mathematics

MDPI

Article

Organized Optimization Integration Validation Model for Internet of Things (IoT)-Based Real-Time Applications

Abdullah Alghuried [1], Moahd Khaled Alghuson [1,*], Turki S. Alahmari [2] and Khaled Ali Abuhasel [3,*]

[1] Department of Industrial Engineering, Faculty of Engineering, University of Tabuk, Tabuk 47512, Saudi Arabia; aalghuried@ut.edu.sa
[2] Department of Civil Engineering, Faculty of Engineering, University of Tabuk, Tabuk 47512, Saudi Arabia; talahmari@ut.edu.sa
[3] Industrial Engineering Department, College of Engineering, University of Bisha, Bisha 61922, Saudi Arabia
* Correspondence: malghuson@ut.edu.sa (M.K.A.); kabuhasel@ub.edu.sa (K.A.A.)

Abstract: Emerging technology like the Internet of Things (IoT) has great potential for use in real time in many areas, including healthcare, agriculture, logistics, manufacturing, and environmental surveillance. Many obstacles exist alongside the most popular IoT applications and services. The quality of representation, modeling, and resource projection is enhanced through interactive devices/interfaces when IoT is integrated with real-time applications. The architecture has become the most significant obstacle due to the absence of standards for IoT technology. Essential considerations while building IoT architecture include safety, capacity, privacy, data processing, variation, and resource management. High levels of complexity minimization necessitate active application pursuits with variable execution times and resource management demands. This article introduces the Organized Optimization Integration Validation Model (O2IVM) to address these issues. This model exploits k-means clustering to identify complexities over different IoT application integrations. The harmonized service levels are grouped as a single entity to prevent additional complexity demands. In this clustering, the centroids avoid lags of validation due to non-optimized classifications. Organized integration cases are managed using centroid deviation knowledge to reduce complexity lags. This clustering balances integration levels, non-complex processing, and time-lagging integrations from different real-time levels. Therefore, the cluster is dissolved and reformed for further integration-level improvements. The volatile (non-clustered/grouped) integrations are utilized in the consecutive centroid changes for learning. The proposed model's performance is validated using the metrics of execution time, complexity, and time lag.

Keywords: integration validation; IoT; k-means clustering; prioritization; real-time applications

MSC: 68M18

check for updates

Citation: Alghuried, A.; Alghuson, M.K.; Alahmari, T.S.; Abuhasel, K.A. Organized Optimization Integration Validation Model for Internet of Things (IoT)-Based Real-Time Applications. *Mathematics* **2024**, *12*, 2385. https://doi.org/10.3390/math12152385

Academic Editor: Ali Shahidinejad

Received: 30 June 2024
Revised: 27 July 2024
Accepted: 30 July 2024
Published: 31 July 2024

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

1. Introduction

Navigating the intricate landscape of IoT integration within real-time applications is crucial for ensuring optimal functionality [1]. Challenges arise from variable execution times and resource management intricacies, necessitating innovative strategies for effective resolution [2]. Data integration processes with optimized energy utilization need to consider an IoT communication system [3]. IoT integrates and organizes sensors and devices for optimal analysis and administration. IoT can manage real-time data from numerous sources for decision-making and analysis. IoT devices monitor and communicate data to improve real-time operations and procedures. Adaptability emerges as a cornerstone in addressing the evolving demands of integration, facilitating seamless operation. Drawing insights from past integration experiences is an asset, informing future strategies and enhancing overall efficiency [4]. Rigorous performance evaluations play a pivotal role in validating the effectiveness of integration approaches and ensuring optimal outcomes [5].

Accurate complexity estimation is fundamental for successful deployment, which guides decision-making processes and resource allocation. Meticulous attention to detail and a commitment to continuous improvement are indispensable in navigating the complexities of IoT integration and ensuring the efficacy of real-time applications [6].

Efficiently managing IoT integration complexities within real-time applications is critical for smooth operation. These complexities, ranging from fluctuating execution times to intricate resource management, present significant challenges that require innovative solutions [7]. Harmonizing integration processes and ensuring validation accuracy are crucial components, providing seamless functionality [8]. Flexibility is essential in adapting to the dynamic nature of integration requirements, facilitating agility and responsiveness [9]. Leveraging insights from past experiences enables the refinement of integration strategies, enhancing overall effectiveness and performance. Rigorous performance evaluations validate the success of integration efforts, guiding future optimizations [10]. Accurate complexity estimation guides decision-making processes and resource allocation, ensuring efficient deployment. Continuous refinement and improvement are vital for achieving seamless integration and optimizing the performance of real-time IoT applications [11]. The associated cloud framework integrates an efficient k-means clustering technique with an accurate diabetes prediction that manages enormous dynamic data [12].

There are many reasons for implementing IoT for real-time applications. Some of these include enhancing the quality of everyday life, utilizing resources more efficiently, fostering a stronger connection between humans and the environment, enhancing transportation for everyone, ensuring global internet connectivity, simplifying use, and enhancing accessibility. These many benefits have propelled IoT to the status of a new industrial revolution. Some of the most lauded uses of IoT include "smart" houses, farms, transportation, healthcare (through the Internet of Medical Things, or IoMT), businesses, and supply chains. Healthcare is one actual use of IoT, which offers many benefits over older medical treatment methods. Optimal hospital administration, cost and risk reduction in medical procedures, faster diagnosis, better use of healthcare resources, telemedicine, self-service Medicare, early illness detection, and support for these applications are all possible thanks to their development and implementation. IoT presents many opportunities as well as numerous obstacles. The architecture is the biggest concern when deploying IoT in real-time. Companies like Cisco, Intel, and IBM have built their reference architectures around how they approach creating IoT technologies; thus, there is no universally accepted IoT architecture. Interchangeability, security, reliability, safety, variation, data management, and connectivity are just a few of the obstacles the IoT architectural standards must overcome. With the proliferation of smart devices comes an explosion in data, which in turn causes problems with data management. Devices from different manufacturers, use of various protocols, and adherence to different architecture standards complicate communication in a heterogeneous environment. Because IoT devices communicate via multiple methods, interoperability is a significant concern. Protecting users' data from disclosure in the IoT ecosystem is crucial for ensuring privacy and building consumers' trust. Ensuring IoT's scalability is critical for devices to be able to adjust to future needs.

Several companies use one service technique to maximize the efficiency of their resources. Unfortunately, few consider the risks of ignoring asset deterioration early in defect creation. The three main types of maintenance are corrective, preventive, and predictive. Predictive and corrective approaches prioritize asset age and maintenance periodicity. These assets often break down between checks, increasing maintenance costs and safety issues. Many companies believe that you can predict when an asset will fail by examining its service life and performing breakdown maintenance. IoT connects physical objects to the internet so they may exchange data. This capacity could simplify massive data collection, boosting organizations' bottom lines and forecasts. Embedded hardware, including sensors and other IoT-managed intelligent equipment, is transforming production and industrial maintenance. When used for predictive maintenance, IoT can be practical.

For real-time applications to run smoothly, it is essential to efficiently manage IoT integration complexity. These intricacies, which include complex resource management and execution durations that might vary greatly, pose big problems that need creative answers. Investigating learning strategies that might reduce integration complexity within IoT paradigms is vital for more efficient operation. These methods provide helpful information for improving the efficacy and efficiency of integration procedures in real-time applications. To achieve this goal of constant improvement, it is essential to embrace fresh approaches and use previous experiences. Organizations can improve their efficiency and adaptability by adjusting and perfecting their integration methods in response to what they learn. Evaluation of integration performance is crucial for driving optimization efforts and ensuring continual improvement and refinement. Users can use the intelligent environment's many interoperable capabilities to communicate with real-time IoT applications. The organized optimization validation model proposed in this paper aims to identify and reduce complexity in IoT integration for large applications. By separating unstructured classes based on integration degree, time latency, and complexity using modified k-means clustering, the performance of the proposed model is validated through evaluation and verification using simulation metrics.

Exploring learning techniques to mitigate integration complexity within IoT paradigms is essential for streamlined operations. These techniques offer valuable insights into optimizing integration processes for real-time applications, fostering efficiency and effectiveness [13,14]. Embracing innovative methodologies and leveraging past experiences are key strategies in this pursuit, facilitating continuous improvement [15]. By adapting and refining integration strategies based on learned insights, organizations can increase efficiency and adaptability. A rigorous evaluation of integration performance ensures constant improvement and refinement, driving optimization efforts [16,17]. Attention to the multiple interoperable features that allow users to communicate with real-time IoT applications exploits the advantages of an intelligent environment [18]. By transferring data processing responsibilities from overloaded edge devices to unused cloud servers, an industrial IoT system and smart agriculture can dynamically reallocate computing power according to the OOIVM paradigm. This helps to reduce processing latency and prevent system bottlenecks.

The proposed model's contribution is presented as follows:

a. The organized optimization validation model identifies and reduces complexity in IoT integration for large-scale applications.
b. Modified k-means clustering is used to segregate non-organized classifications based on complexity, time lag, and integration levels.
c. The verification and assessment of the proposed model are conducted using simulation-related metrics to validate the proposed model's performance.

2. Related Works

Wang et al. [19] analyzed the low-complexity and efficient dependent subtask offloading strategy of IoT. The study investigated the fulfillment of offloading requirements in IoT with multi-access edge computing systems. It also analyzed the problem of task offloading and scheduling in IoT networks. The exact cause of the problems and tasks were identified to solve these issues. The proposed system expands the performance improvement range of the networks. However, an important limitation of the system is that potential solutions may not consider the complex interplay of several contemporary healthcare innovations.

Si-Mohammed et al. [20] studied network simulation (NS) in network digital twin (NDT) in IoT networks. The study explored the advantages and purposes of the tools for their application. It identified the often-prohibitive simulation costs in modeling NS + NDT. It also accelerated the accuracy of decision-making processes. The proposed model reduces the optimization cost and computational complexity ratio of the systems but increases latency overhead.

Brandín et al. [21] proposed a blockchain integration-based framework for data traceability in IoT. The proposed framework is used for offsite manufacturing (OSM), which provides proper manufacturing. It predicts the ledger's behavioral patterns, enhancing the network's convergence range. The proposed framework increases the effectiveness of shifts in market demand, reducing task latency. The main drawback of the proposed model includes planning issues.

Jin et al. [22] designed a multi-objective optimization approach for integrating IoT and blockchain technology. The authors employed a wireless sensor network (WSN) to determine the network's multi-objectiveness. The precise approach employed provides a significant dimension for problem resolution. It minimizes the cost and energy consumption ratio as well as optimization problems but has interoperability issues.

Alulema et al. [23] developed SI4IoT based on model-driven engineering for integration. The developed model identifies the issues that increase the system's workload. The developed model tackles the problems by solving the transformation artifacts in the network. It is used as a new sophisticated model that decreases the diversity range of communication protocols. The model expands the feasibility and increases the accuracy of interaction services.

Irwanto et al. [24] introduced an innovative system based on fuzzy logic integration for IoT networks. The introduced system is used to safeguard the parameters that are used in mushroom farms. It is used to analyze the problems in maintaining the environmental conditions in farming. The fuzzy concept enhances the precision level in the decision-making process, improving agriculture's sustainability level. The introduced system provides sustainable quality development strategies for mushroom farming. The limitation of the proposed model is reduced scalability and reliability.

Chakour et al. [25] developed an innovative approach for integrating IoT networks. It is used to enhance the quality of service (QoS) range of IoT networks. The developed approach uses a tailored resource allocation technique, allocating the resources depending on the requirement. The approach also uses the Monte Carlo control algorithm to address issues in the network. The developed approach increases the performance and feasibility level of the systems.

Narang et al. [26] analyzed the management of an open-loop water value chain (WVC) for IoT networks. The analyzed IoT implementation range in WVC minimizes latency in the decision-making process. A principal component analysis (PCA) is employed here to examine the network's decision-making and planning strategies. The study also identified the configuration and adaptation levels of WVC in IoT applications. The performance efficacy of the system is moderate.

Wu et al. [27] designed an energy-efficient dynamic task offloading (EEDTO) algorithm for blockchain-enabled IoT applications. In this, mobile edge computing (MEC) is employed to reduce the energy consumption in the offloading process. It is applied to elevate the task response time for users online. The designed algorithm decreases the overall computational complexity ratio of IoT networks.

Wang et al. [28] proposed a popularity-aware push-based cache (POPCache) for ICN-based IoT networks. In this system, determining factors are used to make cache decisions that reduce the cost level. The proposed cache strategy achieves low overhead when performing user cache tasks. It also enhances the cache consistency ratio in IoT networks. Experimental results showed that the proposed approach increases the accuracy of network performance. However, the proposed model demands continuous monitoring of the system.

Eroshkin et al. [29] developed a resource-efficient reliability model for multi-agent system (MAS) IoT. In this model, the exact service level agreement (SLA) requirement is analyzed using a mathematical analysis process. The SLA requirements are categorized based on the severity and necessity of the tasks in the network. The developed model enhances the reliability and feasibility range of IoT agents.

Chahed et al. [30] proposed an artificial intelligence-driven network and processing framework (AIDA) for industrial IoT (IIoT) applications. The proposed framework uses a time-sensitive network (TSN) to analyze the data ingestion range in the application. Machine learning (ML) pipelines are used here to make the ratio level of data ingestion flexible. The proposed framework provides effective infrastructural services to build a proper user architecture.

Zhang et al. [31] designed an adaptive differential evolution (ADE) algorithm for fog service placement (FSP) in IoT applications. The developed algorithm identifies the multi-objective optimization problems that cause evolutionary issues. Fog computing technology is used here to ensure the application's processing quality. The ADE algorithm minimizes deadline violations and provides a cost-effective network range.

Wang et al. [32] investigated the multi-branch scalable neural network (MBSNN) for resource-constrained IoT devices. This system investigates the problems that occur in deploying and designing the networks. A threshold selection-based adaptive inference mechanism is implemented to select the necessary threshold for the MBSNN architecture. The investigation detects the exact efficiency and feasibility level of IoT-enabled devices but suffers from the limitation of latency overhead.

Guan et al. [33] developed a deep reinforcement learning (DRL)-based efficient access scheduling algorithm (EASA) for IoT systems. Learning algorithms were employed to train the nodes for access scheduling. The interference features were extracted using a feature extraction technique that minimizes the computational complexity of the network. The developed algorithm enlarges the performance range of the networks.

Kim et al. [34] proposed a hierarchical edge-based architecture for IoT applications. It is used for service orientation, which provides effective communication services to the users. It is also used to increase the scalability and viability ratios of communication systems. It is used in the communication IoT platform to reduce latency in task responses. The proposed architecture improves the reliability of the systems.

Silva et al. [35] developed an agnostic hardware operating system framework for low-end IoT devices. The developed framework uses an open-source instruction set architecture (ISA) to accelerate the services for the users. The developed framework is also used to schedule device resource allocation tasks. Experimental results showed that the developed framework enhances performance, minimizes the energy consumption ratio in operations, and addresses the issue of run-time verification.

The O2IV model proposed in this paper is designed to reduce the complexity of various IoT application integrations through organized optimization and machine learning. In this scenario, the integration is estimated with varying execution times to identify the service level, complexity, and time lag. This integration is validated for minimizing complexity levels based on real-time applications' activeness and resource management.

3. Organized Optimization Integration Validation Model (O2IVM)

Based on these contributions, a diagrammatic illustration of the proposed model's flow is presented in Figure 1. The process flow of the abstract is as follows: The integration of the IoT application under different service levels is clustered based on complexity and lags. The grouped possibilities are used to verify and validate integration improvements. The changes in grouping deviations are used to train the clustering process. The proposed model, OOIVM, must process dynamic, constant information streams coming from a considerable number of IoT devices. This makes it hard to keep processing in real time, which means the system will be slow to respond to new data patterns. The model takes into account the difficulties of grouping and processing overhead. This enhances the chances of integration, reducing complexities (Figure 1).

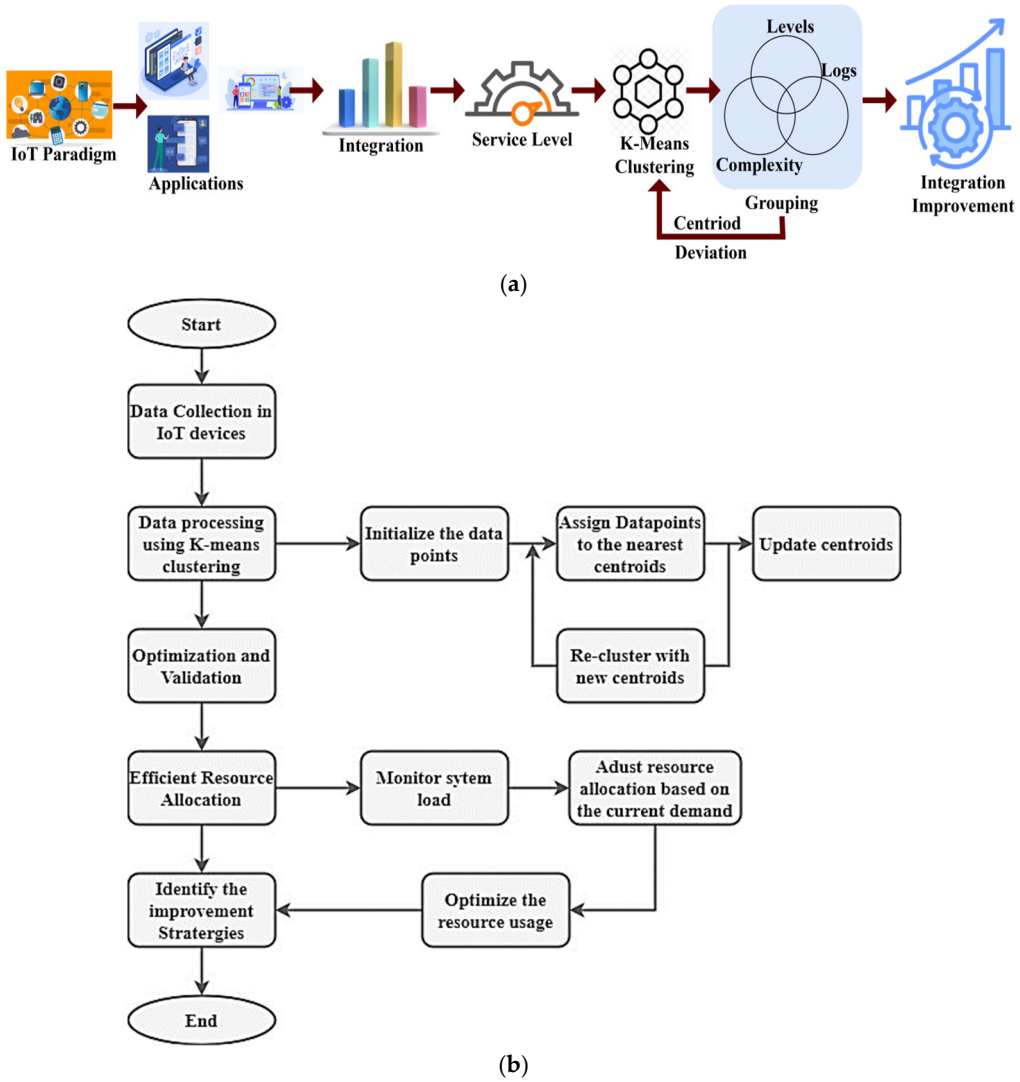


Figure 1. (**a**) Diagrammatic illustration of the proposed model's flow. (**b**) Flow graph representation of the proposed model.

By streamlining application processes, the IoT paradigm solves the complexity occurrence identification challenge. In this case, data from two sources are used: complexity addressing a problem using machine learning and organized optimization modeling of its outputs. IoT in real-time applications usually involves continuous data transmission to the infrastructure. At all times, numerous sensors and gadgets gather data and send it to a centralized location for processing via the web or regional networks. The internet implementation of the optimization challenge is also standard. To respond quickly and efficiently to shifting situations, the system optimizes and adjusts settings in real time using data. This online tuning is essential to keeping IoT applications running well and reliably in ever-changing contexts.

The need for complexity analysis and grouping is determined by the nature of the real-time applications for improving their resource projection quality, modeling, and representation based on interactive devices/interfaces. In the proposed O2IVM, for precise integration evaluation, non-optimized classification and prioritization are devised using the centroid deviation knowledge of the organized integration levels. Moreover, in most organized integration cases, the output of the problem of sequential centroid changes by one criterion and is reduced to prevent validation lags with accurate measures.

This flowchart is connected to the primary objective of enhancing efficiency, scalability, and immediate processing in IoT systems. The data collection phase involves IoT devices

collecting data, which is subsequently analyzed using k-means clustering for improved management. Optimization and validation guarantee the ongoing efficiency and accuracy of the system—the driving force behind the desire to decrease intricacy and enhance data administration. Using k-means clustering, OOIVM organizes comparable data points into groups, resulting in a more controllable and efficient system. OOIVM optimizes resource allocation by adjusting resources according to the current system load. This approach successfully uses computing and network resources, reducing costs and improving performance. The OOIVM undergoes continuous validation and refinement to ensure its adaptability to new data and circumstances, maintaining its effectiveness and reliability.

For using centroids for a single entity, harmonized service levels are required to minimize additional complexity demands and thereby confine complexity lag problems based on centroid deviation.

$$\exists = (\exists_1, \ldots, \exists_n, \ldots, \exists_N) \rightarrow max, n = \overline{1, N} \tag{1}$$

where $\exists = (\exists_1, \ldots, \exists_n, \ldots, \exists_N)$ is the number of real-time applications over the IoT platform. The complexity occurs at the time of device integration based on the service level changes. The model utilizes the complexity analysis, $\omega = (\omega_1, \ldots, \omega_n, \ldots, \omega_N)$, and identification of currently processing applications, $\exists = (\exists_1, \ldots, \exists_n, \ldots, \exists_N)$, for precisely addressing the problems above. The integration is evaluated as the number of service levels observed at different N instances. In this organized integration case, the validation lags, $\exists$, are due to the non-optimized classification issues of n. The variable εt signifies the execution time of currently processing applications based on their requirements. This analysis differentiates the same properties of IoT applications and determines the identification of complexities, $a = (a_1, \ldots, a_s, \ldots, a_S) \in \emptyset$, which contains the same ratio of non-optimized classification, $\bigodot$, and centroid deviation, Δ; for the condition, a' and a'' represent identified complexity levels, and a', $a'' \in \varepsilon t\ (\exists(a'))\bigodot(\exists(a''))$ occurs if and only if

$$\left.\begin{array}{c} \varepsilon t = \frac{1}{n}\left\| \sum_{i=1}^{n} \left(\frac{\exists(a') + \exists(a'')}{\Delta} \right) \right\| \\ where, \\ \exists(a') = \frac{1}{\sqrt{\pi}} \int_{-\infty}^{\infty} \frac{\exists(a') * \varepsilon t}{\varepsilon t} d.\varepsilon t \\ and \\ \exists(a'') = \frac{1}{\sqrt{\pi}} \int_{-\infty}^{\infty} \frac{\exists(a'') * \varepsilon t}{\varepsilon t} d.\varepsilon t \end{array}\right\} \tag{2a}$$

such that

$$\omega\left(\exists\left(a'\right)\right)\bigodot(\exists(a'')) \tag{2b}$$

In Equation (2), IoT application integrations are estimated for the applications defined in Equation (1). The specific service level is represented as $a_s^*, s = \overline{1, \alpha}$; it either has a single entity or a harmonized service level is considered. The mechanism for minimizing the integration complexity can be prevented with the following procedures.

Non-optimized integration classification is performed based on service level changes in real-time application representation; it also contains optimal outputs of $a_s^*, s = \overline{1, \alpha}$ required from the output of integrated optimization for each of the specific service level, $\exists_n, n = \overline{1, N}$. The introduction of a single entity between the harmonized service levels is $\exists_n, n = \overline{1, N}$, which allows the organized integration cases to be addressed and segregated to augment resource projection quality through appropriate optimized integration, $\emptyset$, as a function of the harmonized service level, $\emptyset(\exists) = \emptyset(\exists_1, \ldots \exists_N)$. The condition of $\exists(a')$ and $\exists(a'')$ $(\emptyset\exists(a') \leq \emptyset\exists(a''))$ occurs if and only if

$$\left.\begin{array}{c} \emptyset(\exists) = \exists_N - \omega_N * a_s^* \\ such\ that \\ arg \min_t \sum \omega_N \forall \varepsilon t \end{array}\right\} \tag{3a}$$

where

$$\exists(a') \bigodot \exists(a'') \in \varepsilon t \tag{3b}$$

Thus, the organized optimization integration of $\exists(a)$ is introduced based on grouping the service levels according to condition (2). The identification of accurate integration complexity, $a^0 \in \omega$, in the initial non-optimized classification problem is reduced to minimize the problem of additional complexity demands as follows:

$$\exists(\varnothing_n) = n.\exists(a') \rightarrow \varepsilon t \forall \omega = 0 \tag{4a}$$

and

$$\exists(\varnothing_N) = \frac{\omega}{n} \exists(a'') \rightarrow \omega * \varepsilon t \forall \omega \neq 0 \tag{4b}$$

such that

$$min_{a \in \omega} \varnothing \exists(a) \tag{4c}$$

In this case, the service level is defined by a single entity with weight measurements; the complexity in the organized integration $(\exists = \exists_1, \ldots, \exists_n)$ is minimized and prioritized. This is achieved using k-means clustering with weight measurements, implementing the intense complexity minimization levels for detecting precise complexity levels:

$$a_s^* = \begin{cases} N \times \exists(a) \times \varepsilon t \; \forall \; n :: \varepsilon t \; , \; \forall \; \omega = 0 \\ \exists_n \times N - a^0 \times \varepsilon t \; \forall \; (n, \omega) :: \varepsilon t, \; \forall \; \omega \neq 0 \end{cases} \tag{5a}$$

such that

$$\varnothing(a) = \varnothing \exists(a) \tag{5b}$$

Complexity minimization (5) can be performed to require the varying execution time, εt, only if the harmonized service level, $\exists_n, n = \overline{1, N}$, satisfies the following complexity demands. Specific service levels, $\exists_n, n = \overline{1, N}$, are suitable for minimizing and prioritizing complexity in non-optimized integrations. Each level can be assigned for certain non-optimized classifications, differentiating the non-optimized integrations in grouping to other service levels. Specific service levels, $\exists_n, n = \overline{1, N}$, are used to prevent validation lags or have a single entity.

Figure 2 above presents the analysis for different devices under lags and levels. The lags are computed using $\omega(\exists(a')) \bigodot 1 = \omega(\exists(a'))$ and $\exists(\varnothing_n) = n.(\exists(a''))$ conditions. For the different levels of integration, $\exists(a'') = \omega_n$ is the computable complexity that halts N. In both cases, the lag detection is defined using halted and $N = N - 1$ instances such that the complexity-causing factors are identified. The proportion of lag, level, or both in the identified complexity is used to group and independently identify its cause.

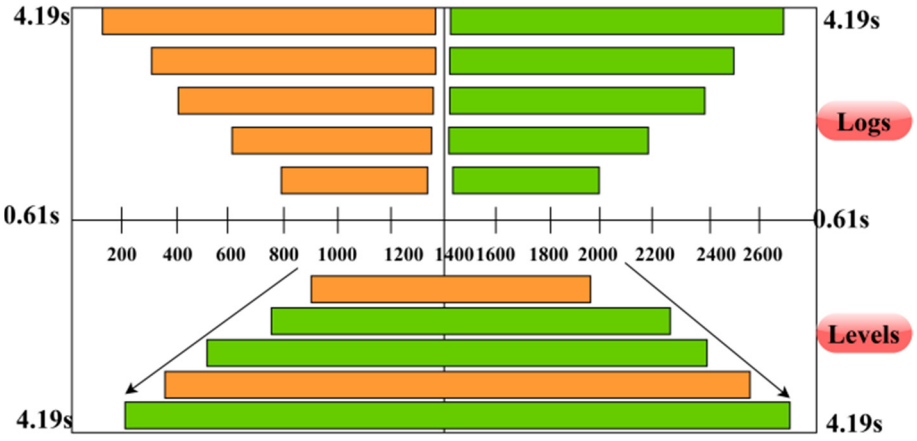


Figure 2. Time-based complexity analysis.

Vital to improving the precision and dependability of current information validation to guarantee the quality of choices and actions supported by IoT data is verification precision. Optimizing the efficiency of algorithmic optimization means making them more responsive to new data patterns and shifting system requirements. With these key performance indicators in mind, the OOIVM sets out to build a solid foundation for handling the intricacies of immediate-time IoT application management, aiming to enhance reliability, scalability, and performance.

The OOIVM expands upon the principles established by prior studies in IoT technology's cooperation, grouping, and resource management. It enhances clustering efficiency, resource consumption, flexibility, computational complexity, implementation speeds, and reliability of validation in a targeted manner. These improvements cumulatively lead to a more robust, more effective, and easily expandable architecture for real-time IoT applications.

3.1. K-Means Clustering Model

It refers to the grouping of similar kinds of complexities in the form of clusters. The real-time applications are responsible for integrating organized optimization from the IoT paradigms. The input data include service levels, complexities, and time lag. In the clustering process, initial data received (A) from the IoT paradigm is evaluated as follows.

$$A = SVL_i \frac{(Compl_{max} - Compl_{min})}{T_{Lag}} + \omega(\exists(a')) \bigodot (\exists(a'')) - \Delta + \alpha \tag{6}$$

and

$$\alpha = \frac{1}{\sqrt{2\pi}} \left[\frac{\left(\frac{Compl_{min}}{Compl_{max}} - \frac{\exists(a')}{\exists(a'')} \right)}{2(\varnothing\exists(a) - \varnothing(a))} \right] \tag{7}$$

where SVL_i is the number of service levels, and $Compl_{min}$ and $Compl_{max}$ are the minimum and maximum complexities detected in different instances. The variable α denotes the segregation of organized integration cases from the centroid for each cluster. The centroids estimate the number of non-optimized classifications (validation lags) observed over the different N service levels. The clustering initialization for grouping is illustrated in Figure 3.

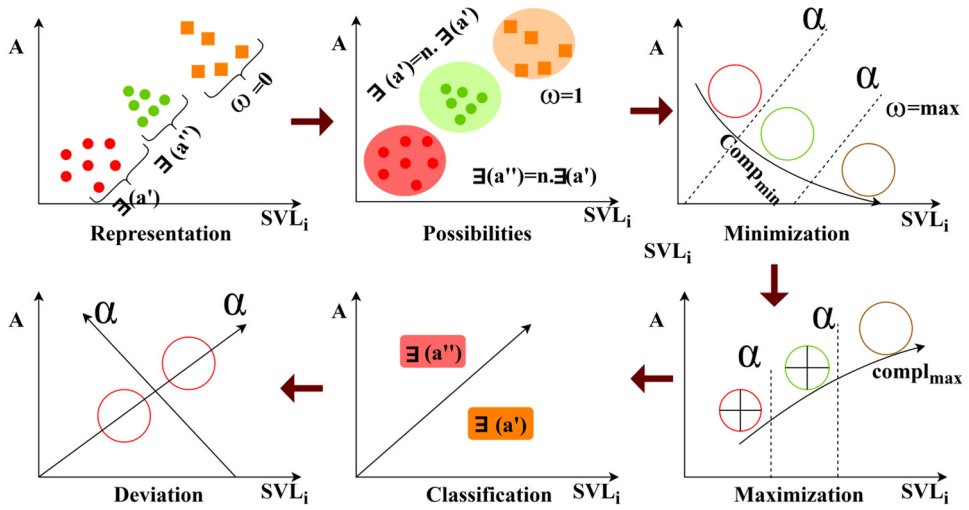


Figure 3. Clustering initialization for grouping.

In Figure 3 above, the A-based representation for grouping initialization is presented. By defining $\exists(a')$ and $\exists(a'')$, the $\bigodot$ condition defines the centroid changes. Both minimization, $min_{a\in\omega}\varnothing\exists(\alpha)$, and maximization, $max_{a\in\omega}\varnothing\exists(a'')$, represent the number of α required. As α increases, $\omega = 0$ is achieved by classifying $\exists(a'')$ and $\exists(a')$. Thus, the grouping

consolidates $\exists$ levels and ties to enable integration feasibility. The deviation is observed for $\exists(a')$ to $\exists(a'')$ switchover or $\omega = 1$ detection. Therefore, the grouping requires multiple considerations to improve integrations. In this organized case, complexity occurs during integration due to non-optimized classification problems. Therefore, the validation lags impact IoT integration at any level of the service for which the organized integration cases are segregated, $\mathbb{S}(INTG)$, using the centroid deviation knowledge. This segregation is represented as follows.

$$\mathbb{S}(INTG) = \frac{devt_{max}}{\left(\frac{Compl_{min}}{Compl_{max}} - devt_{\sigma}\right)^2} + \frac{\exists(a')}{\exists(a'')} - \alpha \tag{8}$$

and

$$devt_{\sigma} = \frac{1}{N}\sqrt{\frac{1}{n-1}\sum_{i=1}^{N}(\mathbb{C}_{dv} - \mathbb{P}_{dv})^2} \tag{9}$$

where the variables $\mathbb{C}_{dv}$ and $\mathbb{P}_{dv}$ signify the current centroid deviation and knowledge of previous deviations. Based on the centroid deviation, the segregation is pursued to follow the maximum centroid deviation $devt_{max}$ and standard deviation $devt_{\sigma}$ for accurately balancing the integration levels. Here, the number of sharing levels and the center point is used to identify the number of clusters $\mathcal{K}$ for which appropriate non-complex processing, integration level balancing, and time-lagging integrations from different levels are required for minimizing integration complexity.

Based on the current and previous deviations, the sequential row of clustering $\alpha[\mathbb{C}_{dv}, \mathbb{P}_{dv}]$ is estimated as follows:

$$\alpha[\mathbb{C}_{dv}, \mathbb{P}_{dv}] = \left[\frac{devt_{\sigma}}{devt_{max}}\right]_1^2 + \left[\frac{devt_{\sigma}}{devt_{max}}\right]_2^2 + \cdots + \left[\left(1 - \frac{\mathbb{C}_{dv}}{\mathbb{P}_{dv}}\right)\Delta\right]_n^2, n \in N \tag{10}$$

In Equation (10), the clustering for a sequence is performed until the changes occur in the centroid. The centroid deviation knowledge confines complexity lags in the service levels. The above clustering sequence is pursued using the k-means clustering process and machine learning to precisely identify complexities. In this analysis and grouping process, k-means clustering is exploited for detecting complexity lags across various IoT application integrations. The harmonized service levels are grouped in appropriate and accurate time instances to reduce validation and complexity lags. In addition, the cluster is dissolved and reformed to be instantaneous to meet complex demands for integration. Therefore, k-means clustering and machine learning are used to prevent validation lags. The sequential clustering for complexity and time lag is analyzed in Figure 4.

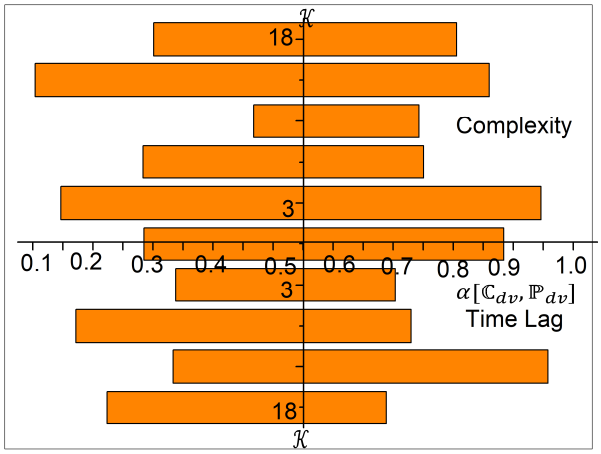


Figure 4. Sequential clustering analysis for complexity and time lag.

The complexity and T_{Lag}-based $\mathcal{K}$ requirements analysis is presented in Figure 4. Considering the $devt_{\sigma}$ and $devt_{max}$, the $[\mathbb{C}_{dv}, \mathbb{P}_{dv}]$ for the sequential process is defined. Further, $\omega = 0$ and $\omega \neq 0$ classifications are performed under $\left[\frac{devt_{\sigma}}{devt_{max}}\right]$ computation to ensure maximum α is identified. The integration complexity is thus estimated for four levels $(N = 4)$, i.e., field level, resource connectivity, response, and storage. Based on the increasing level of A, the above $\mathcal{K}$ is analyzed. In particular, the similar grouping rate is analyzed for four levels, as shown in Figure 5.

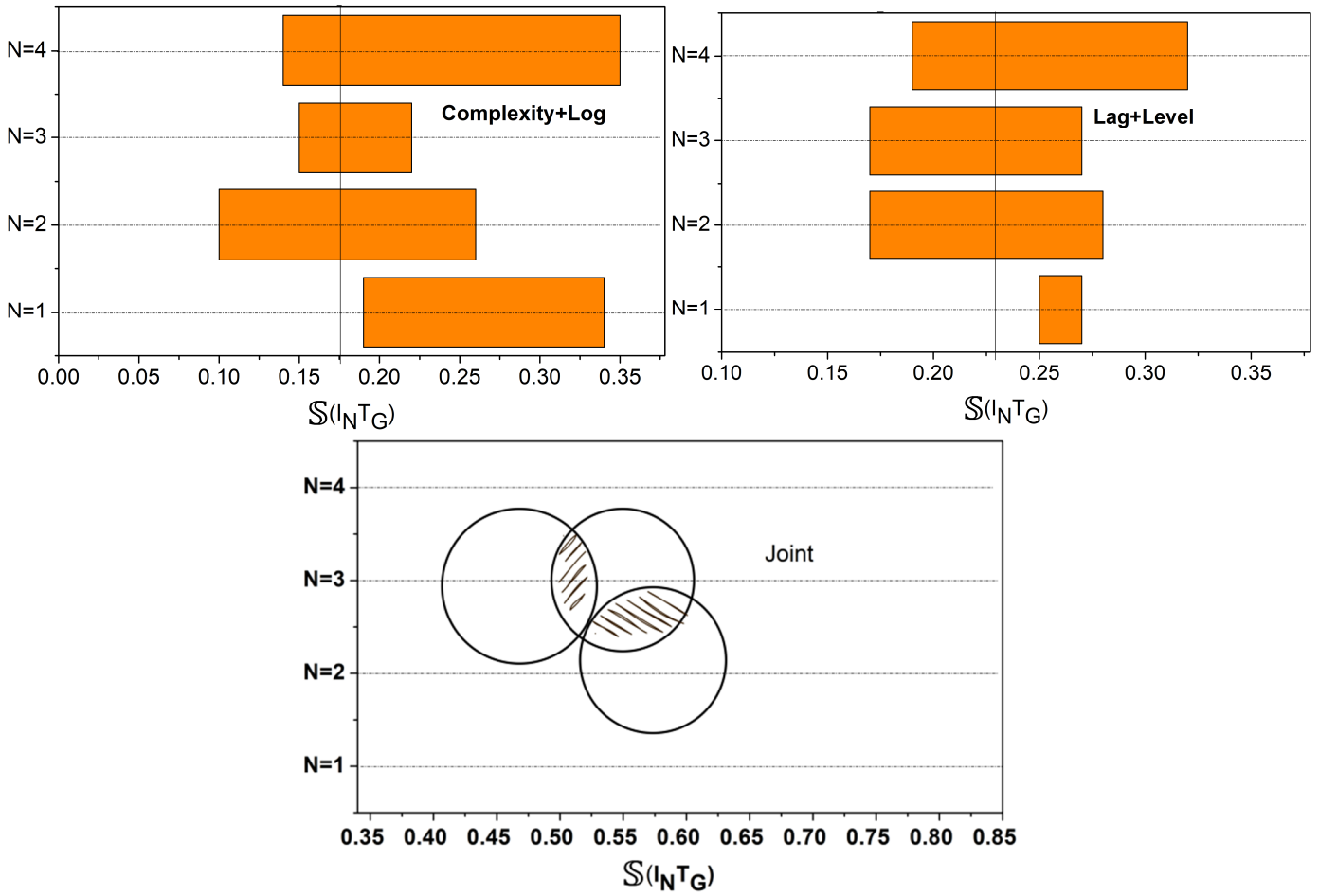


Figure 5. $S(INTG)$ analysis.

The analysis in Figure 5 presents the integration segregation for complexity + lag, lag + levels, and all three (joint). As the $\mathcal{K}$ increases, $\alpha[\mathbb{C}_{dv}, \mathbb{P}_{dv}]$ increases to identify the maximum complexity causing integration. Therefore, the identified grouping is segregated towards multiple instances of organized integration. The clusters are disintegrated and reformed under maximum N and α detected under reduced $devt_{\sigma}$. It also ensures $devt_{max}$ is not reached until a new integration request or complexity arises due to $N = N + 1$. The machine learning output aims to identify and segregate the organized integration cases with standard deviation and previous deviation knowledge updates based on training. In this learning process, the first step is to address the issues above using the clustering process. The number of clusters, $\mathcal{K}$, in service levels is achieved using the condition $\left(1 - \frac{\mathbb{C}_{dv}}{\mathbb{P}_{dv}}\right)\Delta$ for updating the previous deviation knowledge update. O2IVM is implemented to improve the integration level of complexity analysis based on centroid deviation. The centroid deviation is detected from different service levels, i.e., organized integration is observed in various time intervals for further clustering. This proposed model aims to reduce the validation lags in analyzing complexity. The challenge is to update the centroid deviation

and grouping sequence with previous deviation knowledge. The centroid deviations are stored as previously identified centroid deviation knowledge.

This proposed approach effectively meets the need for high availability by utilizing machine learning and k-means clustering to minimize time lags and response time. The centroids are identified from the clustering output to rectify suboptimal classifications and validate delays across various N service levels. Machine learning is utilized to discern the intricacies of fluctuating execution time and resource allocation. The structured integration scenarios are classified based on the centroid deviation to determine and distinguish the maximum centroid deviation (devt_max) and standard deviation (devt_σ) for balance integration levels. The current gadget reduces complexity by employing machine training and clustering techniques, reducing time lag and complexity and improving integration evaluation. This article employs machine learning to mitigate the intricacies in random situations characterized by high availability parameters, in contrast to other aspects.

3.2. Learning Process for Integration Validation

In the service level grouping process, machine learning is used to find out the complexity and centroid deviation estimation. As this machine learning process relies on previous knowledge of centroid deviations, a more accurate complexity lag is achievable. The number of clusters may vary based on service levels, though the previously stored centroid deviation knowledge helps to classify non-optimized integration. Machine learning helps to pursue two types of segments: sequence grouping and centroid deviation identification. Service levels, complexity, and time lags are identified in the sequence grouping process to update the previously stored centroid deviation knowledge. In the centroid deviation identification, the clustering process balances integration levels to augment the same ratio of non-optimized classification $\odot$, thereby reducing centroid deviation Δ along with better grouping. In the grouping process $\mathbb{G}$, the levels, complexity, and time lags are classified independently using k-means clustering. The integration improvement is achieved based on dissolved and reformed clusters, after which volatile integration is used to update the previous centroid deviation knowledge. The machine learning output sequence from $\mathbb{G}_1$ to $\mathbb{G}_T$ is computed as follows:

$$\left.\begin{array}{c} \mathbb{G}_1 = \mathbb{C}_{dv_1} \\ \mathbb{G}_2 = 2\mathbb{C}_{dv_2} - 2SVL_1 - \exists(\varnothing)_1 \\ \mathbb{G}_3 = 3\mathbb{C}_{dv_3} - 3SVL_2 - \exists(\varnothing)_3 \\ \vdots \\ \mathbb{G}_N = n\mathbb{C}_{dv_N} - nSVL_i - \exists(\varnothing)_{N-1} \end{array}\right\} \tag{11}$$

$$\left.\begin{array}{c} \mathbb{U}_1 = \mathbb{C}_1 \\ \mathbb{U}_2 = \mathbb{C}_2 + \left[\left(1 - \frac{\mathbb{C}_{dv}}{\mathbb{P}_{dv}}\right)\Delta\right]_1 \\ \mathbb{U}_3 = \mathbb{C}_3 + \left[\left(1 - \frac{\mathbb{C}_{dv}}{\mathbb{P}_{dv}}\right)\Delta\right]_2 - \omega(\exists(a')) \bigodot (\exists(a''))_1 \\ \vdots \\ \mathbb{U}_N = \mathbb{C}_{\mathcal{K}} + \left[\left(1 - \frac{\mathbb{C}_{dv}}{\mathbb{P}_{dv}}\right)\Delta\right]_{n-1} - \omega(\exists(a')) \bigodot (\exists(a''))_{n-2} \end{array}\right\} \tag{12}$$

The grouping process generates two outputs, namely, levels, complexity, and time lag from $\mathbb{G}_1$ to $\mathbb{G}_N$ and update of the sequence from $\mathbb{U}_1$ to $\mathbb{U}_N$. Based on the number of clusters, the grouping is pursued using k-means clustering based on the complexity of the lag occurring in the sequence. Each service level is analyzed to identify and confirm the validation lags. The number of clusters dissolved must not be equal to the number of clusters reformed, which is the optimal grouping condition, to prevent additional complexity demands. If the occurrence of validation lags is identified in the first sequence, then previous centroid deviation knowledge is updated using $\mathbb{C}_{\mathcal{K}}$. This means the time lag is classified according to the occurrence of non-clustered integrations, and

$\mathbb{C}_{\mathcal{K}} + \left[\left(1 - \frac{\mathbb{C}_{dv}}{\mathbb{P}_{dv}}\right)\Delta\right]_{n-1} - \omega(\exists(a'))\odot(\exists(a''))_{n-2}$ is the previous knowledge updating condition. In the grouping process, from the initial organized integration cases, $(\mathbb{U}_N, \mathbb{C}_{\mathcal{K}})$ is segregated with the appropriate centroid deviation knowledge. The grouping and updating output matching are checked and independently grouped in this clustering process. Previous deviation knowledge occurs in the first level from which that level is grouped for integration improvement. After the grouping process, the harmonized service levels of a single entity are compared with multiple entities based on complexity grouping. Here, each cluster's serving inputs, sharing levels, and midpoint are used to update the previous knowledge.

In the above cluster balancing, the condition of $A > \alpha$ is said to be "1", and the condition of $A < \alpha$ is said to be "0" based on the centroid deviation. If 0 is observed at any level, then the complexity lags are identified; hence, non-optimized classification is performed for further integration-level improvement. Now, the volatile integrations must be verified by machine learning, and the previous centroid deviation knowledge must be updated. In the centroid deviation knowledge data retrieval, the cluster and update are fetched for complexity analysis to identify if any validation lags are taking place. Here, the k-means clustering, $\mathbb{C}_{\mathcal{K}}$, and update is different from the grouping process, represented using (13) and (14):

$$\left.\begin{array}{c} \mathbb{C}_1 = SVL_1 \\ \mathbb{C}_2 = 2SVL_2 - \Delta_1 + \mathbb{G}_1 \\ \mathbb{C}_3 = 3SVL_3 - \Delta_2 + \mathbb{G}_2 \\ \vdots \\ \mathbb{C}_{\mathcal{K}} = nSVL_N - \Delta_N + \mathbb{G}_N \end{array}\right\} \tag{13}$$

$$\left.\begin{array}{c} \mathbb{U}_1 = \frac{1(t_{d1})+t_{s-1}}{t_{d_1}} \\ \mathbb{U}_2 = \left(\frac{(Compl_{max}-Compl_{min})}{T_{Lag}}\right)_1 - \mathbb{C}_1 \\ \mathbb{U}_3 = \left(\frac{(Compl_{max}-Compl_{min})}{T_{Lag}}\right)_3 - \mathbb{C}_2 \\ \vdots \\ \mathbb{U}_N = \left(\frac{(Compl_{max}-Compl_{min})}{T_{Lag}}\right)_{N-1} - \mathbb{C}_{\mathcal{K}} \end{array}\right\} \tag{14}$$

The centroid deviation knowledge update is performed at the end of all service levels or before the start of a new service. The update is pursued along with consecutive centroid changes that are updated using machine learning for integration improvement. It is to be noted that the volatile integrations are utilized and grouped under centroid deviation. Using machine leaning and k-means, clustering complexity and time lags are reduced, and integration is improved.

The analysis of device availability under $compl_{max}$ and $compl_{min}$ is presented in Figure 6. This analysis shows the elasticity of the proposed method for different $\mathbb{U}_N$ instances.

The $\mathcal{K}$ rate improvement increases the $S(INTG)$ for the observed new integration. The $\emptyset$ and Δ of the centroid converges the combination of complexity, lag, and levels to identify α. If α demands are high, then $compl_{max}$ and $compl_{min}$ are segregated at ease using availability. If the availability is high, $\mathbb{U}_N$, for grouping, it is estimated that the time lag is confined. Thus, the feasibility of integration is increased under fewer complexities (Figure 6).

Conventional IoT integration and optimization research has helped us understand this field's issues and solutions. Preliminary data handling and analysis research has revealed the importance of actual time data processing and management in IoT. Researchers have examined dispersed data processing, fog computing, and edge computing to manage the fast influx of IoT data. K-means and its variations have excelled at clustering and analyzing IoT data. These studies illustrate that clustering simplifies data and improves insights. Research has shown that IoT systems need efficient resource management to sustain and

efficiently operate IoT, optimize assets, network bandwidth, and energy use. Real-time system research has focused on latencies and data input responsiveness. The unique timing needs of IoT applications have led to priority-based computing and real-time management.

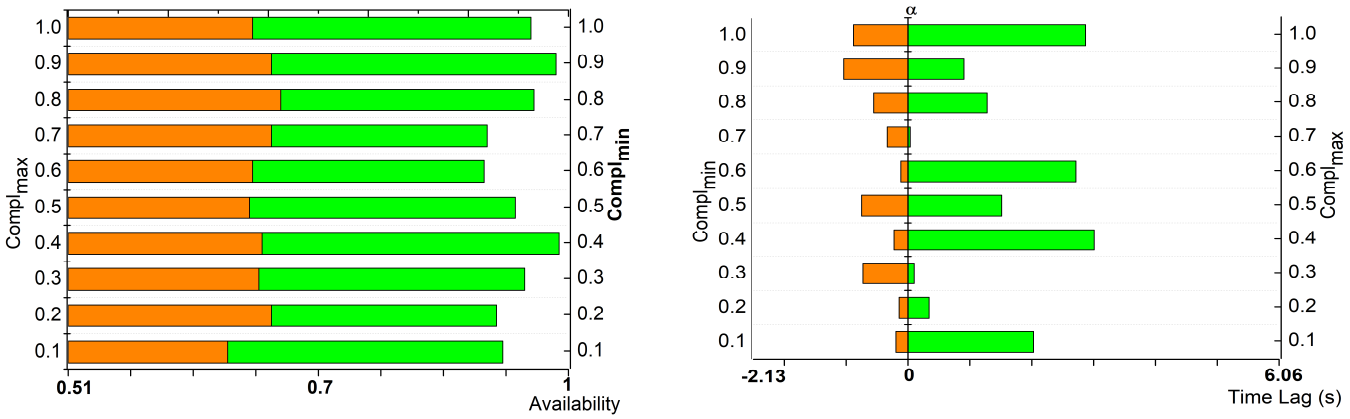


Figure 6. Availability and time lag analysis.

Several practical considerations must be addressed to feasibly and effectively apply OOIVM in real-world IoT applications. To improve the quality of its resource projections, simulations, and depictions based on interactive devices/interfaces, real-world applications are characterized by a need for complexity evaluation and grouping. The proposed O2IVM for precise incorporation evaluation ensures non-optimized categorization and prioritization by utilizing the centroid deviation knowledge of the ordered integration levels. Important factors include harmonized service levels that are needed to decrease complexity demands and limit centroid deviation-based complexity lags for a single entity. OOIVM can streamline and scale real-time IoT applications with proper planning and execution, making it a viable solution for modern IoT.

This proposed model simultaneously categorizes structured and non-organized integrations, utilizing time lag and levels to enhance the response rate. The unmanaged clusters with the highest complexity level are dissolved to decrease the computational complexity. This model successfully decreases complexity by 9.04% and enhances device availability by 12.46% to meet maximum device limits. While this model effectively solves the complexity issue, it is limited by the energy-based difficulties of the devices. Hence, energy harvesting optimization and multi-constraint clustering need to be developed to reduce integration complexity.

Segmentation in IoT typically does not use neural networks or other sophisticated ML models due to the high computing cost, extended training durations, and massive volumes of labeled information that these models require. Regarding the massive, ever-changing characteristics of IoT settings, where flexibility and swift adaptation are of utmost importance, k-means clustering is the way to go because it is more effective, quicker, and simpler to deploy for real-time applications.

4. Results and Discussion

Discussions on execution time, complexity, time lag, response time, and availability are described in this section. The above parameters are obtained from OPNET simulations containing 2600 devices serving 11 applications. The applications are heterogeneous and served from 24 storage units assembled in IoT. A device's maximum request processing capacity is 10–15/h. Integration of multiple devices with IoT paradigms occurs up to a maximum ratio of 60. The simulation is validated for service responses from 300 s to 1800 s devices. Therefore, the devices and the integration demands are the varying parameters set in the discussion for the evaluation metrics. The proposed model is accompanied by EEDTO [27], RERM [29], and ChameLIoT [35] methods from the Related Works section.

IoT relies on k-means clustering because it is scalable, efficient, and simple—three qualities crucial for dealing with the massive, ever-changing datasets in IoT settings. It is perfect for use in real-time situations, primarily due to its rapid data adaptation and simplicity-enhancing centroid-based optimization. The computationally costly or less scalable characteristics of other models, such as hierarchical clustering, DBSCAN, and Gaussian mixture models, make them unsuitable for IoT systems' rapid and expanding characteristics.

4.1. Execution Time

Figure 7 displays the execution time of the proposed model for application integration and service level from the IoT paradigm using k-means clustering to meet integration demand. For accurate issue identification, this approach avoids service level-based adjustments. The non-optimized categories for minimizing complexity address resource management and execution times in protracted device integrations. IoT application integration uses device/interface complexity to improve non-optimized categorization. Thus, the initial O2IVM validation output defines the difficulty of analyzing and grouping for evaluating resource projection quality, modeling, and representation. The conditions $\omega = (\omega_1, \ldots, \omega_n, \ldots, \omega_N)$ and $\exists = (\exists_1, \ldots, \exists_n, \ldots, \exists_N)$ are investigated to reduce execution time at various service levels for resource management and execution. To discover IoT application integration challenges, single entities are grouped. Therefore, the ideal output for 1400 devices involves IoT application device integrations that segregate organized integration scenarios in 1.74 s.

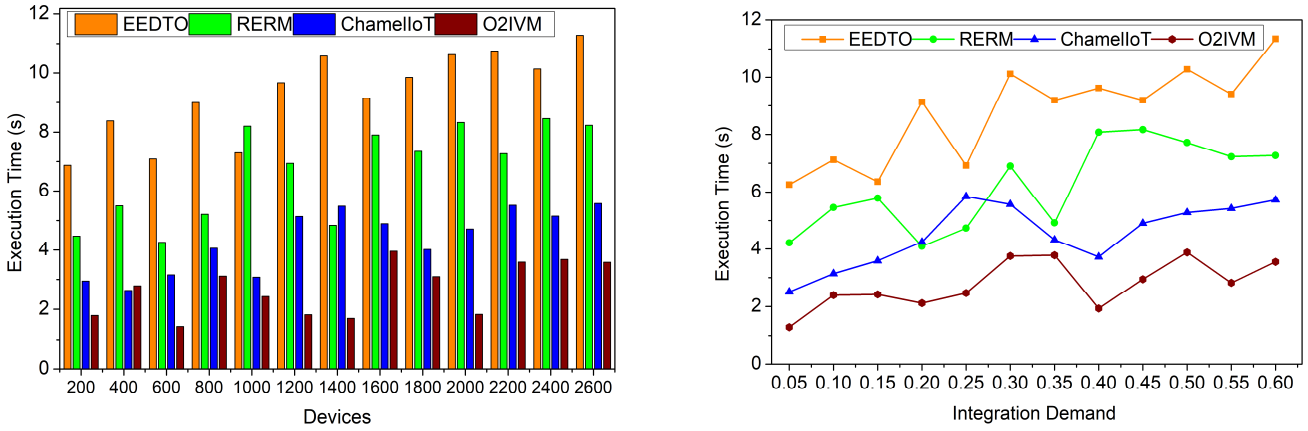


Figure 7. Execution time.

4.2. Integration Complexity

The coordinated optimization integration validation model addresses complexity. K-means clustering reduces complexity, as shown in Figure 8. The resource forecast quality was increased with suitable service level grouping based on execution time and complexity. For exact integration evaluation, we identified service levels, delays, and complications between protracted centroids device integrations. A simpler model took 1.25 s to execute than the other. Clustering to find issues balances integration levels and improves complexity minimization and prioritizing. Minimization helps evaluate complexity gaps between service tiers to prevent time lags. The clustering technique matches centroid changes with service level adjustments for each device integrated to find midpoints and avoid complexity. Better integration improves resource management by minimizing complexity and prioritizing. Therefore, the probability is simpler than the other factors.

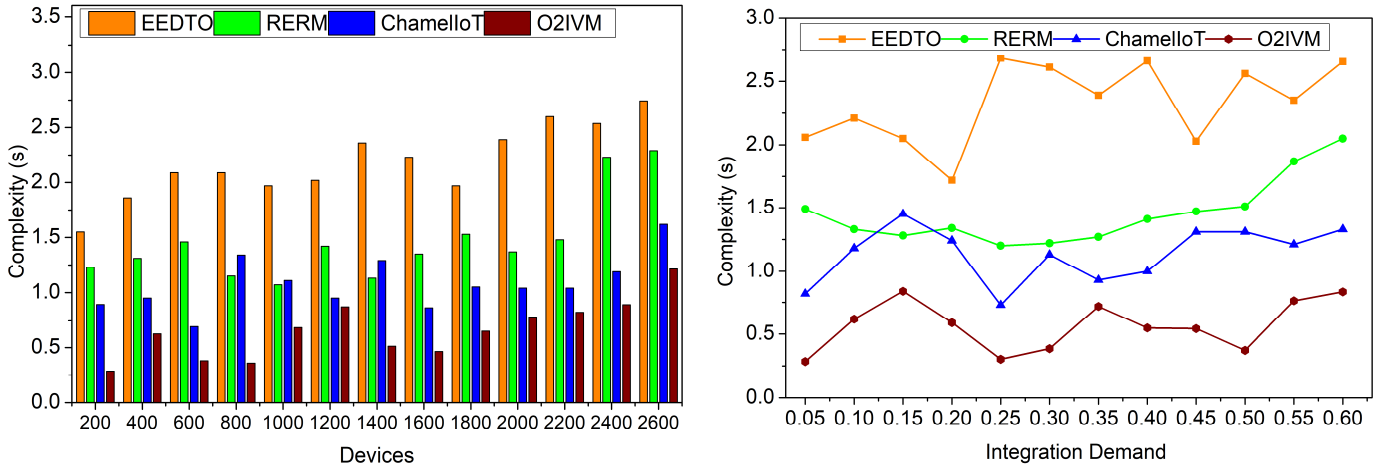


Figure 8. Integration complexity.

4.3. Time Lag

Figure 9 demonstrates the clustering method output service levels, complexity, and latency to fix all device integration issues at different times. Centroid deviation knowledge balances time-lagging integrations, integration levels, and non-complex processing in IoT applications. Matching current and previous centroid deviation knowledge with clustering output prevents time lag and improves resource projection quality. Grouping harmonized service levels with correct centroid deviation knowledge reduces complexity associated with non-optimized classification. Cluster dissolving improves complexity minimization and prioritizing, and reformation reduces time lag. The proposed model addresses midpoints using RL machine learning and organized optimization model output. The centroid reduces complexity and time lags with k-means clustering. The real-time application-based model is considered for combining single entities to reduce prioritization time.

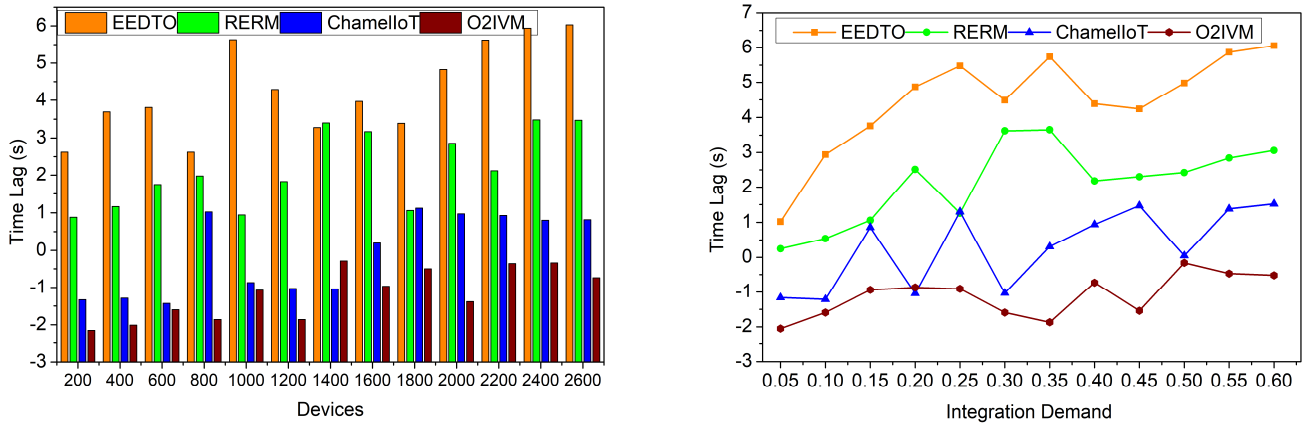


Figure 9. Time lag.

4.4. Response Time

For device integrations with high availability characteristics, the proposed O2IVM is used to achieve less response time. As shown in Figure 10, this is accomplished in this model by simplifying the response time and, as a result, decreasing the time lag. Based on the different service levels, k-means clustering is used to find the complexity using the centroid. The first step in aligning service levels is to group entities into initial single entities, where $n = (1, N)^{-}$. This grouping finds and reduces the complexity to improve resource management using centroid deviation knowledge, which is necessary to meet the criteria mentioned earlier. Therefore, complexity lags are suppressed by the integration evaluation. A reduction in complexity based on centroids is achieved by obtaining the integration

requirement. The incorporation of organized optimization for correctly grouping the service levels according to Equation (2) achieves a high level of complexity minimization. Less response time can be achieved through integration evaluation using the levels, lags, and complexities that result from minimization of complexity.

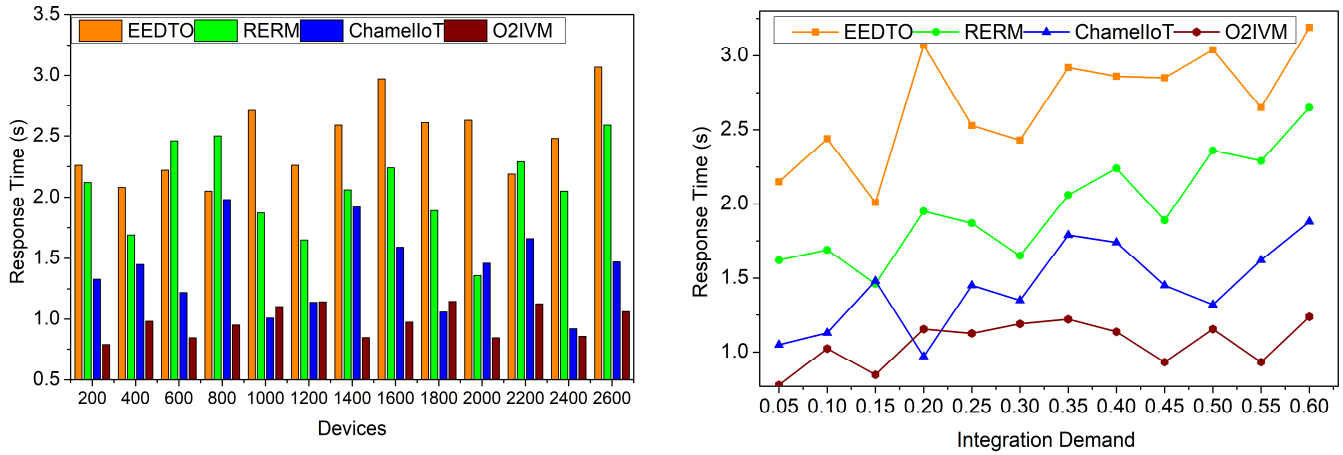


Figure 10. Response time.

4.5. Availability Parameter

Based on minimum and maximum complexity of real-time application at distinct instances, centroid deviation knowledge is analyzed. As seen in Figure 11, this assessment has great availability. With minimal time lags and reaction time, this model meets the high availability parameter using machine learning and k-means clustering. Centroids are recognized from the clustering output to overcome non-optimized classifications and validation lags across N service levels. Machine learning is used to identify execution time and resource management complexities. To determine the maximum centroid deviation, $devt_{max}$, and standard deviation, $devt_{\sigma}$, for balanced integration levels, the organized integration cases are separated based on centroid deviation. The current model reduces complexity and time latency with machine learning and clustering to improve integration evaluation. This method simplifies random instances with high availability characteristics using machine learning.

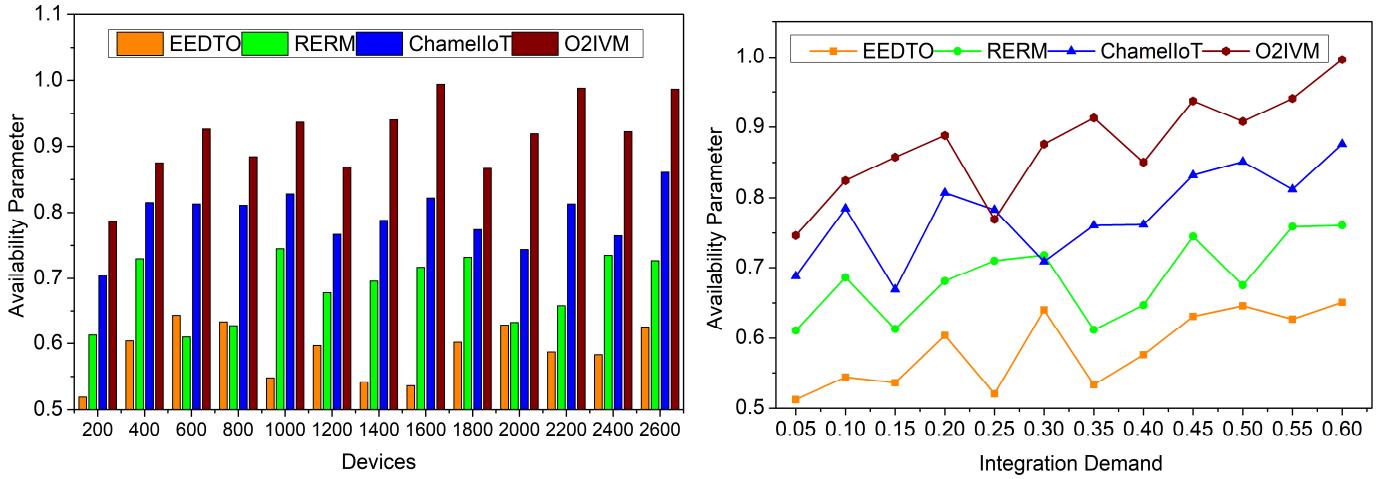


Figure 11. Availability parameters.

Many hurdles prevent widespread deployment of IoT applications and services. Due to IoT technology standard gaps, architecture is the biggest challenge. IoT architecture

design prioritizes safety, capacity, confidentiality, variability, data management, and networking. To quickly make IoT system decisions and maneuvers, the OOIVM addresses criteria such as streamlining data input reception and response time. Better use of computing and network infrastructure reduces waste and improves operations, lowering costs. Scalability lets the network handle more data and IoT devices without slowing down. In terms of computing complexity, the system simplifies IoT data integration and processing and makes systems easier to manage. Verification precision improves information validation to ensure IoT data-supported choices and actions are accurate. Making algorithmic optimization more adaptable to new data patterns and system needs improves efficiency. Reliability, scalability, and performance are OOIVM's goals, as it builds a solid platform for immediate-time IoT application administration.

Substantial energy consumption from the ongoing operation of IoT devices, computational units, and data transportation adds to the functioning, preservation, cloud service, and hardware expenses associated with OOIVM model implementation. Optimization of resource allocation, use of energy-efficient hardware, utilization of edge computing to decrease reliance on the cloud, and adoption of adaptive energy control methods are strategies to mitigate these impacts. Possible obstacles to adoption include safeguarding personal information, establishing interoperability, keeping data consistent and high-quality, controlling latency and bandwidth, and making sure the system can scale. Robust security mechanisms, consistent protocols, effective data management, and scalable system architecture are necessary to overcome these obstacles.

5. Conclusions

The integration of IoT to meet maximum real-time application demands has become of primary interest. This article proposes and discusses an organized optimization integration model for achieving this feature. This model groups the complexity-causing factors, such as response time lag, and the highest integration levels for identifying unisons. A modified k-means clustering method is used for this purpose, performing either disintegration or unison-based grouping with the update. The complexity factors converge towards a single entity through grouping knowledge, thereby reforming the clustering process. The proposed model thus classifies organized and non-organized integrations jointly or independently using time lag and levels to improve the response rate. The unattended clusters with the maximum number of instances are dissolved to reduce the computation complexity. Therefore, this model reduces the complexity by 9.04% and improves device availability by 12.46% for maximum device limits. This model successfully addresses the complexity problem, but its scope is limited by the energy-based complexities of the devices. Therefore, energy harvesting optimization and multi-constraint clustering will need to be improved to sustain less integration complexity.

Author Contributions: Conceptualization, A.A. and M.K.A.; methodology, T.S.A.; validation, K.A.A.; formal analysis, A.A.; investigation, T.S.A.; resources; writing—original draft preparation, K.A.A.; writing—review and editing, K.A.A.; visualization, M.K.A.; funding acquisition, A.A. and M.K.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data will be made available by the authors on request.

Acknowledgments: The authors extend their appreciation to the Deanship of Research and Graduate Studies, at the University of Tabuk for financial and technical support.

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

References

1. Algarni, F.; Khan, M.A.; Alawad, W.; Halima, N.B. P3S: Pertinent Privacy-Preserving Scheme for Remotely Sensed Environmental Data in Smart Cities. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2023**, *16*, 5905–5918. [CrossRef]
2. Liu, L.; Essam, D.; Lynar, T. Complexity measures for IoT network traffic. *IEEE Internet Things J.* **2022**, *9*, 25715–25735. [CrossRef]

3. Alqarni, M.A.; Alharthi, A.; Alqarni, A.; Khan, M.A. A transfer-learning-based energy-conservation model for adaptive guided routes in autonomous vehicles. *Alex. Eng. J.* **2023**, *76*, 491–503. [CrossRef]
4. Shibu, N.S.; Devidas, A.R.; Balamurugan, S.; Ponnekanti, S.; Ramesh, M.V. Optimising Microgrid Resilience: Integrating IoT, Blockchain, and Smart Contracts for Power Outage Management. *IEEE Access* **2024**, *12*, 18782–18803. [CrossRef]
5. Mukherjee, A.; Goswami, P.; Khan, M.A.; Manman, L.; Yang, L.; Pillai, P. Energy-Efficient Resource Allocation Strategy in Massive IoT for Industrial 6G Applications. *IEEE Internet Things J.* **2021**, *8*, 5194–5201. [CrossRef]
6. Liu, H.; Guan, X.; Bai, R.; Qin, T.; Chen, Y.; Liu, T. Designing a medical information diagnosis platform with IoT integration. *Heliyon* **2024**, *10*, e25390. [CrossRef] [PubMed]
7. Hu, X.; Assaad, R.H. BIM-enabled digital twin framework for real-time indoor environment monitoring and visualization based on autonomous LIDAR-based robotic mobile mapping, IoT sensing, and indoor positioning technology. *J. Build. Eng.* **2024**, *86*, 108901. [CrossRef]
8. Varriale, V.; Cammarano, A.; Michelino, F.; Caputo, M. Integrating blockchain, RFID and IoT within a cheese supply chain: A cost analysis. *J. Ind. Inf. Integr.* **2023**, *34*, 100486. [CrossRef]
9. Najim, A.H.; Kurnaz, S. Study of integration of wireless sensor network and Internet of Things (IoT). *Wirel. Pers. Commun.* **2023**, 1–14. [CrossRef]
10. Zhou, Z.; Shojafar, M.; Alazab, M.; Abawajy, J.; Li, F. AFED-EF: An energy-efficient VM allocation algorithm for IoT applications in a cloud data center. *IEEE Trans. Green Commun. Netw.* **2021**, *5*, 658–669. [CrossRef]
11. Abuhasel, K.A.; Khan, M.A. A Secure Industrial Internet of Things (IIoT) Framework for Resource Management in Smart Manufacturing. *IEEE Access* **2020**, *8*, 117354–117364. [CrossRef]
12. Shakeel, P.M.; Baskar, S.; Dhulipala, V.S.; Jaber, M.M. Cloud based framework for diagnosis of diabetes mellitus using K-means clustering. *Health Inf. Sci. Syst.* **2018**, *6*, 16. [CrossRef] [PubMed]
13. Presciuttini, A.; Cantini, A.; Costa, F.; Portioli-Staudacher, A. Machine learning applications on IoT data in manufacturing operations and their interpretability implications: A systematic literature review. *J. Manuf. Syst.* **2024**, *74*, 477–486. [CrossRef]
14. Alghamdi, N.S.; Khan, M.A. Energy-Efficient and Blockchain-Enabled Model for Internet of Things (IoT) in Smart Cities. *Computers, Mater. Contin.* **2021**, *66*, 2509–2524. [CrossRef]
15. Rizvi, S.; Zwerling, T.; Thompson, B.; Faiola, S.; Campbell, S.; Fisanick, S.; Hutnick, C. A modular framework for auditing IoT devices and networks. *Comput. Secur.* **2023**, *132*, 103327. [CrossRef]
16. Jazaeri, S.S.; Asghari, P.; Jabbehdari, S.; Javadi, H.H.S. Composition of caching and classification in edge computing based on quality optimization for SDN-based IoT healthcare solutions. *J. Supercomput.* **2023**, *79*, 17619–17669. [CrossRef] [PubMed]
17. Abohamama, A.S.; El-Ghamry, A.; Hamouda, E. Real-time task scheduling algorithm for IoT-based applications in the cloud–fog environment. *J. Netw. Syst. Manag.* **2022**, *30*, 54. [CrossRef]
18. Sheron, P.F.; Sridhar, K.P.; Baskar, S.; Shakeel, P.M. A decentralized scalable security framework for end-to-end authentication of future IoT communication. *Trans. Emerg. Telecommun. Technol.* **2020**, *31*, e3815. [CrossRef]
19. Wang, H.; Li, W.; Sun, J.; Zhao, L.; Wang, X.; Lv, H.; Feng, G. Low-complexity and efficient dependent subtask offloading strategy in IoT integrated with multi-access edge computing. *IEEE Trans. Netw. Serv. Manag.* **2023**, *21*, 621–636. [CrossRef]
20. Si-Mohammed, S.; Bardou, A.; Begin, T.; Lassous, I.G.; Vicat-Blanc, P. NS+ NDT: Smart Integration of Network Simulation in Network Digital Twin, Application to IoT Networks. *Future Gener. Comput. Syst.* **2024**, *157*, 124–144. [CrossRef]
21. Brandín, R.; Abrishami, S. IoT-BIM and blockchain integration for enhanced data traceability in offsite manufacturing. *Autom. Constr.* **2024**, *159*, 105266. [CrossRef]
22. Jin, S.; Karki, B. Integrating IoT and Blockchain for Intelligent Inventory Management in Supply Chains: A Multi-Objective Optimization Approach for the Insurance Industry. *J. Eng. Res.* **2024**. [CrossRef]
23. Alulema, D.; Criado, J.; Iribarne, L.; Fernández-García, A.J.; Ayala, R. SI4IoT: A methodology based on models and services for the integration of IoT systems. *Future Gener. Comput. Syst.* **2023**, *143*, 132–151. [CrossRef]
24. Irwanto, F.; Hasan, U.; Lays, E.S.; De La Croix, N.J.; Mukanyiligira, D.; Sibomana, L.; Ahmad, T. IoT and Fuzzy Logic Integration for Improved Substrate Environment Management in Mushroom Cultivation. *Smart Agric. Technol.* **2024**, *7*, 100427. [CrossRef]
25. Chakour, I.; Mhammedi, S.; Daoui, C.; Baslam, M. Unlocking QoS Potential: Integrating IoT services and Monte Carlo Control for heterogeneous IoT device management in gateways. *Comput. Netw.* **2024**, *238*, 110134. [CrossRef]
26. Narang, D.; Madaan, J.; Chan, F.T.; Chungcharoen, E. Managing open loop water resource value chain through IoT focused decision and information integration (DII) modelling using fuzzy MCDM approach. *J. Environ. Manag.* **2024**, *350*, 119609. [CrossRef]
27. Wu, H.; Wolter, K.; Jiao, P.; Deng, Y.; Zhao, Y.; Xu, M. EEDTO: An energy-efficient dynamic task offloading algorithm for blockchain-enabled IoT-edge-cloud orchestrated computing. *IEEE Internet Things J.* **2020**, *8*, 2163–2176. [CrossRef]
28. Yang, W.; Qin, Y.; Yi, Z.; Wang, X.; Liu, Y. Providing cache consistency guarantee for ICN-based IoT based on push mechanism. *IEEE Commun. Lett.* **2021**, *25*, 3858–3862. [CrossRef]
29. Eroshkin, I.; Vojtech, L.; Neruda, M. Resource Efficient Real-Time Reliability Model for Multi-Agent IoT Systems. *IEEE Access* **2021**, *10*, 2578–2590. [CrossRef]
30. Chahed, H.; Usman, M.; Chatterjee, A.; Bayram, F.; Chaudhary, R.; Brunstrom, A.; Kassler, A. AIDA—A holistic AI-driven networking and processing framework for industrial IoT applications. *Internet Things* **2023**, *22*, 100805. [CrossRef]

31. Zhang, K.; Zhou, Y.; Wang, C.; Hong, H.; Chen, J.; Gao, Q.; Ghobaei-Arani, M. Towards an automatic deployment model of IoT services in Fog computing using an adaptive differential evolution algorithm. *Internet Things* **2023**, *24*, 100918. [CrossRef]
32. Wang, H.; Li, L.; Cui, Y.; Wang, N.; Shen, F.; Wei, T. MBSNN: A multi-branch scalable neural network for resource-constrained IoT devices. *J. Syst. Archit.* **2023**, *142*, 102931. [CrossRef]
33. Guan, Z.; Wang, Z.; Cai, Y.; Wang, X. Deep reinforcement learning based efficient access scheduling algorithm with an adaptive number of devices for federated learning IoT systems. *Internet Things* **2023**, *24*, 100980. [CrossRef]
34. Kim, E.; Son, T.; Ha, S. A novel hierarchical edge-based architecture for service oriented IoT. *Internet Things* **2023**, *24*, 100939. [CrossRef]
35. Silva, M.; Gomes, T.; Ekpanyapong, M.; Tavares, A.; Pinto, S. ChamelIoT: A tightly-and loosely-coupled hardware-assisted OS framework for low-end IoT devices. *Real-Time Syst.* **2023**, *60*, 1–47. [CrossRef]

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