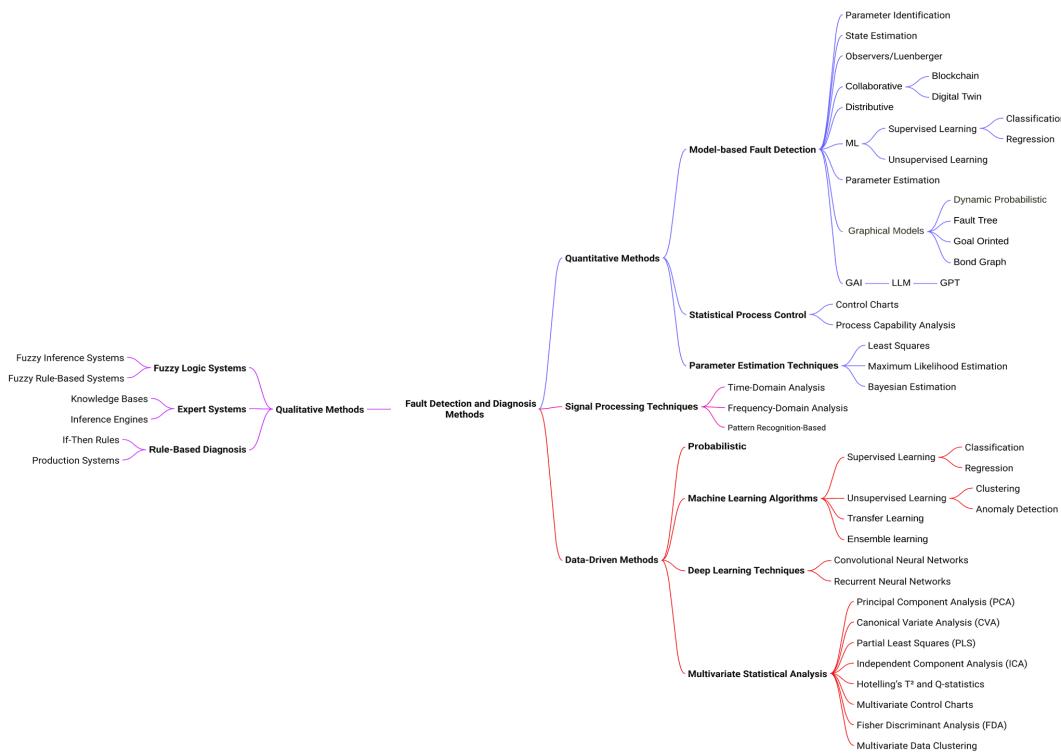


**Figure 1.** MM-LLM architecture demonstrating the integration of diverse data types for comprehensive fault analysis.

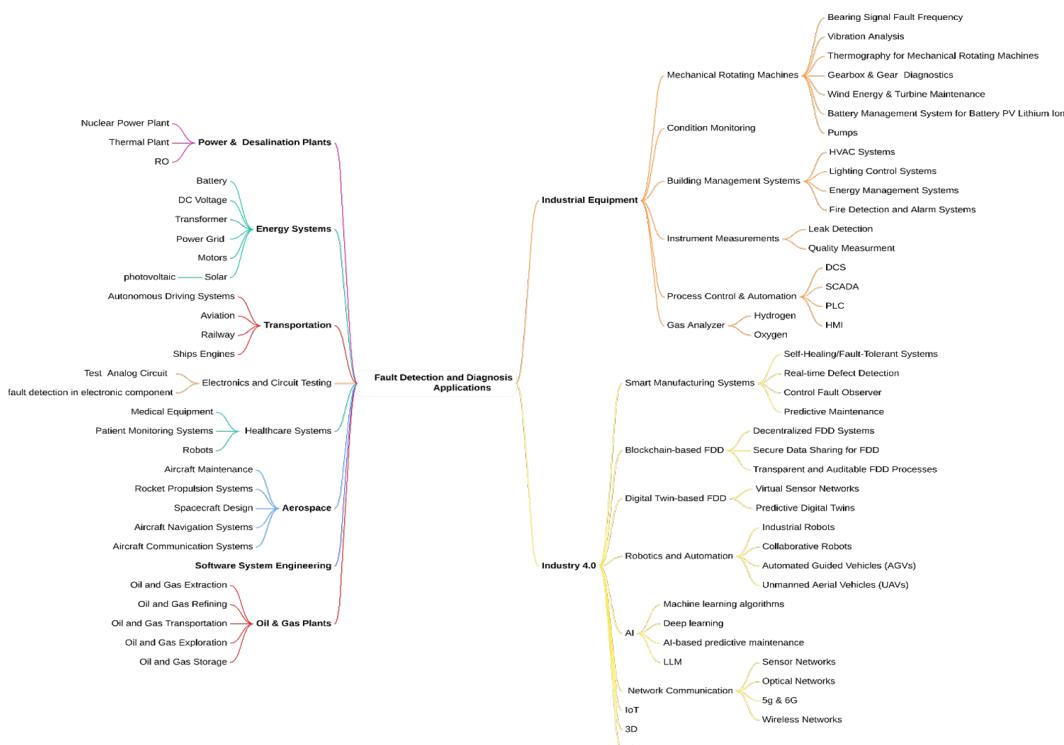
Given the novelty and potential of MM-LLM-based FDD systems, a comprehensive search was carried out using the Scopus database to acquire references for related studies. To narrow down the search, the following query was employed, focusing on the terms “fault diagnosis” and “fault diagnostics” within the title or abstract: TITLE-ABS-KEY (“fault diagnosis” OR “fault diagnostics”) AND PUBYEAR > 2018 AND PUBYEAR < 2025 AND (LIMIT-TO (SRCTYPE, “j”)) AND (LIMIT-TO (LANGUAGE, “English”)) AND ( LIMIT-TO (SUBJAREA, “ENGI”) OR LIMIT-TO (SUBJAREA, “COMP”)). The search was further refined to include publications from 2019 to 2024, ensuring the most recent and relevant findings. To maintain a high standard of research, only journal articles (indicated by “j”) were considered, and the language was restricted to English. Moreover, the subject areas were limited to Engineering (“ENGI”) and Computer Science (“COMP”), aligning with the theme of industrial fault diagnostic support systems. A total of 11,655 documents were identified using this approach.

Subsequently, comprehensive and insightful taxonomies in FDD in terms of methods and applications were utilized, as shown in Figures 2 and 3, respectively. The color coding helps readers quickly distinguish between different methodological categories in fault detection and diagnosis. In Figure 2, the purple lines represent quantitative methods branches, red lines highlight data-driven approaches, and blue lines indicate qualitative methods. In Figure 3, green lines highlights energy system applications, orange lines show industrial equipment branches, yellow emphasizes Industry 4.0 applications, and purple lines indicate power plant applications. With the use of BERTopic and GPT-4, this comprehensive review was essential because of the limited number of articles that explicitly address MM-LLM-based FDD, highlighting its status as an emerging technology and emphasizing the need for further research in this promising area. The resulting taxonomies

provide a structured and easily navigable overview of the diverse FDD methods and their potential applications across various industrial sectors.



**Figure 2.** Comprehensive taxonomy of fault detection and diagnosis methods encompassing qualitative, model-based, signal processing, and data-driven approaches.



**Figure 3.** Application domains of fault detection and diagnosis methods across industrial and functional sectors [5,6].

## 2.1. Industrial Equipment

In mechanical rotating machines, attention mechanisms and deep learning (DL) techniques, including deep convolutional neural networks (DCNNs) and transfer learning, enhance bearing fault diagnosis [4,6,7]. The data-driven condition monitoring of industrial equipment has been reviewed, with a focus on various ML models [16]. A Bayesian network model was proposed for the IoT-based fault diagnosis of intelligent instruments [17]. Industrial equipment heavily depends on FDD methods, such as instrumentation [17], gas analyzers, and oxygen monitoring [18]. Process control applications, including SCADA [19], PLCs [20], and HMIs [21] integrate FDD algorithms with a hybrid customized chatbot HMI using ChatGPT for swift equipment information retrieval [22]. In the realm of hydraulic systems, a novel sensor fault estimation scheme for hydraulic servo actuators was proposed by Djordjevic et al. [23] using sliding mode observers, and its effectiveness was shown via simulations. This approach addresses the difficulties of nonlinear systems and uncertainties in industrial hydraulic equipment. Tao et al. [24] introduced a semi-supervised fault diagnosis method using a dynamic graph attention network with fuzzy K-nearest neighbors (KNNs) for planetary gearboxes. This method effectively addresses the challenge of limited labeled data in gearbox fault diagnosis, which achieves over 99% accuracy with very few labeled samples and outperforms traditional classification networks.

Additionally, a new multi-source partial transfer network model addresses real-world challenges in machinery fault diagnostics, including unknown label spaces and incomplete fault-type coverage. In employing class-level weight mechanisms and particle swarm optimization, the model combines shared feature extraction with domain-specific modules for fault diagnosis and adaptation. It was verified across six datasets, and its effectiveness in industrial equipment fault diagnosis was presented [25].

## 2.2. Industry 4.0

I4.0 integrates AI, data mining, and IoT to revolutionize manufacturing. DL, particularly CNNs, has the potential for FDD in smart manufacturing [26]. Digital twin collaboration aids in the auto-detection of erratic data at Energy 4.0 [27]. A novel approach that combines circular economy, data mining, and AI has been proposed for FDD [28]. Transfer learning has emerged as a valuable technique for predictive maintenance [29]. Innovations have been developed in 3D printing fault diagnostics [30,31] and optimized CNN approaches for industrial robotics [32]. Data security in industrial fault diagnosis is addressed by blockchain-based decentralized, federated transfer learning [33]. Recent advancements in IoT-based fault diagnosis include an industrial IoT platform with an EML for transformer monitoring [34], an IoT-based system using a one-dimensional (1D) CNN for PV plant monitoring [35], and preprocessing techniques with DL models for industrial IoT (IIoT) devices [36]. These approaches have shown improved accuracy, efficiency, and cybersecurity in fault diagnosis across various industrial applications.

In the context of fault estimation and fault-tolerant control, Wang et al. [37] proposed a Q-learning-based approach using an iterative learning control framework. This method addresses the obstacles posed by unknown faults that vary with both the time and trial axes in repetitive tasks, exhibiting the effectiveness of adapting to changing faults and maintaining control performance in MIMO systems.

In the era of big data and advanced networking, the diagnosis of fiber optic cable faults in communication networks has become critical. A combined GAN-CNN approach obtained 98.5% accuracy in addressing these faults [38]. DL enhances fault location and network maintenance efficiency [39].

A Bayesian network-based mechanism ensures efficient real-time fault diagnosis in communication systems [40]. For I4.0, a framework integrating LSTM, CNN, and GNN effectively handled heterogeneous data for fault detection [41]. Advanced orchestration techniques utilizing self-organizing maps and root cause analysis improve fault detection in 5G infrastructure and beyond [42]. A deep neural network-based method accurately diagnoses faults in millimeter-wave active phased arrays for 5G and 6G radios [43]. More-

over, a new federated learning framework known as FedCAE was proposed for edge-cloud collaboration in machine fault diagnosis. In this approach, the challenge of data islands is addressed by allowing multiple data owners to contribute to model training without sharing raw data, thereby exhibiting promising results in bearing fault diagnosis [44]. Furthermore, a comprehensive roadmap for performance-supervised plant-wide process monitoring in Industry 4.0 has been developed. This roadmap highlights the need to evaluate the impact of detected faults on plant-wide performance and proposes a technical route embedded in the cyber–physical–social system framework, which addresses key research questions and future directions in process monitoring [45].

### 2.3. Power Plants

FDD techniques are indispensable in thermal and nuclear power plants. Advancements have focused on intelligent methods, which include model-based, data-driven, and statistical approaches [46], which are applied to crucial equipment, including boilers, turbines, and heat exchangers. A robust soft sensor for estimating heat exchanger fouling in combustion thermal power plants was developed [47]. In nuclear power plants and photovoltaic systems, ML-based fault diagnosis techniques, such as KNNs, support vector machines (SVMs), bagged trees, and ensemble methods, detect faults in pressurized water reactors [48].

### 2.4. Electronic Systems

Numerous advanced techniques have been developed for the fault diagnosis and monitoring of electronic systems. SVMs have been applied for fault diagnosis in analog circuits [49]. FRA-based parameter estimation has also been utilized to diagnose faults in three-phase voltage-source inverters [50]. Recent advancements include the application of transformer-based networks for consumer electronic sensor fault diagnosis in the Internet of Everything era, addressing the challenges posed by environmental factors [51]. In microelectronic manufacturing, a novel hierarchical tree–DCNN structure has been proposed to address unbalanced data diagnosis in flexible integrated circuit substrates, improving the efficiency and accuracy of traditional human visual interpretation methods [52]. Moreover, joint self-attention mechanisms and residual networks have been used for the automated monitoring of intelligent sensors in consumer electronics, showing the potential for enhanced fault detection and classification in complex electronic systems [53].

### 2.5. Energy Sector

Recent advancements in energy sector fault diagnosis include enhanced methods for lithium-ion batteries [54], low-voltage direct current networks [55], and transformers [56]. CNNs perform superior power grid fault diagnosis [57], whereas KG-CNN enhances the motor fault diagnosis [58]. Photovoltaic systems benefit from infrared thermography and ML [59]. Advanced models such as GPT-4 are being explored for building energy management and fault diagnosis systems [60]. For integrated energy systems, a novel multiscale spatial–temporal graph neural network was proposed to address compound fault diagnosis with small sample sizes. This approach employs multiscale graph operators and gated recurrent units to learn label-specific fault features, hence mitigating the challenge of limited compound fault samples [61].

In high-voltage circuit breaker fault diagnosis, a new method combining variational mode decomposition energy entropy and an SVM has been developed. This approach, validated on a 35 kV SF<sub>6</sub> circuit breaker, effectively identifies faults, including actuator issues, buffer spring invalidity, and loose base screws [62].

### 2.6. Transportation

FDD systems are important for transportation, autonomous driving, aviation, railways, and ship engines. In autonomous driving, CNN-based information fusion aids in sensor fault detection [63]. Aviation utilizes biomimetic pattern recognition [64] and deep

neural networks [65] for fault diagnosis. Railways benefit from hybrid methods [66] and RP-CNN [67] for track and wheel fault diagnosis. Digital twin-based methods [68] and advanced fault feature extraction techniques [69] optimize condition monitoring and fault diagnosis for ships.

### 2.7. Healthcare Systems

FDD systems have advanced in various applications. A hybrid HMM-SVM approach enables sensor fault diagnosis in IoT-based healthcare systems [70]. For medical IoT devices, a lightweight autoencoder with KernelSHAP and an ANN classifies anomalies [71]. Medical equipment failure prediction benefits from an optimized least squares SVM algorithm, which demonstrates high accuracy for ECG equipment and incubators [72]. The integration of LLMs has further enhanced healthcare FDD systems, offering improved data management and diagnostic decision-making capabilities [73]. Recent advancements include the development of hierarchical health assessment models for equipment with uncertain fault diagnosis results. Based on Bayesian networks, these models can assess health states at different hierarchical levels by incorporating identified and uncertain fault diagnosis results. This approach has been successfully employed in complex safety-critical systems, including diesel engine combustion systems, showing its potential for more comprehensive health monitoring and risk avoidance in medical equipment [74].

### 2.8. Aerospace

Various innovative approaches have been developed for aerospace FDD applications. A PWM harmonic current-based method detects inter-turn short-circuit faults in FTPMSM drives [75]. For liquid rocket engines, fault diagnosis methods can be categorized as signal processing, model-driven, and AI-based approaches [76]. The LSTM-GAN method effectively detected faults in LOX/kerosene rocket engines [77]. A hybrid AI technique, BGOA-EANNs, is superior for aerospace health diagnosis [78]. The EMFFI approach, which employs bispectrum technology, SURF, LDA, and random forest (RF), attains a high accuracy in gearbox fault diagnosis for aerospace systems [79].

### 2.9. Software System Engineering

For software fault diagnosis, diverse ML, XAI, and data mining methods have been developed. ChatGPT-based fault diagnosis using execution traces enhances the identification of bug causes [80]. Data mining algorithms debug and compare software performance metrics [81]. A combination of adaptive tracing, fault injection, and graph convolutional networks demonstrates superior accuracy in fault diagnosis for various software systems [82]. Among the ML models evaluated for software fault diagnosis, XGBR showed the best performance in terms of accuracy and error metrics [83].

Tables 1 and 2 provide a comprehensive overview of related studies on FDD systems using various models. It details the models employed, their application domains, and sectors of implementation and evaluates them based on ML and DL paradigms. Moreover, the table examines the relevance of I4.0 technologies and their real-time capabilities, generality, and support for multimodal data. The applications range from industrial equipment and power plants to healthcare and aerospace, which reflects the diverse applicability of these models. These studies highlight the diverse range of FDD methods and their broad applicability in various industries. However, its challenges, including generalizability, multimodality, applicability to I4.0, and real-time detection and diagnosis, require further exploration. The novelty of the proposed work lies in addressing these gaps by developing a robust multimodality FDD model that accounts for various types of data from different device applications while utilizing the LLM capability.

**Table 1.** Overview of related work (part 1).

Ref.	Model	Applications	Sector	ML	DL	I4.0	RT	Gen.	MM
Prop. FDD	LLM	Multi-applications	All sectors	✓	✓	✓	✓	✓	✓
[6,50]	KNN	Bearing fault, voltage source inverters	Ind. equip., elec. sys.	✓	✓	✗	✓	✗	✗
[48]		Nuclear power plant	Power plants	✓	✓	✗	✗	✗	✗
[4,16,48,49,56,81]	SVM	Rotating machine, motors, pumps, nuclear power plant, circuits, transformer, software	Ind. equip., power, elec. sys., energy, software	✓	✗	✗	✓	✗	✗
[71]		Healthcare monitoring	Healthcare	✓	✗	✗	✗	✗	✗
[16,20,78]	ANN	Motors, pumps, PLC, tele-echography, satellite, aircraft	Ind. equip., aerospace	✓	✓	✗	✓	✗	✗
[28]		Bridge deck structures	14.0	✓	✓	✓	✗	✗	✗
[26,32,41]		Smart factory, IoT robotics, network, 6G	14.0	✓	✓	✓	✓	✗	✗
[63,67]	CNN	Autonomous driving, railway	Transportation	✓	✓	✗	✓	✗	✗
[4,57]		Rotating machinery, Power grid	Ind. equip., energy	✓	✓	✗	✗	✗	✗
[38]		Fiber optic cable	14.0	✓	✓	✓	✗	✗	✗
[17]	BN	Smart instrument	Ind. equip.	✓	✗	✓	✓	✗	✗
[40]		Communication	14.0	✓	✗	✓	✓	✗	✗
[27,36,41]	LSTM	Micro-electro-mechanical, IoT, IIoT, 6G	I4.0	✓	✓	✓	✓	✗	✗
[77]		Liquid rocket engines	Aerospace	✓	✓	✗	✗	✗	✗

Note: Ref. = references, ML = machine learning, DL = deep learning, I4.0 = Industry 4.0, RT = real time, Gen. = generality, MM = multimodality, BN = Bayesian network, LLM = large language model, IoT = Internet of Things, IIoT = industrial IoT, Ind. equip. = Industrial equipment, ✓ = Available, ✗ = Not available.

**Table 2.** Overview of related work (part 2).

Ref.	Model	Applications	Sector	ML	DL	I4.0	RT	Gen.	MM
[36,39,43]	DNN	IIoT, network, 5G, 6G	14.0	✓	✓	✓	✓	✗	✗
[59,65]		Photovoltaic, aviation	Energy, transport	✓	✓	✗	✗	✗	✗
[38,77]	GAN	Fiber optic, liquid rocket engines	14.0, aerospace	✓	✓	✓	✗	✗	✗
[34]	EML	Power transformer IoT	14.0	✓	✗	✓	✓	✗	✗
[29]	TL	PdM	14.0	✓	✗	✓	✓	✗	✗
[32,33]		IoT robotics, blockchain IoT	14.0	✓	✗	✓	✗	✗	✗
[30]	NN	3D printing	14.0	✓	✓	✓	✓	✗	✗
[64]		Aviation	Transportation	✓	✓	✗	✗	✗	✗
[26,35]	1D-CNN	Smart factory, bearing faults, power transformer IoT	14.0	✓	✓	✓	✓	✗	✗
[41]	GNN	6G	14.0	✓	✓	✓	✓	✗	✗
[82]		Tracing	Software system	✓	✓	✗	✗	✗	✗
[60,80]	LLM GPT	Power building, execution traces	Energy sector, software system	✓	✓	✗	✗	✗	✗
[22]		HMI	Industrial equipment	✓	✓	✓	✗	✓	✗

**Table 2.** Cont.

Ref.	Model	Applications	Sector	ML	DL	14.0	RT	Gen.	MM
[19,21,46,66,68]		SCADA, HMI, power plant, railway, autonomous ships, ship shaft systems, software systems	Industrial equipment, power plants, transportation	✗	✗	✗	✓	✗	✗
[18,47,69]	Other Models	Dissolved oxygen sensor, thermal power plants	Industrial equipment, power plants	✗	✗	✗	✗	✗	✗
[81,83]		Software defect identification	Software system	✓	✗	✗	✗	✗	✗

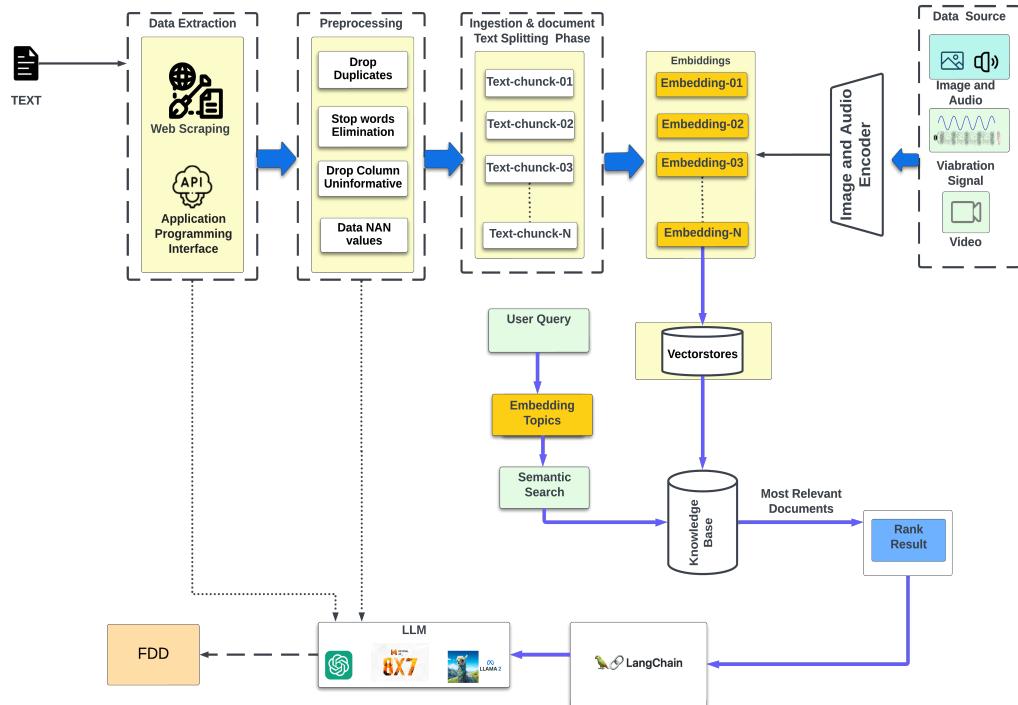
Note: DNN = deep neural network, GAN = generative adversarial network, EML = extreme machine learning, TL = transfer learning, NN = neural network, 1D CNN = one-dimensional convolutional neural network, GNN = graph neural network, LLM GPT = large language model generative pre-trained transformer, PdM = predictive maintenance, IoT = Internet of Things, IIoT = industrial IoT, ✓ = Available, ✗ = Not available.

### 3. Methodology and Design

In this section, the methodology and design of the proposed multimodal LLM-based FDD system, which harnesses the power of multimodal architectures and LLMs, is discussed. The integration of these technologies aims to offer tailored and insightful diagnostic support solutions in the context of the I4.0 transformation and diagnostics for industrial systems. This section contains two main subsections: the design and dataset acquisition.

#### 3.1. A Multimodality Framework and Architecture

Figure 4 presents the proposed MM LLM-based FDD architecture. This novel approach integrates multimodal data sources and advanced ML techniques, mainly focusing on LLMs. The MM-LLM-based FDD framework proposes stages that enable effective FDD in complex systems, which is promising for revolutionizing the industrial domain.



**Figure 4.** MM-LLM-based FDD system architecture showcasing data extraction, preprocessing, and semantic search processes.

The initial stage involves data sourcing and the collection of diverse data types, including images, audio, vibration signals, video, and text. These multimodal data were

extracted via web scraping and application programming interfaces (APIs). LLMs can be utilized for data collection and processing.

In the preprocessing stage, the extracted data undergo various transformations to improve their quality and prepare them for further analysis. This includes drop duplication, stop word elimination, deep column transformations, and the handling of missing values. LLMs can be effectively utilized to enhance these tasks and improve the accuracy of duplicate removal, context-aware stop word elimination, and the intelligent handling of missing data.

The ingestion phase involves embedding techniques for converting text chunks into numerical representations that capture semantic or similarity relationships. These embeddings facilitate the effective utilization of LLMs in subsequent FDD stages. The knowledge base incorporates keywords, topics, embedding topics, and semantic or similarity search capabilities. Leveraging LLMs such as Llama 2 [84], GPT [85], and Mistral [86], the framework extracts relevant information from embedded text chunks, enabling the identification of key insights and patterns for FDD.

The FDD component integrates state-of-the-art techniques such as LangChain [87] or Llamaindex [88], a framework for building applications with language models. These techniques, which work in harmony with LLMs, can accurately detect and diagnose faults within the system. The accuracy and relevance of FDD components were assessed. Unsatisfactory results lead to iterative refinement, which loops back to the data-sourcing step, whereas satisfactory results are saved for decision making and fault management. This feedback mechanism guarantees the framework's long-term efficacy and ongoing improvement. The proposed framework offers generality, flexibility, and scalability to various domains. Its integration of advanced preprocessing, a comprehensive knowledge base, and multimodality LLM ensures adaptability to diverse fault scenarios. This design significantly improves the accuracy and performance. With the LLM-based approach, a substantial improvement can be obtained over traditional methods.

### 3.2. Dataset Acquisition for LLM-Based FDD System

Our multimodal FDD system utilizes a comprehensive data acquisition approach that integrates diverse fault data, contextual information, and enriched technical knowledge. The dataset includes raw fault data, operational parameters, environmental conditions, maintenance records, and expert insights. Our data acquisition strategy encompasses two main types of sources.

#### 3.2.1. Siemens Industry Online Support Forum

This primary source provides rich discussions on various Siemens products [89], including the SIMATIC S5/STEP 5, SIMATIC 505, and SINUMERIK CNC systems. We extracted 240,000 rows across 22 columns from the SIMODRIVE converter system and SIMATIC S7 series, significantly improving the knowledge base of the FDD system.

#### 3.2.2. Diverse Industrial Equipment Manufacturers

To ensure comprehensive training data diversity, we collected fault codes, descriptions, and corrective actions from multiple industrial control system manufacturers. As listed in Table 3, the dataset comprises 1398 fault records gathered from major manufacturers, including Allen Bradley Studio 5000 Logix Designer (420 records), ABB Symphony Control (315 records), CLICK PLUS PLC (298 records), and Yokogawa Centum (365 records). These records encompass a wide range of industrial control system faults, from basic controller errors to complex system diagnostics. Each fault record contains detailed information, including fault codes, diagnostic procedures, corrective actions, equipment model specifications, and required technical expertise levels. The data collection spans various fault types, including controller faults, I/O faults, program faults, motion faults, CPU diagnostics, communication errors, system faults, and analog input issues. Prior to integration, the data underwent standardization preprocessing to ensure consistent terminology and format

across different manufacturers. This comprehensive multi-vendor dataset strengthens our LLM-based framework's capability to handle diverse industrial environments and fault scenarios effectively.

**Table 3.** Data preprocessing and diversity.

Sr.	Device	Data Acquisition Method	Size
1	Siemens online support forum	Web scrubbing	19,335 Pg
2	Diverse industrial equipment manufacturers	Manuals, fault lists	1398 Faults

### 3.3. Data Preprocessing and Diversity

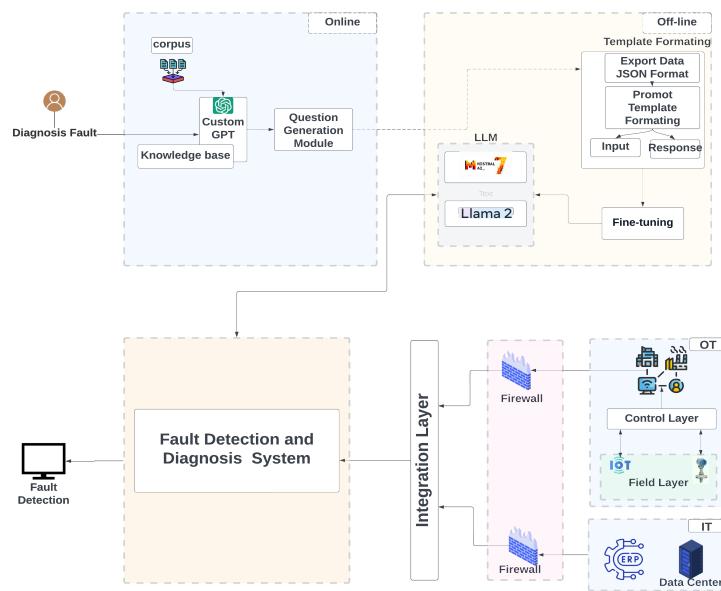
Data preprocessing involves multiple stages to ensure data quality and standardization. The initial preprocessing focused on redundancy removal through automated duplicate detection, eliminating repeated fault discussions, and similar troubleshooting procedures across our dataset. We implemented fault code standardization to ensure consistency across different data sources, which was particularly crucial for Siemens support forum data, where multiple format variations existed for the same fault codes. This standardization process maintained the technical accuracy while creating a uniform format for fault identification and classification.

The key information extraction phase concentrated on identifying critical troubleshooting steps, diagnostic procedures, and corrective actions. We implemented semantic parsing to structure the information hierarchically based on fault severity, complexity, and resolution requirements. The extracted data underwent validation against the original manufacturer documentation to ensure the accuracy and completeness of the technical information.

To enhance data quality, we utilized LLMs for contextual data cleaning and semantic categorization of fault descriptions. This process standardized technical terminology across different manufacturers while maintaining manufacturer-specific diagnostic information. The preprocessing ensured that each fault record, regardless of its source, contained a unique fault identifier, comprehensive fault description with symptoms, required diagnostic steps, and detailed corrective actions with verification procedures. This framework was consistently applied across all manufacturer data, including Allen Bradley, ABB, CLICK, and Yokogawa, preserving the technical integrity of manufacturer-specific characteristics while ensuring format uniformity essential for effective industrial fault diagnosis.

### 3.4. Hybrid LLM-Based Architectures for Online and Offline FDD Systems

Figure 5 shows a hybrid LLM-based FDD architecture that optimizes FDD systems by integrating online and offline models and improving diagnostic capabilities, data privacy, and cybersecurity. The system processes unstructured data, generates embeddings, and utilizes a custom GPT model for query handling. Semantic indexing facilitates efficient knowledge retrieval. Offline LLMs, including Llama 2 and Mistral 7B, are fine-tuned with synthetic data to improve performance and address security concerns. The integration layer incorporates real-time data from various industrial sources, which enriches the diagnostic process. The architecture manages IT and operational technology (OT) complexities and ensures secure data flow through firewalls. This hybrid approach leverages real-time data integration and offline LLM processing, advancing the FDD system's capability to handle real-time data and enhancing fault detection accuracy in I4.0 environments.



**Figure 5.** Hybrid LLM-based FDD architecture featuring online and offline modules with OT-IT integration and secure data flow.

### 3.5. Security Framework and Implementation

In the context of Industry 4.0 environments, security considerations for LLM-based FDD systems are crucial due to the sensitive nature of industrial data. A comprehensive security framework should address data protection through multiple layers. The first layer should focus on API security, ensuring secure authentication and encryption for interactions with models like GPT-4-Preview, alongside data preprocessing mechanisms to anonymize sensitive information before transmission. This approach helps protect proprietary industrial information while maintaining diagnostic accuracy.

Network security represents another critical aspect, particularly the segregation of operational technology (OT) and information technology (IT) networks. This separation, achieved through demilitarized zones (DMZ) and industrial firewalls, helps prevent unauthorized access while enabling secure data exchange for diagnostic purposes. Traffic monitoring and anomaly detection systems can provide additional security layers without impacting system performance.

Knowledge base protection forms the third crucial component, requiring the secure storage of diagnostic information and comprehensive audit logging. The framework should support encrypted storage and secure update mechanisms while ensuring system accessibility for authorized users. This aspect is particularly important for maintaining the integrity of fault diagnostic data while enabling efficient system operation.

## 4. Evaluation

In this section, a comprehensive approach for assessing the performance and effectiveness of the proposed LLM-based FDD system is outlined. The metrics used for the evaluation include inference relevance accuracy, response time, and a custom performance score (PS) that combines accuracy and speed. This section describes our methodology for categorizing diagnostic questions into simple, medium, and complex types, thereby providing a nuanced understanding of a system's capabilities across various fault scenarios. Additionally, the comparative process analysis between different LLMs, including ChatGPT4, Mixtral 8x7B, Llama 2, and our proposed FDD GPT-4-Preview model, is explained.

Moreover, a comparative analysis of conventional FDD and LLM-based FDD was applied. In this subsection, traditional FDD methods against the LLM-based approach are evaluated using the same performance metrics. The comparative analysis helps emphasize the advancements and potential improvements offered by integrating LLMs into FDD

systems. By analyzing both approaches, we aimed to determine the specific benefits and limitations of LLM-based systems over conventional methods.

Additionally, the effectiveness of handling hybrid architecture performance in real-time and offline processing is evaluated. This includes assessing how well the system manages the immediate fault diagnosis needs versus a more detailed, offline analysis. In the section, the manner in which we assessed the system's adaptability to different industrial contexts and its potential impact on I4.0 implementations is also addressed. By detailing these evaluation methods, we aimed to provide a clear and rigorous framework for understanding the strengths and limitations of the proposed FDD system.

#### 4.1. Comparative Analysis of Different Models

In this section, four LLMs are thoroughly examined, all of which were implemented into the FDD system: ChatGPT4, Mixtral 8x7B, Llama 2, and our proposed model using GPT-4-Preview. Our assessment considered various metrics, which include inference relevance, response time, and PSs. Using these metrics, we systematically compared LLMs via quantitative measurements and evaluated their strengths and weaknesses.

##### 4.1.1. Assessment Metrics

This study selected performance evaluation indicators to comprehensively assess LLM use in FDD systems. The proposed metrics included inference relevance categories, response time measurements, and a composite PS, providing a balanced assessment of accuracy and efficiency. The inference relevance categories (irrelevant, generic, relevant, precise) enable the detailed analysis of output quality, whereas the response time metrics address the critical aspect of real-time performance. The PS formula, which combines response quality and speed with adjustable weights for different response categories, allows for flexible prioritization based on specific industrial requirements, demonstrating the adaptability of our research to various contexts. We employed three key metrics to evaluate the model performance.

##### Inference Relevance

There are four categories of relevance metrics in fault diagnosis systems: relevant, irrelevant, generic, and precise. Relevant responses accurately address the specifics of the fault query, effectively understand the fault context, and provide solutions from a knowledge base. Irrelevant responses fail to address the fault, indicating the misinterpretation or retrieval of unrelated information. Precise responses deliver detailed and specific solutions that directly address faults using the targeted information. Although correct, generic responses lack the specificity of precise responses and offer broader and more general information. Equations (1)–(4) are employed to calculate the average inference relevance accuracy by summing the counts for each response type and dividing it by the total number of responses.

$$IR_{generic} = \frac{\sum_{i=1}^m Response_{generic_i}}{m} \quad (1)$$

$$IR_{irrelevant} = \frac{\sum_{i=1}^m Response_{irrelevant_i}}{m} \quad (2)$$

$$IR_{relevant} = \frac{\sum_{i=1}^m Response_{relevant_i}}{m} \quad (3)$$

$$IR_{precise} = \frac{\sum_{i=1}^m Response_{precise_i}}{m} \quad (4)$$

The average value for generic inference relevance, denoted as  $IR_{generic}$ , is calculated using (1), summing all individual counts of generic responses, represented as  $\sum_{i=1}^m Response_{generic,i}$ , where  $Response_{generic,i}$  refers to each generic response generated

by the model and  $m$  is the total number of results in the dataset. The average value for irrelevant inference relevance, denoted as  $IR_{irrelevant}$ , was calculated using (2), where  $\sum_{i=1}^m Response_{irrelevant,i}$  represents the summation of all individual irrelevant responses from the model, and  $m$  is the total number of results in the dataset. Each  $Response_{irrelevant,i}$  refers to an individual irrelevant response generated by the model. This calculation measures the tendency of the model to generate irrelevant responses across the dataset. The average value for the relevant inference relevance, represented as  $IR_{relevant}$ , was determined using (3), where  $\sum_{i=1}^m Response_{relevant,i}$  is the total sum of all relevant responses provided by the model, and  $m$  is the total number of results in the dataset. Each  $Response_{relevant,i}$  corresponds to the individual relevant response generated by the model. This formula effectively quantifies the ability of a model to produce responses that accurately address specific fault queries in a dataset. Equation (4) calculates the precise answer, in which  $\sum_{i=1}^m Response_{precise,i}$  is the total sum of all precise responses provided by the model, and  $m$  is the total number of results in the dataset. Each  $Response_{precise,i}$  refers to the precise response of the model.

### Performance Score

We can establish a performance metric that considers response time and context relevance to evaluate performance. One way to do this is by calculating a PS for each model using (5):

$$PS = \frac{1}{RT_{avg}} \times (W_{irrelevant} \times Irrelevant\% + W_{relevant} \times relevant\% + W_{generic} \times Generic\% + W_{precise} \times Precise\%) \quad (5)$$

A higher PS indicates better performance, which reflects faster average response times and higher relevance in the given context, with a focus on precise and relevant categories. The models with higher PS values were both efficient and effective in their responses. They are also relevant, which makes them ideal for real-time applications, where speed and accuracy are crucial. Conversely, a lower PS value indicates poorer performance, which is characterized by slower average response times and a less relevant context, as evidenced by the higher percentages of irrelevant and generic categories. Such models are less efficient and less relevant and, thus, more suitable for non-real-time applications.

### Response Time Analysis

Our model performance evaluation analyzed the maximum and minimum response times across all the models. These metrics provide insights into the extremes of response latency, which helps us to understand the range of performance variability. To quantify these measures, (6) and (7) were employed:

$$Max_{RT} = \max(RT) \quad (6)$$

$$Min_{RT} = \min(RT) \quad (7)$$

$Max_{RT}$  represents the maximum response time observed across all models, and  $Min_{RT}$  denotes the minimum response time. Function  $\max(RT)$  returns the highest value in response times, whereas  $\min(RT)$  determines the lowest value. These equations allow us to establish the upper and lower bounds of the response time performance, which is crucial for the assessment of the efficiency and consistency of model responses.

#### 4.1.2. Fault Categorization

We suggest organizing diagnostics to assess the troubleshooting requirements of different LLM industrial equipment models, incorporating both complexity levels and token utilization patterns. The questions were divided into three levels based on their computational

demands and token consumption: simple (512–650 tokens), medium (845–1100 tokens), and complex (1280–2048 tokens). Simple diagnostic questions involve basic tests or common issues that can be identified and resolved with minimal technical expertise, such as addressing fault code T18, typically requiring around 580 tokens for processing. Medium diagnostic questions necessitate a more detailed examination, averaging 950 tokens, and potentially involve several steps or specific system knowledge, as observed with fault code T17, where specific runtime diagnostics must be conducted initially. Complex diagnostic questions entail extensive troubleshooting, consuming up to 2048 tokens, requiring an intricate understanding of the system's inner workings and multiple diagnostic procedures. For instance, diagnosing a CPU fault in a Siemens S7-1500 system, as detailed in Table 4, involves comprehensive diagnostic procedures to isolate and resolve the issue, which reflects the high level of complexity involved and demands maximum token allocation. This categorization enables efficient resource allocation and queue prioritization within API constraints while maintaining diagnostic accuracy across different complexity levels.

**Table 4.** Fault types and complexities.

SN	Question	Type
1.	Fault code T17: runtime diagnostics to check first?	Medium
2.	T17:C34 fault: steps to resolve?	Medium
3.	T04:C24 fault: methods to identify correction?	Medium
4.	How to proceed with T18:C08 fault?	Simple
5.	Best way to overcome T04:C31 fault?	Simple
6.	Siemens S7-1500 CPU fault: diagnostic procedures?	Complex
7.	ABB 800XA: failed to activate and connect account dialog. Solution?	Medium
8.	Stardom task overlap error: causes and scheduling adjustments?	Complex
9.	Yokogawa I/O memory usage exceeding limits: optimization strategies?	Complex
10.	Symphony configuration error: steps to verify and correct?	Complex

SN = Serial Number. Questions are categorized as simple, medium, or complex based on the required knowledge depth.

#### 4.2. Comparative Analysis Between Conventional and LLM-Based FDD

In this subsection, the methodology for comparing a conventional FDD system based on traditional ML algorithms with the proposed LLM-based FDD system is outlined. The comparison is grounded in quantitative metrics, including accuracy, precision, recall, F1 score, and computational efficiency. To compare their effectiveness in classifying fault clusters, we utilized three models, namely, logistic regression (LR), RF, and a neural network (NN). The conventional ML-based FDD system was trained on Siemens PLC semantic S7 datasets with five fault categories (CPU/processor issues, power issues, input/output issues, communication issues, and other issues). To measure accuracy, we assessed the proportion of correct predictions made using the model out of the total number of predictions, providing an overall perspective on its performance across all fault categories. It was calculated using (8) in which  $TP$  = true positives,  $TN$  = true negatives,  $FP$  = false positives, and  $FN$  = false negatives.

$$\text{Accuracy} = \frac{(TP + TN)}{(TP + TN + FP + FN)} \quad (8)$$

To measure precision, we evaluated the accuracy of positive predictions by focusing on the ratio of true positives to the total number of positive predictions, which include both true and false positives. The precision was calculated using (9).

$$\text{Precision} = \frac{(TP)}{(TP + FP)} \quad (9)$$

When evaluating the effectiveness of a model, recall (or sensitivity) measures its ability to correctly determine all relevant instances of a fault. This metric captures the ratio of true

positive predictions to the total number of actual positives, including both true and false negatives. This was calculated using (10).

$$\text{Recall} = \frac{(TP)}{(TP + FN)} \quad (10)$$

The data and resource requirements for LLM-based and traditional FDD methods vary significantly, as shown in Table 5. LLM-based approaches can leverage large volumes of unstructured data from diverse sources, excelling at processing natural language. At the same time, traditional methods are typically dependent on structured numerical data from sensors and predefined fault codes. LLMs are pre-trained on vast general knowledge and then fine-tuned for specific tasks, reducing domain-specific training data needs. By contrast, traditional ML models necessitate extensive labeled training data for each fault type and system. LLMs demand significant computational power, especially for training and real-time inference, whereas conventional methods are generally less computationally intensive. Nevertheless, LLMs offer greater adaptability to new fault types or systems through fine-tuning or prompt engineering, whereas traditional models often require retraining from scratch. LLMs provide natural language explanations, enhancing interpretability, and can integrate information across modalities more seamlessly.

**Table 5.** Comparative analysis of key aspects in LLM-based and conventional FDD methods.

Aspect	LLM-Based FDD	Conventional FDD
Data type	Unstructured data (e.g., text, natural language, and diverse sources)	Structured numerical data (e.g., sensor data and fault codes)
Data processing	Excels at natural language processing	Focused on predefined fault codes
Training approach	Pre-trained on general knowledge, fine-tuned for specific tasks	Requires extensive labeled data for each fault type
Computational power	High, especially for training and real-time inference	Generally less computationally intensive
Adaptability	Highly adaptable through fine-tuning or prompt engineering	Often requires retraining from scratch for new faults
Interpretability	Provides natural language explanations	Typically provides structured outputs with limited explanation
Modalities	Seamlessly integrates data across multiple modalities (e.g., text, audio, and images)	Primarily focused on specific, predefined data types
Scalability	Highly scalable to new domains and fault types	Less scalable requires significant effort for new domains

## 5. Results and Discussion

In this section, the findings from our comparative study of four LLMs, namely, ChatGPT-4, Mixtral 8x7B, Llama 2, and our proposed FDD using the GPT-4-Preview model, are presented and analyzed. We evaluated their performance in the context of FDD applications, focusing on three key metrics: inference relevance, response time, and overall PS. This section also assesses the effectiveness of our proposed hybrid architecture in creating a dynamic and responsive FDD system in both online and offline systems. Moreover, we compared conventional FDD methods and our LLM-based approach in terms of accuracy and response time, underscoring the advancements and potential benefits of employing large language models for FDD.

### 5.1. Comparative Analysis of LLMs: Assessment Metrics and Practical Use Cases

We conducted a comparative analysis of various LLMs, which include ChatGPT 4, Mixtral 8x7B, Llama 2, and the proposed FDD GPT-4-Preview. The core objective was to evaluate the efficacy of these models in generating contextual questions derived from a specified knowledge base. Our methodology involved the systematic injection of 10 uniquely constructed questions into each LLM and the proposed models to assess their ability to produce relevant, irrelevant, precise, and generic responses.

### 5.1.1. A Comparison of Different LLMs

In the analysis of the comparative language models, as shown in Tables 6 and 7, the GPT-4-Preview showed superior performance, which generates correct answers with 60% precision. This marks a significant improvement in the specificity of the responses compared with other models, as exemplified in Table 8(a), where a precise response is provided for fault code T04:C31. ChatGPT 4 balanced specific and vague responses at 40% each but lacked precision, as shown in Table 8(b), in which an irrelevant response is given for the same fault code. Llama 2 and Mixtral 8x7B models demonstrated the quickest response times but suffered in contextual relevance, with the highest rate of irrelevant responses at 50%. Llama 2 had the highest generic response rate of 40%, which indicates a more generalized understanding of prompts, as shown in Table 9(b). GPT-4-Preview had a lower rate of irrelevant responses (20%), which suggests better contextual understanding. Although Mixtral 8x7B provided the fastest response times, this speed may come at the cost of contextual accuracy, although it still manages to provide relevant responses, as shown in Table 9(a). Conversely, GPT-4-Preview offered more concise and accurate responses with moderate delay times, which indicates precise comprehension of complex inputs.

**Table 6.** Comparative analysis of response time and inference relevance across different LLM models.

Question	ChatGPT 4		Llama 2		Mixtral 8x7B		FDD GPT-4-Preview	
	RT (s)	IR	RT (s)	IR	RT (s)	IR	RT (s)	IR
Q1	44.04	Irrelevant	10.79	Irrelevant	3.37	Irrelevant	12.7	Precise
Q2	23.58	Generic	12.04	Irrelevant	2.61	Irrelevant	15.29	Precise
Q3	26.75	Irrelevant	15.56	Irrelevant	5.65	Irrelevant	17.17	Precise
Q4	22.31	Irrelevant	17.66	Irrelevant	2.24	Irrelevant	21.27	Precise
Q5	34.66	Irrelevant	18.28	Irrelevant	2.58	Irrelevant	13.34	Precise
Q6	41.46	Relevant	22.93	Relevant	3.81	Relevant	17.71	Relevant
Q7	22.38	Relevant	7.8	Generic	3.2	Relevant	14.68	Precise
Q8	42.18	Generic	10.10	Generic	5.56	Generic	19.88	Generic
Q9	30.04	Generic	12.95	Generic	5.96	Generic	21.33	Irrelevant
Q10	38.52	Generic	13.85	Generic	4.16	Relevant	17.64	Irrelevant

**Table 7.** Time response comparison of different LLM models.

Model	$IR_{irrelevant}$ %	$IR_{relevant}$ %	$IR_{generic}$ %	$IR_{precise}$ %	$Max_{RT}$ s	$Min_{RT}$ s
ChatGPT 4	40	40	20	0	44.04	22.31
Llama 2	50	10	40	0	15.56	7.8
Mixtral 8x7B	50	30	20	0	5.65	2.24
FDD GPT-4-Preview	20	10	10	60	21.33	12.7

**Table 8.** Comparison of LLM responses to fault code T04:C31 (part 1). (a) GPT-4-Preview response, (b) ChatGPT 4 response.

(a)

Q5: For fault code T04:C31, what is the best way to overcome it?

**Response:** The best way to overcome fault code T04:C31 is to ensure that the parameters of the JSR (Jump to SubRoutine) instruction match those of the associated SBR (SubRoutine) or RET (Return from SubRoutine) instruction. If you pass too many parameters, the extra ones are ignored without any error, but you should pass the appropriate number of parameters to avoid the fault

**Table 8.** Cont.

<b>Response time:</b> 13.34 s <b>IR:</b> Precise
(b)
<b>Q5:</b> For fault code T04:C31, what is the best way to overcome it?
<b>Response:</b> To effectively address fault code T04:C31 in an ABB 800xA system, you should follow a structured troubleshooting process. This process will help identify and resolve the issue associated with this specific fault code. Here are the recommended steps: Consult the System Documentation: Begin by referring to the ABB 800xA system's documentation or fault code manual. This documentation often provides detailed information about specific fault codes, including T04:C31, and may offer initial troubleshooting guidance or describe the typical causes of the fault. [Continue...]

**Response time:** 34.66 s    **IR:** Irrelevant

**Table 9.** Comparison of LLM responses for Symphony control system (part 2). (a) Mixtral 8x7B response, (b) Llama 2 response.

(a)
<b>Q10:</b> Symphony control system reports a configuration error; what steps should be taken to verify and correct the configuration settings?
<b>Response:</b> If a Symphony control system reports a configuration error, here are the steps to verify and correct the configuration settings: Check the system status: Check the system status in the Symphony Plus operator interface or in the engineering workstation. Look for any error messages or fault codes that may indicate the cause of the configuration error. Check the configuration files [... Continue ...]
<b>Response time:</b> 4.16 s <b>IR:</b> Relevant
(b)
<b>Q10:</b> Symphony control system reports a configuration error; what steps should be taken to verify and correct the configuration settings?
<b>Response:</b> If a Symphony control system reports a configuration error, it is important to take steps to verify and correct the configuration settings to ensure that the system is operating correctly and efficiently. Here are some steps that can be taken: 1. Review system logs: Review the system logs to identify the source of the configuration error. The logs can provide information about the specific configuration issue and the affected components. [... Continue ...]
<b>Response time:</b> 13.85 s <b>IR:</b> Generic

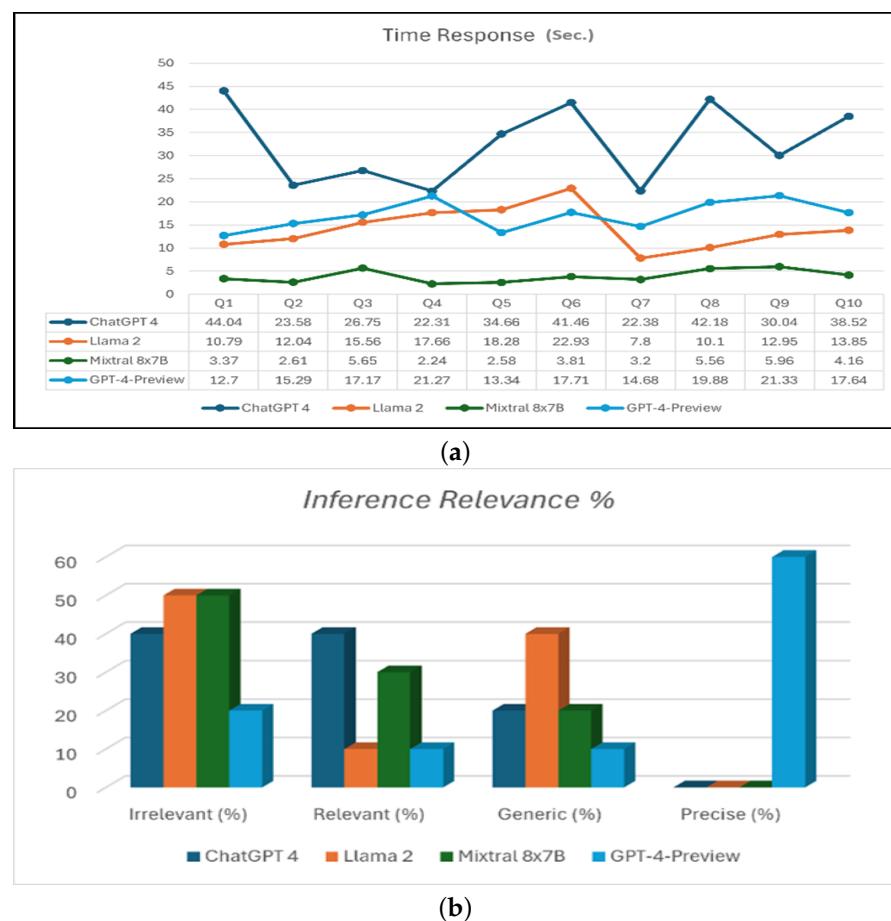
The response time analysis revealed significant variations. ChatGPT 4 presented a maximum response time ( $Max_{RT}$ ) of 44.04 s and a minimum response time ( $Min_{RT}$ ) of 22.31 s, which suggests deeper analysis but slower processing. Llama 2 improved this, with a ( $Max_{RT}$ ) of 22.93 s and ( $Min_{RT}$ ) of 7.8 s, showing enhanced efficiency. Mixtral 8x7B was notably fast, with a ( $Max_{RT}$ ) of only 5.96 s and ( $Min_{RT}$ ) of 2.24 s, showing rapid response generation. GPT-4-Preview balanced comprehensiveness and speed with a ( $Max_{RT}$ ) of 21.33 s and ( $Min_{RT}$ ) of 12.7 s. Figure 6 illustrates response times, inference relevance, and PSs across 10 questions (Q1–Q10) for various LLMs. ChatGPT 4 consistently showed the highest response times, often exceeding 40 s, mainly for Q1, Q6, and Q8. Mixtral 8x7B maintained the lowest response times, staying below 6 s. Llama 2 and GPT-4-Preview exhibited moderate response times, with Llama 2 typically performing faster than GPT-4-Preview.

Table 6 summarizes the comparative analysis based on inference relevance categories and response time ranges. GPT-4-Preview showed higher precision in inference relevance, whereas Mixtral 8x7B excelled in minimum response time. Nevertheless, Mixtral 8x7B's speed did not translate to inference relevance, as it shared the highest percentage of irrelevant responses with Llama 2. This suggests that although Mixtral 8x7B and Llama 2 process queries rapidly, GPT-4-Preview provides more accurate responses at the cost of slightly longer processing times. As a result of PS calculations for all models, ChatGPT 4 (PS = 0.804) has the lowest PS, which indicates that it may be slower and less relevant than the other LLMs. Llama 2 (PS = 0.935) performs better than ChatGPT 4 but is not the best among the given models. Mixtral 8x7B (PS = 1.504) has a higher PS, which indicates that it is more efficient and relevant in its responses than the other two models. GPT-4-

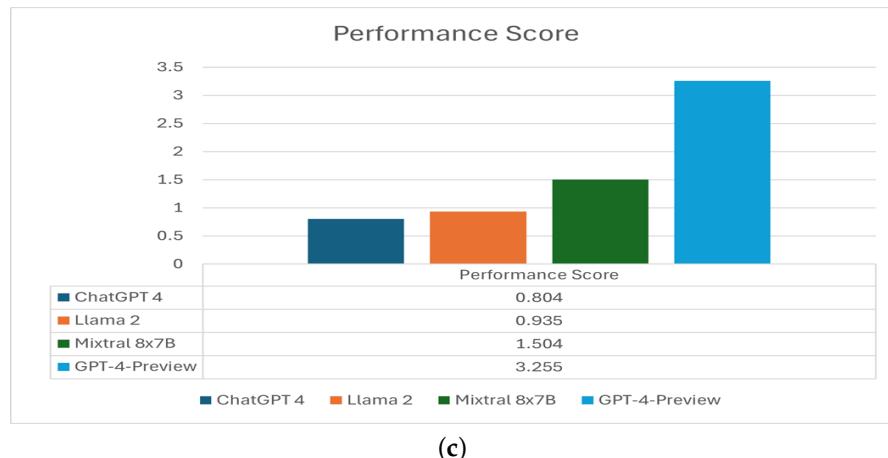
Preview (PS = 3.255) demonstrated the highest PS, which signifies superior efficiency and relevance in its responses. These results are summarized in Figure 6c, which illustrates the LLM PSs.

In terms of scalability, our analysis of token consumption across different query complexities reveals that GPT-4-Preview maintains consistent performance despite the increased load. Simple queries (512–650 tokens) process in 13.77 s; medium queries (845–1100 tokens), in 17.38 s; and complex queries (1280–2048 tokens), in 21.33 s, showing a linear scaling pattern with the token count. This predictable scaling enables effective resource planning and queue management in industrial applications.

The feasibility of implementing the LLM-based FDD method in the industrial domain is substantiated by the results, both from the implementation and computational perspectives. Through existing APIs and open source frameworks, the integration of models such as GPT-4-Preview, ChatGPT 4, Mixtral 8x7B, and Llama 2 is achievable. The observed performance metrics, particularly the superior precision of GPT-4-Preview, indicate that model customization via fine-tuning, enriching the knowledge base, or prompt engineering is feasible and highly effective for FDD tasks. The range of response times (12.7–21.33 s) showed a trade-off between accuracy and computational efficiency. The PS metric exemplifies this trade-off, in which GPT-4-Preview's higher PS of 3.255 justifies a suitable mode selection for critical industrial applications.



**Figure 6. Cont.**



(c)

**Figure 6.** (a) Time response comparison of different LLM models. (b) Inference relevance comparison of LLM models. (c) Performance score comparison of LLM models.

#### 5.1.2. A Comparative Review of Results: LLM Versus Conventional FDD

In our quantitative analysis, we evaluated the performance of conventional and LLM-based FDD models across several key metrics. In Table 10, the three traditional ML models (LR, RF, and NN) are compared with our proposed FDD GPT-4-Preview model. The results reveal a notable trade-off between response time and accuracy.

**Table 10.** Model response times and accuracy comparisons.

Sr.	Model	Response Time (s)	Accuracy (%)
1	GPT-4-Preview	9.3	96.3
2	RF	0.3445	90.36
3	LR	0.0526	86.43
4	NN	2.7168	86.07

RF = random forest, LR = logistic regression, NN = neural network.

Conventional models showed superior speed, with LR leading at 0.0526 s, RF at 0.3445 s, and NN at 2.7168 s. By contrast, our GPT-4-Preview model requires 9.3 s for response. However, this increased processing time was offset by the model's exceptional accuracy of 96.3%, surpassing that of the next best performer, RF, by 5.94 percentage points.

Examining the confusion matrices provides further insight into the strengths and weaknesses of each model. As shown in Figure 7a, the NN model demonstrates strong performance in identifying "Other Issues" (97.75% accuracy) and moderate success with "I/O issues" (40% accuracy). However, it struggles with "Communication Issues" (27.27% accuracy) and "Power Issues" (25% accuracy), which indicates potential areas for improvement.

Figure 7b illustrates the RF performance model, which shows improved results across most categories compared to the NN. It excels in identifying "I/O issues" (70% accuracy) and "Power Issues" (66.67% accuracy). The model maintained high accuracy for "Power Issues" (97.75%) while showing significant improvement in "CPU/Prc. Issues" (72% accuracy).

The LR model, as shown in Figure 7c, exhibits mixed results. It performs exceptionally well in identifying "Power Issues" (99.10% accuracy) but struggles with "Power Issues" (16.67% accuracy) and "CPU/Prc. Issues" (48% accuracy). Its performance on "I/O issues" (40% accuracy) and "Communication Issues" (36.36% accuracy) is moderate.

Neural network confusion matrix						
TARGET OUTPUT	Comm. issues	I/O issues	Power issues	CPU/pro. issues	Other issues	Sum
Comm. issues	3 1.07%	0 0.00%	1 0.36%	1 0.36%	6 2.14%	11 27.27% <b>72.73%</b>
I/O issues	0 0.00%	4 1.43%	1 0.36%	1 0.36%	4 1.43%	10 <b>40.00%</b> <b>60.00%</b>
Power issues	0 0.00%	0 0.00%	3 1.07%	0 0.00%	9 3.21%	12 <b>25.00%</b> <b>75.00%</b>
CPU/pro. issues	0 0.00%	0 0.00%	0 0.00%	14 5.00%	11 3.93%	25 <b>56.00%</b> <b>44.00%</b>
Other issues	0 0.00%	0 0.00%	3 1.07%	2 0.71%	217 77.50%	222 <b>97.76%</b> <b>2.25%</b>
<b>Sum</b>	<b>3 100.00% 0.00%</b>	<b>4 100.00% 0.00%</b>	<b>8 37.50% 62.50%</b>	<b>18 77.78% 22.22%</b>	<b>247 87.85% 12.15%</b>	<b>241 / 280 86.07% 13.93%</b>

(a)

Random forest confusion matrix						
TARGET OUTPUT	Comm. issues	I/O issues	Power issues	CPU/pro. issues	Other issues	Sum
Comm. issues	3 1.07%	0 0.00%	0 0.00%	1 0.36%	7 2.50%	11 27.27% <b>72.73%</b>
I/O issues	1 0.36%	7 2.50%	0 0.00%	0 0.00%	2 0.71%	10 <b>70.00%</b> <b>30.00%</b>
Power issues	0 0.00%	0 0.00%	8 2.86%	0 0.00%	4 1.43%	12 <b>66.67%</b> <b>33.33%</b>
CPU/pro. issues	0 0.00%	0 0.00%	0 0.00%	18 6.43%	7 2.50%	26 <b>72.00%</b> <b>28.00%</b>
Other issues	1 0.36%	1 0.36%	2 0.71%	0 0.36%	217 77.50%	222 <b>97.76%</b> <b>2.25%</b>
<b>Sum</b>	<b>5 60.00% 40.00%</b>	<b>8 87.50% 12.50%</b>	<b>10 80.00% 20.00%</b>	<b>20 90.00% 10.00%</b>	<b>237 91.66% 8.33%</b>	<b>253 / 280 90.36% 9.64%</b>

(b)

Logistic regression confusion matrix						
TARGET OUTPUT	I/O issues	Comm. issues	Power issues	CPU/pro. issues	Other issues	Sum
I/O issues	4 1.43%	0 0.00%	0 0.00%	0 0.00%	6 2.14%	10 <b>40.00%</b> <b>60.00%</b>
Comm. issues	0 0.00%	4 1.43%	0 0.00%	1 0.36%	6 2.14%	11 <b>36.36%</b> <b>63.64%</b>
Power issues	0 0.00%	0 0.00%	2 0.71%	0 0.00%	10 3.57%	12 <b>16.67%</b> <b>83.33%</b>
CPU/pro. issues	0 0.00%	0 0.00%	0 0.00%	12 4.29%	13 4.64%	26 <b>48.00%</b> <b>52.00%</b>
Other issues	0 0.00%	0 0.00%	2 0.71%	0 0.00%	220 78.57%	222 <b>99.10%</b> <b>0.90%</b>
<b>Sum</b>	<b>4 100.00% 0.00%</b>	<b>4 100.00% 0.00%</b>	<b>4 50.00% 50.00%</b>	<b>13 92.31% 7.69%</b>	<b>255 86.27% 13.73%</b>	<b>242 / 280 86.43% 13.57%</b>

(c)

**Figure 7.** (a) NN confusion matrix. (b) RF confusion matrix. (c) LR confusion matrix.

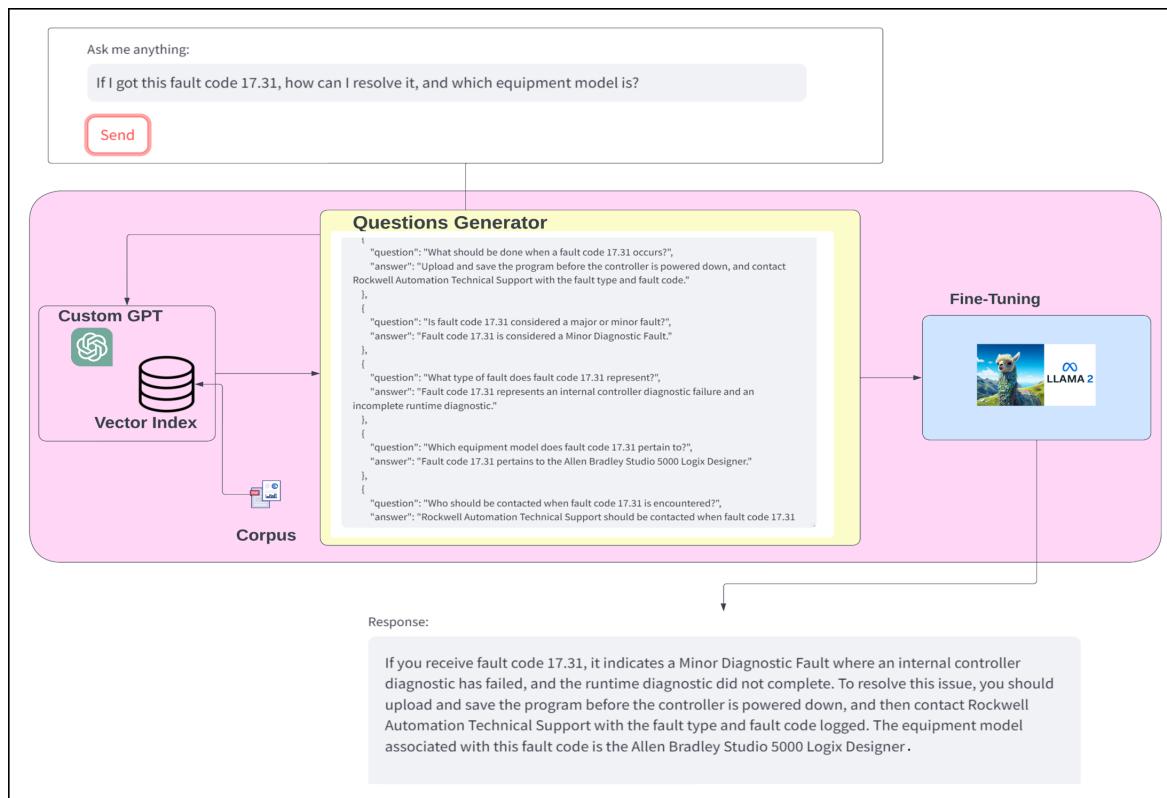
Although these traditional models offer rapid response times, their variable performance across different fault categories highlights the challenges in achieving consistent accuracy across all fault types. Beyond this quantitative measure, the GPT-4-Preview model, despite its longer processing times, offers superior overall accuracy and enhanced explainability. Our adaptable LLM-based approach can address new fault types through fine-tuning or knowledge-based utilization.

Traditional ML models (NN, RF, and LR) require rigorous training, testing, and validation. LLM-based FDD demonstrates superior contextual understanding when compared with conventional models. The combination of high accuracy, interpretability, and adaptability makes the GPT-4-Preview model particularly valuable for complex industrial diagnosis scenarios.

## 5.2. Use Case: Fine-Tuned LLM-Based Multistage FDD Process Workflow

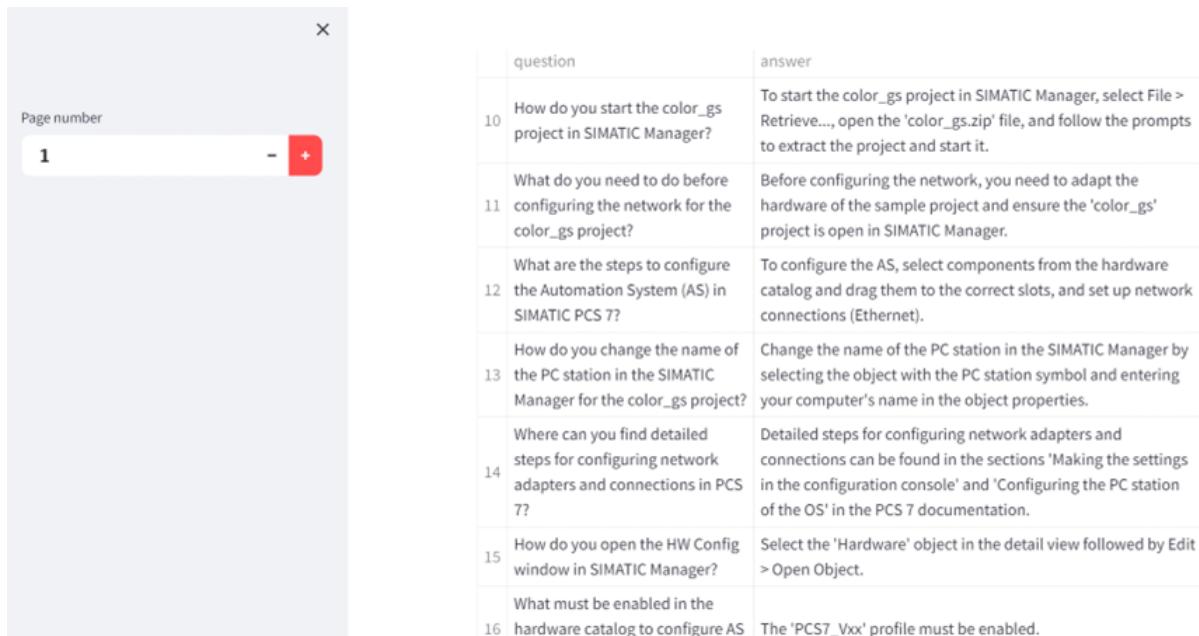
Figure 8 shows the use case of the multistage process workflow to address user queries regarding equipment faults that employ a custom GPT model and Llama 2 to enhance diagnostic precision. In this workflow, a user initiates a query regarding a specific fault code, which is processed by the custom GPT model accessing a vector index constructed from a comprehensive document corpus of unstructured data, such as device manuals and

fault records. The system's question generator module formulates pertinent questions to refine the context and extract detailed information from the knowledge base.



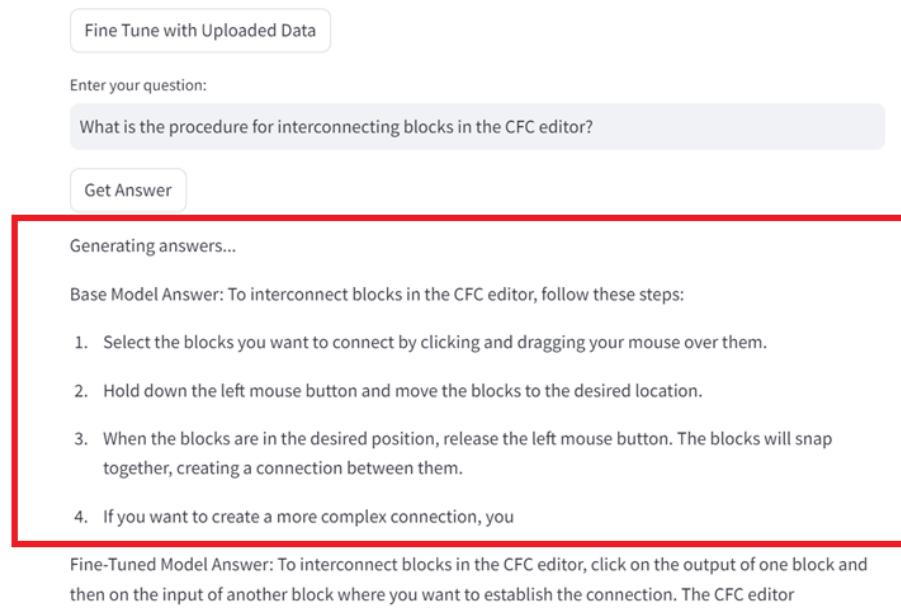
**Figure 8.** Fine-tuned LLM-based workflow for complex fault diagnosis queries.

Then, these refined data were fine-tuned using the Llama 2 model to ensure the accuracy of FDD. The system subsequently generates a detailed response encompassing diagnostic information, recommended resolution steps, and the associated equipment model details. For instance, a query regarding fault code 17.31 for the Siemens STEP device would result in a response that identifies the fault as a minor diagnostic fault, provides steps to resolve the issue, and specifies the related equipment model. This use case demonstrates the integration of advanced AI models into FDD systems, significantly enhancing the accuracy and timeliness of fault diagnostics and recommendations, thereby supporting robust industrial operations. The system generates answers using both the base and fine-tuned models, highlighting the difference in accuracy between them. The process begins when the user enters a query. Custom GPT processes the user's query by accessing the vector index and retrieving relevant information from the corpus, as shown in Figure 8. As part of refining the response, the system includes a question generator module that formulates relevant questions based on the initial user query, ensuring that detailed and relevant information is gathered from the knowledge base. Figure 9 presents a table of questions and answers that are likely utilized as training data for fine-tuning. The system generates answers using both the base and fine-tuned models, emphasizing the difference in accuracy between them. Figure 10 shows the base model answer, irrelevant to the fine-tuned model's response.



question	answer
10 How do you start the color_gs project in SIMATIC Manager?	To start the color_gs project in SIMATIC Manager, select File > Retrieve..., open the 'color_gs.zip' file, and follow the prompts to extract the project and start it.
11 What do you need to do before configuring the network for the color_gs project?	Before configuring the network, you need to adapt the hardware of the sample project and ensure the 'color_gs' project is open in SIMATIC Manager.
12 What are the steps to configure the Automation System (AS) in SIMATIC PCS 7?	To configure the AS, select components from the hardware catalog and drag them to the correct slots, and set up network connections (Ethernet).
13 How do you change the name of the PC station in the SIMATIC Manager for the color_gs project?	Change the name of the PC station in the SIMATIC Manager by selecting the object with the PC station symbol and entering your computer's name in the object properties.
14 Where can you find detailed steps for configuring network adapters and connections in PCS 7?	Detailed steps for configuring network adapters and connections can be found in the sections 'Making the settings in the configuration console' and 'Configuring the PC station of the OS' in the PCS 7 documentation.
15 How do you open the HW Config window in SIMATIC Manager?	Select the 'Hardware' object in the detail view followed by Edit > Open Object.
16 What must be enabled in the hardware catalog to configure AS	The 'PCS7_Vxx' profile must be enabled.

**Figure 9.** Sample questions and answers for fine-tuning the LLM model.



Fine Tune with Uploaded Data

Enter your question:

What is the procedure for interconnecting blocks in the CFC editor?

Get Answer

Generating answers...

Base Model Answer: To interconnect blocks in the CFC editor, follow these steps:

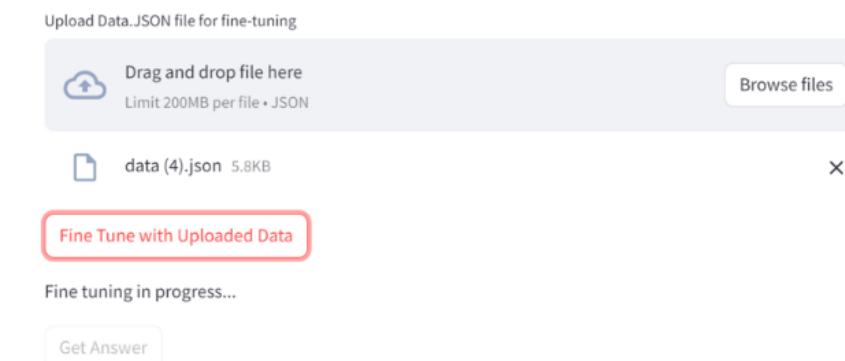
1. Select the blocks you want to connect by clicking and dragging your mouse over them.
2. Hold down the left mouse button and move the blocks to the desired location.
3. When the blocks are in the desired position, release the left mouse button. The blocks will snap together, creating a connection between them.
4. If you want to create a more complex connection, you

Fine-Tuned Model Answer: To interconnect blocks in the CFC editor, click on the output of one block and then on the input of another block where you want to establish the connection. The CFC editor automatically creates a line indicating the interconnection.

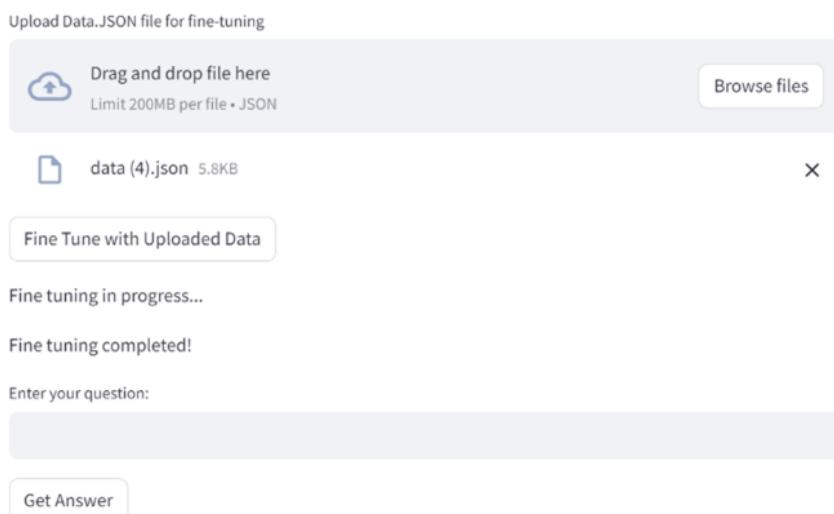
**Figure 10.** Comparison of base model and fine-tuned model answers. The red rectangle highlights the base model's irrelevant response compared to the fine-tuned model.

Furthermore, the system allows users to upload a JSON data file to fine-tune the model. The interface indicates the progress and completion of the fine-tuning process, as shown in Figure 11a,b, giving the user complete control of the fine-tuning process. The fine-tuned Llama 2 model then processes this refined data further, enhancing the diagnostic accuracy and relevance of the system's responses. With the integration of these components, the workflow ensures that the FDD system can provide precise and contextually relevant responses.

# Fine Tuning and Question Answering Interface



(a)



(b)

**Figure 11.** (a) Fine-tuning dataset ingestion. (b) Fine-tuning process.

### 5.3. Parameter Tuning and PS Optimization

Optimizing the performance of the proposed method requires a systematic approach to parameter tuning, focusing on maximizing the PS. Key strategies include reducing the average response time ( $RT_{avg}$ ) by selecting appropriate LLMs, employing effective preprocessing techniques, using high-quality datasets for fine-tuning, and optimizing vector database configurations. Fine-tuning is critical, as parameters such as the learning rate, number of epochs, and batch size significantly affect model performance. Moreover, the choice of embedding technique and its dimensionality can significantly influence the quality of fault representation and retrieval efficiency from the vector database. The indexing method and similarity search algorithms within the vector database must be carefully considered to achieve an optimal balance between speed and accuracy.

### 5.4. Limitations and Challenges of LLM-Based FDD

Despite the GPT-4-Preview model achieving a 96.3% accuracy in fault diagnosis, outperforming conventional models, several limitations were determined. A significant challenge lies in the model's lack of true understanding and commonsense reasoning. In approximately 3.7% of cases, the model produced responses that were syntactically correct

but contextually inaccurate. Moreover, instances of hallucination were observed, where the model generated irrelevant or factually incorrect responses in approximately 10% of cases.

Our analysis revealed specific biases in the GPT-4-Preview model's fault diagnosis capabilities. When handling ambiguous cases, the model demonstrated a tendency to attribute faults more frequently to CPU/processor issues rather than input/output issues, indicating a systematic bias in fault classification. Bias manifestation occurs primarily through hallucination in complex fault scenarios, over-attribution to common fault types, and inconsistent response patterns in ambiguous cases. While our current implementation demonstrates high overall accuracy, addressing these biases is crucial for industrial applications.

Future implementations should consider training data diversification with balanced fault type representation, the implementation of post-generation validation, and prompt engineering optimization for balanced fault attribution. These bias patterns indicate the importance of human oversight in critical diagnostic decisions, particularly in cases where the model shows uncertainty or inconsistent response patterns. The integration of human expertise with AI capabilities remains essential for ensuring reliable fault diagnosis in industrial settings.

Computationally, fine-tuning a GPT-4 model on a dataset comprising 52,464 words (approximately 69,952 tokens), including fault data from diverse industrial equipment manufacturers over 10 epochs, is computationally intensive. With an NVIDIA V100 GPU, this operation is estimated to take around 3 h. Moreover, the fine-tuning process incurs significant financial costs due to the resources required for computation and model optimization. For resource-constrained environments, several optimization approaches could be considered. The local preprocessing and caching of common fault patterns could reduce dependency on external API calls. Token optimization strategies could allocate resources based on fault complexity: simple (512–650 tokens), medium (845–1100 tokens), and complex diagnoses (1280–2048 tokens). Such approaches would enable smaller enterprises to balance diagnostic capability with resource constraints.

Context management emerged as a significant challenge, but we were able to overcome it. We were initially limited by the model's token window size of 8192 tokens. However, we integrated tools, including LangChain and LlamaIndex, which enable dynamic document retrieval and sophisticated text-splitting and vector storage techniques.

Finally, reliability in real-time diagnosis is still a significant area for improvement. The model exhibited inconsistent response times, with the most extended delay recorded at 21.27 s for specific fault scenarios. Such variability, especially in real-time industrial applications, can negatively impact reliability, which highlights the need to optimize both model selection and response strategies to meet the strict timing demands of industrial FDD systems.

While our framework demonstrates adaptability across multiple industrial systems, its performance in novel environments requires further validation. Future work should focus on expanding the training dataset diversity and implementing continuous learning mechanisms to improve generalization to unseen fault scenarios.

## 6. Conclusions

In conclusion, in this research, a novel approach to FDD in industrial systems is presented, leveraging LLMs and multimodal data analysis within Industry 4.0. Our proposed multimodal framework, which incorporates diverse data sources, has significantly advanced industrial diagnostics. Specifically, our FDD GPT-4-Preview model showed superior performance with 96.3% accuracy in fault diagnosis, surpassing traditional methods such as RF (90.36%) and NNs (86.07%).

The proposed FDD GPT-4-Preview model achieved a PS of 3.255, significantly outperforming other models such as Mixtral 8x7B (1.504) and ChatGPT 4 (0.804). It also demonstrated superior performance in relevant diagnostic information across various fault scenarios, with 60% of the responses classified as precise compared to 0% for other models.

This study significantly contributes to the application of AI techniques in industrial settings. The proposed framework is a substantial step toward developing more sophisticated diagnostic support tools, offering a 5.94-percentage-point improvement in accuracy over the next best conventional model while providing enhanced explainability through natural language responses.

By contrast, a hybrid architecture was proposed that combines real-time data streams with fine-tuned LLMs to effectively create a dynamic and responsive FDD system. Our fine-tuned model consistently outperformed the base model, which exhibits enhanced accuracy and contextual understanding in fault diagnosis scenarios.

Although the results are promising, further research is necessary to fully realize the potential of LLM-based FDD systems in various industrial contexts. To enhance data security and confidentiality in industrial settings, future work should explore integrating emerging technologies, including blockchain. The decentralization and immutability of blockchain could provide a robust foundation for securing sensitive diagnostic data and ensuring the integrity of FDD processes.

Moreover, fostering vital collaboration between humans and machines through human-in-the-loop systems can further enhance the outcomes of LLM-based diagnostics. This collaborative approach leverages human expertise to validate and refine AI-generated insights, potentially leading to more accurate and contextually appropriate fault diagnosis. Through combining advanced AI techniques with secure data management and human oversight, more trustworthy and effective FDD systems that meet the complex demands of Industry 4.0 environments can be created.

**Author Contributions:** K.M.A. contributed significantly to the conception and design of the study, wrote the main manuscript, and conducted the data analysis and interpretation. M.A.K., F.E.E., and A.A.A. provided valuable feedback, which was essential for the critical revision of the manuscript, and all authors approved the final version for publication. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** The original data supporting the findings of this study are included within the article. Additional inquiries can be directed to the corresponding author.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

- Park, Y.J.; Fan, S.K.S.; Hsu, C.Y. A review on fault detection and process diagnostics in industrial processes. *Processes* **2020**, *8*, 1123. [[CrossRef](#)]
- Yao, Y.; Duan, J.; Xu, K.; Cai, Y.; Sun, Z.; Zhang, Y. A survey on large language model (LLM) security and privacy: The good, the bad, and the ugly. *High-Confid. Comput.* **2024**, *4*, 100211. [[CrossRef](#)]
- Hang, C.N.; Yu, P.D.; Morabito, R.; Tan, C.W. Large Language Models Meet Next-Generation Networking Technologies: A Review. *Future Internet* **2024**, *16*, 365. [[CrossRef](#)]
- Zhang, D.; Zhou, T. Deep convolutional neural network using transfer learning for fault diagnosis. *IEEE Access* **2021**, *9*, 43889–43897. [[CrossRef](#)]
- Ran, M.; Zhao, X.; Li, Y.; Liang, H. Development of fault diagnosis expert system based on wastewater treatment systems of intelligent substations. In Proceedings of the 2023 IEEE International Conference on Control, Electronics and Computer Technology (ICCECT), Jilin, China. 28–30 April 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 1070–1073.
- Xue, L.E.I.; Ningyun, L.U.; Chuang, C.; Tianzhen, H.U.; Bin, J. Attention mechanism based multi-scale feature extraction of bearing fault diagnosis. *J. Syst. Eng. Electron.* **2023**, *34*, 1359–1367. [[CrossRef](#)]
- Tang, S.; Yuan, S.; Zhu, Y. Convolutional neural network in intelligent fault diagnosis toward rotatory machinery. *IEEE Access* **2020**, *8*, 86510–86519. [[CrossRef](#)]
- Makridakis, S.; Petropoulos, F.; Kang, Y. Large language models: Their success and impact. *Forecasting* **2023**, *5*, 536–549. [[CrossRef](#)]
- Xu, T.; Tang, X.S. Electrical Equipment Fault Diagnosis: A Technique Combining Fuzzy Logic and Large Language Models. In Proceedings of the 2023 IEEE International Symposium on Product Compliance Engineering-Asia (ISPCE-ASIA), Shanghai, China, 3–5 November 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 1–4. [[CrossRef](#)]

10. Qin, X.; He, Y.; Ma, J.; Peng, W.; Zio, E.; Su, H. An Effective Knowledge Mining Method for Compressor Fault Text Data Based on Large Language Model. In Proceedings of the 2023 International Conference on Computer Science and Automation Technology (CSAT), Shanghai, China, 6–8 October 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 44–48.
11. Xia, Q.; Zhao, H.; Liu, M. Prompt Engineering Approach Study for Supervised Fine-Tuned (SFT) Large Language Models (LLMs) in Spacecraft Fault Diagnosis. In Proceedings of the 2024 3rd Conference on Fully Actuated System Theory and Applications (FASTA), Shenzhen, China, 10–12 May 2024; IEEE: Piscataway, NJ, USA, 2024, pp. 819–824.
12. Peifeng, L.; Qian, L.; Zhao, X.; Tao, B. Joint knowledge graph and large language model for fault diagnosis and its application in aviation assembly. *IEEE Trans. Ind. Inform.* **2024**, *20*, 8160–8169.
13. Li, L.; Li, Y.; Zhang, X.; He, Y.; Yang, J.; Tian, B.; Ai, Y.; Li, L.; Nüchter, A.; Xuanyuan, Z. Embodied Intelligence in Mining: Leveraging Multi-modal Large Language Model for Autonomous Driving in Mines. *IEEE Trans. Intell. Veh.* **2024**, *9*, 4831–4834. [[CrossRef](#)]
14. Li, F.; Feng, S.; Yan, Y.; Lee, C.H.; Ong, Y.S. Virtual Co-Pilot: Multimodal Large Language Model-enabled Quick-access Procedures for Single Pilot Operations. *arXiv* **2024**, arXiv:2403.16645.
15. Zhang, D.; Yu, Y.; Li, C.; Dong, J.; Su, D.; Chu, C.; Yu, D. Mm-llms: Recent advances in multimodal large language models. *arXiv* **2024**, arXiv:2401.13601.
16. Qi, R.; Zhang, J.; Spencer, K. A review on data-driven condition monitoring of industrial equipment. *Algorithms* **2022**, *16*, 9. [[CrossRef](#)]
17. Liu, Q.; Wang, C.; Wang, Q. Bayesian Uncertainty Inferencing for Fault Diagnosis of Intelligent Instruments in IoT Systems. *Appl. Sci.* **2023**, *13*, 5380. [[CrossRef](#)]
18. Xu, X.; Du, Z.; Bai, Z.; Wang, S.; Wang, C.; Li, D. Fault diagnosis method of dissolved oxygen sensor electrolyte loss based on impedance measurement. *Comput. Electron Agric.* **2023**, *212*, 108123. [[CrossRef](#)]
19. Zhang, G.; Li, Y.; Zhao, Y. A novel fault diagnosis method for wind turbine based on adaptive multivariate time-series convolutional network using SCADA data. *Adv. Eng. Inform.* **2023**, *57*, 102031. [[CrossRef](#)]
20. Ghosh, A.; Wang, G.N.; Lee, J. A novel automata and neural network based fault diagnosis system for PLC controlled manufacturing systems. *Comput. Ind. Eng.* **2020**, *139*, 106188. [[CrossRef](#)]
21. Durrani, S.; et al. Smart Fault Detection and Generator Protection scheme using Arduino and HMI. In Proceedings of the 2019 International Conference on Electrical, Communication, and Computer Engineering (ICECCE), Swat, Pakistan, 24–25 July 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–6. [[CrossRef](#)]
22. Kiangala, K.S.; Wang, Z. An experimental hybrid customized AI and generative AI chatbot human machine interface to improve a factory troubleshooting downtime in the context of Industry 5.0. *Int. J. Adv. Manuf. Technol.* **2024**, *132*, 2715–2733. [[CrossRef](#)]
23. Djordjevic, V.; Dubonjic, L.; Morato, M.M.; Pršić, D.; Stojanović, V. Sensor fault estimation for hydraulic servo actuator based on sliding mode observer. *Math. Model. Control* **2022**, *2*, 34–43. [[CrossRef](#)]
24. Tao, H.; Shi, H.; Qiu, J.; Jin, G.; Stojanovic, V. Planetary gearbox fault diagnosis based on FDKNN-DGAT with few labeled data. *Meas. Sci. Technol.* **2023**, *35*, 025036. [[CrossRef](#)]
25. Yu, Y.; Guo, L.; Tan, Y.; Gao, H.; Zhang, J. Multisource partial transfer network for machinery fault diagnostics. *IEEE Trans. Ind. Electron.* **2021**, *69*, 10585–10594. [[CrossRef](#)]
26. Neupane, D.; Kim, Y.; Seok, J.; Hong, J. CNN-based fault detection for smart manufacturing. *Appl. Sci.* **2021**, *11*, 11732. [[CrossRef](#)]
27. Gopal, L.; Singh, H.; Mounica, P.; Mohankumar, N.; Challa, N.P.; Jayaraman, P. Digital twin and IOT technology for secure manufacturing systems. *Meas. Sens.* **2023**, *25*, 100661. [[CrossRef](#)]
28. Gordan, M.; Sabbagh-Yazdi, S.R.; Ghaedi, K.; Ismail, Z. A damage detection approach in the era of industry 4.0 using the relationship between circular economy, data mining, and artificial intelligence. *Adv. Civ. Eng.* **2023**, *2023*, 3067824. [[CrossRef](#)]
29. Azari, M.S.; Flammini, F.; Santini, S.; Caporuscio, M. A systematic literature review on transfer learning for predictive maintenance in industry 4.0. *IEEE Access* **2023**, *11*, 12887–12910. [[CrossRef](#)]
30. Yen, C.T.; Chuang, P.C. Application of a neural network integrated with the internet of things sensing technology for 3D printer fault diagnosis. *Microsyst. Technol.* **2022**, *28*, 13–23. [[CrossRef](#)]
31. Pineda-Pineda, A.E.; Madrid-Rivera, Ó.B. Design and development of an integrated diagnostic pneumatic end effector using soft robotics and IOT. In Proceedings of the 2023 IEEE Central America and Panama Student Conference (CONESCAPAN), Guatemala, Guatemala, 26–29 September 2023 ; pp. 145–150.
32. Liu, Y.; Chen, C.; Wang, T.; Cheng, L. An attention enhanced dilated CNN approach for cross-axis industrial robotics fault diagnosis. *Auton. Intell. Syst.* **2022**, *2*, 11. [[CrossRef](#)]
33. Zhang, W.; Wang, Z.; Li, X. Blockchain-based decentralized federated transfer learning methodology for collaborative machinery fault diagnosis. *Reliab. Eng. Syst. Saf.* **2023**, *229*, 108885. [[CrossRef](#)]
34. Ali, M.N.; Amer, M.; Elsisi, M. Reliable IoT paradigm with ensemble machine learning for faults diagnosis of power transformers considering adversarial attacks. *IEEE Trans. Instrum. Meas.* **2023**, *72*, 1–13. [[CrossRef](#)]
35. Elsisi, M.; Tran, M.Q.; Mahmoud, K.; Mansour, D.E.A.; Lehtonen, M.; Darwish, M.M. Effective IoT-based deep learning platform for online fault diagnosis of power transformers against cyberattacks and data uncertainties. *Measurement* **2022**, *190*, 110686. [[CrossRef](#)]
36. Venkatasubramanian, S.; Raja, S.; Sumanth, V.; Dwivedi, J.N.; Sathiaparkavi, J.; Modak, S.; Kejela, M.L. Fault diagnosis using data fusion with ensemble deep learning technique in IIoT. *Math. Probl. Eng.* **2022**, *2022*, 1682874. [[CrossRef](#)]

37. Wang, R.; Zhuang, Z.; Tao, H.; Paszke, W.; Stojanovic, V. Q-learning based fault estimation and fault tolerant iterative learning control for MIMO systems. *ISA Trans.* **2023**, *142*, 123–135. [[CrossRef](#)] [[PubMed](#)]
38. Zhang, L.; Gao, W.; Yan, L. Deep learning-based fault diagnosis and localization method for fiber optic cables in communication networks. *Appl. Math. Nonlinear Sci.* **2023**, *9*, 1–14. [[CrossRef](#)]
39. Feng, C.; Zhao, Y. Fault Identification and Analysis of Communication Network Based on Deep Learning. *Mob. Inf. Syst.* **2022**, *2022*, 1456425.
40. Sun, L. Fault Diagnosis Design Mechanism of Communication System Based on Bayesian Networks. In Proceedings of the 2023 IEEE 3rd International Conference on Electronic Communications, Internet of Things and Big Data (ICEIB), Taichung, Taiwan, 14–16 April 2023; pp. 41–44.
41. Mezair, T.; Djenouri, Y.; Belhadi, A.; Srivastava, G.; Lin, J.C.W. A sustainable deep learning framework for fault detection in 6G Industry 4.0 heterogeneous data environments. *Comput. Commun.* **2022**, *187*, 164–171. [[CrossRef](#)]
42. Chondroulis, I.; Ntogkas, C.; Belikaidis, I.; Georgakopoulos, A.; Kosmatos, E.; Tsagkaris, K.; Demestichas, P. Performance-Aware Orchestration and Management over 5G and Beyond Infrastructures Based on Diagnostic Mechanisms. In Proceedings of the 2022 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit), Grenoble, France, 10 June 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 381–386.
43. Nielsen, M.H.; Zhang, Y.; Xue, C.; Ren, J.; Yin, Y.; Shen, M.; Pedersen, G.F. Robust and efficient fault diagnosis of mm-wave active phased arrays using baseband signal. *IEEE Trans. Antennas Propag.* **2022**, *70*, 5044–5053. [[CrossRef](#)]
44. Yu, Y.; Guo, L.; Gao, H.; He, Y.; You, Z.; Duan, A. FedCAE: A new federated learning framework for edge-cloud collaboration based machine fault diagnosis. *IEEE Trans. Ind. Electron.* **2024**, *71*, 4108–4119. [[CrossRef](#)]
45. Jiang, Y.; Yin, S.; Kaynak, O. Performance supervised plant-wide process monitoring in industry 4.0: A roadmap. *IEEE Open J. Ind. Electron. Soc.* **2020**, *2*, 21–35. [[CrossRef](#)]
46. Khalid, S.; Song, J.; Raouf, I.; Kim, H.S. Advances in fault detection and diagnosis for thermal power plants: A review of intelligent techniques. *Mathematics* **2023**, *11*, 1767. [[CrossRef](#)]
47. Neuvonen, M.; Selek, I.; Ikonen, E.; Aho, L. Heat exchanger fouling estimation for combustion–thermal power plants including load level dynamics. In Proceedings of the 2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Kuching, Malaysia, 7–10 October 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 2987–2992.
48. Naimi, A.; Deng, J.; Doney, P.; Sheikh-Akbari, A.; Shimjith, S.R.; Arul, A.J. Machine learning-based fault diagnosis for a PWR nuclear power plant. *IEEE Access* **2022**, *10*, 126001–126010. [[CrossRef](#)]
49. Yuan, X.; Liu, Z.; Miao, Z.; Zhao, Z.; Zhou, F.; Song, Y. Fault diagnosis of analog circuits based on IH-PSO optimized support vector machine. *IEEE Access* **2019**, *7*, 137945–137958. [[CrossRef](#)]
50. Luo, Y.; Zhang, L.; Chen, C.; Li, K.; Yu, T.; Li, K. FRA-Based Parameter Estimation for Fault Diagnosis of Three-Phase Voltage-Source Inverters. *IEEE Access* **2023**, *11*, 113836–113847. [[CrossRef](#)]
51. Lin, W. Intelligent fault diagnosis of consumer electronics sensor in IoE via transformer. *IEEE Trans. Consum. Electron.* **2024**, *70*, 1259–1267. [[CrossRef](#)]
52. Zeng, Y.; Mei, Y.; Hu, Y.; Sheng, Z. A Novel Hierarchical Tree-DCNN Structure for Unbalanced Data Diagnosis in Microelectronic Manufacturing Process. *IEEE Trans. Instrum. Meas.* **2024**, *73*, 1–11. [[CrossRef](#)]
53. Zhong, S.; Ali, R. Joint Self-Attention Mechanism and Residual Network for Automated Monitoring of Intelligent Sensor in Consumer Electronics. *IEEE Trans. Consum. Electron.* **2024**, *70*, 1302–1309. [[CrossRef](#)]
54. Yu, Q.; Wang, C.; Li, J.; Xiong, R.; Pecht, M. Challenges and outlook for lithium-ion battery fault diagnosis methods from the laboratory to real world applications. *ETransportation* **2023**, *17*, 100254. [[CrossRef](#)]
55. Zhang, C.; Wang, H.; Wang, Z.; Li, Y. Active detection fault diagnosis and fault location technology for LVDC distribution networks. *Int. J. Electr. Power Energy Syst.* **2023**, *148*, 108921. [[CrossRef](#)]
56. Zhou, X.; Liu, Z.; Shi, Z.; Ma, L.; Du, H.; Han, D. Fault Diagnosis of the Power Transformer Based on PSO-SVM. In Proceedings of the International Conference in Communications, Signal Processing, and Systems, Changbaishan, China, 22–23 July 2022; Springer: Berlin/Heidelberg, Germany, 2022; pp. 95–102.
57. Qu, L.; Zhang, J.; Gao, T. Fault Diagnosis of Power Grid Based on Convolutional Neural Network. In Proceedings of the 2022 21st International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES), Chizhou, China, 14–18 October 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 82–86.
58. Geng, D.; Zhou, L.; Zhao, Y.; Li, H. Fault Diagnosis of Motor based on Knowledge Graph and CNN. In Proceedings of the 2022 3rd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE), Xi'an, China, 15–17 July 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 742–745.
59. Boubaker, S.; Kamel, S.; Ghazouani, N.; Mellit, A. Assessment of machine and deep learning approaches for fault diagnosis in photovoltaic systems using infrared thermography. *Remote. Sens.* **2023**, *15*, 1686. [[CrossRef](#)]
60. Zhang, C.; Lu, J.; Zhao, Y. Generative pre-trained transformers (GPT)-based automated data mining for building energy management: Advantages, limitations and the future. *Energy Built. Environ.* **2024**, *5*, 143–169. [[CrossRef](#)]
61. Zhang, J.; He, X. Compound-fault diagnosis of integrated energy systems based on graph embedded recurrent neural networks. *IEEE Trans. Ind. Inform.* **2024**, *20*, 3478–3486. [[CrossRef](#)]

62. Wu, W.; Dou, L. Fault diagnosis for high voltage circuit breaker based on VMD energy entropy and support vector machine. In Proceedings of the 2023 IEEE 3rd International Conference on Electronic Technology, Communication and Information (ICETCI), Changchun, China, 26–28 May 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 1044–1049.
63. Hou, W.; Li, W.; Li, P. Fault diagnosis of the autonomous driving perception system based on information fusion. *Sensors* **2023**, *23*, 5110. [CrossRef]
64. Zhao, X. Aviation application of dual-weighted neural network based on biomimetic pattern recognition. *J. Phys. Conf. Ser.* **2022**, *2252*, 012059. [CrossRef]
65. Wang, S.; Li, P.; Niu, W. Application and challenges of deep neural network in fault diagnosis of aviation equipment. In Proceedings of the International Conference on Neural Networks, Information, and Communication Engineering (NNICE 2022), Qingdao, China, 25–27 March 2022; SPIE: Weinheim, Germany, 2022; pp. 111–115.
66. Liu, X.; Wang, X.; Han, G. Adaptive Fault Diagnosis System for Railway Track Circuits. In Proceedings of the International Conference on Electrical and Information Technologies for Rail Transportation, Beijing, China, 19–21 October 2023; Springer: Berlin/Heidelberg, Germany, 2021; pp. 491–498.
67. Chung, K.J.; Lin, C.W. Condition monitoring for fault diagnosis of railway wheels using recurrence plots and convolutional neural networks (RP-CNN) models. *Meas. Control* **2024**, *57*, 330–338. [CrossRef]
68. Hasan, A.; Asfihani, T.; Osen, O.; Bye, R.T. Leveraging digital twins for fault diagnosis in autonomous ships. *Ocean Eng.* **2024**, *292*, 116546. [CrossRef]
69. He, Y.; Ni, C. A fault diagnosis method for ship shaft system based on empirical wavelet transform and particle filtering. In Proceedings of the 2023 IEEE 2nd International Conference on Electrical Engineering, Big Data and Algorithms (EEBDA), Changchun, China, 24–26 February 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 942–947.
70. Anandhalekshmi, A.V.; Rao, V.S.; Kanagachidambaresan, G.R. Hybrid approach of baum-welch algorithm and SVM for sensor fault diagnosis in healthcare monitoring system. *J. Intell. Fuzzy Syst.* **2022**, *42*, 2979–2988. [CrossRef]
71. Abououf, M.; Singh, S.; Mizouni, R.; Otrok, H. Explainable AI for Event and Anomaly Detection and Classification in Healthcare Monitoring Systems. *IEEE Int. Things J.* **2024**, *11*, 3446–3457. [CrossRef]
72. Ma, H.; Xu, C.; Yang, J. Design of Fine Life Cycle Prediction System for Failure of Medical Equipment. *J. Artif. Intell. Technol.* **2023**, *3*, 39–45. [CrossRef]
73. Yu, P.; Xu, H.; Hu, X.; Deng, C. Leveraging generative AI and large Language models: A Comprehensive Roadmap for Healthcare Integration. *Healthcare* **2023**, *11*, 2776. [CrossRef]
74. Zhang, S.; Luo, X.; Li, L.; Yang, Y. Hierarchical health assessment of equipment with uncertain fault diagnosis result. In Proceedings of the 2019 Prognostics and System Health Management Conference (PHM-Qingdao), Qingdao, China, 25–27 October 2019; IEEE: Piscataway, NJ, USA, 2019. [CrossRef]
75. Xu, J.; Tian, X.; Jin, W.; Guo, H. Pwm harmonic-current-based interturn short-circuit fault diagnosis for the aerospace ftpmsm system even in the fault-tolerant operation condition. *IEEE Trans. Power Electron.* **2023**, *38*, 5432–5441. [CrossRef]
76. Wang, T.; Ding, L.; Yu, H. Research and development of fault diagnosis methods for liquid rocket engines. *Aerospace* **2022**, *9*, 481. [CrossRef]
77. Deng, L.; Cheng, Y.; Shi, Y. Fault detection and diagnosis for liquid rocket engines based on long short-term memory and generative adversarial networks. *Aerospace* **2022**, *9*, 399. [CrossRef]
78. Ezzat, D.; Hassani, A.E.; Darwish, A.; Yahia, M.; Ahmed, A.; Abdelghafar, S. Multi-objective hybrid artificial intelligence approach for fault diagnosis of aerospace systems. *IEEE Access* **2021**, *9*, 41717–41730. [CrossRef]
79. Min, Z.; Ke, W.; Yang, W.; Kaijia, L.U.O.; Hongyong, F.U.; Liang, S.I. Online condition diagnosis for a two-stage gearbox machinery of an aerospace utilization system using an ensemble multi-fault features indexing approach. *Chin. J. Aeronaut.* **2019**, *32*, 1100–1110.
80. Sakura, T.; Soga, R.; Kanuka, H.; Shimari, K.; Ishio, T. Leveraging Execution Trace with ChatGPT: A Case Study on Automated Fault Diagnosis. In Proceedings of the 2023 IEEE International Conference on Software Maintenance and Evolution (ICSME), Bogotá, Colombia, 1–6 October 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 397–402.
81. Khurana, R.; Batra, S.; Sharma, V. Software Fault Diagnosis via Intelligent Data Mining Algorithms. In Proceedings of the Proceedings of International Conference on Recent Trends in Computing: ICRTC 2022, Delhi, India, 3–4 June 2022; Springer: Berlin/Heidelberg, Germany, 2023; pp. 655–667.
82. Zhang, W.; Hu, Y.; Tan, B.; Shi, X.; Jiang, J. Adaptive Tracing and Fault Injection based Fault Diagnosis for Open Source Server Software. In Proceedings of the 2023 IEEE 23rd International Conference on Software Quality, Reliability, and Security (QRS), Chiang Mai, Thailand, 22–26 October 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 729–740.
83. Begum, M.; Shuvo, M.H.; Ashraf, I.; Mamun, A.A.; Uddin, J.; Samad, M.A. Software Defects Identification: Results Using Machine Learning and Explainable Artificial Intelligence Techniques. *IEEE Access* **2023**, *11*, 132750–132765. [CrossRef]
84. AI, M. Meta Llama Guard 2. 2023. Available online: <https://llama.meta.com/docs/model-cards-and-prompt-formats/meta-llama-guard-2> (accessed on 7 February 2024).
85. Assistants Overview—OpenAI API. Available online: <https://platform.openai.com/docs/assistants/overview> (accessed on 20 June 2024).
86. AI, M. Mistral AI Documentation. 2023. Available online: <https://docs.mistral.ai/> (accessed on 7 February 2024).
87. Introduction | LangChain. Available online: <https://python.langchain.com/v0.2/docs/introduction/> (accessed on 20 June 2024).

88. LlamaIndex—LlamaIndex. Available online: <https://docs.llamaindex.ai/en/stable/> (accessed on 19 June 2024).
89. Siemens Industry Support. 2024. Available online: <https://support.industry.siemens.com/cs/start?lc=en-US> (accessed on 5 January 2024).

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.