Uncertainty Discourse and Applications

www.uda.reapress.com

Uncert. Disc. Appl. Vol. 1, No. 2 (2024) 186–194.

Paper Type: Original Article

AI-Based Fault Detection in IoT Cloud Systems

Aryan Kapoor*

School of Computer Science Engineering, KIIT University, Bhubaneswar, India; 2205885@kiit.ac.in.

Citation:

Received: 28 May 2024 *Revised: 19 July 2024* *Accepted:25 September 2024*	Kapoor, A. (2024). AI-based fault detection in IoT cloud systems. *Uncertainty discourse and applications, 1*(2), 186-194.

Abstract

This paper investigates the use of Artificial Intelligence (AI) technologies for identifying faults in Internet of Things (IoT) cloud systems. By utilizing machine learning and deep learning models, the suggested method seeks to improve fault detection accuracy, minimize downtime, and enhance resource allocation in IoT-enabled cloud settings. The research reviews a range of AI models, assesses their effectiveness on IoT cloud data, and introduces an optimized hybrid model. The findings indicate significant improvements in fault detection rates and management of cloud resources. The study also discusses the implications for the robustness of cloud systems and the monitoring of real-time IoT applications.

Keywords: Artificial intelligence, Internet of things cloud systems, Fault detection, Machine learning, Cloud computing.

1 | Introduction

Combined with the Internet of Things (IoT), cloud computing has radically transformed how we collect, process, and analyze data across industries [1]. This interconnected ecosystem of devices—from simple sensors to complex industrial systems—enables massive volumes of data to be generated and transmitted in real-time. These data streams are processed and stored on cloud platforms, providing the computational power, storage capabilities, and scalability needed to support IoT environments' diverse and dynamic nature.

The fusion of IoT and cloud computing has numerous applications across various sectors, including smart cities, healthcare, manufacturing, transportation, and agriculture [2]. In smart cities, for instance, IoT devices collect data on traffic flow [3], air quality, and energy usage, which is then processed in real time by cloud-based systems to optimize infrastructure and enhance city services. Similarly, in healthcare, connected medical devices transmit patient data to cloud systems for continuous monitoring and diagnostics, improving patient care while reducing the workload on healthcare providers [4].

Corresponding Author: 2205885@kiit.ac.in

https://doi.org/10.48313/uda.v1i2.40

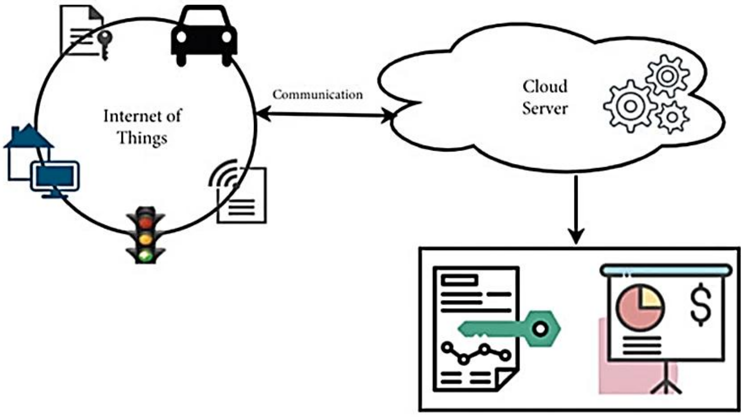

Fig. 1. Integration of internet of things with cloud.

Artificial Intelligence (AI)-based fault detection is emerging as a crucial technique to address these issues [5]. By leveraging machine learning and deep learning, cloud systems can analyze large volumes of data, detect patterns, and identify faults in real-time. Implementing AI in fault detection offers the potential to predict failures before they occur, mitigating downtime and ensuring smooth operations [6].

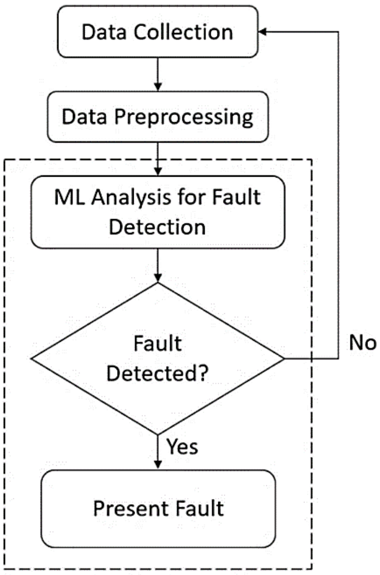

Fig. 2. Flowchart illustration of the machine learning-based analysis procedure.

Table 1. Common faults in internet of things cloud systems.

Fault Type	Description	Impact
Device malfunction	Physical or software failure in IoT devices	Loss of data, downtime
Communication failure	Loss of connectivity between devices and the cloud	Disrupted data flow
Resource overload	Cloud resources overwhelmed by data traffic	Slow response, crashes
Security breach	Unauthorized access or data tampering	Data theft, downtime
Software bug	Programming errors in IoT applications	Unexpected system behavior

IoT systems, when integrated into cloud environments, introduce new challenges [7-9]:

I. Scalability: With an increasing number of devices, the system must scale efficiently without losing performance.

II. Real-time processing: IoT data needs to be processed instantly for effective fault detection and recovery.

III. Resource management: Efficient use of cloud resources is critical to handle the diverse and massive data influx from IoT devices.

IV. Fault tolerance: System components must handle unexpected failures and continue operations without disrupting service.

2 | Literature Review

2.1 | Traditional Fault Detection Methods

Fault detection in IoT cloud systems relies on rule-based or threshold-based systems [10]. These systems analyze incoming data streams from IoT devices and flag any deviations from predefined parameters as potential faults. While effective in austere environments, these methods have significant limitations, particularly in detecting complex, unknown fault types.

Fault diagnosis is a critical component of maintaining the reliability and performance of IoT cloud systems. While fault detection aims to identify when a failure or anomaly has occurred, fault diagnosis goes a step further by determining the cause and location of the fault. This process is essential for rapid fault recovery, system optimization, and preventive maintenance. In the context of AI-based fault detection in IoT cloud systems, some key fault diagnosis methods are employed (*Fig. 3*).

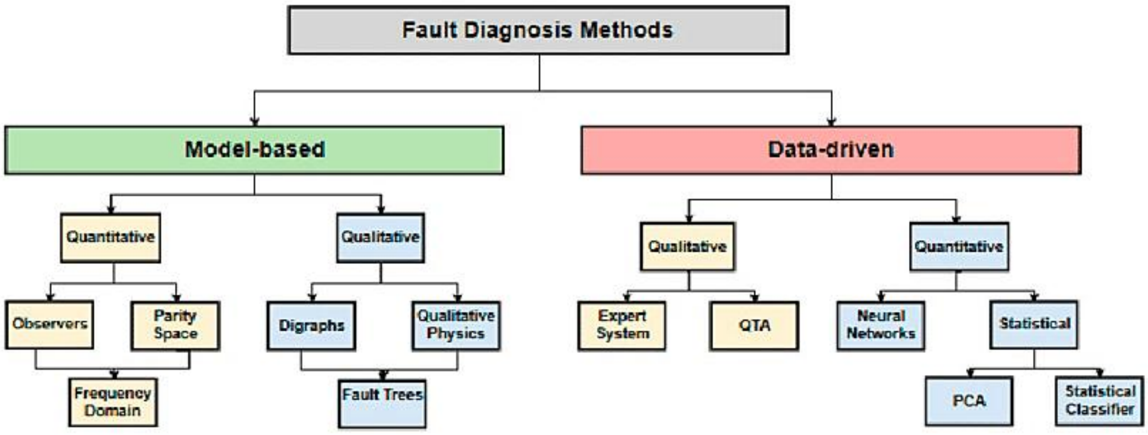

Fig. 3. Fault diagnosis methods.

2.2 | Artificial Intelligence-Based Fault Detection Techniques

AI has revolutionized fault detection by allowing systems to learn from historical data, adapt to changing environments, and make accurate predictions in real time [11-13].

2.2.1 | Supervised learning techniques

Supervised learning models such as decision trees, random forests, and Support Vector Machines (SVMs) are often used in fault detection [14], [15]. These models are trained on labeled datasets that include historical instances of faults and normal behavior.

Decision trees

Useful for classifying faults based on predefined features.

Support vector machines

Highly effective for binary classification problems, such as fault/no-fault scenarios.

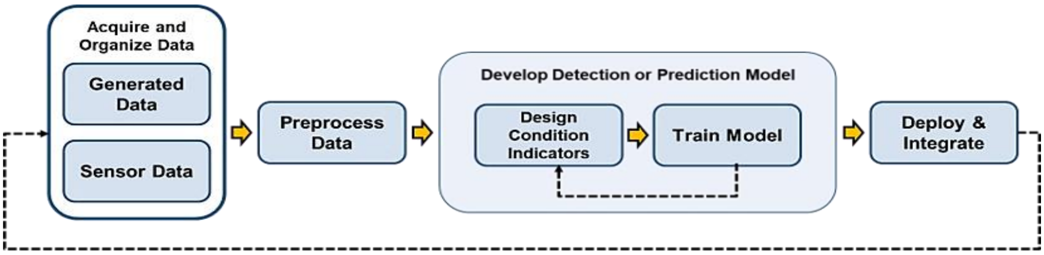
Acquire and Organize Data
Generated Data
Sensor Data
Preprocess Data
Develop Detection or Prediction Model
Design Condition Indicators
Train Model
Deploy & Integrate

Fig. 4. Decision models for fault detection and diagnosis.

2.2.2 | Unsupervised learning techniques

When labeled data is scarce, unsupervised learning models [16], such as K-means clustering and autoencoders, are employed [17]. These models identify patterns and anomalies without prior knowledge of fault types.

K-means clustering

Useful for grouping data points into clusters, enabling detection of abnormal behavior as outliers.

Autoencoders

Neural networks designed to detect anomalies by reconstructing input data and flagging deviations.

2.3 | Hybrid Artificial Intelligence Models for Fault Detection

Recent advancements suggest combining supervised and unsupervised learning techniques in hybrid AI models yields superior results. The hybrid approach leverages the strengths of both methods, offering a more robust fault detection mechanism.

Table 2 highlights how AI-based methods, especially hybrid models, outperform traditional methods in scalability, accuracy, and the ability to handle unknown faults. This demonstrates why AI-driven approaches are critical for fault detection in complex IoT cloud environments.

Table 2. Comparison of traditional and artificial intelligence-based fault detection techniques.

Method	Detection Speed	Accuracy	Scalability	Handling Unknown Faults
Traditional rule-based	Moderate	Low	Low	No
Threshold-based systems	Fast	Moderate	Moderate	No
Supervised learning	Fast	High	High	No
Unsupervised learning	Moderate	Moderate	High	Yes
Hybrid AI model	Fast	Very high	Very high	Yes

3 | Proposed Artificial Intelligence-Based Model for Fault Detection

The proposed AI-based fault detection model integrates both supervised and unsupervised learning methods. The primary objective is to maximize detection accuracy while minimizing computational overhead in large-scale IoT cloud systems.

3.1 | Architecture of the Proposed Model

The hybrid model consists of two main components:

I. Supervised learning component: Trained on historical data to identify known fault types.

II. Unsupervised learning component: Monitors real-time IoT data to identify new, previously unseen fault patterns.

Data from IoT devices is collected and processed in the cloud. The supervised component classifies known faults, while the unsupervised component looks for anomalies that deviate from expected patterns.

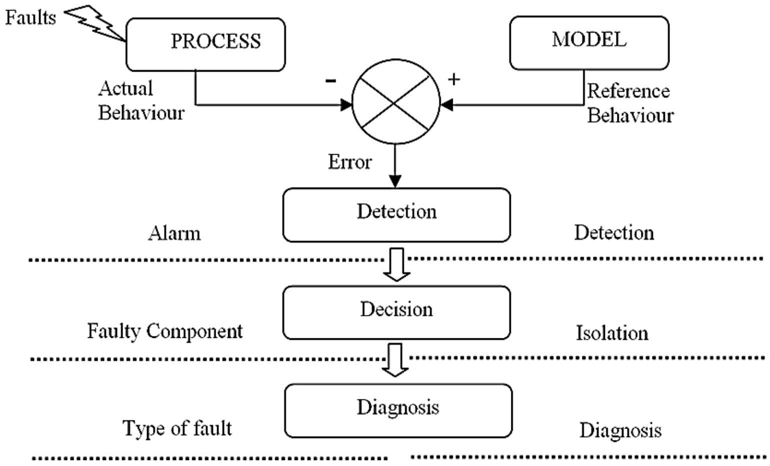

Fig. 4. Concept of model-based fault detection and isolation.

3.2 | Fault Detection Algorithm

The following steps outline the proposed fault detection algorithm:

I. Data collection: Real-time IoT data is continuously streamed to the cloud.

II. Supervised learning: Known fault types are classified based on historical data.

III. Unsupervised learning: Anomalies in real-time data are detected using clustering techniques.

IV. Fault identification: Faults are identified and flagged for further action.

V. Action: Alerts are generated, and automated recovery mechanisms are triggered.

Example of a fault detection algorithm flow:

I. Input: IoT sensor data is fed into the system.

II. Preprocessing: Data cleaning and normalization.

III. Feature extraction: Relevant features (e.g., temperature, voltage) are identified.

IV. Model: Supervised or unsupervised learning models analyze the data.

V. Anomaly detection: Any deviation from standard patterns triggers fault detection.

VI. Output: the system flags an anomaly and triggers an alert or automated action.

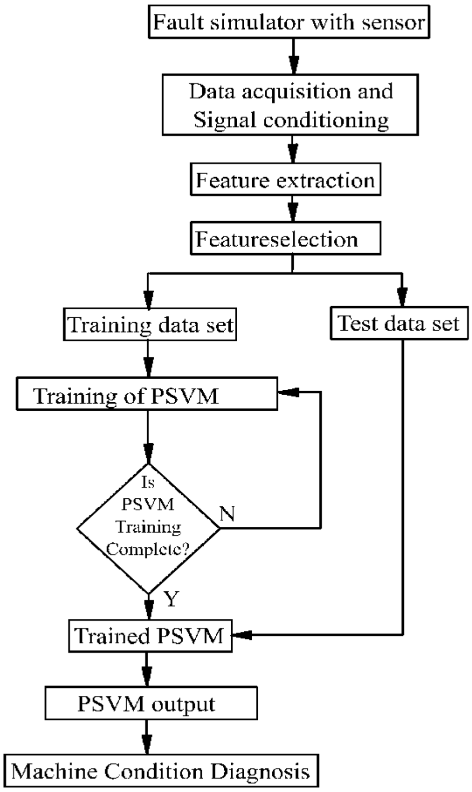
Fault simulator with sensor
Data acquisition and Signal conditioning
Feature extraction
Featureselection
Training data set
Test data set
Training of PSVM
Is PSVM Training Complete?
N
Y
Trained PSVM
PSVM output
Machine Condition Diagnosis

Fig. 5. Fault detection algorithm workflow.

Internet of things dataset characteristics

In AI-based fault detection in IoT cloud systems, the quality and scope of the dataset play a crucial role in training the models and achieving accurate fault detection results. The effectiveness of machine learning and deep learning models depends on the volume, variety, and veracity of the data collected from IoT devices. A well-structured dataset allows models to learn and distinguish normal behavior patterns from anomalies or faults.

For fault detection, the dataset typically includes sensor readings, event logs, performance metrics, and historical fault data collected from various IoT devices in real-time. These devices could range from temperature sensors in industrial settings to smart home devices connected via cloud platforms. The data is often time-series, reflecting continuous monitoring of environmental conditions or device performance.

Key characteristics of the dataset, such as the number of devices, data points collected, data types, and the volume of data, directly influence the performance of the AI models. A comprehensive dataset ensures that the model can accurately identify known faults (using supervised learning) and unknown anomalies (using unsupervised learning).

Table 3. Internet of things dataset characteristics.

Parameter	Value
Number of IoT devices	1,000
Data points collected	500,000
Fault types recorded	10
Data types	Sensor data, log files, system alerts
Average data size	100 GB

4 | Results and Analysis

4.1 | Evaluation Metrics

The model was evaluated using the following performance metrics:

I. Precision: Percentage of correctly identified faults.

II. Recall: Ability of the model to detect all relevant faults.

III. F1 score: Harmonic mean of precision and recall.

Table 4. Performance metrics of different artificial intelligence models.

Model	Precision	Recall	F1 Score	Training Time (min)
Decision tree	89%	88%	88.5%	30
SVM	91%	90%	90.5%	45
K-means clustering	85%	80%	82.5%	20
Autoencoders	93%	92%	92.5%	50
Hybrid AI model	95%	93%	94%	40

4.2 | Experiment Results

The proposed hybrid AI model was tested on a dataset from various IoT devices integrated into a cloud system. The model demonstrated a high degree of accuracy in detecting both known and unknown faults.

The results significantly improved over traditional rule-based systems, particularly regarding recall and fault identification speed.

5 | Discussion

Implementing AI in fault detection for IoT cloud systems presents several advantages.

The hybrid approach provides:

I. Improved accuracy: By combining supervised and unsupervised learning, the model can detect known and unknown faults accurately.

II. Scalability: The model adapts well to an increasing number of IoT devices without significant performance loss.

III. Real-time detection: Faults can be detected and addressed in real-time, minimizing downtime and preventing major system failures.

However, there are challenges regarding computational overhead and the need for large datasets to train the supervised models. Future work could explore methods to further optimize the model for real-time applications.

Table 5. Advantages and challenges of artificial intelligence techniques for fault detection.

Technique	Advantages	Challenges
Supervised learning	High accuracy for known fault types	Requires labeled data
Unsupervised learning	Can detect unknown anomalies	Less accurate for known faults
Hybrid models	Combines strengths of both approaches	Computationally intensive

6 | Conclusion

AI-based fault detection in IoT cloud systems offers a promising solution to system reliability and fault tolerance challenges. The hybrid model proposed in this paper integrates supervised and unsupervised learning techniques to deliver superior performance in real-time fault detection.

By enhancing fault detection accuracy, reducing false positives, and improving response times, AI can play a key role in maintaining the stability of IoTs cloud systems as they continue to grow in complexity and scale.

Data Availability

All data are included in the text.

Funding

This research received no specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

[1] Parida, B. R., Rath, A. K., & Mohapatra, H. (2022). Binary self-adaptive salp swarm optimization-based dynamic load balancing in cloud computing. *International journal of information technology and web engineering (IJITWE)*, *17*(1), 1-25. https://doi.org/10.4018/IJITWE.295964

[2] Botta, A., De Donato, W., Persico, V., & Pescapé, A. (2016). Integration of cloud computing and internet of things: a survey. *Future generation computer systems*, *56*, 684-700. https://doi.org/10.1016/j.future.2015.09.021

[3] Mohapatra, H., Rath, A. K., & Panda, N. (2022). IoT infrastructure for the accident avoidance: an approach of smart transportation. *International Journal of Information Technology*, *14*(2), 761-768. https://doi.org/10.1007/s41870-022-00872-6

[4] Altawaiha, I., Atan, R., Yaakob, R. B., & Abdullah, R. B. H. (2024). Assessing and prioritizing crucial drivers for cloudiot-based healthcare adoption: an analytic hierarchy process approach. *International Journal of Information Technology*, 1-18. https://doi.org/10.1007/s41870-024-01742-z

[5] Ghasemi Parvin, B., & Ghasemi Parvin, L. (2023). Applications of artificial intelligence in fault detection and prediction in technical systems. *14th international conference on recent developments in management and industrial engineering*. Figshare. http://dx.doi.org/10.6084/m9.figshare.25180289

[6] Thakfan, A., & Bin Salamah, Y. (2024). Artificial-intelligence-based detection of defects and faults in photovoltaic systems: A survey. *Energies*, *17*(19), 4807. https://doi.org/10.3390/en17194807

[7] Díaz, M., Martín, C., & Rubio, B. (2016). State-of-the-art, challenges, and open issues in the integration of Internet of things and cloud computing. *Journal of network and computer applications*, *67*, 99–117. https://doi.org/10.1016/j.jnca.2016.01.010

[8] Stergiou, C., Psannis, K. E., Kim, B. G., & Gupta, B. (2018). Secure integration of IoT and cloud computing. *Future generation computer systems*, *78*(Special Issue), 964–975. https://doi.org/10.1016/j.future.2016.11.031

[9] Botta, A., de Donato, W., Persico, V., & Pescapé, A. (2016). Integration of cloud computing and internet of things: A survey. *Future generation computer systems*, *56*(Special Issue), 684–700. https://doi.org/10.1016/j.future.2015.09.021

[10] Uppal, M., Gupta, D., Juneja, S., Dhiman, G., & Kautish, S. (2021). Cloud-based fault prediction using IoT in office automation for improvisation of health of employees. *Journal of healthcare engineering*, *2021*(1), 8106467. https://doi.org/10.1155/2021/8106467

[11] Kashpruk, N., Piskor-Ignatowicz, C., & Baranowski, J. (2023). Time series prediction in industry 4.0: A comprehensive review and prospects for future advancements. *Applied sciences*, *13*(22), 12374. https://doi.org/10.3390/app132212374

[12] Ahmed, I., Jeon, G., & Piccialli, F. (2022). From artificial intelligence to explainable artificial intelligence in industry 4.0: A survey on what, how, and where. *IEEE transactions on industrial informatics*, *18*(8), 5031–5042. https://doi.org/10.1109/TII.2022.3146552

[13] Lu, Y. (2019). Artificial intelligence: A survey on evolution, models, applications and future trends. *Journal of management analytics*, *6*(1), 1–29. https://doi.org/10.1080/23270012.2019.1570365

[14] Noshad, Z., Javaid, N., Saba, T., Wadud, Z., Saleem, M. Q., Alzahrani, M. E., & Sheta, O. E. (2019). Fault detection in wireless sensor networks through the random forest classifier. *Sensors*, *19*(7), 1568. https://doi.org/10.3390/s19071568

[15] Teles, G., Rodrigues, J. J. P. C., Rabelo, R. A. L., & Kozlov, S. A. (2021). Comparative study of support vector machines and random forests machine learning algorithms on credit operation. *Software: practice and experience*, *51*(12), 2492–2500. https://doi.org/10.1002/spe.2842

[16] Usama, M., Qadir, J., Raza, A., Arif, H., Yau, K. A., Elkhatib, Y., Hussain, A., & Al-Fuqaha, A. (2019). Unsupervised machine learning for networking: techniques, applications and research challenges. *IEEE access*, *7*, 65579–65615. https://doi.org/10.1109/ACCESS.2019.2916648

[17] Berahmand, K., Daneshfar, F., Salehi, E. S., Li, Y., & Xu, Y. (2024). Autoencoders and their applications in machine learning: A survey. *Artificial intelligence review*, *57*. https://doi.org/10.1007/s10462-023-10662-6