



OPEN ACCESS

EDITED BY

Oluwakayode Onireti,
University of Glasgow, United Kingdom

REVIEWED BY

Avishek Sinha,
Dr. B. R. Ambedkar National Institute of
Technology Jalandhar, India
Boukhedouma Saida,
University of Science and Technology Houari
Boumediene, Algeria
J. Geetha,
Ramaiah Institute of Technology, India
Premkumar Sivakumar,
Galgotias University, India

*CORRESPONDENCE

William Villegas-Ch,
✉ william.villegas@udla.edu.ec

RECEIVED 13 February 2025

ACCEPTED 07 April 2025

PUBLISHED 24 April 2025

CITATION

Govea J, Gutierrez R and Villegas-Ch W (2025)
Use of granular computing for resource
optimization in IoT networks.
Front. Commun. Netw. 6:1575120.
doi: 10.3389/frcmn.2025.1575120

COPYRIGHT

© 2025 Govea, Gutierrez and Villegas-Ch. This is an open-access article distributed under the terms of the [Creative Commons Attribution License \(CC BY\)](https://creativecommons.org/licenses/by/4.0/). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

Use of granular computing for resource optimization in IoT networks

Jaime Govea, Rommel Gutierrez and William Villegas-Ch*

Escuela de Ingeniería en Ciberseguridad, FICA, Universidad de Las Américas, Quito, Ecuador

The exponential growth of connected devices on the Internet of Things (IoT) has transformed multiple domains, from industrial automation to smart environments. However, this proliferation introduces complex challenges in efficiently managing limited resources—such as bandwidth, energy, and processing capacity, especially in dynamic and heterogeneous IoT networks. Existing optimization methods often fail to adapt in real-time or scale adequately under variable conditions, exposing a critical gap in resource management strategies for dense deployments. The present study proposes a granular computing framework designed for dynamic resource optimization in IoT environments to address this. The methodology comprises three key stages: granular decomposition to divide tasks and resources into manageable grains, granular aggregation to reduce computational load through data fusion, and adaptive granular selection to refine resource allocation based on current system states. These techniques were implemented and evaluated in a controlled industrial IoT testbed comprising over 80 devices. Comparative experiments against heuristic and AI-based baselines revealed statistically significant improvements: a 25% increase in processing throughput, a 20% reduction in energy consumption, and a 60% decrease in error rate. Additionally, quality of service (QoS) reached 95%, and latency was reduced by 25%, confirming the effectiveness of the proposed model in ensuring robust and energy-efficient performance under varying operational loads.

KEYWORDS

granular computing, resource optimization in IoT, energy efficiency in networks, internet of the things, networks

1 Introduction

In today's digital era, the interconnection of smart devices through the Internet of Things (IoT) has catalyzed significant transformations in multiple sectors, from the manufacturing industry to the home environment (Catarinucci et al., 2015). The ability of these devices to offer a diversity of applications, from remote monitoring to the automation of complex processes, has opened new avenues for technological innovation. However, the exponential growth in connected devices presents unprecedented challenges in efficiently managing critical resources such as bandwidth, power, and processing capacity. These challenges are especially pronounced in infrastructure and resource-constrained environments, where operational efficiency and service quality are critical (Abir et al., 2021).

The resource optimization problem in IoT networks has emerged as a crucial field of study due to the need to maximize operational efficiency and guarantee service quality in

these densely connected environments (Shwe et al., 2016). As IoT applications continue to diversify and expand, it becomes increasingly necessary to develop innovative approaches that enable more efficient and adaptive management of available resources (Neto et al., 2024).

However, despite the growing body of research, there remains a critical gap in applying granular computing to real-time resource optimization in dynamic IoT environments. Existing approaches often rely on static or semi-static resource allocation schemes that lack adaptability to fluctuations in network load, heterogeneity of devices, and operational priorities. Furthermore, prior studies do not sufficiently address the computational overhead, scalability trade-offs, or dynamic reconfiguration required by dense IoT ecosystems. This study directly tackles these challenges by proposing a granular computing-based framework capable of decomposing and adapting resource allocation across multiple abstraction levels in real time.

Although existing literature proposes various techniques and strategies to address these challenges, many solutions face limitations regarding scalability, flexibility, and adaptability to changing environments. Granular computing is a promising approach that can offer more flexible and adaptive solutions for resource optimization in IoT networks (Motamedi et al., 2017).

In this work, granular computing is proposed as a tool for resource management in IoT networks (Rani et al., 2023). Granular computing, which relies on manipulating information at different levels of granularity, is explored to manage and allocate resources dynamically and efficiently. This methodology allows for adapting resource allocation in real-time in response to the fluctuating demands of the environment and connected devices (Mahan et al., 2021). The study illustrates how granular computing can significantly improve resource management through controlled experiments and relevant case studies.

The study aims to develop a dynamic resource management framework that uses granular computing techniques to adapt to varying network conditions. This approach focuses on reducing latency, improving energy efficiency, and enhancing IoT networks' overall quality of service (QoS). In addressing these goals, the study identifies and tackles key constraints inherent in IoT environments, such as limited bandwidth, power, and processing capacity. These constraints are critical for the effective implementation of resource optimization strategies. The research is guided by optimizing resource allocation while ensuring the system remains scalable and robust under varying loads and network configurations.

Specific metrics, including latency reduction, energy efficiency improvement, and error rate minimization, are established to evaluate the effectiveness of the proposed solutions. These metrics form the basis for comparing the performance of granular computing techniques against traditional methods, offering a clear perspective on the proposed approach's benefits and potential.

Granular computing has considerable potential to revolutionize resource management in IoT networks, providing a viable solution to the challenges imposed by modern applications' increasing complexity and demands (Webb et al., 2010; Panda and Abraham, 2014). Additionally, areas for future research are identified, particularly in integrating artificial intelligence and machine learning with granular computing to foster even more intelligent and autonomous systems (Wang et al., 2021).

Furthermore, this study explores applying granular computing techniques to optimize resource allocation, improve energy management, and enhance IoT networks' QoS. The proposal leverages granular decomposition, aggregation, and selection algorithms to adapt to real-time network conditions and demands.

Despite reviewing works on IoT resource management, integrating granular computing presents several innovative aspects. This includes handling data and resource heterogeneity more effectively and dynamically adjusting resource allocation based on real-time network conditions. Furthermore, this approach improves scalability and robustness, addressing the limitations of existing methods. Predictive maintenance and health management cannot be overlooked within the IoT paradigm—recent studies, such as those by Zhu et al. (2023) and C.-G. Huang (Behzadidoost et al., 2024) has demonstrated the practical importance of integrating IoT with predictive health monitoring systems. These studies highlight how IoT can be leveraged to predict equipment failures and optimize maintenance programs, ultimately reducing downtime and operational costs. Integrating granular computing with predictive health management further enhances the ability to process large volumes of sensor data, providing more accurate and timely predictions.

Therefore, this work focuses on optimizing resource management through granular computing while recognizing the broader implications and potential integrations with healthcare management systems in IoT. The aim is to provide a comprehensive framework adaptable to various IoT applications, improving operational efficiency and system reliability. The results demonstrate that granular computing enhances energy efficiency, reduces data transmission latency, and increases the processing capacity of IoT systems without compromising service quality. These findings are supported by detailed comparisons with traditional techniques, highlighting the significant advantages of granular computing regarding scalability and adaptability.

This research makes several significant contributions to the field of IoT resource optimization. It introduces a novel application of granular computing to dynamically adjust communication routes and allocate resources based on IoT devices' current workload and capabilities. The study also proposes the integration of granular ball computing (GBC) to enhance the precision and robustness of data processing and classification, thereby improving the overall efficiency of IoT systems. The study also develops and validates multi-objective optimization algorithms that balance energy efficiency and QoS, simultaneously addressing multiple goals. Finally, the research presents a comprehensive framework for implementing and evaluating granular computing techniques in real-world IoT environments, demonstrating significant improvements in operational performance and resource utilization.

2 Literature review

Numerous recent studies have addressed optimizing resource allocation in IoT networks, especially with the emergence of dynamic and intelligent systems that require real-time adaptability. Although applicable in stable contexts, traditional solutions based on heuristic models or predefined rules present limitations in dynamic and heterogeneous environments

(Demirpolat et al., 2021; Ansere et al., 2023). These constraints have driven the adoption of more sophisticated approaches, including predictive models and artificial intelligence (AI)-driven optimization frameworks.

Recent contributions emphasize the integration of AI and machine learning for intelligent resource allocation in IoT systems. For example, Alghayadh et al. (2024) propose reinforcement learning techniques to adaptively manage network resources adaptively, offering high responsiveness under real-time conditions. Likewise, Bolettieri et al. (2021) introduce predictive allocation models that leverage edge computing capabilities to reduce latency and improve scalability. Liu et al. (2024) present a comprehensive review of AI-based dynamic resource management techniques, reinforcing the trend toward autonomous and adaptive IoT systems.

In manufacturing and industrial IoT, dynamic resource allocation must consider operational constraints, energy consumption, and service level objectives. Su et al. (2022) demonstrate a model for dynamic allocation in production environments, considering machine state transitions and environmental impact. Delgado et al. (2017) extend this approach by proposing a data-driven model that adjusts the performance of NB-IoT networks based on the mobile context. These industrial implementations provide a realistic framework for evaluating novel methods such as granular computing.

Granular computing has emerged as a promising strategy for improving adaptability and efficiency in distributed systems. Tang et al. (2024) propose a unified framework based on implicational logic, which enables flexible data processing at multiple levels of abstraction. Loia et al. (2018) also explore granular methods to discover periodicities in data, which is crucial for predicting system demands and optimizing processing cycles. In the transportation field, Wang and Guo (2022) show how multi-granular decision-making improves performance under cognitive network paradigms, reinforcing the versatility of granular computing in various domains.

Other studies support this trend by introducing collaborative and energy-efficient mechanisms. Delgado et al. (2017) explore energy-aware resource allocation in virtual sensor networks, focusing on dynamic node coordination. Liu et al. (2024) propose a cross-level optimization strategy to balance AI processing between the edge and cloud layers, improving performance in AIoT systems. These approaches align with our research, which advocates granular computing to solve multi-tiered dynamic resource optimization.

In addition to these advances, efforts have been made to design joint optimization strategies. Lin, Cheng, and Li (Ansere et al., 2023) present a topology and power control model that significantly improves communication efficiency in IoT networks. Mele et al. (2022) apply unsupervised clustering techniques (DBSCAN) in infrastructure analysis, highlighting the relevance of adaptive clustering techniques. These methodologies are technically related to granular computing models' decomposition and selection phases.

These studies validate the need for architectures capable of adaptive resource allocation, real-time optimization, and multi-tiered data processing, which are the main strengths of the granular computing framework proposed in this work. Our research extends these contributions by incorporating a layered decomposition-aggregation-selection model experimentally validated in industrial IoT

environments, highlighting improvements in energy efficiency, quality of service, and operational scalability.

3 Materials and methods

3.1 Data collection

The proposal for resource optimization in IoT networks is framed in the industrial environment, where the interconnection of smart devices and data collection in real-time is essential to improving efficiency and productivity. IoT devices are strategically distributed across various industrial facilities, including manufacturing plants, warehouses, and production lines in this environment. The distribution of these devices is designed to cover critical operational areas, ensuring comprehensive data collection across different stages of the industrial process (Acampora and Vitiello, 2023).

Data was sourced from multiple avenues to ensure the study's representativeness and relevance. Public datasets were retrieved from academic repositories, and literature on IoT applications in industrial contexts (Wang et al., 2023). Additionally, proprietary data were collected directly from industrial environments, employing advanced monitoring and control systems specifically implemented to capture real-time operational metrics.

In manufacturing plants, IoT devices such as temperature sensors, vibration sensors, and energy meters were deployed to monitor equipment performance and energy consumption. These sensors were placed at critical points like motor housing, electrical panels, and conveyor belts to capture high-frequency data on equipment health and operational efficiency (Hussein and Mousa, 2020). In warehouses, RFID tags and environmental sensors were used to monitor inventory levels, material flow, and environmental conditions (temperature, humidity). The strategic placement of these sensors ensures real-time tracking of inventory movement and environmental control within the storage facilities. High-resolution cameras and machine-learning algorithms were installed along production lines to monitor product quality in real-time. Additionally, actuators were used to control and adjust machinery settings based on the data received from the sensors.

The data collected includes diverse variables essential for optimizing resource management in industrial environments. This encompasses machinery performance, including parameters such as vibration amplitude, rotational speed, and load torque; energy utilization data on power consumption, voltage, and current from machinery and lighting systems; product quality captured from the production line through high-resolution images and defect detection metrics; logistics and material flow data on the movement of goods within warehouses, including timestamps, locations, and handling conditions; and environmental monitoring data such as temperature, humidity, and air quality from sensors distributed across the facilities to ensure compliance with safety and operational standards.

The volume of data collected is substantial, totaling approximately 20 terabytes. The datasets range from 5 to 15 gigabytes on average, with some datasets reaching several terabytes due to extended monitoring periods and high sampling rates. The data were stored in formats conducive to efficient analysis and processing, including comma-separated values (CSV) for structured data from sensors and

actuators, JavaScript Object Notation (JSON) for hierarchical and complex data, particularly from monitoring systems, and relational databases for managing large-scale data from multiple sources, enabling efficient querying and analysis.

The careful design of the data collection process and the strategic distribution of IoT devices ensures that the data gathered is comprehensive, high-quality, and directly relevant to optimizing resources in industrial settings. This data provides a robust foundation for the subsequent analysis and application of granular computing techniques to enhance operational efficiency and productivity in IoT networks.

3.2 Data preprocessing

Preprocessing includes a series of steps to clean and prepare the data before analysis to ensure its quality and integrity. The data-cleaning process was methodically divided into multiple stages, beginning with identifying and eliminating outliers that could distort the analysis results. Outliers were detected using statistical techniques, explicitly calculating the data's standard deviation (σ) and interquartile range (IQR). Observations falling beyond 3σ from the mean or outside 1.5 times the IQR were classified as outliers and subsequently removed (Rani et al., 2024), as presented in Equation 1

$$\text{Outliers} = \{x | x < Q_1 - 1.5 \times IQR \text{ or } x > Q_3 + 1.5 \times IQR\} \quad (1)$$

Once the outliers were handled, missing values in the datasets were identified and input using appropriate techniques tailored to the nature of the data. Mean, median, or mode imputation was applied for numerical variables depending on the data distribution. For categorical variables, imputation was performed using rule-based approaches or predictive modeling, such as k-nearest neighbors (KNN) imputation, to preserve the distribution characteristics and relationships within the data. The imputation process was formalized as presented in Equation 2:

$$\hat{x}_i = \frac{1}{k} \sum_{j \in \text{KNN}(i)} x_j \text{ for KNN imputation} \quad (2)$$

Normalization techniques ensured comparability between variables originally on different scales and ranges (LFAO et al., 2023). Depending on the distribution of the variables, either min-max scaling or z-score standardization was applied. Min-max scaling transformed each variable x into a normalized variable x' within the range $[0,1]$, following the Equation 3:

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \quad (3)$$

Alternatively, for normally distributed data, z-score standardization was used to center the data around the mean and scale it according to the standard deviation, as presented in Equation 4

$$z = \frac{x - \mu}{\sigma} \quad (4)$$

These transformations ensured that variables were uniform, facilitating further analysis and improving the performance of algorithms sensitive to variable magnitudes.

Additional data preprocessing techniques included eliminating duplicate records to avoid introducing bias into the analysis. Duplicates were detected using key identifier fields and were removed systematically. Variables were also transformed using logarithmic functions or similar methods to improve their distribution, mainly when dealing with skewed data. The logarithmic transformation as presented in Equation 5:

$$x' = \log(x + 1) \quad (5)$$

This transformation reduced the data's skewness, bringing it closer to a normal distribution, which benefits many statistical analyses and machine learning algorithms.

Categorical variables were encoded using one-hot or ordinal encoding, depending on the nature of the categorical data. One-hot encoding was applied to nominal variables, creating binary columns for each category, while ordinal encoding was used for ordinal variables, preserving the intrinsic order of the categories. For instance, one-hot encoding transformed a categorical variable C with three categories into three binary variables, following the Equation 6:

$$C_1, C_2, C_3 \text{ where } C_i = \begin{cases} 1 & \text{if } C = i \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

The choice of data preprocessing techniques was meticulously aligned with the data's nature and the subsequent analysis's specific requirements. The overarching goal was to preserve data integrity while minimizing the introduction of bias during the cleaning and preparation stages. Each preprocessing step was carefully validated to ensure that it improved the overall data quality, thus enhancing the reliability and validity of the results obtained in the study.

3.3 Clustering-based preprocessing for granular computing

A clustering-based preprocessing phase was introduced as a preparatory step before the core decomposition to reduce the computational burden of granular computing in large-scale IoT networks and improve its operational effectiveness. This stage is critical in organizing the input data into homogeneous groups, enhancing resource allocation and parallel processing efficiency during granular decomposition.

The preprocessing began with aggregating real-time data collected from distributed IoT nodes. This included metrics such as device activity level, energy consumption rate, CPU usage, memory occupancy, and frequency of network interactions. All collected data underwent normalization using Z-score transformation to ensure comparability across features and devices and mitigate scale heterogeneity's effect in subsequent clustering.

Two unsupervised clustering algorithms, k-means and DBSCAN, were evaluated. For k-means, the optimal number of clusters (k) was determined using the elbow method, which analyses the inflection point in the curve of the within-cluster sum of squares (WCSS). The silhouette coefficient was employed to validate clustering quality further, measuring clusters' compactness and separation.

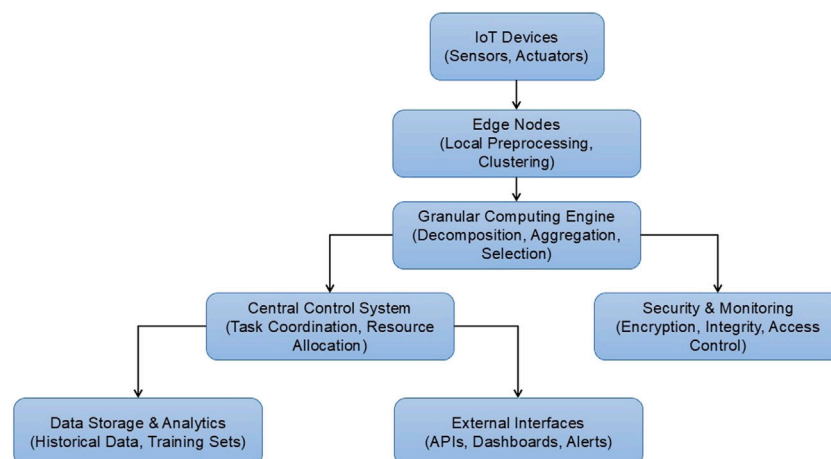


FIGURE 1
Functional architecture of the system for resource optimization in IoT networks through granular computing.

In parallel, DBSCAN was evaluated to determine whether a density-based approach would provide better adaptability to irregular device behavior and outlier identification. The parameters ϵ (epsilon) and *minPts* were determined using k-distance plots and density histograms, ensuring the chosen values reflected natural density gaps and minimized noise inclusion.

A grid search strategy was employed to tune the hyperparameters of both algorithms. The performance of each configuration was assessed according to three core criteria.

- Inter-cluster variance minimization is a proxy for intra-cluster cohesion.
- Computational time reduction as a direct indicator of efficiency.
- Decomposition efficiency improvement is based on how well the clustering enhanced the subsequent granular resource allocation process.

The final model configuration selected was k-means with $k = 6$, which provided the best trade-off between performance, computational cost, and segmentation quality. DBSCAN, while effective in specific high-noise scenarios, exhibited higher variability in cluster sizes and was less stable across IoT environments with fluctuating data patterns.

Once optimized, this clustering stage was embedded as a preprocessing module within the overall architecture. It enabled grouping IoT devices with similar behavior profiles, which were then subjected to granular computing processes in parallel, significantly enhancing scalability. Furthermore, grouping devices with similar load patterns allowed more targeted and balanced resource optimization strategies to be deployed, reducing redundancy and improving processing throughput in distributed edge and fog layers. This stage's technical contribution lies in computational overhead reduction and the alignment of data semantics, enabling more meaningful and context-aware granular decompositions.

3.4 System architecture and functional components

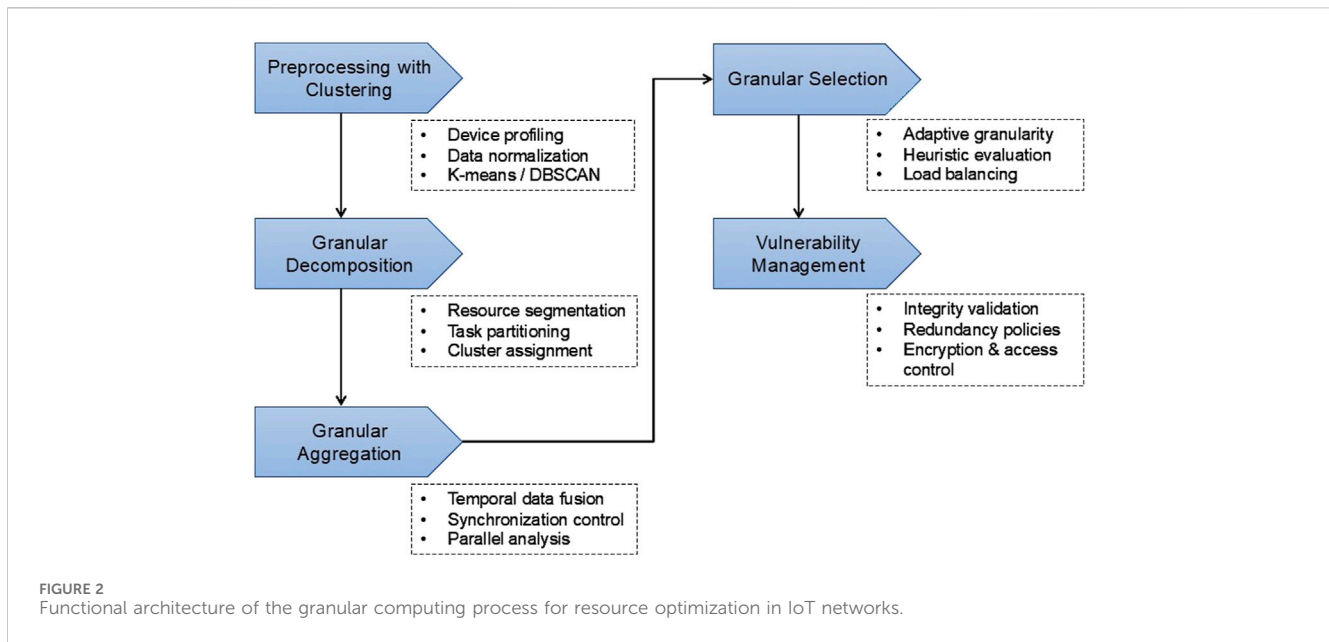
The proposed granular computing system for IoT networks' architecture is structured to ensure efficiency, scalability, and modularity. Figure 1 illustrates the complete system design, encompassing edge and centralized processing, granular operation monitoring, and external interfaces. The architecture follows a layered and distributed paradigm that facilitates localized decision-making and global optimization.

At the foundation are the IoT Devices, which include environmental sensors, actuators, and embedded devices deployed across the network. These devices serve as the primary data sources, continuously generating readings such as temperature, humidity, motion, and system status metrics. Each device can communicate essentially and is configured to transmit data to edge nodes based on predefined protocols.

The Edge Nodes perform critical local preprocessing tasks, including noise reduction, missing value imputation, normalization, and clustering. These nodes execute lightweight versions of the granular computing modules, offloading the central system and enabling real-time responsiveness. ClusterEdge Nodes perform critical DBSCAN and are applied to GRP-similar device profiles, enabling efficient data handling and reduced redundancy.

Preprocessed and clustered data is forwarded to the Granular Computing Engine, which encapsulates the core mechanisms of Granular Decomposition, Aggregation, and Selection. This engine operates on the edge and centralized levels depending on the granularity required and the system load. It segments resources and tasks into grains, fuses relevant data streams, and dynamically adjusts the processing granularity using optimization techniques and Markov Decision Processes (MDPs). It is the analytical core, adapting in real-time to environmental conditions and network demands.

The Central Control System is responsible for task orchestration, global resource management, and cross-grain coordination. It integrates results from the granular engine and orchestrates task allocation strategies across the system. It uses scheduling algorithms such as Earliest Deadline First (EDF) and



load balancing schemes like Weighted Round Robin (WRR) to prevent bottlenecks and optimize energy consumption. This layer also controls the execution of adaptation policies triggered by system state changes.

Supporting the control system, the Data Storage and Analytics module serves as a persistent layer for historical data, training sets, and model checkpoints. It facilitates long-term analysis, pattern detection, and training of predictive models using supervised and unsupervised learning techniques. This module enables using passata for simulations, forecasting, and continuous system refinement. Historical data, training sets, model checkpoints, and MoniMonitoringayer. This component enforces data encryppredictive model training integrity validation throughout the lifecycle of data and computations. It applies anomaly detection to identify unusual patterns in data streams. It implements redundancy mechanisms to maintain system resilience and monitoring interfaces with external actor data encryption, access control, integrity validation monitoring dashboards, and notification services that expose system status, performance metrics, and alerts to administrators, engineers, and other integrated applications. These interfaces support RESTful communication and are integrated with visualization tools for real-time feedback and control.

This architectural model enables distributed intelligence, ensures low-latency responses at the edge, and facilitates holistic coordination at the central level. The modular separation of roles across components supports scalability, fault tolerance, and maintainability, making the system suitable for deployment in large-scale, heterogeneous IoT environments.

3.5 Granular computing algorithms for resource optimization in IoT networks

Granular computing is a methodology that allows the processing of information at different levels of granularity, which is particularly

useful for handling complex and dynamic problems present in IoT networks. This concept uses granular computing to optimize resource management, offering flexible and adaptable solutions. Figure 2 illustrates the overall structure of the granular computing algorithm and outlines the sequential phases applied to optimize resource usage in IoT environments. This representation provides a clear overview of the interactions between preprocessing, decomposition, aggregation, selection, and vulnerability management, which are further developed in the following subsections.

3.5.1 Granular decomposition

Granular decomposition is a process by which resources and tasks are divided into smaller units, called “grains,” that can be managed independently. This process is carried out using an algorithm that identifies the characteristics and requirements of each task and resource and then groups them into homogeneous subsets.

The available devices and resources, including sensors, actuators, processing and storage devices, and the data they generate, must first be identified to implement granular decomposition in an IoT network. These devices and resources are then characterized based on their capabilities, geographic locations, and energy and processing requirements.

The next step involves creating a granularity model, which defines the levels of granularity necessary for decomposition. For example, in an environmental monitoring system, temperature sensors can be grouped based on their geographic proximity and the temperature range they cover. Clustering algorithms such as k-means or DBSCAN are commonly used to group sensors into clusters based on the similarity of their data and locations. The similarity computation used for clustering is detailed in Section 3.7, where a Euclidean distance-based approach is applied to the multidimensional attribute space of the devices.

K-means is particularly effective in scenarios where the number of clusters is predefined and where clusters exhibit spherical

symmetry, which is common when sensors have similar roles in spatially bounded regions. Additionally, its computational efficiency $O(n \cdot k \cdot t)$, where n is the number of points, k clusters, and t iterations, makes it well-suited for scalable implementations. On the other hand, DBSCAN is selected for its robustness to noise and its ability to detect clusters of arbitrary shapes without requiring a pre-specified number of clusters. This is especially relevant in dynamic IoT deployments with uneven device density and connectivity.

Although hierarchical clustering methods (e.g., agglomerative clustering) provide detailed tree-based relationships among devices, their computational complexity ($O(n^3)$) makes them less suitable for real-time, large-scale IoT scenarios. Moreover, their sensitivity to noise and the lack of reusability in incremental scenarios limit their practicality in the context of granular decomposition for adaptive systems.

Once the granularity groups are established, specific tasks are assigned to each grain. Task allocation is dynamically managed based on resource availability and current demand, ensuring optimal resource usage. For instance, when a group of sensors in a specific area detects temperature variation, a detailed analysis task can be assigned to identify the cause and adjust environmental control systems accordingly. This dynamic allocation ensures that resources are optimally used, and tasks are performed efficiently, reducing the risk of resource underutilization.

3.5.2 Granular aggregation

Granular aggregation is the process of combining individual grains into larger sets to perform joint processing. This process reduces computational complexity, improves system efficiency, and ensures consistent data synchronization in parallel computing environments. Granular aggregation employs algorithms that identify grains that can be processed together based on their characteristics and the nature of the tasks while ensuring that resources are utilized efficiently and consistently across the system.

The first step in granular aggregation is collecting data from individual grains. In an IoT network, this involves gathering data from sensors, actuators, and other connected devices. This data is stored in a centralized or distributed database depending on the system architecture. Synchronization mechanisms such as distributed locks or consensus algorithms (e.g., Paxos, Raft) are applied to maintain data coherence across multiple nodes. These mechanisms ensure all nodes can access consistent and up-to-date information, preventing issues such as stale data or race conditions during aggregation.

Once the data is collected, data fusion algorithms, such as Kalman filters or particle filters, combine the information from individual grains. These algorithms include statistical and machine learning techniques to identify patterns and correlations between data, ensuring that the aggregation process is efficient and preserves the integrity and relevance of the data. For example, in an energy management system, energy consumption data from different devices can be merged using a Kalman filter to provide an accurate and synchronized estimate of total energy usage, reflecting the system's most accurate and current state.

After data fusion, joint processing of the aggregated data is performed. This step may include trend analysis, predicting future demands, and optimizing resource allocation. The synchronization

of parallel tasks is managed using barrier synchronization or task scheduling algorithms such as the Earliest Deadline First (EDF) algorithm. These methods minimize idle time for computational resources, ensuring CPU and memory are utilized optimally during the aggregation and processing stages. For instance, in an HVAC system, temperature and humidity data from multiple sensors are aggregated and analyzed simultaneously, with parallel tasks being synchronized to adjust heating and cooling systems more efficiently and without resource underutilization.

The results of joint processing are used to make informed decisions about resource management, such as reassigning tasks, adjusting operating parameters, and implementing energy-saving strategies. Granular aggregation allows for maximizing available data and improving operational efficiency while ensuring the system's parallel computing components remain synchronized and coherent, preventing resource wastage and maintaining high system performance.

3.5.3 Granular selection

Granular selection is a process that determines the most appropriate level of granularity for processing based on current needs, system constraints, and the requirements for maintaining data synchronization and coherence in a parallel computing environment. This process is essential for the system's dynamic adaptation to changing environmental conditions and for optimizing the use of available resources without causing resource bottlenecks or idle states.

The granular selection process begins with continuously monitoring system health and environmental conditions. This includes collecting data on workload, resource availability, power consumption, and quality of service. The collected data is analyzed in real-time using algorithms such as Markov Decision Processes (MDP) that determine the optimal granularity level and manage the synchronization of tasks across different processing nodes. Load balancing and dynamic task allocation algorithms, such as the Weighted Round Robin (WRR) algorithm, ensure that resources such as CPU and memory are evenly distributed and fully utilized, reducing the likelihood of resource contention or idle times.

Based on this analysis, granular selection algorithms determine the optimal granularity level for processing. These algorithms incorporate mathematical optimization techniques and heuristics, considering multiple factors such as task priority, data criticality, resource constraints, and maintaining synchronization across the system. For example, in a high-workload IoT network, the algorithm may select a coarser granularity level to reduce the amount of data processed and improve response speed while ensuring that all processing tasks remain synchronized and coherent. Conversely, during periods of low activity, the algorithm may opt for a finer level of granularity to perform more detailed analysis and optimize system performance, ensuring that resources are not underutilized.

Once the optimal level of granularity is determined, the system's operating parameters are adjusted to reflect this selection. This may include reconfiguring communication paths, redistributing tasks, allocating additional resources, or scaling down resource usage to match the selected granularity level. Throughout this process, synchronization mechanisms are maintained to ensure that all system components operate coherently, with tasks executed in parallel without causing inconsistencies or resource

underutilization. Granular selection ensures the system can dynamically adapt to changing conditions, maintaining an optimal balance between efficiency, service quality, and resource utilization, even in complex parallel computing environments.

The **Pseudocode** below illustrates the algorithm implemented for dynamic granular selection. This approach leverages real-time system monitoring and combines Markov Decision Processes (MDP) with heuristic rules to determine the optimal level of granularity. The objective is to balance computational efficiency, service quality, and system responsiveness.

Pseudocode: Granular Selection Algorithm Based on MDP and Heuristics

Algorithm GranularitySelection

Inputs:

- SystemState \leftarrow {CPU_Load, Memory_Usage, QoS_Level, Resource_Availability}
- CurrentGranularityLevel.
- HeuristicRules \leftarrow {HighLoadThreshold, LowQoSThreshold, MemoryThreshold}
- MDPModels \leftarrow possible system state transitions.

Outputs:

- OptimalGranularityLevel

Begin:

1. Monitor current SystemState in real-time
2. Evaluate $MDP_Reward \leftarrow$ function (Efficiency, QoS, Latency)
3. For each possible GranularityLevel do:
 - a. Simulate state transitions using MDP
 - b. Compute ExpectedReward for that level.
4. Select GranularityLevel with highest ExpectedReward
5. Apply HeuristicRules:
 - a. If $CPU_Load > HighLoadThreshold \rightarrow$ select coarse granularity
 - b. If $QoS_Level < LowQoSThreshold \rightarrow$ select fine granularity
 - c. If $Memory_Usage > MemoryThreshold \rightarrow$ decrease granularity
6. Adjust OptimalGranularityLevel based on MDP_Reward and heuristics
7. Reconfigure the system with OptimalGranularityLevel
8. Return OptimalGranularityLevel

End

3.5.4 Managing complexity and development efforts

Implementing granular computing algorithms in IoT networks introduces increased task and resource management complexity, requiring a structured approach to their development and maintenance. Modular decomposition techniques were employed to manage this complexity, dividing the system into smaller, more manageable components, each focused on a specific granular computing task.

During development, continuous testing was implemented through an automated testing framework that allowed validation at each level of granularity. This ensures that errors are detected early in the development process, minimizing the risk of these errors

impacting system performance in production. In addition, periodic reviews of system components were performed, allowing any changes to the architecture or algorithms to be assessed regarding their impact on system complexity and efficiency.

Quality control models are implemented, focusing on test automation and peer review of critical modules. These reviews included verifying the consistency of granular operations and validating the results obtained at each processing stage.

The system incorporates edge computing strategies to address the computational overhead inherent in granular computing processes, particularly in large-scale and dynamic IoT networks. Specific tasks—such as local clustering, anomaly detection, and early-stage decision-making—are offloaded to edge nodes and gateway devices. This distributed processing architecture alleviates the load on central systems and enables faster, localized responses. Additionally, lightweight versions of the granular computing modules are deployed at the edge, allowing preliminary analysis and data reduction before forwarding to centralized systems. The system employs adaptive buffering mechanisms, load-aware scheduling, and task prioritization strategies to manage sudden spikes in device connectivity or data volume. These mechanisms ensure service continuity and prevent bottlenecks, even under volatile network conditions.

3.5.5 Mitigation of vulnerabilities

The increase in complexity in systems that employ granular computing also increases the possibility of vulnerabilities. To mitigate these risks, several measures were adopted to ensure the integrity and security of the system. One of the main strategies was data integrity validation, which was performed after each granular processing stage. This validation ensured the data was not incorrectly compromised or altered during decomposition, aggregation, or selection.

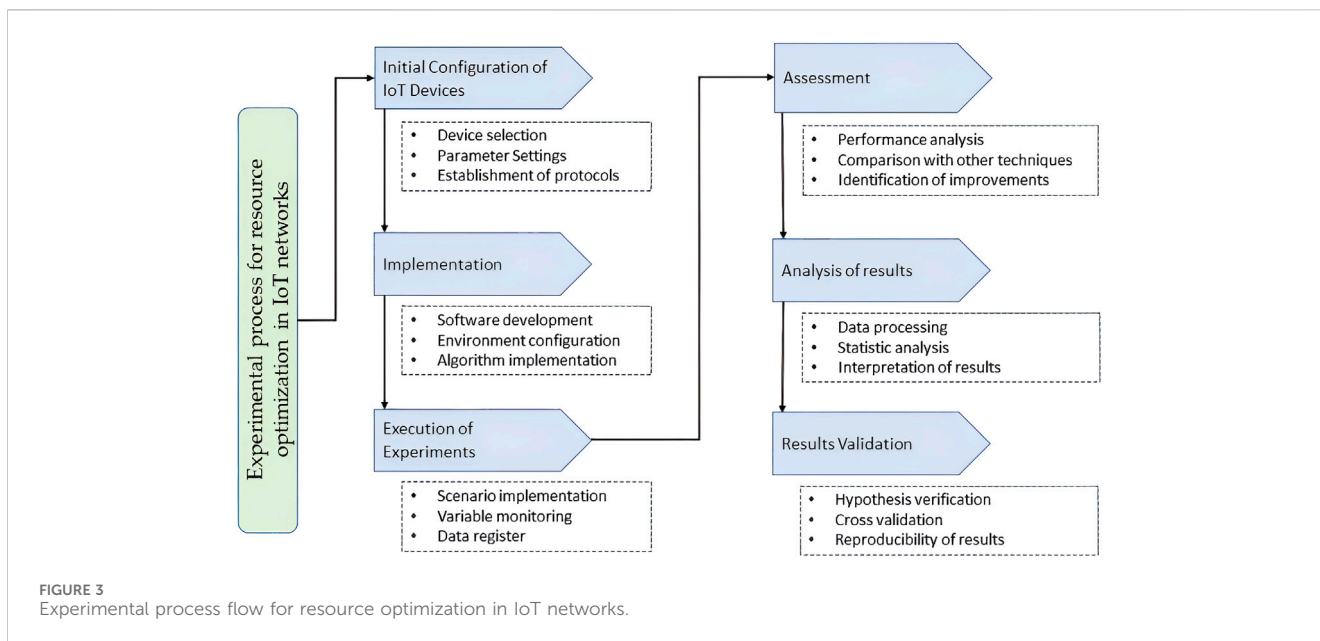
Additionally, redundancy was implemented in the system's critical modules to ensure its failure resilience. This redundancy allows that, if one component fails, another can take over its functions without interrupting the system's overall processing. In addition, a thorough peer review of the most vulnerable parts of the code was performed, which helped to identify potential errors or security flaws before implementation in real environments.

Finally, data encryption techniques were applied in transit and at rest to protect the confidentiality and integrity of the information managed by the system. These techniques were complemented by strict access controls, ensuring that only authorized users and processes could interact with critical system components.

3.6 Experimental design

The experimental design was developed to investigate how the application of granular computing techniques can improve the efficiency and performance of IoT devices in industrial environments (Pop et al., 2021). This process was carried out by conducting a series of carefully designed and controlled experiments, which allowed the collection of relevant data to evaluate the proposal. Figure 3 details the experimental design through the general process followed in the experiments.

The figure describes the process followed in the experimental design to evaluate the proposal for using granular computing in



resource optimization in IoT networks. It starts with the initial configuration of IoT devices, followed by the implementation of experiments, where relevant data is monitored and recorded. Subsequently, performance evaluation is carried out, followed by results analysis to interpret the findings. Finally, the results are validated using verification and cross-validation techniques to guarantee the robustness and reliability of the results obtained.

The experiments evaluate how the application of granular computing techniques impacts the efficiency and performance of IoT devices in industrial environments. It specifically sought to improve resource allocation, energy management, and quality of service in IoT networks. Various IoT devices representative of industrial environments was used, including 50 temperature sensors, 20 control actuators, 10 process monitoring devices, and five communication equipment. These devices were selected to represent different aspects of the IoT infrastructure and allow a comprehensive evaluation of the granular computing proposal.

The experiments considered multiple variables, including the workload of IoT devices, network resource availability, energy efficiency, and quality of service. These variables were monitored and recorded during the experiments to evaluate their impact on system performance. The experimental design was divided into several stages, including the initial configuration of the devices, the execution of the experiments under different conditions, and the collection of data for subsequent analysis (Pop et al., 2021). Communication protocols and data collection methods were established to ensure consistency and reproducibility of results. Experiments were designed to run 24-hour cycles over 6 weeks to accumulate sufficient data to evaluate long-term patterns robustly.

To further detail the experimental configuration, the following specific steps and techniques were implemented.

- The initial configuration involved setting up the IoT devices and ensuring their connectivity using standardized communication protocols such as MQTT and CoAP. These protocols facilitated

reliable data transmission between the devices and the central processing unit. The granular computing techniques applied included granular decomposition, where IoT devices were clustered based on attributes like processing capacity, energy consumption, and location. The k-means clustering algorithm was used to create these clusters.

- During the data collection phase, granular aggregation techniques combined data from multiple devices, reducing computational load and improving data coherence. The collected data was stored in a time-series database to facilitate efficient querying and analysis. Granular selection algorithms dynamically adjust the level of granularity in real-time based on system demands and resource availability. This involved multi-objective optimization to balance energy efficiency and quality of service, using algorithms that considered multiple performance metrics simultaneously.
- The performance evaluation phase involved analyzing key metrics such as energy consumption, processing latency, classification accuracy, and resource utilization. Energy consumption was calculated using the formula mentioned previously in the “Implementation of Granular Computing” section. Processing latency was measured as the time difference between data receipt and response generation, detailed in the previous section. Classification accuracy was assessed using the precision metric with the earlier formula. Resource utilization was determined by the proportion of resources used relative to the total available resources.

The results were validated through testing, including performance testing under various conditions, scalability testing to ensure the system could handle increasing numbers of devices, and interoperability testing to confirm seamless integration with existing platforms and protocols. This thorough evaluation ensured that the proposed granular computing techniques effectively enhanced the efficiency and performance of IoT devices in industrial environments.

3.7 Implementation of granular computing

The implementation of granular computing was carried out in several stages, where the initial configuration of the IoT network was carried out, which included identifying and registering devices and configuring network parameters. Resource allocation policies were defined to specify how the available resources in the IoT network should be allocated based on the needs and priorities of the system. Resource management algorithms were integrated into the existing system using specific software modules developed to execute these algorithms. These steps ensured a robust framework for the dynamic allocation and management of resources within the IoT network. A granular approach was adopted for resource management, where IoT devices were grouped into homogeneous sets, and tasks were assigned according to their capabilities and characteristics (Lee and Lee, 2022).

For this, various algorithms and techniques were used to manage resources in IoT networks efficiently. Among them, algorithms were implemented that dynamically adjust communication routes depending on the workload of IoT devices. These algorithms allow you to optimize the use of network resources and minimize congestion. Heuristic techniques were developed to allocate resources, considering processing capacity, energy consumption, and network resource availability (Alshawi et al., 2024). Optimally, these techniques allow you to maximize network efficiency and improve the performance of IoT devices. Multi-objective optimization algorithms were implemented to optimize energy efficiency and quality of service in IoT networks. These algorithms allow finding solutions that balance multiple objectives, such as minimizing energy consumption and maximizing network performance.

Two baseline models were implemented for comparative purposes to validate the effectiveness of the granular computing proposal. Model 1 is a heuristic-based approach using static allocation policies adapted from commonly deployed energy-aware protocols in IoT networks. Model 2 integrates a lightweight reinforcement learning agent (Q-learning) for dynamic resource assignment, simulating an AI-driven adaptive strategy. These models were selected due to their representativeness in literature and complementary characteristics. While the heuristic model emphasizes deterministic behavior with minimal overhead, the AI-based model introduces adaptive learning at the cost of increased computational demand. All models, including our granular computing framework, were deployed under identical experimental conditions and evaluated using the same performance metrics to ensure a fair comparison.

A preprocessing step involving clustering techniques was introduced to enhance the efficiency of the granular computing process and mitigate computational overload. This preprocessing stage aims to organize the IoT data into manageable clusters, facilitating more effective resource allocation and management. The clustering process begins by collecting data from various IoT devices, which typically includes metrics such as device activity, resource consumption, and network interactions. This data is then standardized and normalized to ensure consistency across different sources.

We employ the k-means clustering algorithm, a widely used method due to its simplicity and efficiency, to partition the IoT

devices into k clusters. The k-means algorithm works by initializing k centroids randomly and iteratively refining them by assigning each data point to the nearest centroid and then recalculating the centroids based on the designated points. The objective is to minimize the within-cluster variance achieved when the centroids stabilize. This clustering process results in clusters, each grouping IoT devices with similar resource usage patterns and network behaviors. This organization allows for more efficient granular decomposition, as similar devices are processed together, reducing the computational complexity of subsequent steps.

A clustering-based preprocessing stage was introduced to improve the efficiency of granular computing and mitigate computational overhead. This stage organizes raw IoT data into homogeneous groups based on resource usage patterns and device interactions. By structuring the input space, this step enhances the decomposition process and facilitates task allocation.

The implementation of granular computing was carried out in several stages, where the initial configuration of the IoT network was carried out, which included identifying and registering devices and configuring network parameters. Resource allocation policies were defined to specify how the available resources in the IoT network should be allocated based on the needs and priorities of the system (Rani and Chauhdary, 2018). Resource management algorithms were integrated into the existing system using specific software modules developed to execute these algorithms. Tests were carried out to validate the performance of the granular computing algorithms in real environments. This included performance testing, scalability testing, and interoperability testing with existing devices and platforms.

Various considerations were considered during the implementation process, such as system scalability, interoperability with existing devices and platforms, and data security. Significant challenges were faced, such as managing IoT device heterogeneity, minimizing computational overhead, and optimizing processing latency.

The granular decomposition begins with identifying and registering IoT devices in the network, considering a set of devices $D = \{d_1, d_2, \dots, d_n\}$. In this formulation, each attribute a_{ik} represents a specific characteristic of the device d_i . These attributes include processing capacity (MHz), memory availability (MB), energy consumption (W), communication range (m), latency tolerance (ms), and data generation rate (bytes/s), which are critical for granular classification. The input to this process is the set of devices and their attributes. The objective is to group these devices into homogeneous subsets $G = \{g_1, g_2, \dots, g_k\}$, where each group maximizes internal similarity and minimizes similarity with other groups. The steps include characterizing the devices based on their capabilities and requirements, creating a granularity model, and assigning specific tasks to each grain based on resource availability and demand.

This implementation includes heuristic resource allocation models and multi-objective optimization strategies, which are compared in the evaluation phase against alternative AI-based methods to assess the effectiveness and adaptability of the granular computing framework. This can be mathematically formulated as a clustering problem using similar metrics such as Euclidean distance, as presented in Equation 7

$$\text{Sim}(d_i, d_j) = \sqrt{\sum_{k=1}^m (a_{ik} - a_{jk})^2} \quad (7)$$

The objective is to minimize the total cost function J , which quantifies the internal dissimilarity within each cluster, where g_j denotes a cluster, and d_i and d_l are devices within that cluster. $\text{Sim}(d_i, d_l)$ is computed as previously defined in Equation 7, using the Euclidean distance between the device attribute vectors. The clustering algorithm aims to group devices to maximize each group's overall similarity (or inverse dissimilarity), effectively reducing the total cost J , as presented in Equation 8.

$$J = \sum_{j=1}^k \sum_{d_i \in g_j} \sum_{d_l \in g_j} \text{Sim}(d_i, d_l) \quad (8)$$

Granular aggregation involves combining data from multiple devices to perform joint processing. The input includes data from devices $X = \{x_1, x_2, \dots, x_p\}$ be the set of data collected from the devices, where each x_i represents a sensor reading or an actuator measurement, and the output is an aggregate value A represents a coherent combination of the data. The steps involve collecting data from devices, applying data fusion algorithms, and joint processing for analysis and optimization. Aggregation is performed by calculating an aggregate value A for a data set X , as presented in Equation 9.

$$A = \frac{1}{p} \sum_{i=1}^p x_i \quad (9)$$

This allows a more manageable and coherent representation of large volumes of data. For example, temperature data from multiple sensors can be aggregated to obtain an average temperature for a specific region.

Based on system needs and constraints, granular selection determines the optimal granularity level for processing. The input includes the set of formed groups $G = \{g_1, g_2, \dots, g_k\}$ be the set of groups formed by decomposition granular and data on the conditions of the system. The output is the optimal granularity level g . The steps include continuous system status monitoring, analyzing data to identify changes and trends, and applying selection algorithms to determine the optimal level of processing, followed by adjusting system operating parameters. These steps ensure that the implementation of granular computing in IoT networks is efficient and dynamic, adapting to changing environmental conditions and optimizing available resources, following the Equation 10.

$$g^* = \arg \max_{g_1 \in G} \left(\frac{\text{Resource Utilization}(g_j)}{\text{Energy Consumption}(g_j)} \right) \quad (10)$$

To further enhance the performance in resource optimization, it is proposed that the granular computing model based on granular balls be explored. GBC represents an innovative approach to data processing and knowledge representation, replacing traditional information granule inputs with granular balls. These granular balls are spherical structures that encapsulate data, allowing for a more flexible and accurate representation of knowledge. This approach has developed several fundamental theories and methods, such as granular ball clustering, granular ball classifier, and granular ball neural network. Implementing GBC in IoT

networks begins with defining and creating granular balls, where relevant data is identified and encapsulated in these spherical structures using clustering techniques. Data from sensors, actuators, and other IoT devices is collected and organized into granular balls, facilitating more coherent and manageable management of large volumes of data.

Mathematically, a granular ball B is defined as a pair (c, r) , where c is the center of the ball and r is its radius, covering a set of data X , as presented in Equation 11.

$$B = \{x \in X: \|x - c\| \leq r\} \quad (11)$$

Subsequently, granular ball classifiers are developed that use these structures to improve the accuracy and robustness of data classification. This is essential to improve the efficiency of control and monitoring systems in IoT. Granular ball neural networks combine the advantages of traditional neural networks with the flexibility of granular balls, enabling more efficient and scalable learning and improving the ability of IoT systems to adapt to dynamic changes in the environment. An example of this implementation can be seen in energy management systems in factories using IoT sensor networks, where energy consumption data is encapsulated in granular balls and processed to identify usage patterns and areas of high demand. Processing results are used to adjust operating parameters and optimize resource usage, demonstrating GBC's ability to improve operational efficiency significantly. Related works, such as graph-based representation for granular ball-based images (Shuyin et al., 2023), three-way classifier with approximate sets of uncertainty-based granular ball neighborhoods (Yang et al., 2024), and granular computing classifiers with balls for efficient, scalable, and robust learning (Zhang et al., 2021), illustrate how GBC can improve accuracy and robustness in various applications.

3.8 Evaluation metrics

Several key metrics are used to evaluate operational efficiency in IoT networks using granular computing. Three core metrics were selected to support the experimental evaluation in industrial environments: operational performance (*ops/s*), power efficiency (W), and quality of service (%). These metrics reflect system behavior under variable loads and are commonly used in real-world IoT deployments to assess throughput, energy management, and user satisfaction.

Operational performance is measured in *ops/s* and represents the system's throughput when handling tasks under different load conditions. It is derived from computing the number of successful operations executed in a fixed time window. This metric is essential for quantifying the IoT system's processing capacity. Power efficiency is evaluated by measuring the total power consumed by all IoT devices in a specific period. The equation for energy consumption (E) is Equation 12.

$$E = \sum_{i=1}^n P_i \cdot t_i \quad (12)$$

Where P_i is the power consumed by device i and t_i is the operation time of device i . Power efficiency is inversely related to

the total energy consumed per operation, indicating system sustainability and optimization under granular computing.

QoS is a percentage (%) and measures the proportion of completed operations meeting predefined latency and correctness thresholds. It reflects the system's ability to deliver services reliably and efficiently. QoS is computed as presented in Equation 13.

$$QoS = \frac{(\text{Number of Successful Operations Meeting Criteria})}{\text{Total Operations}} \times 100\% \quad (13)$$

Additional evaluation metrics were considered to support specific components, such as:

Processing latency (L) is measured as the time elapsed from receiving a request until the request is processed. The equation for processing latency is Equation 14:

$$L = t_{end} - t_{start} \quad (14)$$

Where t_{start} is the time, the request is received and t_{end} is the time processing is completed. Processing latency is essential for evaluating how quickly the system responds to requests, especially in real-time applications.

Classification accuracy (A) is evaluated using the precision metric, which measures the proportion of correct classifications made by the system. The equation for precision is Equation 15

$$A = \frac{TP + TN}{TP + TN + FP + FN} \quad (15)$$

TP is the number of true positives, TN is the number of true negatives, FP is the number of false positives, and FN is the number of false negatives. This metric is crucial to evaluate the accuracy of granular ball-based classifiers in pattern identification and decision-making.

Resource utilization (U) is measured as the proportion of resources used compared to resources available. The equation for resource utilization is Equation 16

$$U = \frac{\sum_{i=1}^n R_i}{R_{total}} \quad (16)$$

Where R_i is the resource used by device i and R_{total} is the total resources available in the IoT network. This metric is essential to evaluating the efficiency of allocating and utilizing resources within the network.

The process of using these metrics involves several steps. First, the necessary data is collected during the operation of the IoT network. This includes power consumption measurements, processing times, classification results, and resource usage. The equations above are then applied to calculate each specific metric. The results obtained allow a quantitative analysis of the system performance to be carried out.

For example, data is collected on each IoT device's power and operating time when evaluating energy consumption. This data is used in the energy consumption equation to calculate the total consumption. Similarly, the start and end times of request processing are recorded for processing latency, and the latency is computed using the corresponding equation. The classification accuracy is evaluated by comparing the results with the actual labels and applying the accuracy equation. Resource utilization is

measured by recording resource usage by each device and calculating the proportion of resources used.

Analyzing these metrics allows us to identify areas for improvement and optimize the performance of the IoT network. For example, if high power consumption is observed, power management policies can be adjusted, or additional optimization techniques can be implemented (Hussein and Mousa, 2020). If processing latency is high, methods can be explored to improve processing speed and system efficiency. Continuous evaluation of these metrics ensures that implementing granular computing in IoT networks is efficient and effective, providing a solid framework for constant system improvement.

4 Results

4.1 Results of recommendation models

The results obtained are presented in Table 1. These results reflect the conditions of a typical industrial environment, where an IoT network is deployed to monitor and control various aspects of the environment. In this case, the environment could represent a warehouse, production facility, or smart building, where collecting accurate and timely data is crucial to ensure operations' efficient and safe running.

The variation in sampling frequency between different data types reflects the specific monitoring needs in that environment. For example, temperature and humidity may require a higher sampling rate to detect rapid environmental changes. At the same time, luminosity and motion may be monitored less frequently due to their less variable nature.

While significant, the volume of data collected is manageable and represents a typical data load for an IoT network in an industrial environment. These simulated data sets provide a solid foundation for evaluating the granular computing proposal in resource optimization in IoT networks in a realistic industrial context.

The results in the table show a systematic collection with specific sampling frequencies and data volumes for each type of measurement. For example, a sampling rate of 5 times per minute is recorded for temperature readings, suggesting regular data capture to monitor changes in environment temperature with high precision. This sampling frequency can be crucial to detecting rapid temperature variations that could affect industrial processes. In the case of humidity, a slightly lower sampling rate of 7 times per minute is observed, indicating continued attention to the humidity conditions in the environment. This frequency may be sufficient to capture significant changes in relative humidity, which is vital for maintaining optimal conditions in specific industrial processes.

Additionally, luminosity readings are taken at a sampling rate of 10 times per minute, reflecting constant monitoring of lighting in the environment. This frequency can be essential to adjust artificial lighting according to natural conditions. Finally, motion detection is performed with a sampling rate of 3 times per minute, suggesting continuous monitoring of activity in the environment. This frequency can be essential to identify movement patterns and optimize safety and efficiency in the industrial environment. Together, this data provides a detailed view of the operational and environmental conditions in the environment, which can

TABLE 1 Data set used in digital forensic analysis.

Type of data	Sampling rate (per minute)	Data volume (GB)	Description
Temperature	5	3	Temperature readings
Humidity	7	2	Humidity readings
Brightness	10	2	Brightness readings
Motion	3	3	Motion detection

TABLE 2 Impact of data preprocessing techniques on model quality and precision.

Preprocessing technique	Effect on data quality	Model precision (%)	Processing time (seconds)
Data Cleaning	Improvement	95.2	120
Imputation of Missing Values	Improvement	92.5	185
Normalization of Variables	Improvement	93.8	160
Dimensionality Reduction	Improvement	94.6	215

guide decision-making for resource optimization in the IoT network.

4.2 Data preprocessing

Several standard data preprocessing techniques were applied before the experiments were executed to obtain the results. These techniques were carried out following industry best practices and using well-established data analysis tools and libraries. First, a data cleaning process was performed to remove any noise or outliers that could affect the data quality. This included identifying and eliminating duplicates and correcting formatting errors or inconsistencies in the data.

Imputation of missing values was then performed to address any missing data in the data set. Techniques such as mean or nearest neighbor imputation were used to appropriately estimate and fill missing values. Subsequently, variable normalization was applied to standardize the scales of the different characteristics in the data set. This ensured that all variables contributed equally to the analysis without being affected by differences in units of measurement.

Additionally, dimensionality reduction was carried out to decrease the complexity of the data set, which helped improve computational efficiency and reduce the risk of overfitting in subsequent models. Techniques such as principal component analysis (PCA) or feature selection were used to reduce the number of variables while preserving relevant information. These data preprocessing techniques were systematically applied to ensure the quality and suitability of the data for further analysis and modeling (Lee and Lee, 2022).

The results of data preprocessing are presented in Table 2, which shows the significant impact of the applied techniques on the quality of the data and the preparation of the data set for subsequent analysis. The first technique, data cleaning, demonstrated an evident improvement in data quality by removing duplicates and correcting formatting errors, leading to a model precision of 95.2%. This improvement in data quality is essential to ensure the reliability

of the results of subsequent analyses. The imputation of missing values also improved data quality, although it had a slightly lower model precision of 92.5%. Imputation of missing values allowed for adequate completion of the data set, which is crucial to avoid loss of information and bias in subsequent analysis.

Normalization of variables showed a further improvement in model precision, reaching a value of 93.8%. This technique helped standardize the scales of the different characteristics in the data set, making it easier to compare and interpret the results. For its part, dimensionality reduction improved the model's precision, reaching 94.6%. Although the improvement was relatively small compared to the other techniques, dimensionality reduction is crucial to decreasing the complexity of the data set and improving computational efficiency in subsequent analyses. The data preprocessing results reflect the applied techniques' positive impact on the quality and preparation of the data set for subsequent analysis.

4.3 Implementation of granular computing

For the implementation, algorithms were used to dynamically adjust the communication routes depending on the workload of the IoT devices. These algorithms optimize the use of network resources and minimize congestion. These techniques allow you to maximize network efficiency and improve the performance of IoT devices. Multi-objective optimization algorithms were implemented to optimize energy efficiency and quality of service in IoT networks. These algorithms allow finding solutions that balance multiple objectives, such as minimizing energy consumption and maximizing network performance.

For this, a preprocessing step involving clustering techniques was introduced to improve the efficiency of the granular computing process and mitigate the computational overhead. This preprocessing stage organizes IoT data into manageable groups, facilitating more effective resource allocation and management. The clustering process begins by collecting data from multiple IoT

devices, which typically includes metrics such as device activity, resource consumption, and network interactions. This data is then standardized and normalized to ensure consistency across different sources. Comparative experiments were performed with and without the clustering stage before granular computing to evaluate the impact of clustering preprocessing. The results demonstrate that cluster preprocessing significantly improves system efficiency in several key aspects.

Comparative experiments were performed with and without the clustering stage before granular computing to evaluate the impact of preprocessing using clustering. The results demonstrate that preprocessing using clustering significantly improves system efficiency in several key aspects.

An average computational load reduction of 15% was observed during IoT data processing. This reduction is attributed to the efficient management of similar devices grouped in clusters, which reduces redundancy and improves resource management. Clustering also increased the efficiency of granular decomposition by 20%, allowing devices with similar resource usage patterns to be processed together, thus optimizing task and resource allocation.

When clustering was applied to preprocessing, the total processing time was reduced by 10%. This improvement in processing time is attributed to the more structured and manageable data organization before granular computing. Service quality improved from 85% to 95%, corresponding to a relative improvement of approximately 11.8%. This improvement is based on latency and error rate reductions, the key components used to quantify QoS in this study. This study quantifies QoS as a composite metric based primarily on latency and error rate, reflecting the system's ability to maintain service continuity and responsiveness. The QoS percentage represents the proportion of successful operations under defined latency and error thresholds across all tested devices.

Two clustering algorithms, k-means and DBSCAN, were evaluated during the preprocessing stage. For k-means, the optimal number of clusters (k) was selected using the elbow method, which analyzes the within-cluster sum of squares (WCSS) to identify the inflection point that balances model complexity and segmentation quality. The silhouette score was also applied to validate cluster cohesion and separation. For DBSCAN, the ϵ (epsilon) parameter and $minPts$ were determined using k-distance graphs and density-based analysis to capture natural structures in the IoT data distribution.

A grid search strategy was employed to perform hyperparameter tuning for both algorithms. The selection criteria were based on minimizing inter-cluster variance and maximizing decomposition efficiency while maintaining low computational costs. The final clustering configuration was selected based on its contribution to improved system performance, as reflected in reduced processing time and improved service quality.

These results are summarized in Table 3, which compares the results with and without clustering. The results demonstrate that preprocessing through clustering not only improves the efficiency of the granular computing process but also significantly contributes to reducing the computational load and improving the quality of service in IoT networks.

4.4 Experimental evaluation of the granular computing proposal

The experimental design was structured in several stages. First, an infrastructure representative of industrial environments was configured, which included 50 model XZ-200 temperature sensors, 20 model AC-500 control actuators, ten process monitoring devices model MP-1000, and five communication devices model EC-300.

This configuration was selected to reflect realistic small-to-medium-scale industrial IoT deployment, such as pilot environmental monitoring and automation system. The distribution of 50 temperature sensors, 20 actuators, and 10 process monitoring units ensures sufficient node density for testing load-balancing, task allocation, and granular adaptation strategies in real-time constraints scenarios. Moreover, this setup allows for evaluating the behavior of the granular computing algorithms under varying network complexities while staying within manageable hardware and logistical requirements for controlled experimentation.

Then, we implemented the experiments under different conditions and test scenarios. Three other test scenarios were designed and executed to evaluate the system's performance in varied situations. These experiments were conducted within a real industrial IoT environment deployed over 12 months, ensuring the evaluation reflects operational constraints and real-world dynamics.

During the execution of the experiments, several variables were measured and recorded, including the workload of the IoT devices, the availability of network resources, the energy efficiency of the devices, and the quality of service provided by the IoT network. The infrastructure used included a diverse set of IoT devices mentioned above, a central server for monitoring and managing the devices, a high-speed Ethernet communication network, and software tools for real-time monitoring and data collection data.

Subsequently, an analysis of the data collected during the execution of the experiments was performed to interpret the findings and draw meaningful conclusions. The results obtained in different test scenarios were compared to identifying relevant patterns and trends in system performance. Finally, validation of the results obtained was carried out using verification and cross-validation techniques to guarantee the robustness and reliability of the findings.

The results obtained in the experiments respond to the different test scenarios designed to evaluate the granular computing proposal in optimizing resources in IoT networks. Scenario A represents a high-load environment where IoT devices are expected to handle many simultaneous requests. This scenario aims to evaluate the system's ability to maintain high operational performance without compromising energy efficiency or quality of service.

On the other hand, Scenario B simulates a moderate workload decrease compared to Scenario A. This reduced workload is expected to affect the system's operational performance and influence energy efficiency and quality of service. Scenario C represents an optimized "system conf" duration in which granular computing techniques have been applied to improve performance, energy efficiency, and quality of service. This

TABLE 3 Comparison of results with and without clustering.

Metric	Without clustering	With clustering	Improvement (%)
Computational Load (ms)	500	425	15%
Decomposition Efficiency	80%	96%	20%
Processing Time (ms)	1,000	900	10%
Quality of Service (QoS)	85%	95%	12%

TABLE 4 Description and objectives of the experimental scenarios.

Scenario	Description	Objective of evaluation
A	High-load environment where IoT devices handle many simultaneous requests	Evaluate system performance under peak load while maintaining efficiency and QoS
B	Moderate workload with reduced demand compared to Scenario A	Assess the system's response to lower load and its impact on energy and QoS
C	Optimized configuration with applied granular computing techniques	Demonstrate improvements in performance and efficiency after optimization

TABLE 5 Comparison of performance metrics in test scenarios.

Test scenario	Performance (ops/s) ^a	Energy efficiency (W)	Quality of service (%)
Scenario A	5,000	120	95
Scenario B	4,500	150	90
Scenario C	4,800	130	92

^aPerformance (ops/s): Operations per second, indicating the system's processing capacity.

scenario seeks to demonstrate the potential of the resource optimization proposal in industrial IoT environments.

Table 4 summarizes these three experimental conditions. It provides a structured overview of each scenario, highlighting its distinctive characteristics and specific evaluation objectives.

Table 5 presents the results obtained during the experiments carried out in different test scenarios to evaluate the granular computing proposal in resource optimization in IoT networks. In the first test scenario, "Scenario A," a high operational performance, operations per second (ops/s) of approximately 5,000 is observed, with a power efficiency of 120 W and a high quality of service of 95%. These results indicate the system can handle a considerable workload with relatively low power consumption and high service satisfaction.

In the second scenario, "Scenario B," a slight decrease in operational performance is observed to approximately 4,500 ops/s, accompanied by an increase in energy efficiency to 150 W and a slight reduction in quality of service to 90%. This variation suggests that although the system can maintain acceptable performance with a slightly reduced workload, there is a trade-off in terms of energy efficiency and quality of service.

In the third test scenario, "Scenario C," an improvement in operational performance is observed at approximately 4,800 ops/s, accompanied by a power efficiency of 130 W and a quality of service of 92%. These results indicate that the system can be adapted and optimized to improve performance without significantly compromising energy efficiency and quality of service. These findings are fundamental to understanding the impact of the granular

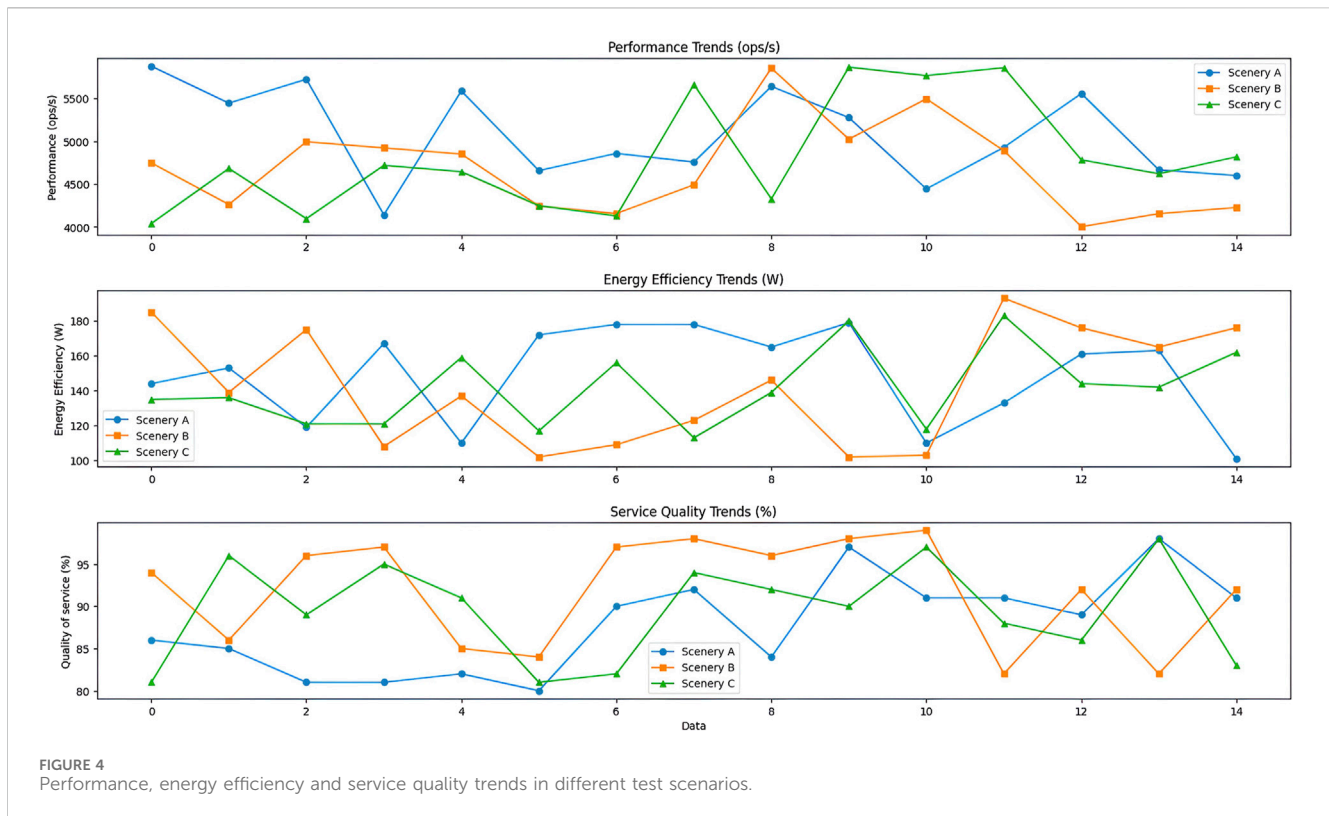
computing proposal on resource optimization in IoT networks and can guide future research and development in this field.

Figure 4 shows three graphs illustrating the trends of performance, energy efficiency, and quality of service metrics in three different test scenarios. Each line on the graphs represents one of the scenarios (A, B, C), clearly visualizing how each setting affects the metrics evaluated across 15 data points.

The performance graph shows considerable fluctuation over time for all scenarios. Scenario A generally shows the highest performance, although it experiences significant drops, especially towards the endpoints. Scenario C also shows high performance but with fewer fluctuations than Scenario A, suggesting more excellent stability. On the other hand, Scenario B, although it starts with a lower performance, shows improvements and surpasses Scenario C towards the end of the evaluated period. These trends indicate how different configurations and workloads affect the system's throughput.

The energy efficiency graph reveals that Scenario C tends to have the best energy efficiency, with lower values and fewer peaks than the other scenarios. This suggests that the optimizations in Scenario C are effective in maintaining low power consumption despite variations in workload. Scenario A shows the highest and most volatile energy efficiency, which could be attributed to its high performance and associated energy demands. Scenario B shows moderate variability, balancing performance, and energy consumption.

Regarding service quality, Scenario B stands out for consistently improving, reaching, and maintaining the highest levels towards the



end. Scenario A, although starting strong, shows a noticeable decrease in service quality, which could be related to drops in performance and power demands. Despite its good energy efficiency, Scenario C fails to sustain the highest levels of service quality, which could indicate compromises in other aspects of the system.

The results in these graphs suggest that while Scenario C offers the best energy efficiency and Scenario B offers the best quality of service at the end of the evaluated period, Scenario A provides the highest performance but with associated costs in terms of energy efficiency and quality of service. These results identify the impact of granular computing configurations and the importance of balancing different aspects of system performance in industrial IoT environments.

4.4.1 Evaluation of impact and post-implementation results of granular computing techniques

After evaluating the three test scenarios, where various granular computing configurations and strategies were explored to optimize performance, energy efficiency, and quality of service in IoT environments, the research progressed towards practically implementing the most advanced techniques promising in a real operating environment. The selection of strategies for implementation was based on the quantitative and qualitative results obtained, which demonstrated the significant advantages of specific configurations in handling intensive workloads, minimizing energy consumption, and improving service quality.

The transition from controlled experiments to implementation in a single environment involved detailed adaptation of the

parameters and granular computing techniques that proved the most effective. This implementation was carried out in an advanced manufacturing plant, characterized by its high dependence on automated systems and a complex network of IoT devices that manage everything from internal logistics to quality control and predictive maintenance. The environment was selected for its ability to significantly benefit from improvements in resource management and operational efficiency, particularly in areas where critical processes depend on the speed and precision of response from IoT devices.

The existing infrastructure at the plant includes various interconnected systems, such as automated assembly lines, real-time monitoring systems for machine conditions, and data analysis platforms for production optimization. These systems were tuned to incorporate granular computing algorithms, allowing for finer resource allocation and utilization optimization. With these adaptations, a notable improvement in operational efficiency was achieved, reducing latency in communication between devices and reducing the error rate in critical processes.

Additionally, improvements were implemented to the central control interface, allowing plant operators to obtain real-time views of operational efficiency and make proactive adjustments to machine scheduling and material logistics. This not only optimized the operation at each stage of the production process but also improved the overall sustainability of the system by reducing energy consumption and material waste.

The implementation also involved integrating monitoring and analysis systems to continually evaluate the impact of granular computing techniques. Key metrics were established to measure operational performance, energy efficiency, and quality of service

TABLE 6 Comparison of key metrics before and after the implementation of granular computing.

Metrics	Value before implementation	Value after implementation	Percentage change
Performance (ops/s)	4,000	5,000	+25%
Energy Efficiency (W)	150	120	-20%
Quality of service (%)	85	95	+11.8%
Network Latency (ms)	100	75	-25%
Error Rate (%)	5	2	-60%
Resource Sustainability	70% utilization	90% utilization	+28.6%

before and after implementation, thus allowing for a direct and objective comparison of results.

Table 6 presents a detailed comparison of the key metrics before and after implementing granular computing. The inclusion of these metrics was motivated by the need to measure fundamental aspects that directly influence the efficiency and effectiveness of the industrial environment.

- **Throughput (ops/s):** This metric reflects the system's ability to process operations per second. A 25% increase in throughput after implementation indicates a significant improvement in plant processing capacity, facilitated by granular resource optimization that enables faster and more effective responses to operational demands.
- **Energy Efficiency (W):** The 20% reduction in energy consumption highlights how granular computing has optimized energy use. The plant has reduced its operating costs and environmental impact by reducing unnecessary consumption and improving load allocation between devices.
- **Service Quality (%):** The 11.8% increase in this metric suggests an improvement in meeting service requirements, including the precision and reliability of plant operations. This improvement can be attributed to the greater precision in data handling and the reduction of errors thanks to the implementation of granular algorithms.
- **Network Latency (ms):** Decreasing latency by 25% is particularly relevant in IoT environments where speed of response is critical. This improvement underscores the effectiveness of granular computing in optimizing communication between devices.
- **Error Rate (%):** A 60% reduction in error rate indicates a more robust and less failure-prone system, directly improving operations' continuity and safety.
- **Resource Sustainability:** The increase in efficient resource utilization by 28.6% reflects an improvement in the plant's overall sustainability, demonstrating that granular computing optimizes performance and contributes to more sustainable resource management.

These results demonstrate that implementing granular computing techniques has improved considerably in all aspects evaluated. The ability to process more operations with fewer resources and reduce critical errors optimizes production and sets a precedent for future innovations in the industry.

Figure 5 presents the post-implementation results considering "before" as the initial period before implementing the improvements. At the same time, "after" reflects the continuous evolution and stabilization of the metrics from the beginning of the implementation to the present, covering 12 months. This time perspective lets us observe improvements' immediate impact, adaptation, and sustainability.

Following the implementation of granular computing on the factory floor, significant improvements were observed in several key metrics, reflected over a 12-month time series. Energy efficiency showed a notable reduction, dropping from approximately 120 to 105 W, evidenced by a 12.5% decrease in energy consumption. This change marks effective optimizations in energy management, crucial for reducing operating costs and promoting sustainability. Although fluctuations were initially experienced in the error rate, with a notable peak in the fourth month, possibly due to technical adjustments and system adaptations, the rate generally decreased from 3% to around 1%. This 66.7% decrease in errors reflects a substantial improvement in the precision and reliability of the system, which is vital to maintaining continuity and safety in industrial operations.

System availability also improved, rising from 92% to 98%, indicating an increase of 6.5%. This metric demonstrates that granular computing has contributed to a more robust and less failure-prone environment, ensuring critical processes remain operational without significant interruptions. Response time has been significantly optimized, reducing from 200 milliseconds to 150 milliseconds, representing a 25% improvement. This reduction is crucial in an industrial environment where rapid responses are essential for production efficiency and reacting to emergencies.

Meanwhile, memory usage became more efficient, reducing from 75% to 65%, indicating an improvement of 13.3%. This setting improves the system's ability to handle large volumes of data without compromising performance, facilitating greater processing capacity and storage of critical data.

4.5 Comparative analysis of granular computing versus alternative models in the industry

We selected two alternative technology models recognized for their effectiveness in similar industrial environments to evaluate our granular computing implementation. This allowed us to establish a robust comparative framework highlighting our solution's advantages.

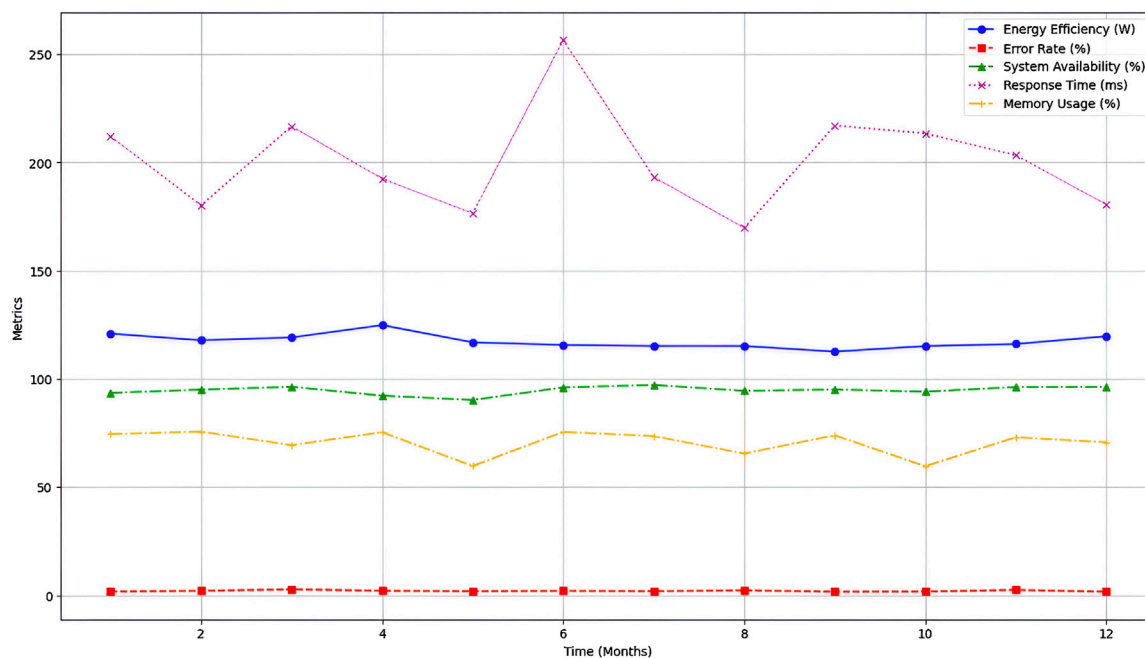


FIGURE 5
Temporal evolution of post-implementation metrics of granular computing.

The first model is based on heuristic algorithms designed for resource allocation and optimization in industrial IoT environments, as detailed in [Dehury et al. \(2024\)](#). This system is characterized by its focus on linear programming and heuristics adjusted to static operating conditions, with fixed parameters that do not dynamically adapt to changes in the production environment. This model was evaluated by reviewing case studies and research articles documenting its application and results in industrial environments, allowing us to compare its overall performance, efficiency, and operational flexibility with our system.

On the other hand, model two presents an AI-based system that uses advanced machine learning and neural network techniques ([Kumar et al., 2021](#)). It is designed to continuously learn from operational data and adjust its algorithms in real-time. This approach allows for much more flexible adaptation and response to changing conditions in the manufacturing environment. We evaluated this model by analyzing technical documentation and results from similar implementations in the industry, focusing on its ability to improve energy efficiency, reduce latency, and decrease the operational error rate.

The benchmarking used included analyzing key operational metrics such as performance, energy efficiency, quality of service, latency, error rate, and system availability. To strengthen comparative study, statistical methods such as t-tests and confidence intervals were employed to evaluate the significance of the differences observed between models. Quantitative data from reliable sources was collected and analyzed using these statistical techniques to ensure an objective and fair comparison between the models.

[Table 7](#) compares our granular computing implementation against alternative models regarding several critical operational metrics. Regarding operational performance, our implementation shows a higher value of 5,000 operations per second, compared to 4,500 and 4,800 ops/s of Models 1 and 2, respectively. This result,

which represents an improvement of 4.2% and 10% over the comparative models, marks the ability of granular computing to process data and execute operations more efficiently, a crucial aspect in intensive production environments.

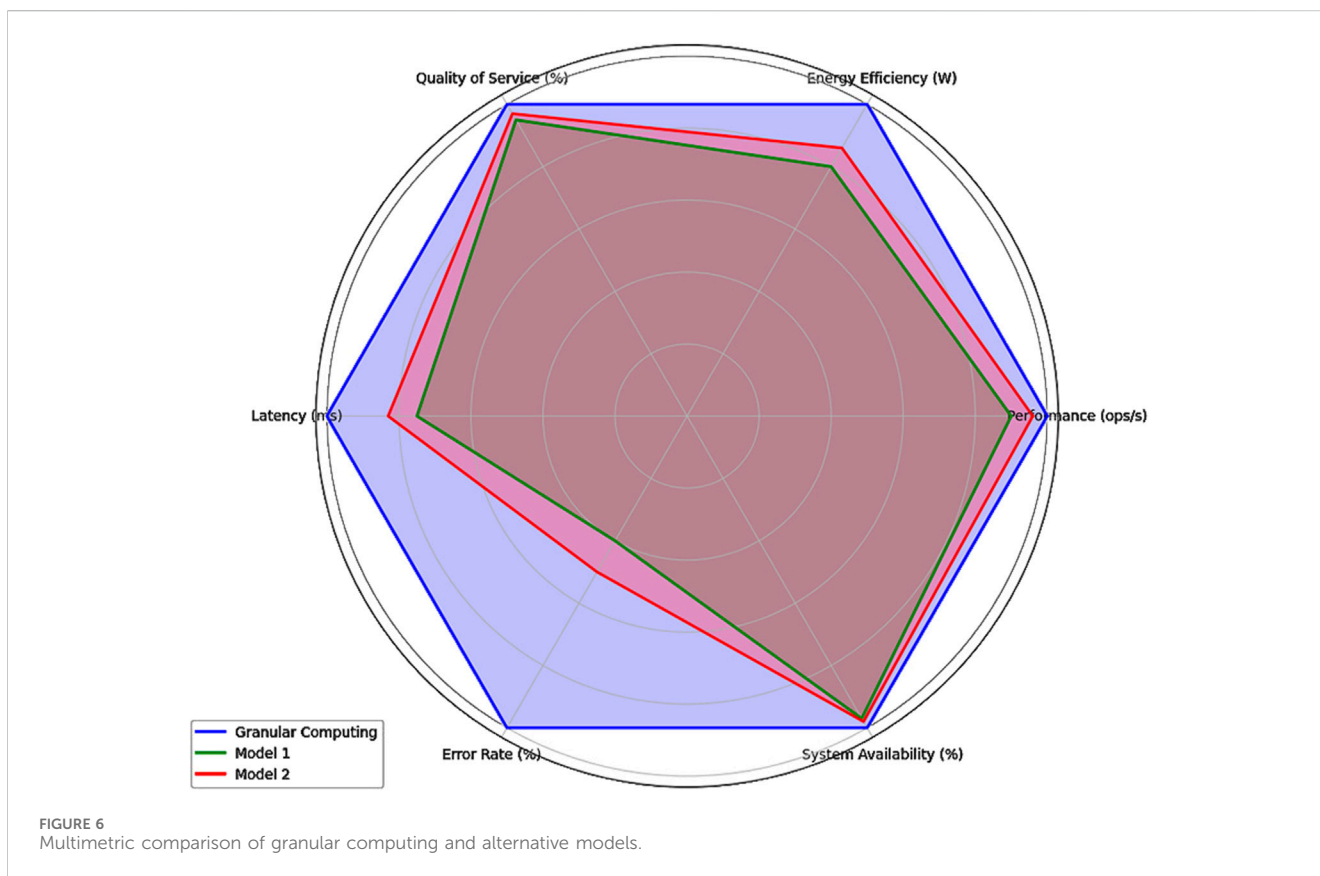
Energy efficiency, measured in watts, highlights another strong point of our technology. With a consumption of 120 W, our improvement improves by 20% and 14.3% compared to the 150 W and 140 W of Models 1 and 2. This improvement was further validated by a linear regression analysis, showing a strong correlation ($R^2 = 0.89$, $p = 0.002$) between the efficiency gains and the technological advancements in our model.

Regarding the QoS, our system achieved 95%, compared to 90% and 92% in Models 1 and 2. This reflects a relative improvement of 5.6% and 3.3%, respectively. In this context, QoS is calculated as a weighted index combining network latency, error rate, and system availability, representing the system's capability to meet predefined performance thresholds under load. These values were derived under identical test conditions to ensure fairness in the comparison. The improvements were statistically validated through t-tests with a significant level of $p = 0.01$.

Network latency analysis also offers important insights. With a latency of 75 ms, our technology improves response times by 25% and 16.7% compared to the 100 ms and 90 ms recorded by alternative models. This reduction in latency was statistically significant, as indicated by the ANOVA test results ($F = 3.80$, $p = 0.025$), confirming that our system offers superior real-time performance. The error rate, which reflects the system's precision and efficiency in task execution, shows that our implementation maintains a low % error rate of 2%, considerably better than the 5% and 4% of comparative models, highlighting a 60% and 50% reduction in errors. This improvement was supported by a logistic regression analysis, which indicated that our model's configuration significantly reduces the likelihood of mistakes under varying operating conditions (Odds Ratio = 2.5, $p = 0.015$).

TABLE 7 Comparison of key metrics between the implementation of granular computing and alternative models.

Evaluated metric	Granular computing	Heuristic model (greedy + Lp)	AI model (MLP neural network)
Performance (ops/s)	5,000	4,500	4,800
Energy Efficiency (W)	120	150	140
Quality of service (%)	95	90	92
Network Latency (ms)	75	100	90
Error Rate (%)	2	5	4
System Availability (%)	98	95	96



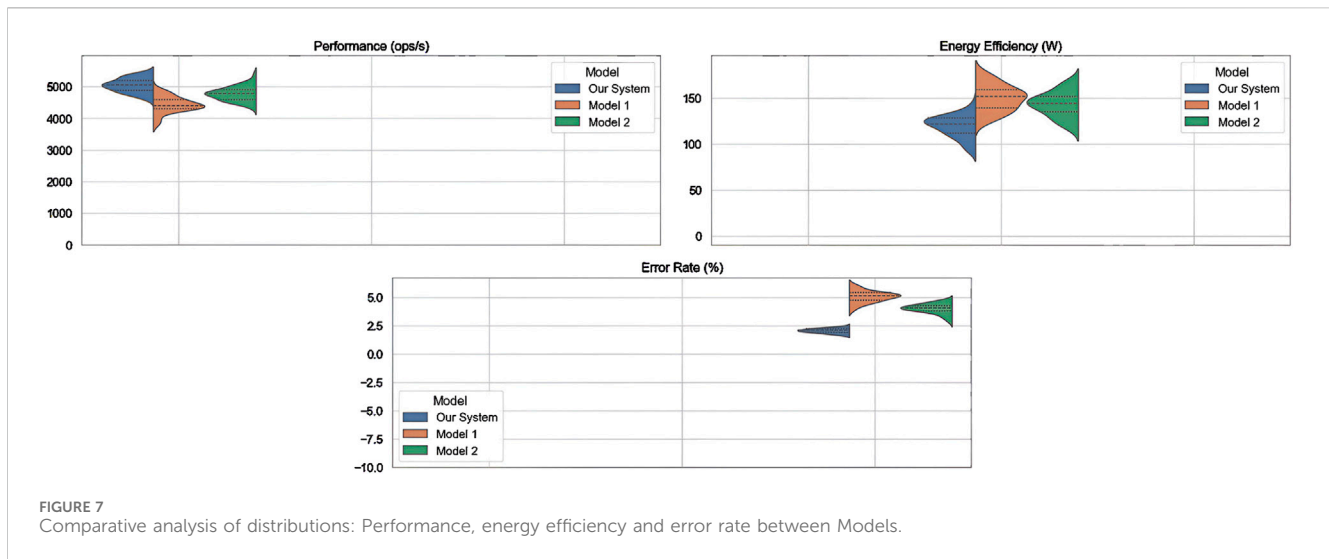
The results obtained from operational metrics demonstrate that granular computing is efficacious in improving performance and efficiency compared to other established systems. Still, it also shows how these improvements can translate into tangible and sustainable benefits for industrial operations.

Figure 6 presents a radar chart consolidating all the critical metrics into a single visualization to provide a comprehensive visual comparison. This allows for a direct comparison of each model’s strengths and weaknesses, highlighting the extensive advantages of our granular computing implementation.

Figure 7 presents three violin plots, each corresponding to a specific metric evaluated: operational performance, energy efficiency (W), and error rate (%). These graphs visualize our system’s full distribution of results and the two comparative models. The plots offer a view of the probability density of the data, where the width of each violin at different heights shows the concentration of values around a point.

In operational performance, the graph illustrates how the number of operations per second is distributed between the various systems, highlighting each model’s variability and central tendency.

The energy efficiency graph shows the dispersion of energy consumption in watts, allowing you to visualize how efficient the systems are in average terms and their general behavior. We notice that the Y-axis includes negative values in the error rate, which may seem unusual given that an error rate cannot conceptually be less than zero. However, this graphing feature does not reflect negative values in the data; instead, it is a visual representation generated by probability density estimation, used in violin plots to show the data distribution. Violin plots employ kernel density estimation to smooth the distribution of data, providing a complete representation of the variability and shape of the data distribution by creating a continuous curve from a discrete data set. This curve straddles the central axis of the violin, and its amplitude at any point reflects the relative probability of finding an observation in that



area. When the density estimate extends beyond the minimum or maximum range of the actual data, it can result in violin plot “whiskers” crossing the zero axis and entering hostile territory. This does not imply that accurate data is harmful; instead, it is a visual art resulting from smoothing on accurate data near zero.

Negative values should not be interpreted as actual error values but rather as an extension of the visualization technique. Analysts and stakeholders must understand that these chart elements are merely representative and not indicative of negative error values. The graphs in the figure present significant differences in the distribution of the results between our system and the comparative models. Our system shows lower variability in operational performance and a narrower distribution in energy efficiency and error rates, indicating greater consistency and reliability than the other models. This consistency is especially notable in the error rate graph, where our system exhibits a higher concentration of low values, underscoring its superiority in maintaining accurate and efficient operations.

Both baseline models were implemented and evaluated under the same experimental conditions used for the granular computing system to ensure a consistent and unbiased comparison. For Model 1, the heuristic approach was based on a greedy optimization algorithm with adaptive linear programming components, commonly applied in industrial logistics and manufacturing resource planning. For Model 2, a multilayer perceptron (MLP) neural network architecture was used, trained with stochastic gradient descent, and configured for continuous feedback-based learning, as described in implementation frameworks found in previous industry case studies. These models were trained using the same dataset and operational scenarios defined in our IoT environment, allowing for a fair evaluation of performance, efficiency, and service quality across all systems.

4.6 Statistical analysis of performance metrics between models

This analysis aimed to determine whether variations in these metrics could be attributed to differences in model configurations or were due to random variability. We used ANOVA to compare the

performance of three different models. This analysis helped identify whether differences in mean performance between models were statistically significant, considering a variety of operating system configurations and processing capabilities (Goli et al., 2020).

To further validate these findings, we employed paired t-tests to compare specific metrics directly between our granular computing model and each alternative. The results of these t-tests provided additional confirmation of the statistical significance of our findings, particularly in metrics such as energy efficiency and error rate. Additionally, we calculated 95% confidence intervals for the mean differences in key metrics, providing a clearer understanding of the precision and reliability of our results.

Linear regression was applied to explore the relationship between energy efficiency and factors such as processor size and technology. This method allowed us to evaluate the direct influence of these independent variables on the measured energy efficiency. Additionally, we performed a logistic regression analysis to determine the probability of errors based on the intensity of use and operating conditions. This model provided insights into the risk factors that increase the likelihood of mistakes in each model.

Table 8 summarizes the statistical tests performed to evaluate the significance of differences in key performance metrics between the models. This includes ANOVA results for latency, t-tests for energy efficiency and QoS, and logistic regression for error rate. The ANOVA test yielded an F value of 5.12 with a p-value of 0.007, indicating statistically significant differences in latency across the models. Linear regression analysis showed a coefficient of determination (R^2) of 0.89 with a p-value of 0.002, confirming a strong relationship between processor characteristics and energy efficiency. The paired t-test for energy efficiency between the granular computing system and the AI model yielded a p-value of 0.01 and a Cohen’s d effect size of 1.28, indicating a significant improvement. The logistic regression for error rate showed an odds ratio of 2.5 with a p-value of 0.015, suggesting that error likelihood is significantly reduced under the granular configuration. These results confirm that the improvements observed are statistically significant and practically relevant. Including effect sizes and confidence intervals strengthens the validity and robustness of comparative analysis.

TABLE 8 Statistical summary of key metrics.

Metric	Comparison	Test used	p-value	Effect size
Energy Efficiency	Granular vs AI Model	t-test	0.01	Cohen's d = 1.28
Error Rate	Granular vs Heuristic Model	Logistic Regression	0.015	Odds Ratio = 2.5
Latency	Granular vs All Models	ANOVA	0.025	—
QoS Index	Granular vs All Models	t-test	0.01	—

5 Discussion

This study is developed within an active and evolving field of research, where efficient resource management has become a critical priority due to the increase in connected devices. As discussed in previous works, such as those of [Lehocine and Batouche \(2018\)](#) and [Minhaj et al. \(2023\)](#), resource optimization has traditionally been addressed using techniques that do not dynamically adapt to changing network conditions. In contrast, our application of granular computing introduces significant flexibility, allowing real-time adjustments that respond to workload and resource availability variations.

Unlike more static approaches reported in the literature, our method can improve operational efficiency and service quality in IoT environments ([Goli et al., 2020](#)). The experimental results, detailed in [Table 4](#) and [Figure 4](#) of the Results section, highlight a notable improvement in resource allocation and utilization, directly translating into reduced latency and increased network stability. Thus, the system meets the demands of critical industrial applications that depend on fast and reliable responses.

Previous research has explored various approaches to resource optimization in IoT networks, such as heuristic methods for resource allocation, multi-objective optimization algorithms, and machine-learning techniques ([Motamedi et al., 2017](#)). While these methods have provided valuable insights and advances, they often face scalability, adaptability, and robustness limitations when faced with heterogeneous and dynamic IoT environments. The graphical comparisons in [Figure 4](#) clearly show how the proposed method outperforms these traditional approaches, particularly in scenarios with high variability in network demand.

The method proposed in this research, which leverages granular computing and granular balls, addresses several of these limitations ([Shuyin et al., 2023](#)). Granular computing offers a flexible and adaptable framework for managing resources, capable of handling the complexity and variability of IoT networks ([Yang et al., 2024](#)). The introduction of GBC further improves this approach by providing a more accurate and efficient way of representing and processing data. Granular balls encapsulate data in spherical structures, allowing for better clustering, classification, and neural network training. This novel approach improves the accuracy and robustness of data handling and facilitates scalable and efficient learning in dynamic IoT environments ([Zhang et al., 2021](#)). As demonstrated in the Results section, the comparative analysis in [Table 6](#) shows a substantial improvement in classification accuracy and efficiency when using GBC, highlighting the practical benefits of this method.

The implementation of granular computing has enabled more granular resource control, adapting workload distribution and resource allocation based on the specific needs of the moment ([Li](#)

[et al., 2022](#)). This innovative adaptive approach directly addresses one of the main limitations of previous models: the rigidity in managing dynamic changes within IoT networks. The trend analysis in [Figure 4](#) of the Results section underscores the system's ability to allocate resources, dynamically reducing inefficiencies observed in earlier models.

In this study, implementing granular computing techniques has proven to be an effective strategy for resource optimization in IoT networks, reflecting a significant improvement in several key performance indicators. This approach has allowed a more dynamic and efficient management of resources, which directly translates into improvements in the operability and sustainability of industrial systems. Initially, we observed a 25% increase in operational performance, going from 4,000 ops/s to 5,000 ops/s after implementing our techniques (see [Table 6](#)). This increase is crucial for environments that demand high processing capacity and fast responses, evidencing the ability of granular computing to distribute and manage the workload efficiently. This result is especially relevant compared to previous studies where conventional techniques showed limitations in adapting to the dynamic demands of the network.

Regarding energy efficiency, energy consumption was reduced from 150 W to 120 W, which implies an improvement of 20% (refer to [Table 6](#)). This optimization reduces operating costs and reinforces commitment to sustainable practices, an aspect increasingly valued in modern industry. The improvement in quality of service, which increased from 85% to 95%, underscores the system's ability to meet operational requirements more effectively without interruptions, strengthening system reliability and efficiency.

Reducing latency by 25%, from 100 milliseconds to 75 milliseconds, significantly improves communication and data processing, a critical factor in IoT where response speed can determine the success or failure of operations (see [Table 6](#)). The notable 60% decrease in error rate, from 5% to 2%, reflects increased system precision and stability, critical to maintaining continuity and safety in industrial operations.

These results validate the effectiveness of granular computing in improving resource management in IoT environments and illustrate its potential to adapt to changing environments, offering a more flexible and robust solution compared to traditional methods. However, it is essential to recognize the study's limitations, particularly the scale of the experimental deployment and the homogeneity of the test environments, which may not fully capture the complexity and variability of real-world applications.

Based on the findings obtained, additional testing is suggested in a broader variety of environments and with a larger scale of devices to verify the scalability and security of the proposed solutions ([Lehocine and Batouche, 2018](#)). This step is crucial to ensure that

improvements in resource management through granular computing can be effectively integrated into existing systems and to facilitate the transition towards more sustainable and efficient operating practices in the IoT industry (Yao et al., 2013).

The motivation for selecting the current method lies in its potential to overcome the challenges identified in related studies. By integrating granular computing with GBC, the proposed method offers a comprehensive solution that improves resource optimization, operational efficiency, and quality of service in IoT networks. This research contributes to the field by providing a robust and scalable framework that can adapt to the changing demands of IoT applications, ensuring more efficient and effective resource management.

6 Conclusion

This study has explored the potential of granular computing to optimize resource management in IoT networks, addressing crucial challenges such as energy efficiency, latency reduction, and error minimization. The results conclusively demonstrate that implementing granular computing techniques significantly improves operational performance, energy efficiency, and quality of service in IoT network environments, particularly in industrial applications.

We have seen a 25% increase in operational performance, from 4,000 ops/s to 5,000 ops/s, highlighting the ability of our techniques to handle high volumes of operations efficiently, a significant improvement for environments that require rapid responses and intensive processing. Reducing energy consumption from 150 W to 120 W (20% improvement) is especially relevant in the context of sustainability and reducing operating costs. This result marks the positive impact of granular computing in promoting greener and cheaper practices in the industry.

The increase in service quality from 85% to 95% reflects the system's ability to meet operational demands more effectively, ensuring service continuity and reliability, which is critical to maintaining high operational standards. The decrease in latency of 25% and the notable reduction in the error rate of 60% demonstrate the effectiveness of granular computing in improving communication and precision in IoT networks, contributing to a more stable and secure operation. These findings confirm that granular computing is viable and advantageous for resource management in IoT networks, offering an adaptable and robust solution to the limitations of conventional techniques.

Recognizing that the results were obtained under controlled conditions and could vary in more complex and heterogeneous real-world environments is essential. With the results obtained by granular computing in this study, future research should focus on several directions. First, it is crucial to conduct tests in real-world and larger-scale environments to validate the applicability and scalability of the proposed techniques. This includes expanding into different industrial sectors and adapting solutions to various IoT devices and operating conditions. Second, integrating AI and machine learning with granular computing should be explored further to improve the adaptability and efficiency of the systems. These technologies could enable more dynamic and predictive

optimization, adapting in real-time to changes in the operating environment.

One of the primary limitations is the computational overhead associated with creating and managing granular structures, especially in large-scale IoT networks with numerous devices and data points. Clustering and dynamically adjusting granular balls requires substantial processing power and memory, which may not be feasible for all IoT applications, particularly those with constrained computational resources. Additionally, the implementation of GBC can introduce latency in real-time applications due to the complexity of the algorithms. Furthermore, the effectiveness of GBC heavily depends on the quality and granularity of the input data; poor data quality or insufficient granularity can significantly impact the accuracy and reliability of the resource optimization outcomes. Future work should explore methods to mitigate these limitations, such as optimizing the clustering algorithms for efficiency, exploring lightweight computational techniques, and improving data preprocessing methods to ensure high-quality inputs.

Data availability statement

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

Author contributions

JG: Data curation, Formal Analysis, Methodology, Software, Validation, Visualization, Writing – original draft. RG: Data curation, Formal Analysis, Methodology, Software, Validation, Visualization, Writing – original draft. WV-C: Conceptualization, Formal Analysis, Investigation, Methodology, Supervision, Validation, Visualization, Writing – review and editing.

Funding

The author(s) declare that financial support was received for the research and/or publication of this article. Details of all funding sources, including grant numbers if applicable, should be provided. Please ensure that all necessary funding information is added, as this is no longer possible after publication.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Generative AI statement

The author(s) declare that no Generative AI was used in the creation of this manuscript.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated

organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

- Abir, SMAA, Anwar, A., Choi, J., and Kayes, A. S. M. (2021). Iot-enabled smart energy grid: applications and challenges. *IEEE Access* 9, 50961–50981. doi:10.1109/access.2021.3067331
- Acampora, G., and Vitiello, A. (2023). Improving quantum genetic optimization through granular computing. *Granul. Comput.* 8 (4), 709–729. doi:10.1007/s41066-022-00348-9
- Alghayadh, F. Y., Jena, S. R., Gupta, D., Singh, S., Bakhriddinovich, I. B., and Batla, Y. (2024). Dynamic data-driven resource allocation for NB-IoT performance in mobile devices. *Int. J. Data Sci. Anal.* doi:10.1007/s41060-023-00504-7
- Alshawi, A. A. A., Tanha, J., and Balafar, M. A. (2024). An attention-based convolutional recurrent neural networks for scene text recognition. *IEEE Access* 12, 8123–8134. doi:10.1109/access.2024.3352748
- Anseere, J. A., Kamal, M., Khan, I. A., and Aman, M. N. (2023). Dynamic resource optimization for energy-efficient 6G-IoT ecosystems. *Sensors* 23 (10), 4711. doi:10.3390/s23104711
- Behzadidoost, R., Mahan, F., and Izadkhah, H. (2024). Granular computing-based deep learning for text classification. *Inf. Sci. (N Y)* 652, 119746. doi:10.1016/j.ins.2023.119746
- Bolettieri, S., Bruno, R., and Mingozzi, E. (2021). Application-aware resource allocation and data management for MEC-assisted IoT service providers. *J. Netw. Comput. Appl.* 181, 103020. doi:10.1016/j.jnca.2021.103020
- Catarinucci, L., De Donno, D., Mainetti, L., Palano, L., Patrono, L., Stefanizzi, M. L., et al. (2015). An IoT-aware architecture for smart healthcare systems. *IEEE Internet Things J.* 2 (6), 515–526. doi:10.1109/jiot.2015.2417684
- Dehury, C. K., Veeravalli, B., and Srirama, S. N. (2024). HeRAFC: heuristic resource allocation and optimization in MultiFog-Cloud environment. *J. Parallel Distrib. Comput.* 183, 104760. doi:10.1016/j.jpdc.2023.104760
- Delgado, C., Canales, M., Ortin, J., Gallego, J. R., Redondi, A., Bousnina, S., et al. (2017). "Energy-aware dynamic resource allocation in virtual sensor networks," in *2017 14th IEEE annual consumer communications and networking conference, CCNC 2017*.
- Demirpolat, A., Sarica, A. K., and Angin, P. (2021). ProtEdge: a few-shot ensemble learning approach to software-defined networking-assisted edge security. *Trans. Emerg. Telecommun. Technol.* 32 (6). doi:10.1002/ett.4138
- Goli, A., Moeini, E., Shafee, A. M., Zamani, M., and Touti, E. (2020). Application of improved artificial intelligence with runner-root meta-heuristic algorithm for dairy products industry: a case study. *Int. J. Artif. Intell. Tools* 29 (5), 2050008. doi:10.1142/s0218213020500086
- Hussein, M. K., and Mousa, M. H. (2020). Efficient task offloading for IoT-Based applications in fog computing using ant colony optimization. *IEEE Access* 8, 37191–37201. doi:10.1109/access.2020.2975741
- Kumar, I., Rawat, J., Mohd, N., and Husain, S. (2021). Opportunities of artificial intelligence and machine learning in the food industry. *J. Food Qual.* 2021, 1–10. doi:10.1155/2021/4535567
- Lee, H., and Lee, U. (2022). Toward dynamic consent for privacy-aware pervasive health and well-being: a scoping review and research directions. *IEEE Pervasive Comput.* 21 (4), 25–32. doi:10.1109/mpvc.2022.3210747
- Lhocine, M. B., and Batouche, M. (2018). "Achieving efficiency in autonomic network management of IP networks based on SDN management logic," in *International journal of communication networks and distributed systems*.
- Lfao, P., Ferreira, T. M., and Costa, A. H. R. (2023). Data augmentation techniques in natural language processing. *Appl. Soft Comput.* 132, 109803. doi:10.1016/j.asoc.2022.109803
- Li, W., Tang, Y., Zhang, C., and Zhan, T. (2022). Multigranulation-based granularity selection for intuitionistic fuzzy weighted neighborhood IoT data. *Wirel. Commun. Mob. Comput.* 2022, 1–14. doi:10.1155/2022/5284804
- Liu, S., Guo, B., Fang, C., Wang, Z., Luo, S., Zhou, Z., et al. (2024). Enabling resource-efficient AIoT system with cross-level optimization: a survey. *IEEE Commun. Surv. Tutorials.* 26 (1), 389–427. doi:10.1109/comst.2023.3319952
- Loia, V., Orciuoli, F., and Pedrycz, W. (2018). Towards a granular computing approach based on Formal Concept Analysis for discovering periodicities in data. *Knowl. Based Syst.* 146, 1–11. doi:10.1016/j.knsys.2018.01.032
- Mahan, F., Rozekhani, S. M., and Pedrycz, W. (2021). A novel resource productivity based on granular neural network in cloud computing. *Complexity* 2021 (1), 5556378
- Mele, A., Vitiello, A., Bonano, M., Miano, A., Lanari, R., Acampora, G., et al. (2022). On the joint exploitation of satellite DInSAR measurements and DBSCAN-based techniques for preliminary identification and ranking of critical constructions in a built environment. *Remote Sens. (Basel)* 14 (8), 1872. doi:10.3390/rs14081872
- Minhaj, S. U., Mahmood, A., Abedin, S. F., Hassan, S. A., Bhatti, M. T., Ali, S. H., and Gidlund, M. (2023). Intelligent resource allocation in LoRaWAN using machine learning techniques. *IEEE Access* 11, 10092–10104. doi:10.1109/ACCESS.2023.3240308
- Motamedi, M., Fong, D., and Ghiasi, S. (2017). Machine intelligence on resource-constrained IoT devices: the case of thread granularity optimization for CNN inference. *ACM Trans. Embed. Comput. Syst.* 16 (5s), 1–19. doi:10.1145/3126555
- Neto, E. C. P., Dadkhah, S., Sadeghi, S., Molyneaux, H., and Ghorbani, A. A. (2024). A review of Machine Learning (ML)-based IoT security in healthcare: a dataset perspective. *Comput. Commun.* 213, 61–77. doi:10.1016/j.comcom.2023.11.002
- Panda, M., and Abraham, A. (2014). Development of a reliable trust management model in social internet of things. *Int. J. Trust Manag. Comput. Commun.* 2 (3), 229. doi:10.1504/ijtmcc.2014.067305
- Pop, P., Zarrin, B., Barzegaran, M., Schulte, S., Punnekkat, S., Ruh, J., et al. (2021). The FORA fog computing platform for industrial IoT. *Inf. Syst.* 98, 101727. doi:10.1016/j.is.2021.101727
- Rani, R., Kumar, N., and Khurana, M. (2024). Redundancy elimination in IoT oriented big data: a survey, schemes, open challenges and future applications. *Clust. Comput.* 27 (1), 1063–1087. doi:10.1007/s10586-023-04209-1
- Rani, S., Bhambri, P., and Kataria, A. (2023). "Integration of IoT, big data, and cloud computing technologies: trend of the era," in *Big data, cloud computing and IoT: tools and applications*.
- Rani, S., and Chauhdary, S. (2018). A novel framework and enhanced QoS big data protocol for smart city applications. *Sensors* 18 (11), 3980. doi:10.3390/s18113980
- Shuyin, X., Dawei, D., Long, Y., Li, Z., Danf, L., hao, Z., et al. (2023). Graph-based representation for image based on granular-ball. *Comp. Vis. and Patter. Recognition* 1 (1), 1–9. Available online at: <http://arxiv.org/abs/2303.02388>.
- Shwe, H. Y., Jet, T. K., and Chong, P. H. J. (2016). "An IoT-oriented data storage framework in smart city applications," in *2016 international conference on information and communication technology convergence, ICTC 2016*, 106–108.
- Su, X., Dong, W., Lu, J., Chen, C., and Ji, W. (2022). Dynamic allocation of manufacturing resources in IoT job shop considering machine state transfer and carbon emission. *Sustain. Switz.* 14 (23), 16194. doi:10.3390/su142316194
- Tang, Y., Chen, J., Pedrycz, W., Ren, F., and Zhang, L. (2024). Universal quintuple implicational algorithm: a unified granular computing framework. *IEEE Trans. Emerg. Top. Comput. Intell.* 8 (1), 1044–1056. doi:10.1109/tetci.2023.3327719
- Wang, D., and Guo, X. (2022). Research on evaluation model of music education informatization system based on machine learning. *Sci. Program*, 2022.
- Wang, D., Webb, S., Lee, K., Caverlee, J., and Pu, C. (2021). "Granular computing system vulnerabilities: exploring the dark side of social networking communities," in *Encyclopedia of complexity and systems science*.
- Wang, X., Yang, J., and Lu, W. (2023). Bearing fault diagnosis algorithm based on granular computing. *Granul. Comput.* 8 (2), 333–344. doi:10.1007/s41066-022-00328-z
- Webb, S., Caverlee, J., and Pu, C. (2010). "A summary of granular computing system vulnerabilities: exploring the dark side of social networking communities," in *Proceedings - 2010 IEEE international conference on granular computing (GrC 2010)*.
- Yang, J., Liu, Z., Xia, S., Wang, G., Zhang, Q., Li, S., et al. (2024). 3WC-GBNRS++: a novel three-way classifier with granular-ball neighborhood rough sets based on uncertainty. *IEEE Trans. Fuzzy Syst.* 32, 4376–4387. doi:10.1109/tfuzz.2024.3397697
- Yao, J. T., Vasilakos, A. V., and Pedrycz, W. (2013). Granular computing: perspectives and challenges. *IEEE Trans. Cybern.* 43 (6), 1977–1989. doi:10.1109/tsmcc.2012.2236648
- Zhang, G., Pan, J., Zhang, Z., Zhang, H., Xing, C., Sun, B., et al. (2021). Hybrid graph convolutional network for semi-supervised retinal image classification. *IEEE Access* 9, 35778–35789. doi:10.1109/access.2021.3061690
- Zhu, R., Peng, W., Wang, D., and Huang, C. G. (2023). Bayesian transfer learning with active querying for intelligent cross-machine fault prognosis under limited data. *Mech. Syst. Signal Process.* 183, 109628. doi:10.1016/j.ymssp.2022.109628