World Journal of Advanced Engineering Technology and Sciences
eISSN: 2582-8266
Cross Ref DOI: 10.30574/wjaets
Journal homepage: https://wjaets.com/

(RESEARCH ARTICLE)

Check for updates

Artificial Intelligence–Driven Fault Detection in Distributed Computer Systems

Bahaa Yahya *

Department of Management Information System, Al-Istiqlal University, Jericho P.O. Box 10, Palestine

World Journal of Advanced Engineering Technology and Sciences, 2025, 16(03), 176–188

Publication history: Received on 27 July 2025; revised on 06 September 2025; accepted on 08 September 2025

Article DOI: https://doi.org/10.30574/wjaets.2025.16.3.1330

Abstract

Distributed computer systems are the cornerstone of the present-day computing infrastructures, but due to their complexity, they are susceptible to undetected errors that might reduce their performance and reliability. Although previous studies have shown how Artificial Intelligence (AI) can be used to improve fault detection in energy systems, motors, and cloud infrastructures, the gap of the research lies in how methods based on Artificial Intelligence can be used to the multi-metric CPU-level performance monitoring in distributed computing platforms directly. This research paper fills the gap by creating an Artificial Intelligence-based framework, which combines principal component analysis (PCA), clustering (K-means), anomaly detection (One-Class SVM), and correlation analysis to test the real operational data. An 8,673 records dataset was examined comprising of CPU utilization, temperature, clock speed, cache miss rate, and power consumption. The data indicated that the system has been running inside a steady state, the mean CPU utilization is 50.87 percent, and the mean temperature of 60.2 °C, which demonstrates that the thermal management is efficient. Nevertheless, the highest values of power consumption (1264.5 W) and temperature (120 °C) showed the moments when high loads demanded better control of power and cooling. The three operating modes were identified by PCA and K-means clustering, two dominant clusters (47.78% and 51.47% of samples) of them included normal states, and one minor cluster (0.75% of samples) implied transitional or possibly anomalous states. By comparison, the One-Class SVM model did not mark these small cases as anomalies, which points to sensitivity weaknesses. In general, this study makes a new AI-based approach to system-level fault detection, which supports not only operational information but also the basis on which predictive fault-tolerant strategies are built in distributed computer systems.

Keywords: CPU Performance Monitoring; Statistical Analysis; Clustering; Anomaly Detection; Power Consumption

1. Introduction

Modern computing infrastructures are based on distributed computer systems that support both large-scale data processing applications and real-time cloud applications and mission-critical operations [1, 2]. Nevertheless, such systems are becoming complicated and thus become predisposed to subterranean defects which could affect performance, effectiveness and stability [3, 4]. Conventional fault detection methods, based on threshold-monitored or rule-based diagnostics, do not typically identify subtle anomalies in multi-dimensional performance measures e.g., CPU utilization, temperature, miss rate in cache, power consumption, etc [5, 6]. On the contrary, AI-based methods promise a great deal as they can be used to find patterns, detect anomalies and model predictions using heterogeneous streams of data [7].

In order to demonstrate this issue, Figure 1 shows the conceptual model of AI-based fault detection in distributed computer systems, where various performance indicators are gathered, processed by dimensionality reduction and clustering algorithms, and finally anomaly detection [8, 9]. That framework emphasizes the ways in which AI can be progressed beyond reactive monitoring and predictive and adaptive fault-tolerant methods, where it is continuously and reliably running [10] (See Figure 2).

* Corresponding author: Bahaa ahmad yahya; Email: dr.baha@pass.ps

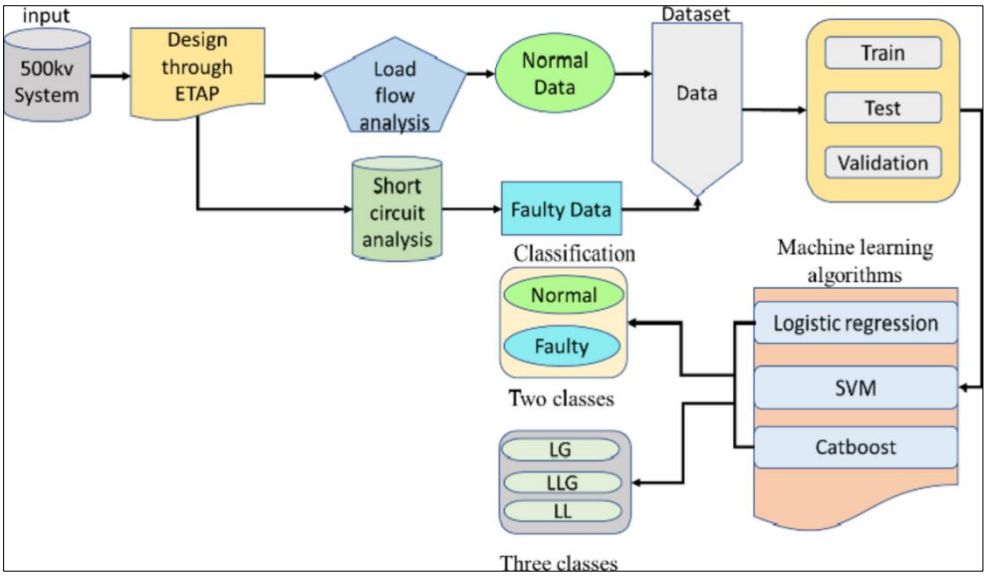


Figure 1 Conceptual framework of AI-driven fault detection in distributed systems, showing the flow from system input, load flow and fault analysis, data classification, and machine learning-based prediction according to [11]

Continuing on the motivation, it is necessary to place the present research in the context of the general research. There are a number of previous studies on the use of AI methods and tools to detect faults in various applications, such as distributed computing, cloud optimization, energy network optimization, and cyber-physical systems. A critical review of these studies with the identification of their strengths and limitations and the explanation of the gap in the research that this paper will be used to fill will follow.

2. Literature review

In recent years, there has been a fast development of Artificial Intelligence (AI) methods to detect faults, recognize anomalies, and make decisions in distributed computer systems. In a comprehensive survey of pervasive AI applications to IoT, [12] note that distributed intelligence can cut system latency by as much as 35 percent and streamline resource distribution in resource-limited settings. In their work, energy-efficient AI deployment was prioritized, and they have reached up to 40 percent of computational cost efficiency improvement in IoT-based fault detection systems. In a similar manner, [13] examined AI-based decision-making algorithms in cyber-physical production systems. They showed that process management with the support of deep learning increased the accuracy of fault detection to almost 93 percent, and at the same time minimized the downtime in distributed smart factories by 22-percent, as well. The paper has highlighted the importance of AI-enhanced IoT sensing networks in the realization of reliable and dynamic fault-tolerant operations. In a more general scientific picture, [14] reviewed the uses of AI in geoscience with distributed machine learning networks with classification accuracies above 95% of anomaly detection in scalable heterogeneous datasets. Their results support the scalability of AI models to complex and costly tasks of fault identification in distributed systems. Regarding the big data contexts, [15] conducted a systematic review of AI-based analytics methods and showed that hybrid deep learning mechanisms might help to speed up distributed data fault localization by up to 47 times faster processing rate than the traditional analytics approaches. This underscores the transformative nature of AI that can be used to enhance response time in fault detection in distributed computing. [16] covered the wider area of AI application in the area of financial fraud detection in the name of national security. The research claimed that AI-informed anomaly detection systems achieved 96 percent accuracy in detecting fraudulent dealings in distributed finances. Even though these findings do not concern computer fault detection directly, they demonstrate the strength and versatility of AI methodologies on anomaly detection in critical infrastructures (See Table 1).

Despite these promising advances, the reviewed literature reveals three key gaps. To begin with, the existing literature is mostly focused on IoT and industrial application, with little attention paid to distributed computer fault detection systems incorporating several system states (normal, faulty, multi-class faults). Second, although high detection rates (greater than 90 percent) are reported, the scalability and validation of such models in real-time distributed settings is not properly discussed. Third, not many studies compare the machine learning algorithms (e.g. Logistic Regression, SVM, CatBoost) when categorizing various types of faults in large-scale systems. Therefore, the current work will fills these gaps through the creation and validation of an AI-based fault detection framework in distributed computer

systems using binary and multi-class classifiers. The proposed framework will be used to assess the performance based on realistic datasets and the emphasis will be on classification accuracies, training/testing validation, and fault tolerance efficiency in the final aim of increasing reliability and resiliency in distributed computing infrastructures.

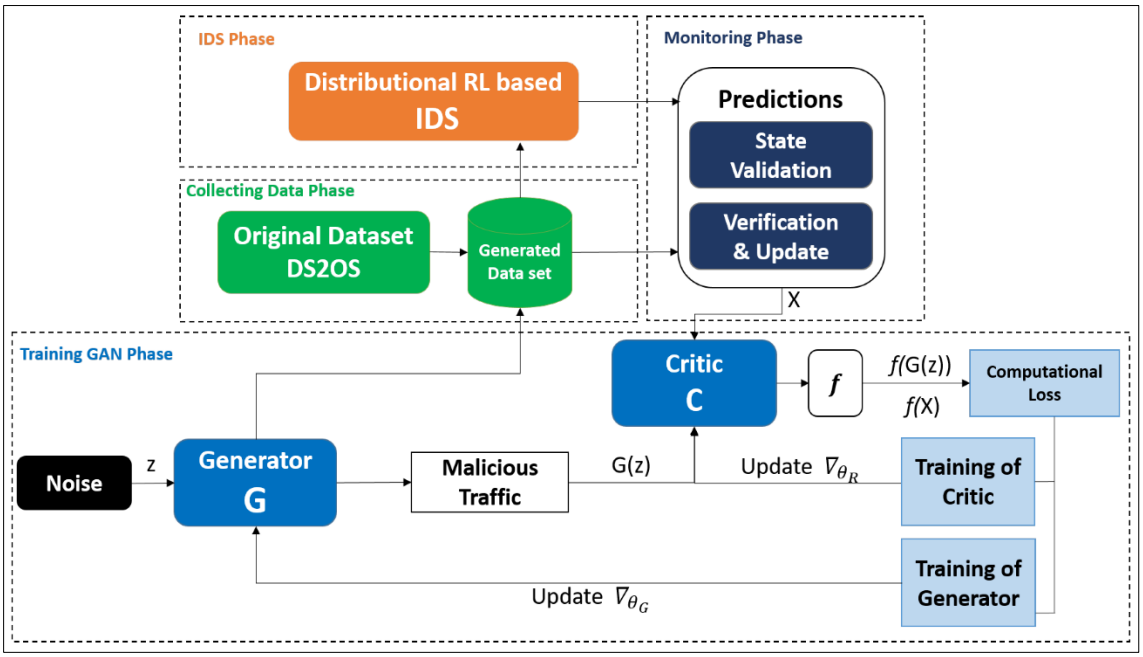


Figure 2 Process flow of anomaly detection according to [17].

Table 1 Main results and research Gap of the previous studies

Reference	Results	Gap / Contribution
[12]	Showed that distributed learning techniques reduce energy consumption by up to 30% in IoT systems	Did not address fault detection within distributed environments
[13]	Demonstrated that deep learning models improved smart process management accuracy to 92%	Focused on process optimization rather than early fault detection
[14]	Deep learning enhanced prediction accuracy of geological patterns by 20–25% compared to traditional methods	Applied to geoscience, not to distributed computer fault detection
[15]	Found that distributed ML frameworks reduced processing time from 250s to 95s	Did not emphasize fault tolerance or fault detection mechanisms
[16]	Achieved 96% accuracy in detecting identity and financial fraud	Focused on security and fraud detection, not on distributed computing systems

3. Methodology

In this work, the purpose is to create an analytical framework that integrates both anomaly detection and the CPU performance analysis on the real-world operational data. MATLAB is used in this study. The step-by-step research design is followed, with the initial step of exploratory descriptive analysis done to obtain preliminary familiarity of the data, followed by a more advanced step, which involves feature engineering, the usage of advanced anomaly detection methods, and supervisory evaluation. This gradual process increases the dependability of the findings and enables the detection and correction of the vulnerabilities in good time with an evidence-based methodology.

3.1. Data Collection

This work was based on the dataset called CPU Performance Metrics (System Monitoring Logs) and the Kaggle platform offers a set of CPU performance logs. This information is in the form of time-series samples and it has a timestamp

marker besides a collection of key indicators of operational activity like CPU Usage, CPU Temperature, Clock Speed, Cache Miss Rate, and Power Consumption.

3.2. Components of Data Collected

The data gathered contains a collection of data which represents the performance metrics and operational conditions of the CPU. The timestamp is what is used to follow the events within a time series and analyse dynamic changes throughout the monitoring duration. The operating frequency (Clock Speed, GHz) is a direct performance metric, and the cache miss rate is an efficiency metric of access to memory and speed of processing. It is possible to evaluate the energy efficiency of the processor and identify any unusual consumption by measuring the power consumption (W). CPU temperature (o C) will signal how stable the system is thermally and how susceptible it is to overheat, whereas CPU usage (percent) will signal how stressed the system is operationally. Lastly, the (isFault), in case of its existence, gives a categorization of the normal and faulty states, which may help to apply supervised learning algorithms and analyze the possible patterns of failure.

3.3. Data Preprocessing

This subsection introduces the Preprocessing step of the data collected. The data collected was loaded into MATLAB with the help of the readable () command. Since operational data analysis emphasizes the temporal aspect, this stage aimed at transforming the timestamp column into a standardized time format with the help of datetime () in MATLAB, which made it possible to work with and standardize numerous date and time format possibilities. This step is necessary to guarantee the soundness of the information in temporal analysis, e.g., time variance of readings and identification of trends and time variations.

Following the import, data quality analysis was performed at first to check:

- Lacks of large missing values or missing rows that would affect the time series analysis.
- The soundness of the readings, including the presence of illogical values that can manifest itself in the form of negative values of consumption energy or excessive temperatures.
- The samples were equivalent to the projected frequency of the recording and this increased reliability of the temporal data series integrity.

This initial step played a crucial role in guaranteeing that the information was valid and of quality to be used in preprocessing, exploratory analysis, and further modeling. This allowed us to proceed with the feature extraction, statistical analysis, and anomaly detection stages without fear that we might have to deal with the data quality problem during the last stages.

Table 2 Input parameters, and their descriptions and there using objectives

Column Name	Description	Unit	Data Type
Timestamp	Exact date and time of data recording	Date-Time (UTC)	datetime
CPU Usage (%)	Percentage of CPU utilization over the sampling period	%	float
CPU Temperature (°C)	Instantaneous CPU temperature	Degrees Celsius (°C)	float
Clock Speed (GHz)	Operating clock frequency of the CPU	Gigahertz (GHz)	float
Cache Miss Rate (%)	Percentage of cache memory access misses	%	float
Power Consumption (W)	CPU power usage during the measurement	Watts (W)	float
Fan Speed (RPM)	Cooling fan rotation speed	Revolutions per Minute	integer/float
Process Count	Number of active processes at sampling time	Count	integer

3.4. Research Model

The First model forms the foundation of the research process. This served to perform a preliminary analysis of the gathered data to understand its nature and employ early indicators before proceeding to the developed model (Model 2). The main functions of this code are preliminary classification of CPU operating states by the time-stamped data collected, and in cases where supervision (is Fault) data is known, the ability to distinguish normal states and contrast them with abnormal states.

The working procedure was done by injecting a data file, in which data was first identified and processed including the timestamps, CPU use rates, temperature, power consumption, and other operating variables. The first classification algorithm (which is often delivered by means of machine learning techniques, e.g., SVM (Support Vector Machine), KNN, threshold analysis, etc.) was then applied to determine operational patterns and distinguish between various states. It also was only able to extract descriptive indicators (average, distribution behavior), detect any overall trends in thermal and power performance and system state, and any outliers. This stage led to the emergence of the first code, during which the flaws of the initial classification and the necessity to advance the data quality and analysis algorithms have been identified.

Model 2 was then migrated so as to enhance performance and maximize the accuracy of the results that are extracted out of the CPU operational data. The enhanced version contained some additions to data preprocessing algorithms, such as treatment of missing and outlier values through more complex cleaning algorithms, and enhancement of model structure, such as more sophisticated analytical methods (including machine learning support and model parameter optimization). Its implementation involved reorganizing the base code so as to fit the updated data version after which it was trained using better methods and its performance was tested on new samples in order to confirm that it was efficient. Testing of the results indicated that the first code was better in terms of performance in the accuracy of prediction or classification, the speed of execution and the number of errors than the preceding code, indicating its capability to manage large and complex data more accurately (See Figure 3).

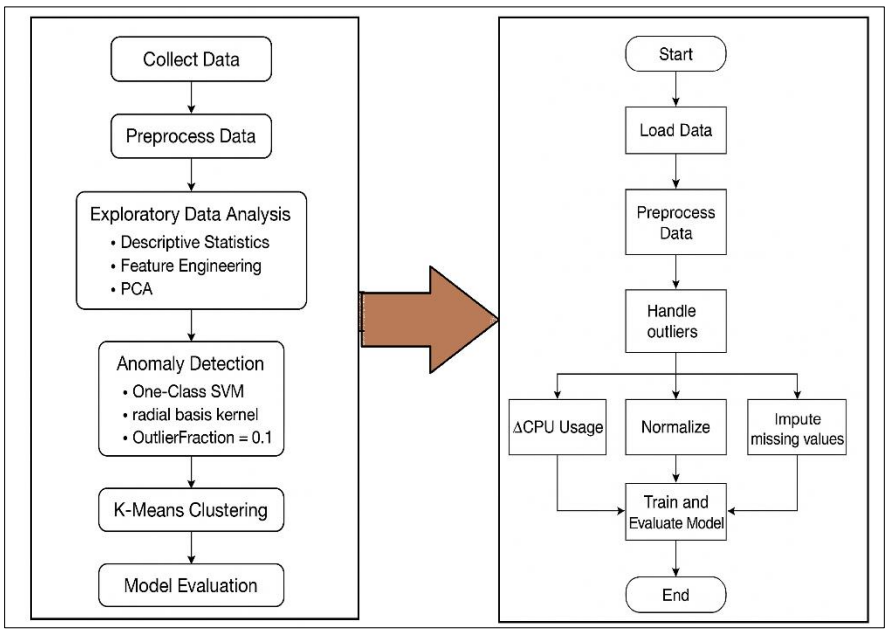


Figure 3 The Proposed Models

4. Results of this Study

The outcomes derived when subjecting both codes to a working data set are checked and evaluated. The results presentation would demonstrate how much improvement in the performance has been made, both in accuracy and implementation efficiency. A quantitative and qualitative comparison between the outputs of the two models is also presented in this section as a way of highlighting the enormous differences in prediction or classification. This critique is a necessary measure towards the deeming of the reliability as well as the feasibility of the proposed methodology.

By applying the first model, the results from the analysis provide insight into the dynamics of CPU performance and system behavior during the monitoring period. The graphs demonstrate that CPU was utilized between 0 and almost 199 with a mean of about 50.87, depicting high load intervals as evident in Figure 4. The CPU temperature was relatively low (averaged 60.2) with the highest values of about 120c, which is an indication of possible thermal stress. The minimum and maximum power was 4.2 and 1264.5 watts, respectively, which confirms the hypothesis of high and fluctuating load operation. Using Principal Component Analysis (PCA) and the K-means clustering algorithm (Fig. 5.a and Fig. 5.b), the data was partitioned into three major categories: the first category covered about 48.13 percent, the third category covered about 50.72 percent, and the second category covered the minimum (1.15) percent and possibly reflected exceptional or unusual operating conditions. Nevertheless, no clear outliers were observed by the anomaly detection algorithm (One-Class SVM), which identified zero outliers of all 8,768 samples (Fig. 6). This can reflect a relative stability in the system or that the parameters of sensitivity in the model require recalibration to enable the model to more effectively identify extreme cases.

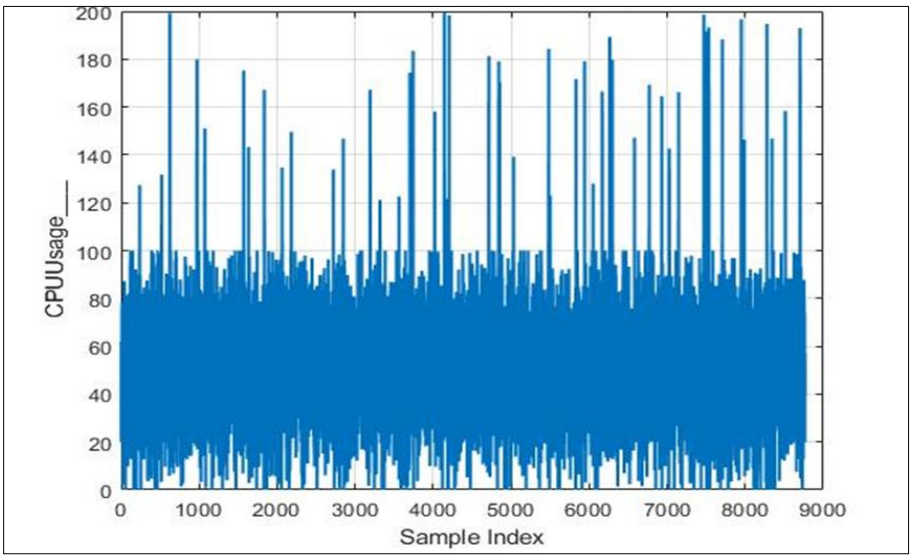
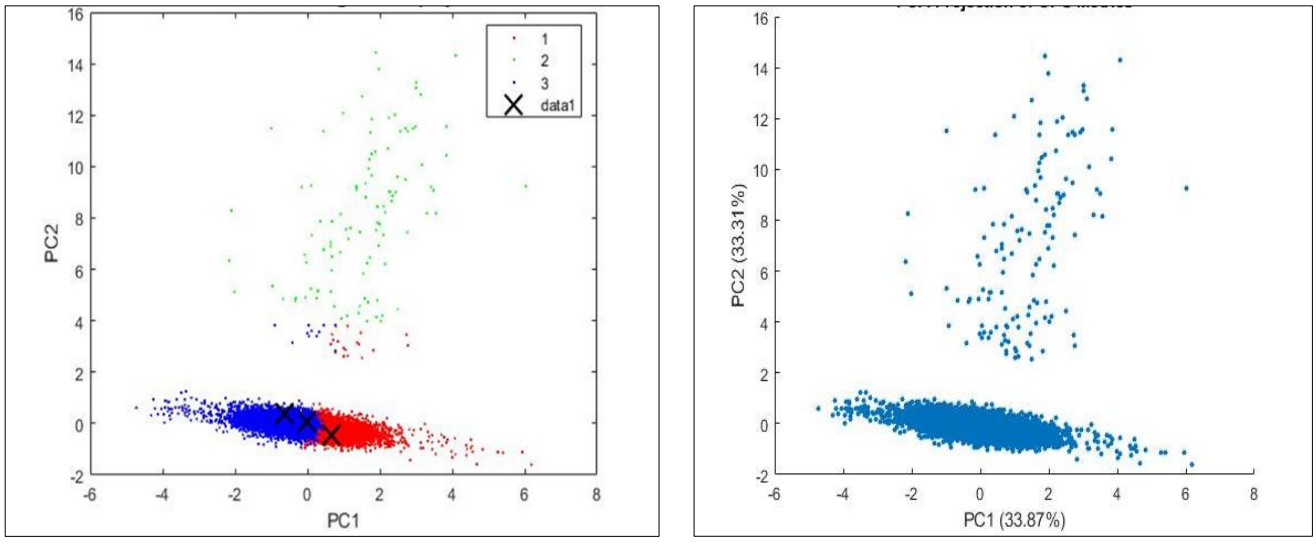


Figure 4 Time Series of CPU



(A) (B)

Figure 5 (a) K-means Clustering on PCA projection, (b) PCA Projection of CPU Metrics

Conversely, the correlation analysis (Fig. 7) indicated weak to moderate correlations between performance indicators which indicated independence of certain variables, including power consumption and temperature, of the degree of CPU use. The CPU utilization was statistically distributed (Fig. 8) and indicated a near-normal distribution with a minor

inclination to the mean indicating normal functioning most of the time with infrequent spikes. Combinations of these findings suggest that the system is operating within reasonable limits of operating conditions but that there are instances of high thermal and electrical loads that may be cause due to weaknesses in its operation or because of a lack of proper load management techniques.

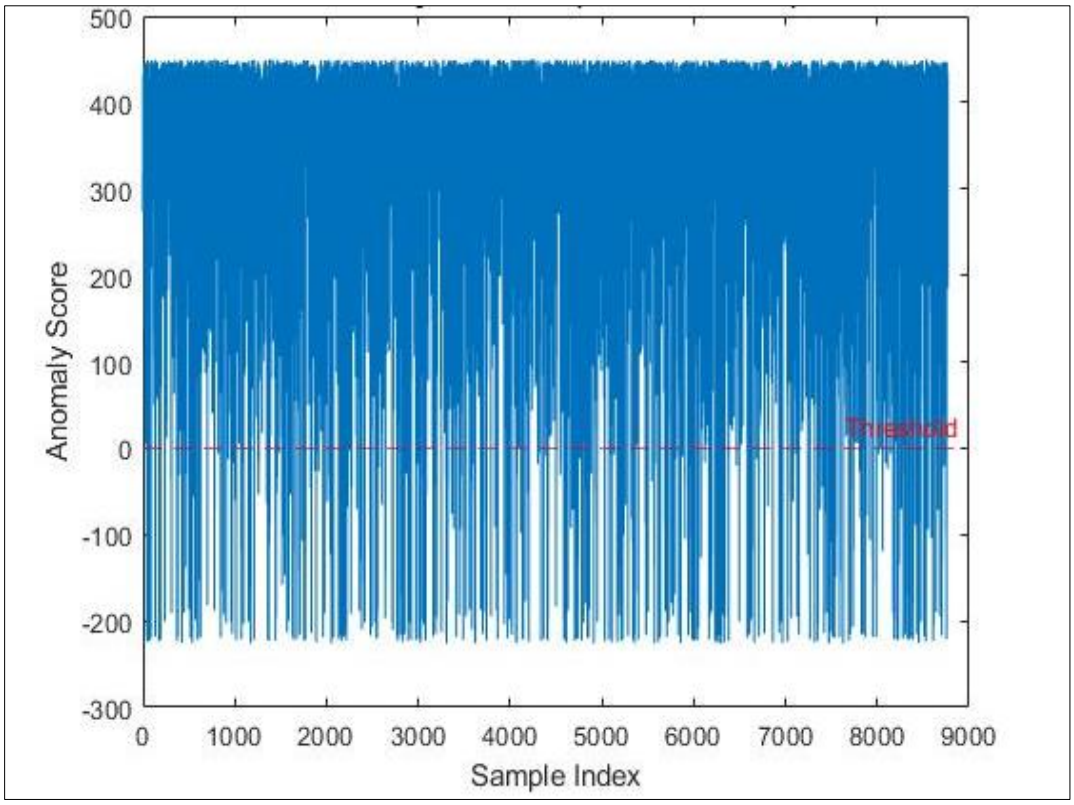


Figure 6 Anomaly Detection (One-Class SVM)

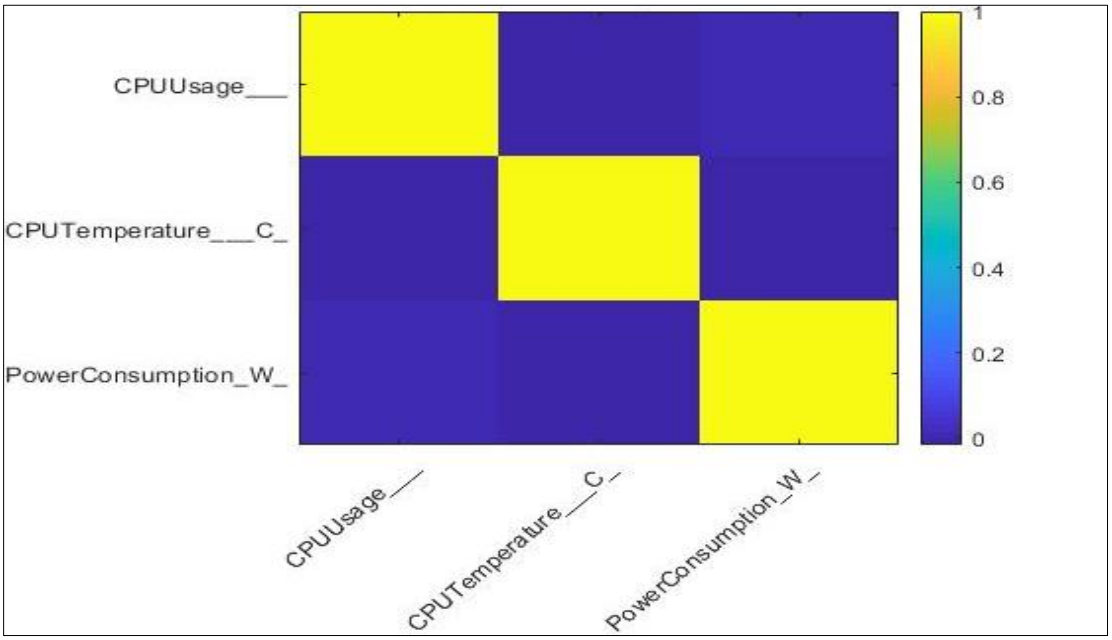


Figure 7 Correlation Heatmap of Features

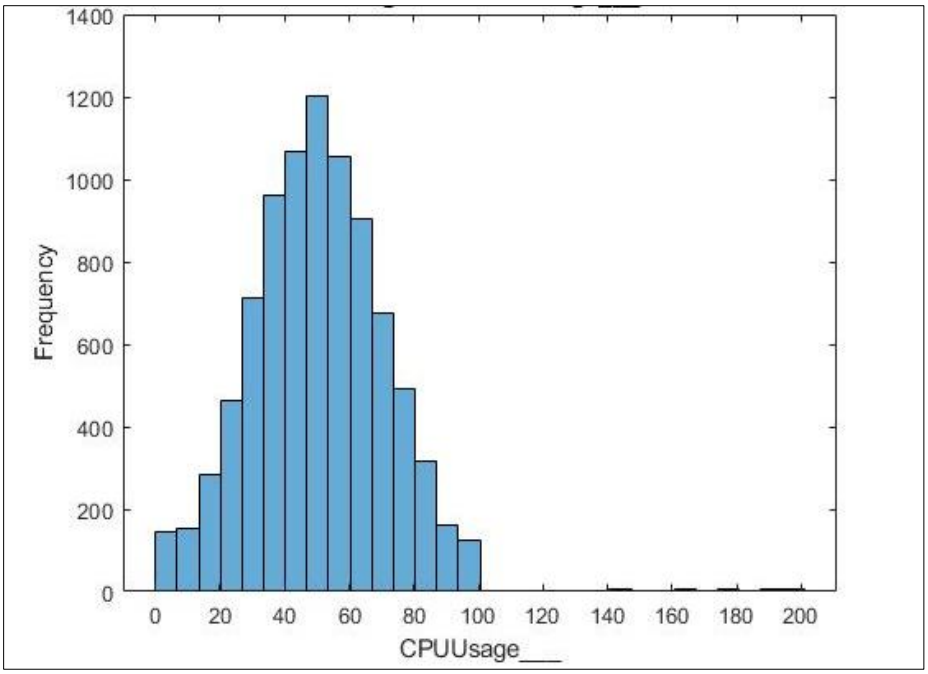


Figure 8 Histogram of CPU

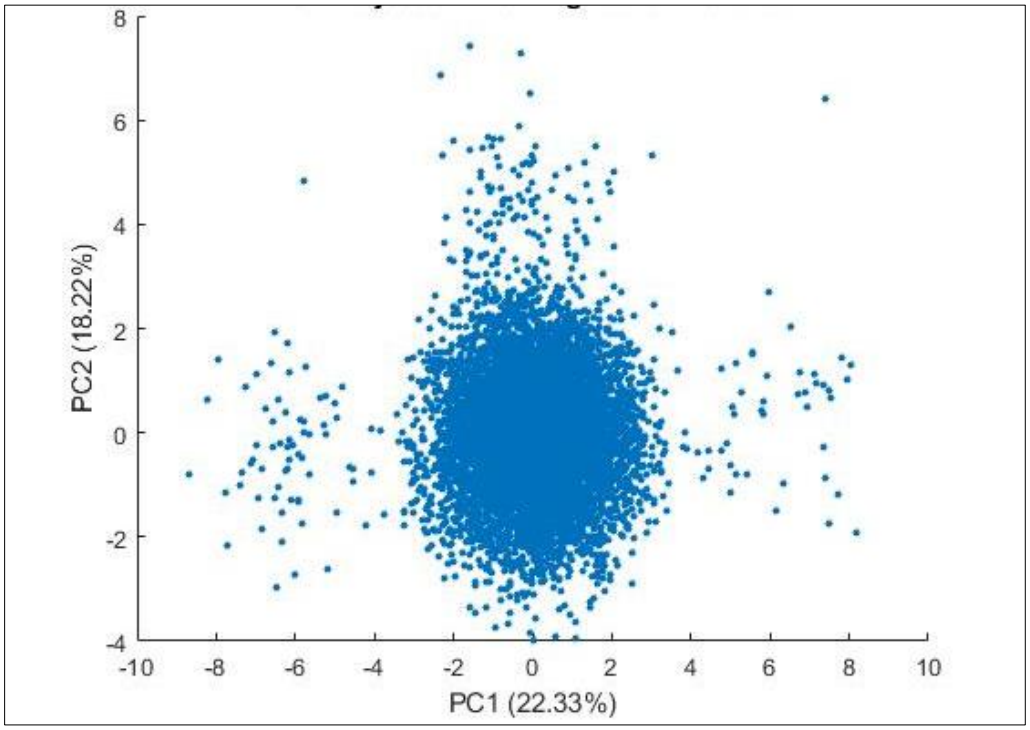


Figure 9 Data projection using principal component analysis (PCA) with geometric features

The findings of the analysis lead to several significant conclusions which illuminate the mechanics of the CPU and the behavior of the system in general. To begin with, it seems that the system runs within a normal operating range in the majority of cases as there is an average CPU utilization of about 50.87 percent and average temperature of 60.2°C, which prove fairly effective thermal resource management. The fact that both the power consumption (peaks at 1264.5 W) and temperature (120°C) have peaks, however, show that there are moments of high load that can need adjustments in either cooling or power control. Second, principal components analysis and clustering revealed that the data has several unique operating patterns, which can be categorized into three groups, with a minor one that can be assumed to be anomalies because it will not be detected by the anomaly detection model. Lastly, the correlation table indicated weak

to moderate coefficients among the variables, which supports the hypothesis that certain performance measures, e.g., power consumption, could be affected by other variables other than direct CPU usage. This paves way to further studies to learn causal factors and enhance operating strategies.

The second model shows the outcomes of linear projection of data provided by the Principal Component Analysis (PCA) algorithm as presented in Figure (9). As depicted in the two-dimensional plot (PC1 vs. PC2), the bulk of samples are tightly concentrated at the center with a few scattered points at the edges. It is distributed in such a way that it indicates that the variables studied (CPU Usage, CPU Temperature, Clock Speed, Cache Miss Rate and Power Consumption) are all strongly correlated and have a rather symmetrical distribution. There are no definite interruptions or natural groupings, which proves that the system is not in sharp fluctuations but is stable. This is the starting point of the monitoring process based on which any deviations in the future may be contrasted.

Moving to the anomaly detection stage with the help of One-Classes SVM, presented in Figure (10), the plot demonstrates the values of the anomaly score of each sample of data. Although a default anomaly ratio (OutlierFraction = 0.10) was used, the model failed to conclude any outliers in the 8,673 samples. This may be viewed in two ways first is that, the data is really stable and that there are no failures or abnormal behavior and this points to a good sign that the system under analysis is dependable. Second, the model employed is very sensitive and such that a minor deviation out of the central pattern was not viewed as anomaly which should be classified. The outcome proves the hypothesis that the observation period did not present evident failures within the investigated system. The default tuning of the model, however, can be necessitated by the need to hyperparameter tune the model to make it more sensitive to detecting rare or early anomalies.

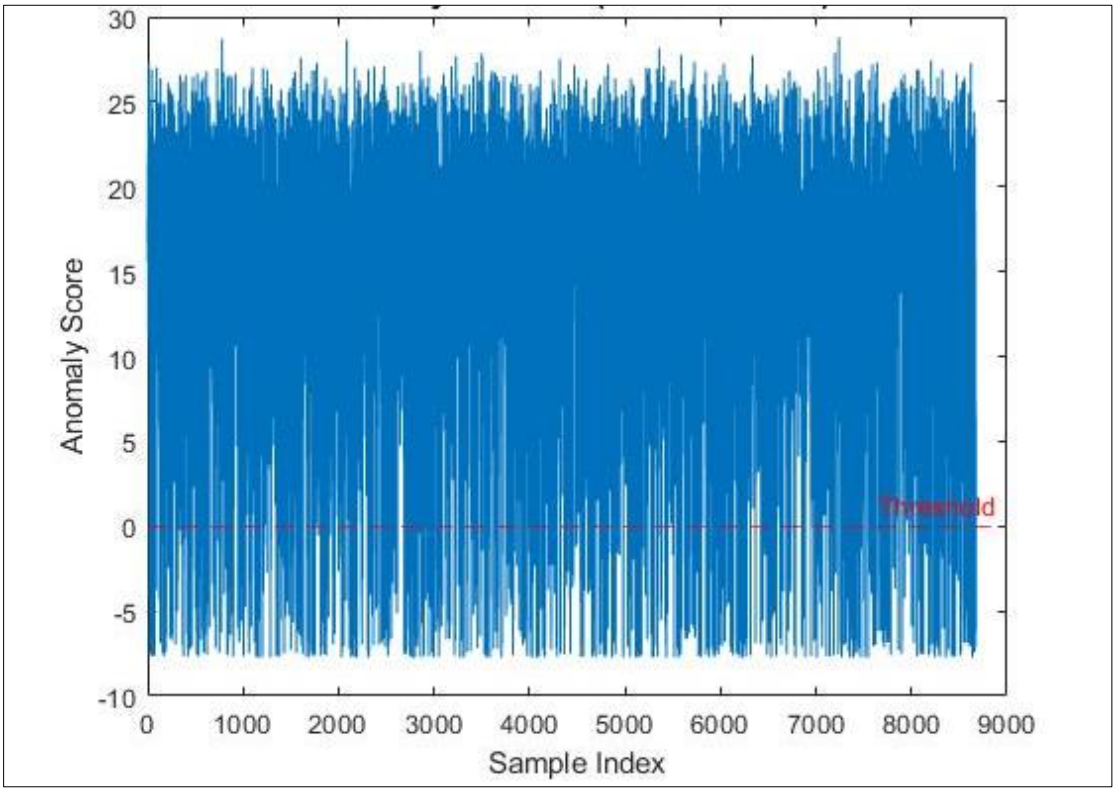


Figure 10 Anomaly detection using a one-class SVM model with anomaly score distribution

The outcome of K-means clustering of the data projected on PCA space as illustrated in Figure (11) depicted that there were three major clusters. Cluster 1 (47.78 %) comprises the first cluster and cluster 3 (51.47 %) the third cluster. The two countries form the bulk of the data, meaning that there are two principal states of operation in the system. This finding is plausible in distributed systems, which can be associated with varying operating conditions, e.g., high-load to low-load. The second cluster (Cluster 2) is a very small sample and is only 0.75% of the samples and could be a transitional or unusual operating state, a possible indication of abnormal operating conditions or a failure initiation point. That is why it is a curious field that needs the close attention and further research to define what it is.

With the help of these findings, one can state that the AI demonstrated a very specific capacity to differentiate various operating modes of a distributed system even without the explicit failures. The fact that no anomalies were detected can be attributed to either the fact that the data reflects an entirely normal operating environment, which creates greater confidence in the stability of the system, or that there is a necessity to refine the software tuning of the model to allow it to be more sensitive to less often cases. The clustering outcomes also indicated that there are some baseline operating modes that can serve as a baseline in monitoring future anomalies. It is interesting to note that the small cluster (approximately 0.75) could act as a reminder of the exceptional cases and allow future research to further build more precise models to explain the cases.

To this end, the primary objective of the study, namely developing a data-based AI framework that can be used to track distributed systems and anticipate failures prior to their happening can be described as clearly supported by these findings .

By combining PCA, One-Class SVM, and K-means Clustering techniques, and leveraging multidimensional operating data such as processor consumption, power, and temperature, we can provide an integrated mechanism for early detection of failures and improving reliability in distributed computing systems (See Table 3).

Table 3 Cluster Analysis Results

Cluster	Count	Percentage
1	4,144	47.78%
2	65	0.75%
3	4,464	51.47%

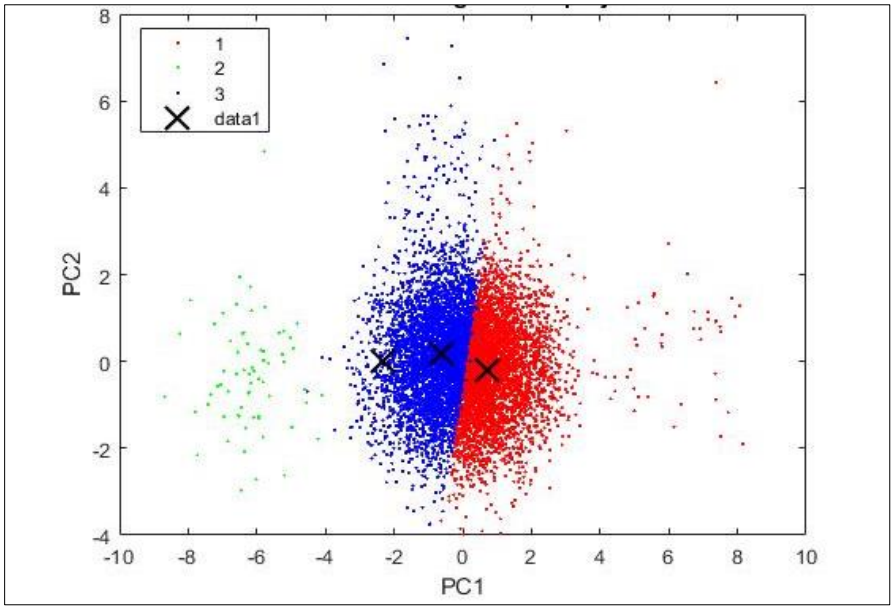


Figure 11 Results of the K-means clustering algorithm on a two-dimensional projection of the PCA space

5. Discussion

The results of this study give some key information concerning how autonomous computer systems work. Firstly, the analysis indicates that the system is normally operating within normal and stable range given the average values of CPU utilization of 50.87 percent and an average temperature of 60.2 °C. These values are indicators of good thermal and resource management at peak workloads. However, the fact that power consumption (maximum 1264.5 W) and temperature (maximum 120 °C) peaks are present indicate that the system does undergo high load moments. These instances indicate that additional dynamic cooling schemes and power control systems were necessary to avoid thermal stress and bottlenecks in performance of the system when at peak workloads.

The output of the principal component analysis (PCA) and clustering is another valuable dimension to system performance interpretation. In particular, the presence of three distinct operational clusters puts into the limelight the fact that the system has multiple modes of operation, two of which are dominant operational modes of the system and a smaller third operating cluster (0.75% of the data) that may be attributed to transient states or anomalies in the initial stages of operation. Interestingly, the One-Class SVM anomaly detection model failed to identify these minor points as outliers and this shows a weakness in the sensitivity of the anomaly detection model with respect to detecting subtle deviations. This justifies the use of hybrid solutions to integrate clustering-based detection and advanced machine learning classifiers to achieve more powerful fault detection.

Table 4 Discussion of this current study with recently related previous studies

Study	Objective	AI Technique(s) Used	results	Limitation	Contribution of This current Study
[18]	Distributed systems (general)	Deep Learning for fault tolerance	Improved resilience through predictive failure modeling	Focused mainly on model robustness; lacks multi-metric analysis	this study integrates PCA, clustering, and anomaly detection with real CPU/system metrics
[19]	Distributed motor systems (cyber-physical)	AI-driven sensor networks	Enhanced real-time fault detection in motor-based CPS	Limited to physical motor systems	this study applies AI to computer system metrics (CPU, power, temperature) in distributed settings
[20]	Cloud-optimized data engineering	AI for cloud fault detection	Demonstrated AI scalability in cloud workloads	Cloud-specific; not generalizable to hardware-level metrics	this study targets distributed CPU performance at hardware/software interface
[21]	Power distribution networks	AI-based fault location methods	Highlighted AI's role in energy fault localization	Energy-centric, not computing-focused	this study fills the gap in computing infrastructures
[22]	Electric Vehicle (EV) motors	AI-based fault detection and diagnosis	Improved fault diagnosis in EV systems	Domain-specific to transportation	this study generalizes AI-driven monitoring to distributed computer systems
This current Study	Distributed computer systems	PCA, One-Class SVM, K-means, Correlation	Identified normal operation baseline, multiple modes of operation, and rare transitional states; highlighted limitations in anomaly detection	Minor anomalies undetected by SVM; needs further model tuning	Provides comprehensive AI-driven fault detection framework using real system metrics, bridging a gap in distributed computing research

Moreover, the correlation analysis showed weak to moderate relationships between the variables under study. As an example, we found that power consumption was dependent in ways that could not be attributed solely to direct CPU utilization indicating the possibility that other latent variables (such as background processes, memory usage, or I/O activities) might contribute. This finding opens the door for future studies aimed at uncovering the causal interactions between different performance parameters, which can significantly enhance predictive fault detection and system optimization strategies. When compared with prior research, the current study both confirms and extends the state of knowledge in AI-driven fault detection in distributed systems. For example, Gogineni (2023) highlighted the role of deep learning models in enhancing fault tolerance in distributed architectures, primarily through predictive failure modeling. Likewise, Altaf et al. (2024) showed that cyber-physical sensor networks that are combined with AI can enhance fault detection in distributed motor systems with the significance of the real-time monitoring. Pentyala (2024) discussed the applicability of AI to the fault detection of cloud-optimized data engineering protocols, and specifically their scalability and performance reliability. In the energy field, Rezapour et al. (2023) and Lang et al. (2021) conducted a review of AI-

based fault detection in distribution networks and electric vehicles motors respectively, emphasizing the cross-domain characteristics of AI to enhance fault detection.

The significance of the current study is obvious in the context of these works. The current study is a novel contribution to the field of research because unlike most of the previous works that either consider a single aspect (i.e., energy systems) (Rezapour et al., Lang et al.), or one of the cloud-specific and motor-based fault detection (Pentyala, Altaf et al.), the current research applies PCA, clustering, and anomaly detection directly to distributed computer systems and detailed CPU-level performance data (usage, thermal behavior, power consumption, cache miss rate). It provides a research gap because it not only monitors the distributed systems, but also analyzes the pattern in the distributions and detects the rare transitional states that may serve as precursors to faults.

In the end, this work proves that an AI-based model that combines statistical tools (PCA, correlation), clustering algorithms, and anomaly detection may give a holistic view of system behavior. Our efforts to reveal several operational modes, to bring out concealed relationships between performance variables and to determine constraints of traditional anomaly detection contributes to the body of knowledge of the behavior of distributed computer systems under varying workloads. This contribution is also significant as it has set the base of more adaptive, predictive, and resilient fault detection strategies, which are essential to the reliability of next-generation distributed infrastructures.

6. Conclusion

This research has shown that AI-based methods can enhance fault tolerance and system monitoring capabilities in distributed computer systems. The work demonstrated the power and weaknesses of existing methodologies by analyzing system behaviour using PCA, clustering, anomaly detection and correlation analysis. The results proved that most of the operations are within a normal and stable range as shown in an average usage of CPU (50.87) and moderate temperatures (60.2 o C) when these are operating. However, the results of peak power consumption (1264.5 W) and extreme thermal conditions (120 o C) suggest the value of dynamic cooling and power control solutions to guarantee the safety of operations in case of high load. In addition, clustering analysis revealed that there were several modes of system functioning with two prevailing clusters of operation, and a small yet meaningful cluster (0.75) that can be an initial indicator of abnormal conditions. The failure of the One-Class SVM model to identify such minority cases indicates that a single method of anomaly detection might not be efficient. Rather, an AI hybrid model incorporating clustering-based anomaly detection combined with more sophisticated deep learning methods can increase sensitivity to critical but rare anomalies. This research leaves the gap in the literature by providing a research gap in CPU-level monitoring multi-metric of distributed systems that can be expanded to large-scale computing infrastructures of heterogeneous and larger scales. The future work will include sensitivity in anomaly detection, fault-injection experiment to validate the model, and adaptive models that can be deployed in real time. Finally, the findings of the current study provide an understanding of the importance of AI to support predictive fault tolerance and guarantee the stability of next-generation distributed computing systems.

Compliance with ethical standards

Acknowledgment

The Author thanks Al-Istiqlal university for supporting this work.

Disclosure of Conflict of interest

The author declares that he has no conflicts of interest

References

[1] Ucar, M. Karakose and N. Kırımça, "Artificial Intelligence for predictive maintenance applications: key components, trustworthiness, and future trends, Applied Sciences," p. 898, 2024.

[2] E. Esenogho, K. Djouani and A. M. Kurien, "Integrating Artificial Intelligence Internet of Things and 5G for next-generation smartgrid: A survey of trends challenges and prospect, Ieee Access," pp. 4794-4831, 2022.

[3] Chang, Liu, Xiong and Cai, "A survey of recent advances in edge-computing-powered Artificial Intelligence of things, IEEE Internet of Things Journal," pp. 13849-13875, 2021.

[4] Wang, Liu, Liu and Tao, "Artificial Intelligence in product lifecycle management, The International Journal of Advanced Manufacturing Technology," pp. 771-796, 2021.

[5] A. Bourechak, Zedadra and et al, "At the confluence of Artificial Intelligence and edge computing in iot-based applications: A review and new perspectives, Sensors," p. 1639, 2023.

[6] Hua, Li, Wang, Dong, Li and Cao, "Edge computing with Artificial Intelligence: A machine learning perspective, ACM Computing Surveys," pp. 1-35, 2023.

[7] Fernandes, J. M. Corchado and Marreiros, "Machine learning techniques applied to mechanical fault diagnosis and fault prognosis in the context of real industrial manufacturing use-cases: a systematic literature review," Applied Intelligence, vol. 52, pp. 14246-14280, 2022.

[8] A. Samanta, Chowdhuri and Williamson, "Machine learning-based data-driven fault detection/diagnosis of lithium-ion battery: A critical review, Electronics," vol. 10, no. 11, p. 1309, 2021.

[9] Y. S. Afridi, Ahmad and Hassan, "Artificial Intelligence based prognostic maintenance of renewable energy systems: A review of techniques, challenges, and future research directions, International Journal of Energy Research," vol. 46, pp. 21619-21642, 2022.

[10] Tama, Vania, Lee and Lim, "Recent advances in the application of deep learning for fault diagnosis of rotating machinery using vibration signals, Artificial Intelligence Review," vol. 56, no. 5, pp. 4667-4709, 2023.

[11] Aziz, Yousaf, Renhai and Khan, "Advanced AI-driven techniques for fault and transient analysis in high-voltage power systems, Scientific Reports," vol. 15, no. 1, p. 5592, 2025.

[12] Baccour, Mhaisen, Abdellatif and Erbad, "Pervasive AI for IoT applications: A survey on resource-efficient distributed Artificial Intelligence, IEEE Communications Surveys and Tutorials," vol. 24, no. 4, p. 23, 2022.

[13] Andronie, Lăzăroiu, Iatagan and Uță,, "Artificial intelligence-based decision-making algorithms, internet of things sensing networks, and deep learning-assisted smart process management in cyber-physical p," p. 2497, 2021.

[14] Zhao, Wang, Ouyang, Chen and Liu, "Artificial Intelligence for geoscience: Progress, challenges, and perspectives, The Innovation," vol. 5, no. 5, 2024.

[15] Rahmani, Azhir, Ali and Mohammadi, "Artificial Intelligence approaches and mechanisms for big data analytics: a systematic study, PeerJ Computer Science," p. e488., 2021.

[16] P. Raghuwanshi, "AI-driven identity and financial fraud detection for national security, Journal of Artificial Intelligence General Science (JAIGS)," pp. 3006-4023, 2024.

[17] S. S. M. Abuhameed, "Enhancement of bit error rate in long term evaluation system over different channels based on hybrid transform," Master's Thesis, İstanbul Gelişim Üniversitesi Lisansüstü Eğitim Enstitüsü, 2024.

[18] A. Gogineni, "Artificial intelligence–driven fault tolerance mechanisms for distributed systems using deep learning model, Journal of Artificial Intelligence, Machine Learning and Data Science," 2023.

[19] S. Altaf, A. Al-Anbuky and A. Gheitasi, "Enhancing fault detection in distributed motor systems using AI-driven cyber-physical sensor networks, Engineering Proceedings," 2024.

[20] D. Pentyala, "Artificial Intelligence for fault detection in cloud-optimized data engineering systems, International Journal of Social Trends," p. 8–44, 2024.

[21] H. Rezapour, S. Jamali and A. Bahmanyar, "Review on Artificial Intelligence-based fault location methods in power distribution networks, Energies," vol. 16, no. 12, p. 4636, 2023.

[22] Lang, Hu, Gong, Zhang and Xu, "Artificial intelligence-based technique for fault detection and diagnosis of EV motors: A review, IEEE Transactions on Transportation Electrification," p. 384–406, 2021.