electronics

MDPI

Article

Two-Level Dynamic Programming-Enabled Non-Metric Data Aggregation Technique for the Internet of Things

Syed Roohullah Jan [1], Baraq Ghaleb [2], Umair Ullah Tariq [3,*], Haider Ali [4], Fariza Sabrina [3] and Lu Liu [5]

[1] School of Technology, Business and Arts, University of Suffolk, Ipswich IP4 1QJ, UK; syed.jan@uos.ac.uk
[2] School of Computing, Engineering, and the Built Environment, Edinburgh Napier University, Edinburgh EH10 5DT, UK; b.ghaleb@napier.ac.uk
[3] School of Engineering and Technology, Central Queensland University, Rockhampton, QLD 4701, Australia; f.sabrina@cqu.edu.au
[4] School of Computing, University of Derby, Derby DE22 3AW, UK; h.ali@derby.ac.uk
[5] School of Informatics, University of Leicester, Leicester LE1 7RH, UK; l.liu@leicester.ac.uk
* Correspondence: u.tariq@cqu.edu.au

Abstract: The Internet of Things (IoT) has become a transformative technological infrastructure, serving as a benchmark for automating and standardizing various activities across different domains to reduce human effort, especially in hazardous environments. In these networks, devices with embedded sensors capture valuable information about activities and report it to the nearest server. Although IoT networks are exceptionally useful in solving real-life problems, managing duplicate data values, often captured by neighboring devices, remains a challenging issue. Despite various methodologies reported in the literature to minimize the occurrence of duplicate data, it continues to be an open research problem. This paper presents a sophisticated data aggregation approach designed to minimize the ratio of duplicate data values in the refined set with the least possible information loss in IoT networks. First, at the device level, a local data aggregation process filters out outliers and duplicates data before transmission. Second, at the server level, a dynamic programming-based non-metric method identifies the longest common subsequence (LCS) among data from neighboring devices, which is then shared with the edge module. Simulation results confirm the approach's exceptional performance in optimizing the bandwidth, energy consumption, and response time while maintaining high accuracy and precision, thus significantly reducing overall network congestion.

Keywords: Internet of Things; data aggregation; longest common subsequence; QoS; accuracy

Citation: Jan, S.R.; Ghaleb, B.; Tariq, U.U.; Ali, H.; Sabrina, F.; Liu, L. Two-Level Dynamic Programming-Enabled Non-Metric Data Aggregation Technique for the Internet of Things. *Electronics* **2024**, *13*, 1651. https://doi.org/10.3390/electronics13091651

Academic Editors: Wanfu Gao, Yuzhou Liu and Ping Zhang

Received: 27 March 2024
Revised: 19 April 2024
Accepted: 23 April 2024
Published: 25 April 2024

1. Introduction

Due to recent technological advancements in micro-electromechanical systems (MEMS), the Internet of Things has been extensively utilized in various application domains, i.e., home automation, manufacturing, smart cities, etc. These networks consist of smart devices, preferably equipped with embedded sensors and actuators, enabling them to interact with physical phenomena and report them to the respective server module after a defined time interval [1,2]. However, data values captured by these resource-limited devices may likely contain noise or duplicate data, which need to be refined prior to the transmission process in the IoT network. It is important to note that the refinement process should be robust enough, i.e., not only should it have the capacity to minimize the ratio of the outliers or noisy data values, but it should also be smart enough to carry out this task with the minimum possible information loss ratio, ideally negligible [3,4]. In addition to the outliers, the refinement process should be useful for reducing the ratio of duplicate data values, especially those captured by devices deployed nearby. Duplicate data inflate data storage requirements, strain network bandwidth, and increase power consumption, particularly in resource-constrained Internet-of-Things (IoT) environments. Moreover, they can distort data analysis and decision-making processes, potentially leading to incorrect conclusions

or actions. Efforts to address the duplicate data issue in IoT have become increasingly critical, prompting researchers and engineers to develop innovative methods to minimize its impact [5–8]. The primary causes of duplicate data values in IoT networks can be attributed to network congestion, device malfunction, and software errors. It is important to mitigate this issue because duplicate data can lead to inaccurate analysis, wasted resources, and compromised decision-making in IoT systems. Efficient data handling ensures reliable insights and optimal performance [9]. Specifically, the major causes of duplicate data values include the following:

1. Dense deployment of member devices, which can overlap in their data collection.
2. Proximity of neighboring devices, especially those residing in close vicinity, which may inadvertently capture and transmit similar data.

One promising avenue to mitigate this challenge is through data fusion and aggregation techniques. These techniques aim to not only reduce the redundancy of data but also enhance the overall quality of information by combining and processing data from multiple sources intelligently.

To address the issue of duplicate data in IoT effectively, researchers and engineers have been developing advanced data fusion and aggregation techniques. These methodologies offer a promising path to reduce data redundancy while preserving essential information. Data fusion involves the integration of information from multiple sources to create a more comprehensive and accurate representation of the underlying phenomena. Data aggregation, on the other hand, focuses on combining data points in a manner that reduces the volume of transmitted data without compromising the overall quality [10]. In both cases, the minimum possible information loss ratio should be ensured as long as it is relaxed by the application requirements [11]. In multi-sensor environments, the pre-aggregation function has been extended through an effective methodology to enhance the precision and accuracy ratio of the concerned data aggregation model. In this approach, a smart approach has been reported to reduce the influence of values, preferably high density, on the expected model's outcome. For this purpose, functions are divided into monotone or directionally monotone whereas data refinement is carried out through a more powerful scheme [12]. Similarly, a robust and hybrid fusion model, which is based on multi-modal and deep learning schemes, was presented for the diagnosis of faults where a feature-level extraction model has been utilized to extract accurate features from two signals in the time domain [13].

A lightweight data fusion and vulnerability-aware communication approach has been presented for the efficient utilization of the available resources where decision support systems refine data values before making a decision [14]. However, duplicate data values are kept, i.e., both from the same and neighboring nodes, to preserve the integrity and accuracy of the underlying decision support system. An efficient and effective data aggregation methodology was reported by Fitzgerald et al. [15] to resolve the duplicate data value problem, where two different approaches were developed. For scenarios involving multiple sensors, the 1K aggregation model was developed. However, if both sensors and actuators are part of the operational IoT networks, then the nK model is applicable. Building on this, a multi-modal aggregation methodology was developed and reported by Islam et al. [16] to refine data values captured through wearable devices in the model of human activity recognition. This model was realized through a multi-head architectural system, which was specifically designed to process visual data in the IoMT. In the healthcare domain, the precision and accuracy of both captured and transmitted data are more important as false data may lead to severe losses. To address this issue, a lightweight fusion methodology was presented to detect and correct, if applicable, false readings captured by the concerned wearable devices [17]. A well-known propagation model, i.e., extended belief, was utilized for perceiving the expected quality of the transmission media. Moreover, the precision and accuracy ratio of refined data is improved by using Archimedes' optimization. To classify an accident, a specialized fusion approach, i.e., a multi-mode approach, was adopted along with other mathematical measures.

A distributed and Automata learning-enabled aggregation model was reported in the literature to enhance the throughput of the entire network by intelligently reducing traffic, i.e., congestion control, through an effective filtering process for the duplicate data values in the IoT networks [14]. Likewise, two distinct aggregation methods were specifically designed to refine vibration signals captured through wearable body sensors. The second model applied Bayesian fusion, incorporating three interval estimators into every active channel [18]. While these data aggregation models demonstrate effectiveness within their respective domains, they also face significant challenges. Moreover, a significant number of these models have predominantly utilized either in-node data aggregation or server-level aggregation strategies, making them less applicable across diverse domains. Consequently, there is a pressing need to propose a robust data fusion methodology capable of meeting the requirements of both homogeneous and heterogeneous IoT network infrastructures.

The sophisticated data aggregation approach can be deployed in IoT networks for scenarios such as smart city management, industrial automation, and healthcare monitoring. In smart city management, it could optimize traffic flow by aggregating data from various sensors, leading to reduced congestion and improved urban planning. In industrial automation, it can enhance production efficiency by aggregating data from sensors across the manufacturing process, facilitating predictive maintenance, and minimizing downtime. In healthcare monitoring, it can enable better patient care by aggregating data from wearable devices and medical sensors, allowing for early detection of health issues and personalized treatment plans. Overall, the expected benefits include improved resource utilization, enhanced decision-making, and cost savings in real-world implementations [19,20].

In this paper, a two-tier data aggregation methodology is proposed that aims at enhancing the quality of refining captured data values while minimizing information loss in IoT networks. In this model, every captured data value undergoes local refinement to ensure consistency with previously transmitted data. Additionally, a local server model conducts a comprehensive aggregation process to fine-tune collected data values before the decision-making phase, ultimately improving the accuracy and precision ratios. Thanks to these modifications, the proposed approach proves instrumental in mitigating network congestion by preventing the transmission of duplicate data values, both within local segments and across the broader IoT networks. The key scientific contributions of this paper are highlighted as follows:

1. Introduction of a novel data aggregation approach for IoT networks, utilizing a two-tier and dynamic programming-based non-metric method, to refine every captured data value.
2. Development of an in-node or local data aggregation model, which refines captured data values by addressing both noise and duplicate data issues before transmitting the data.
3. A server-enabled aggregation model, i.e., the longest common subsequence-based model, where data values received from multiple source devices are further refined, and where duplicate data values are discarded.

The rest of the paper is organized as follows: In Section 2, we provide a comprehensive review of the existing literature, focusing specifically on approaches that pertain to the problem addressed in this paper. Section 3 broadly describes how the proposed approach works both locally and globally to minimize the ratio of duplicate data values with minimum possible overhead. Section 4 presents a detailed performance evaluation of the proposed approach compared to the state-of-the-art approach. Finally, our concluding remarks and future research directives are described in Section 5.

2. Literature Review

The Internet of Things (IoT), thanks to its remarkable features, has found extensive application in automating and controlling various activities, especially activities demanding round-the-clock availability, within diverse application domains where devices with suitable embedded sensors are positioned in proximity to physical phenomena [21,22].

Even though these devices can effectively accomplish their designated tasks with limited resources, the central issue revolves around outliers or noisy data values, which have the potential to significantly degrade the performance of the underlying applications and need to be resolved on a priority basis with the available resources. These issues are broadly divided into two different categories, as follows:

1. Duplicate data values captured by those devices or modules that are deployed in neighborhoods.
2. Outliers or noisy data values generated by the respective embedded sensors due to malfunctioning.

To address these issues, data aggregation was presented in the literature with schemes reported to minimize (if elimination was not possible) occurrences of duplicate data values in the actual domain. Likewise, fusion methodologies were presented to detect and correct outliers or noisy data values (if applicable) captured by resource-constraint embedded sensors in IoT infrastructure. For instance, the authors in [12] proposed an extended pre-aggregation method to enhance the precision and accuracy of aggregation in multi-sensor environments. In a multi-sensor environment, the pre-aggregation function has been extended through an effective methodology to enhance the precision and accuracy ratio of the concerned data aggregation model. In this approach, a smart approach has been reported to reduce the influence of high-density values on the expected model's outcome. For this purpose, functions are divided into monotone or directionally monotone functions, whereas data refinement is carried out through a more powerful scheme proposed in [12]. Similarly, a robust and hybrid fusion model, integrating both multi-modal and deep learning approaches, was introduced for the purpose of fault diagnosis. In reference [13], this model utilizes a feature-level extraction technique to precisely extract features from two signals within the time domain. A lightweight data fusion and vulnerability-aware communication approach has been presented for the efficient utilization of available resources. In this approach, decision support systems refine data values before making a final decision, as detailed in [23]. Nonetheless, duplicate data values, whether from the same node or neighboring nodes, are retained to preserve the integrity and accuracy of the underlying decision support system. An efficient and effective data aggregation methodology was reported by E. Fitzgerald et al. in their work on data aggregation to resolve duplicate data values [15]. This approach encompasses the development of two distinct strategies. In cases involving multiple sensors, the 1K aggregation model was applied. Conversely, when both sensors and actuators are integral to IoT networks in operation, the nK model is the applicable choice. Building upon this, the authors in [16] introduced a multi-modal aggregation approach designed to refine data values acquired from wearable devices in the context of human activity recognition. This model has been developed and realized via a multi-head architectural system, which was specially designed to process visual data in the IoMT. In the healthcare domain, the precision and accuracy of both captured and transmitted data are more important as false data may lead to severe losses. To address this issue, a lightweight fusion methodology was presented to detect and correct, if applicable, false readings captured by wearable devices, as proposed in [17]. A well-known propagation model, which is based on extended belief, was utilized to perceive the expected quality of the transmission media. Moreover, the precision and accuracy ratio of the refined data are improved by using Archimedes' optimization. To classify an accident, a specialized multi-modal fusion approach was adopted along with other mathematical measures. A distributed and Automata learning-enabled aggregation model was reported in the literature to enhance the throughput of the entire network by intelligently reducing congestion, through an effective filtering process for the duplicate data values [14]. Likewise, two distinct aggregation methods, particularly designed to refine vibration signals captured through wearable body sensors. The second model has applied Bayesian fusion with embedded three interval estimators to every active channel [18]. In the healthcare domain, an entropy-enabled fusion methodology has been proposed to enhance or at least examine the accuracy level of the underlined captured data values along with its dependability ratio

in a real environment. This task is completed before any other processing on the respective data, such as determining the number of injuries, classifying illnesses based on numerous parameters, and ultimately predicting future demand for medical supplies [24]. Although these data aggregation models are effective in their particular domain, issues are linked with every model, such as the ratio of duplicate data even after the refinement process. Additionally, the majority of these models have either used in-node data aggregation or server-level aggregation models, which are not applicable in every domain. Thus, there is a need to develop a robust data fusion approach that is applicable to both homogeneous and heterogeneous IoT networking infrastructures. Reference [25] has highlighted numerous potentials of a well-known procedure, i.e., truncated bits, where extra bits are disregarded to minimize information loss, particularly in devices deployed nearby or neighborhood settings. An effective methodology proposed and described in [26], known as bidirectional least mean square methodology-enabled aggregation, ensures that weights are updated according to optimal values collected through the proposed procedure in the Internet of Things (IoT). Apart from data aggregation, an effective multiple-path routing methodology has also been presented to increase the lifetime of the underlying network without compromising other performance metrics. Likewise, an energy-aware, cluster-enabled data aggregation mechanism, along with an optimal routing mechanism, has been presented for the Internet of Things, where capuchin and fuzzy logic algorithms are used to form a hybrid aggregation model [26]. Similarly, a privacy-preserving and lightweight data aggregation approach was developed and presented by [27] to collect noise and duplicate free datasets in a realistic IoT environment. A lightweight structure-based routing-enabled aggregation methodology was presented for the next-generation wireless sensor networks [28]. Although these approaches are extremely useful in resolving either fusion or aggregation issues, a robust methodology is needed that addresses both of these concerns and can be implemented equally well on both individual devices and server modules.

In short, the challenges in the existing literature regarding duplicate data values in IoT include the need for robust data refinement processes to minimize duplicate data while preserving essential information, and developing effective data fusion and aggregation techniques to address redundancy and enhance data quality.

3. Proposed Aggregation Methodology

Data fusion and aggregation techniques are commonly applied in both traditional and resource-constrained networking infrastructures to refine collected data without compromising their integrity. These approaches are primarily aimed at enhancing the accuracy and precision ratios of the underlying decision support system, which uses these captured data values as input for certain activities. These approaches are designed to perform one or both of the following tasks, depending on the application requirements:

1. Detection and correction of false data values captured by the respective embedded sensor module deployed near the phenomenon.
2. Minimizing (or, if not possible, eliminating) the ratio of duplicate data values in the entire database, which is carried out either through in-node processing or server-level data aggregation and fusion.

The proposed approach can address both of these issues through a two-level aggregation methodology, involving both in-node and server-focused data aggregation. Additionally, these approaches are sophisticated enough to refine captured data values without compromising other important metrics, such as information loss and integrity. A detailed flow chart of the working mechanism is presented in Figure 1. Sections 3.1 and 3.2 describe in detail our proposed approach.

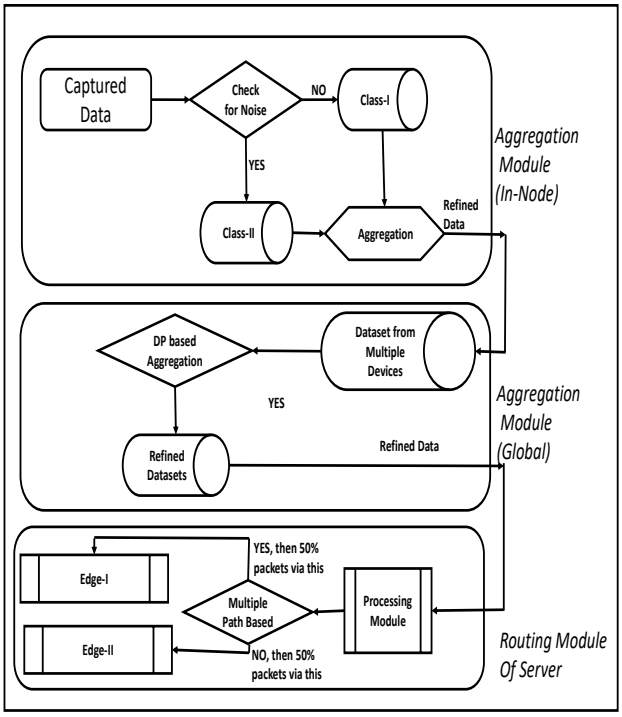

Figure 1. Working mechanism of the proposed two-tier data aggregation approach.

3.1. Proposed In-Node Data Aggregation: A Smart Approach

Generally, in IoTs, every device, C_i, is equipped with appropriate sensors to capture real-time data values by directly interacting with the underlying phenomenon. Subsequently, these devices share these data with the intended destination module, such as a server or edge module, preferably those residing nearby where important decisions are made. However, the accuracy and precision ratios of these decisions, i.e., at the server or edge module, have a direct correlation with the accuracy of the collected data. Therefore, the integrity and accuracy of the captured data values must be maintained throughout both processes, i.e., data capturing (detection and correction of outliers) and transmission via wireless media. Hence, in-node data processing, i.e., the refinement process carried out at the source devices, is proposed to refine captured data values, preferably in their raw form, and initiate the transmission process only if the respective data value is found to be correct. The proposed in-node data fusion or aggregation methodology is based on a realistic approach where every source device is required to ensure that the captured data value falls within the defined inner and outer bounds of the concerned electronic module. Let us assume that X and Y are used as the upper and lower bounds of a given device, then the accuracy of the captured data, Z, is ensured through Equation (1).

$$Z_i \in (X, Y) \tag{1}$$

If the captured data value falls within the defined bounds, it is assumed to be accurate, irrespective of the difference between it and previously transmitted data values, which are considered as benchmarks in some existing state-of-the-art approaches. However, if data values do not fall within the respective bounds, they need to be refined before transmission, as sharing faulty data is a waste of resources, as depicted in Figure 2. Therefore, the proposed approach forces the concerned sensor module to abort its waiting state, i.e., the defined time interval to capture and transmit data and become active to recapture the data value. As soon as a new data value is captured, it undergoes the same procedure as described above. The data value is transmitted if found correct; otherwise, the concerned sensor module is informed to recapture it. By following this procedure, the probability of information loss is almost negligible, while accuracy is approximately 100 percent. For

example, let us assume that $X = 0$ °C and $Y = 100$ °C are the upper and lower bounds for a particular device, respectively. If the most recently captured value, i.e., $Z = 50$ °C, then according to Equation (1), it is an accurate data value as it falls within the defined bounds. However, if the value of $Z = 105$ °C, which does not belong to the reference range, it is discarded, and the respective module needs to recapture it.

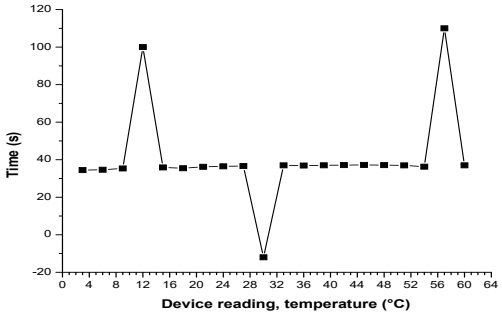

Figure 2. Sample graphical overview of outliers and accurate data values (sampling time interval = 4 s).

Apart from the detection and correction of outliers or noisy data values, the proposed in-node data fusion enables devices to reduce duplicate data values, i.e., controlling local data duplication. A common scenario arises when a device, C_i, captures a data value identical to one previously transmitted. For this purpose, each device must keep track of the five most recent or application-dependent previously transmitted data values in memory and compare every captured data value with these readings using an equation. See Equations (2) and (3); data are transferred only if they are correct and not recently transmitted.

$$\forall_{i=1\ldots n} Z_i \notin X_{1\ldots 5} \tag{2}$$

where variable Z represents the current captured value and $X_{1\ldots5}$ are the previously transmitted duplicate-free data values.

$$d(Z, X) = \sqrt{\sum_{i=1}^{n} (Z - X_i)^2} \tag{3}$$

In this approach, if a value is a duplicate, i.e., an exact copy of another value, then it is simply discarded. This not only helps to reduce congestion across the network but also leads to the efficient utilization of resources. For example, assume that the embedded sensor is a temperature sensor and its previously captured value is 37 °C, which has been communicated to the intended server module and a copy is stored at the local memory unit for onward processing as described above. Now, let us assume that the newly captured value of the concerned embedded sensor module is 37.5 °C. As it is a different value from the previously transmitted value, the transmission process is initiated because the current value is accurate and is neither an outlier nor a duplicate. Furthermore, let us assume that the newly captured value is 200 °C. In-node processing will match against the defined boundary values of the respective embedded sensor module where it is confirmed that this value is out of bounds as described in Equation (1). Therefore, this value is discarded without matching it against the previously transmitted refined values, and the transmission process is not activated as it is a noisy value. Finally, if the concerned sensor module captures a value of 37 °C, then initially it falls within the defined bounds, which means it is an accurate value. Next, the proposed setup has to check for a duplicate data value by comparing it against previously transmitted refined values. In this case, a match is encountered and, therefore, a timestamp of the concerned value is retained, and transmission activity is aborted as this value has already been communicated with

the intended server module. The proposed node-level data fusion algorithm is presented below in (Algorithm 1).

Algorithm 1: Proposed Device-Oriented Data Aggregation Algorithm

```
Input: Captured Data at a Particular Time Stamp
Output: Return Refined (Aggregated) Data
Array1 ← ∈ ;
Previous-Val ← 37.5°C;
Captured-Val ← Embedded Sensors' Reading;
for (Every Captured-Val) do
    if Captured-Valu ∈ (Lower-Bound, Upper-Bound) then
        if Captured-Val == Previousl-Val then
            Discard Captured-Val;
            Maintain Time Stamp of Captured-Val in Array1;
        else
            Previous-Val ← Captured-Val;
            Initiate Transmission Process;
        end
    else
        Discard Captured-Val;
        Wait for the Next Captured Data Value from Sensor;
    end
end
return Refined (Aggregated) Data
```

3.2. Dynamic Programming-Enabled Non-Metric-Based Data Aggregation: Server Level

Generally, in IoT, aggregation not only needs to be carried out locally (e.g., the in-node process, as described in Section 3.1), but an intelligent and effective procedure is required to be implemented on the respective server or edge module, where data values are captured by different devices, which are deployed in close proximity to the underlying phenomenon. As in-node processing has eliminated a considerable amount of outliers or noisy data values, which are captured by a respective embedded sensor module. Additionally, duplicate data values were removed through a sophisticated procedure, and each device was ensured to transmit refined data values to the concerned server or edge module. However, it is highly likely that data values captured by two devices, especially those deployed near each other, are duplicates, as this is beyond the operational capacity of the in-node processing proposed in the previous subsection of this paper. Additionally, refined data values may be corrupted during the transmission activity and need to be rechecked for the integrity and accuracy of the underlined system.

To address the aforementioned issues with the refined data values, a robust and precise global data aggregation mechanism is presented to enhance the accuracy and precision ratios of the underlined system. The proposed approach is based on the well-known concept of the dynamic programming-enabled similarity measure, which is pruned against numerous outliers or noise data values, especially those that occur during the transmission process. This approach is based on the concept of identifying common subsequences, more specifically the longest ones, among two or more datasets, preferably collected through different devices in the IoT infrastructure. In this approach, two or more datasets (preferably those collected through different devices), such that datasets X_n *and* Y_m are collected through devices C_i *and* C_{i+1}, respectively, and are represented as $a_1, a_2, \ldots, a_n \in X_n$, and $b_1, b_2, \ldots, b_m \in Y_m$, where $m \leq n$. Initially, symbol one, i.e., $a_1 \in X_n$, is matched with every data value, i.e., $b_1, b_2, \ldots, b_m \in Y_m$, of the other device's captured data values; let us assume that the similarity metric is represented by $v_1, v_2, \ldots, v_k \in Z_k$. Now, if $X_n : Y_m$, then $Z_k : X_n : Y_m$, and Z_{k-1} is the LCSS of X_{n-1} & Y_{m-1} OR if $X_n ! : Y_m$, then $Z_k ! : X_n$, and Z is the LCSS of $Xn - 1$ &Y OR if $X_n ! : Y_m$, then $Z_k ! > Y_m$, and Z is the LCSS of X &Y_{m-1}. Furthermore, the recursive solution for any matching problem is represented as given below in Equation (4).

$$Z_{i,j} = \begin{cases} i = 0 & if\ i = 0\ or\ j = 0 \\ Z_{i-1,j-1} + 1 & if\ i,j > 0\ \&\ X_i = Y_j \\ Max[Z_{i,j-1}, Z_{i-1,j}] & if\ i,j > 0\ \&\ X_i ! = Y_j \end{cases} \quad (4)$$

The above recursive algorithm computes the longest common subsequences (LCSSs), i.e., similarity indexes of two or more datasets collected through different devices, by solving a particular sub-problem exactly once. For example, when it finds $X_n : Y_m$, it uses the remaining sub-problems to compute the similarity measures of X_{n-1} and Y_{m-1}. However, if $X_n \neq Y_m$, it computes similarity indexes between X_{n-1} and Y_m or X_n and Y_{m-1}, respectively. In this process, every element of the first dataset is matched with all elements of the other dataset, captured through different devices. After this process, a tabular form is generated where different entries are filled according to Equation (4). In the next step, a simplified algorithm is used to find the longest similarity measure among two datasets along with the index information. As soon as the similarity measure is found, a single copy of it is retained, whereas duplicate data values are discarded. For example, if we have two datasets, i.e., $X : a, b, c, b, d, a, b$ and $Y : b, d, c, a, b, a$, the dynamic programming-enabled algorithm works by comparing elements of two datasets starting from the first element and continues until a match is encountered. It is important to note that an entry in the respective table (as presented in Figure 3) occurs according to the matching elements, which are either equal, less, or greater. In this table, a match, i.e., when two values are equal ($X_i = Y_j$), is represented by digits greater than zero. Similarly, an arrow represents a match, which points to either the upper or left neighbor if it is greater.

To find similarity indexes or LCSSs of two datasets, i.e., X_n and Y_m, in the aforementioned table, a simplified procedure is used. This procedure begins the search from the lower right corner of the given table, i.e., cell (7, 6), and backtracks in the reverse direction to compute the longest common subsequence between these two datasets. During this process, the cursor shifts to either the left neighbor cell or the upper neighbor cell, depending on the cell's values. For example, in the first step, it will move to the upper neighbor as its value is greater than the left neighbor's value, i.e., cell (6, 6). However, if the values of both neighbors are the same, it will move to the diagonally upward neighbor cell, as in the second phase, i.e., cell (5, 5). This process is repeatedly applied until a long common sub-sequence is found. This scheme can identify duplicate data values even in the presence of outliers or noisy data values. Once a common sub-sequence is found, a single copy of it is retained, while another copy is discarded. Additionally, this process is not only limited to two datasets; it can be applied to multiple datasets as well. Finally, the proposed approach requires every server module to compute the similarity of the neighboring devices only, which is realized through a simplified procedure, where every device shares its neighborhood information with the concerned server or edge module prior to actual communication. The algorithm for the dynamic programming-enabled data aggregation approach is presented below in Algorithm 2. It is important to note that the proposed scheme utilizes a state-of-the-art dynamic programming-enabled similarity-measuring algorithm to determine how similar two datasets are.

i		0	1	2	3	4	5	6
i		y_j	b	d	c	a	b	a
0	x_i	0	0	0	0	0	0	0
1	a	0	↑ 0	↑ 0	↑ 0	↖ 1	← 1	↖ 1
2	b	0	↖ 1	← 1	← 1	↑ 1	↖ 2	← 2
3	c	0	↑ 1	← 1	↖ 2	← 2	↑ 2	↑ 2
4	b	0	↖ 1	↑ 1	↑ 2	↑ 2	↖ 3	← 3
5	d	0	↑ 1	↖ 2	↑ 2	↑ 2	↑ 3	↑ 3
6	a	0	↑ 1	↑ 2	↑ 2	↖ 3	↑ 3	↖ 4
7	b	0	↖ 1	↑ 2	↑ 2	↑ 3	↖ 4	↑ 4

Figure 3. A simplified example of the dynamic programming-enabled algorithm for finding similarity indexes.

Algorithm 2: Dynamic programming-enabled data aggregation algorithm for IoT infrastructure.

Input: Datasets X_n & Y_m Collected through Neighboring Devices C_i & C_{i+1}
Output: Aggregated Datasets
$Z_k \leftarrow$Similarity Measure(X_n, Y_m);
m: length of dataset X;
n: length of dataset Y;
Table-B[1, 2..m, 1, 2... n];
Table-A[0,1,... m, 0,1,... n];
Function-LCS(m,n);
Function-Length(B,A,i,j);
Discard Duplicates either from X_n or Y_m;
Refined datasets X_n and Y_m;
Function-LCS(arg1, arg2){
  for $i = 0...m$ **do**
  | $c_{i,0} = 0$
end
for $j = 0...n$ **do**
  | $c_{0,j} = 0$
end
for $i = 1...m$ **do**
  | **for** $j = 1...n$ **do**
  | | **if** $X_i = Y_j$ **then**
  | | | Table-A$_{i,j}$ = Table-A$_{i-1,j-1}$+1;
  | | | Table-B$_{i,j}$ = "↖";
  | | **else**
  | | | **if** *Table-A*$_{i,j-1} \leq$ *Table-A*$_{i-1,j}$ **then**
  | | | | Table-A$_{i,j}$ = Table-A$_{i-1,j}$;
  | | | | Table-B$_{i,j}$ = "↑";
  | | | **else**
  | | | | Table-A$_{i,j}$ = Table-A$_{i,j-1}$;
  | | | | Table-B$_{i,j}$ = "←";
  | | | **end**
  | | **end**
  | **end**
end
}
  Function Length (Arg1, Arg2, Arg3, Arg4){
  if $i = 0$ & $j = 0$ **then**
  | return 0;
end
if *Table-B*$_{i,j}$ == ↖ **then**
  | print-LCS(B, A, i-1, j-1);
  | print A_i;
else
  | **if** *Table-B*$_{i,j}$ == ↑ **then**
  | | print-LCS(B, A, i-1, j);
  | **else**
  | | print-LCS(B, A, i, j-1);
  | **end**
end
}**return** *Outliers with Location Information*

4. Performance Evaluation

In this section, we evaluate the performance of the proposed data aggregation method and compare it with state-of-the-art approaches using the well-known open-source OMNeT++ simulator. Various performance metrics, such as data fusion efficiency, duplicate data ratio, system longevity, packet transmission, and congestion control, were examined. The simulation scenario features several devices that were randomly deployed and are controlled by several server modules, with each device placed within the coverage area of at least one server module. Additionally, every server module must communicate directly with a centralized edge, where actual decisions are made. A device is assumed to be within the coverage area of a particular server module if its distance is less than or approximately equal to 450 m, which is a standard distance for the Waspmote board from Libelium. A summary of the general characteristics of the simulation and its parameters is described in Table 1. Furthermore, we assume that every server module does not have to have an equal number of member devices due to the random deployment nature of these networks. However, if these server modules somehow have an equal number of devices, it is also supported in the proposed setup. Finally, the onboard batteries of these devices are thoroughly

checked with different power measurements, such as 1150, 2300, 6600, and 13,000 mAh to make them consistent with the actual deployment infrastructures of IoT. Each metric, along with its approximate values, which are precise and consistent with the Waspmote boards, is presented in Table 1.

Table 1. Simulation set-up along with important parameters and their values.

Parameters	Assumed Values
Coverage area where IoT is deployed	1000 m × 1000 m
Active devices C_i	Approximately 97, 194, 288, 386
Server S_j	4, 8, 16, 24
Edge module	1
Initial or on-board power (E_i)	1150, 2300, 6600, 13,000 mAh
Residual energy (E_r)	E_i—E_{cons}
Power required for the packet transmission (P_{T_x})	91.4 mW
Power required for the packet receiving (P_{R_x})	59.1 mW
Coverage area (T_r)	500 m
(T_r)	0
Beacon length S_j	70 to 100 bytes
Back-off timer S_j	random
Signal-to-noise ratio (SNR) p	10 dB
Channel Delay (Ch_{delay})	10 ms
Power consumption (idle mode)	1.27 mW
Power consumption (sleep mode)	15.4 µW
Energy consumed by transceiver (T_i)	1 mW
Coverage area of the transmitter (T_r)	500 m
Power Threshold for reception (RTS_n)	1024 bits
Packet Size (P_{num})	128 bytes
Distance between server S_j and devices C_i	300 m
Sampling interval	10 s
Typologies checked	Static and Random

4.1. Evaluation Metrics

To describe the exceptional power of the proposed two-tier data aggregation approach, several parameters (metrics) were utilized for the performance evaluation under different conditions, such as when devices in IoT are static and mobile. These performance evaluation metrics include the aggregation ratio (both in-node and global), false and outlier data values (both in-node and at the server), energy and power consumption of both the devices and server module, the lifetime, throughput, and drop ratio of devices in the IoT. These metrics, in terms of both proposed and existing approaches, are described in detail, as given below.

4.1.1. Accuracy and Precision Ratio Metric: In-Node Aggregation

Generally, in IoT infrastructure, the accuracy and precision ratios are among the vital performance evaluation metrics, which are used to thoroughly examine the suitability of a newly developed algorithm in the context of a real deployment setup. Figure 4 presents the accuracy and precision ratios of the proposed method in comparison to the state-of-the-art approaches. It is evident from the figure that the proposed metric outperforms the existing approaches with the performance showing no correlation with the size of the dataset as it utilizes only a fixed portion of the captured data, i.e., the most recent and refined data. The proposed approach has achieved the highest possible ratios of accuracy and precision. Importantly, attaining these high levels does not come at the expense of other evaluation metrics. The new approach is highly effective at utilizing available resources, specifically bandwidth. This efficiency in resource utilization ensures that the overall service quality within the IoT infrastructure remains unaffected and does not degrade.

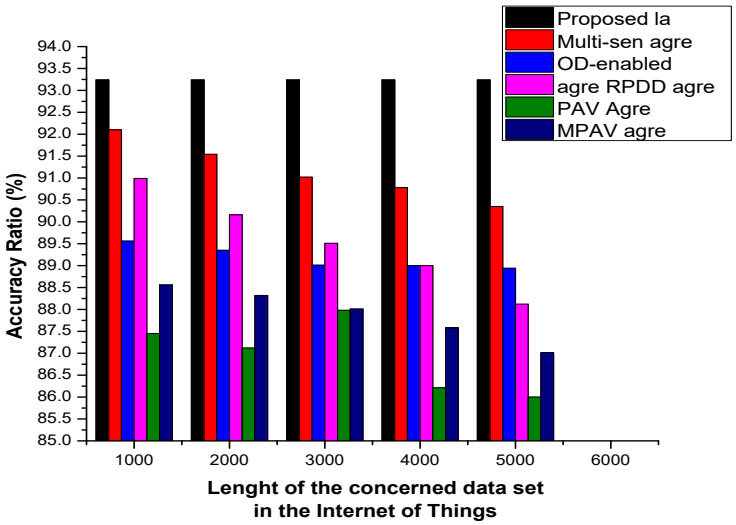

Figure 4. In-node data aggregation accuracy and precision ratios.

4.1.2. Accuracy and Precision Ratio Metric: Dynamic Programming-Enabled Data Aggregation

The proposed in-node data aggregation method operates at the device level to improve the quality of captured data by addressing issues such as duplicates and outliers. However, once the data are sent to the server, this method no longer has any control over it. Hence, this approach is only applicable to refining irregularities in the captured data values before the transmission process. Indeed, the refined data values could still potentially be corrupted during the transmission process due to several factors. Moreover, devices deployed in close vicinity may have captured and transmitted duplicate data values, which are far beyond the operational capabilities of the proposed in-node data aggregation approach. Thus, the dynamic programming-enabled global data aggregation approach was introduced as an extension to the in-node technique to resolve such issues on the server side. The performance of the proposed dynamic programming-enabled approach, compared to existing techniques, is illustrated in Figure 5. The proposed approach performed exceptionally well in terms of accuracy and precision ratios. Additionally, this approach minimized the ratio of duplicate data values, especially those captured and transmitted by neighboring devices in the IoT infrastructure.

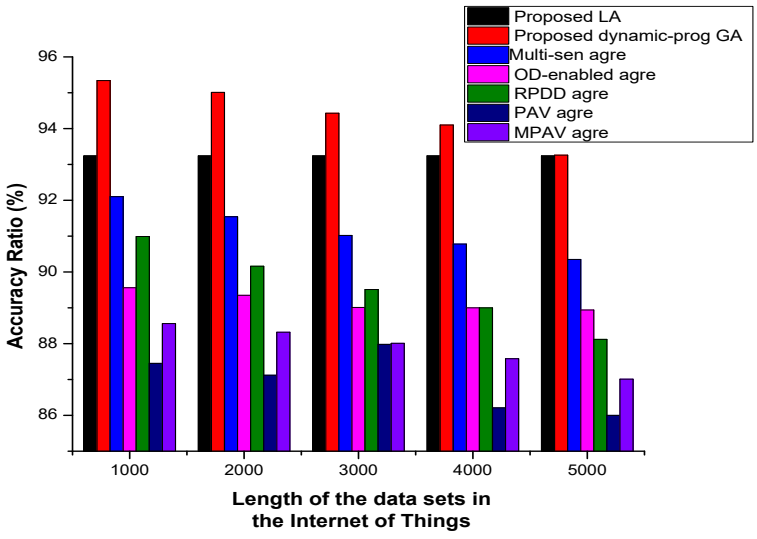

Figure 5. Dynamic programming-enabled data aggregation accuracy and precision ratios.

4.1.3. Refinement Ratio: Device-Level Data Aggregation

It is clear that the proposed technique significantly outperforms existing schemes, including methods for outlier detection (OD), multiple-sensor-based strategies, multiple-pattern anomaly values (MPAVs), and rare pattern drift detection (RPDD). It is important to note that these existing algorithms are resource-intensive and can degrade the performance of the devices or modules involved in the IoT network. In contrast, the proposed in-node data aggregation approach is efficient and size-independent, using only a small portion of the data. It accurately refines data values at the device level, reducing the need for extensive processing at central modules within the IoT network.

It is evident from Figure 6 that the proposed in-node aggregation technique significantly excels in refining data at the device level within the IoT framework, maintaining a high refinement ratio across various dataset sizes. This method's efficiency is highlighted by its consistent performance compared to other techniques such as sensor-enabled aggregation, OD aggregation, MPAV aggregation, and RPD aggregation, reaffirming its suitability for in-node processing. The effectiveness of this approach is especially notable as it achieves high levels of data refinement without the extensive resource consumption typically associated with these processes in central IoT network modules.

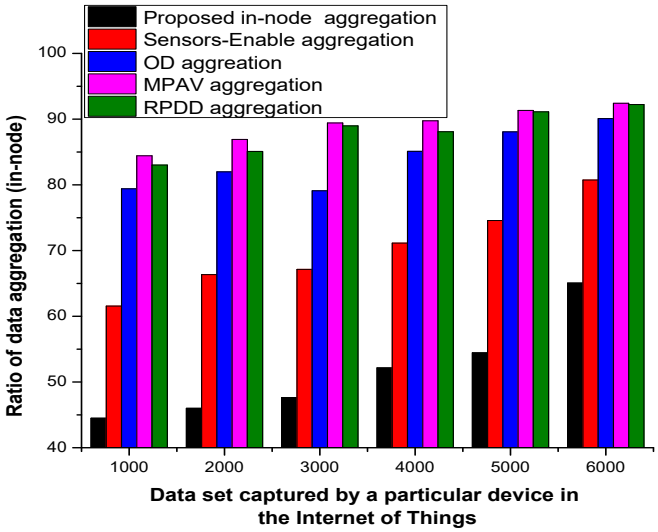

Figure 6. In-node data aggregation: refinement ratio.

4.1.4. Refinement Ratio: Dynamic Programming-Enabled Server-Based Data Aggregation

Figure 7 shows the refinement ratio of the server-based aggregation compared to existing approaches. It is evident from Figure 7 that the minimum possible aggregation (if needed) needs to be carried out at the respective edge module in the IoT infrastructure as a maximum portion of the duplicate and outlier data values are refined either locally through the proposed in-node aggregation or globally via a dynamic programming-enabled aggregation approach. Finally, in both cases, the information loss ratio is negligible as it only eliminates outliers or noisy data values, and for duplicate data values, at least one copy is retained.

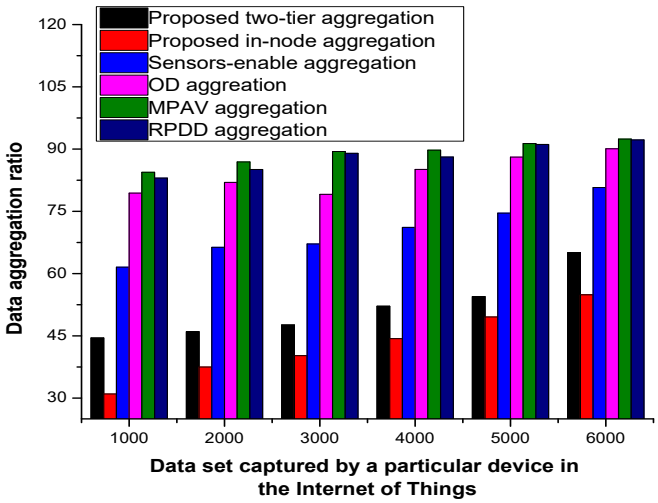

Figure 7. Dynamic programming-enabled data aggregation: refinement ratio.

4.1.5. Computational OR Processing Time Metric

The processing (or computation) time is another vital performance evaluation metric in the context of data aggregation in IoT infrastructure, both locally, i.e., in-node processing, and globally, i.e., server-oriented. It is the approximate time required to complete the respective aggregation process for the detection and refinement of outliers and duplicate data values, either locally or globally. The evaluation results of the proposed method in terms of the processing time compared to the existing techniques are shown in Figure 8, where it is visible that the proposed approach has performed exceptionally well and completed the required aggregation process within the minimum possible time. The computational time of in-node data aggregation is constant as it does not correlate with the dataset size in the operational IoT.

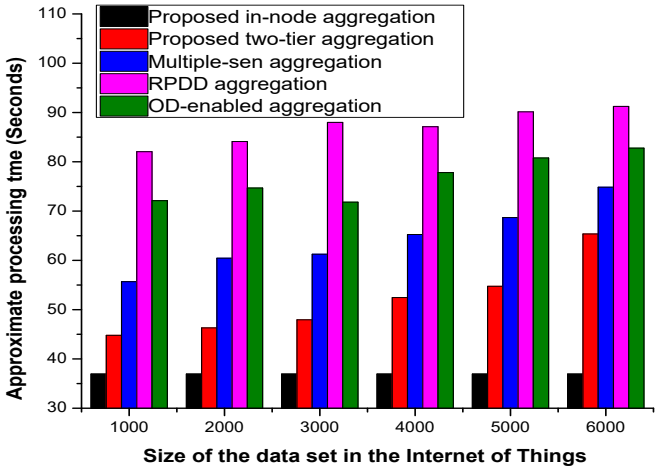

Figure 8. Computational time or processing time.

4.1.6. Drop Ratio of Outliers or Noisy Data Value

The proposed two-tier data aggregation mechanism has a built-in capacity to prohibit the transmission of false readings or outliers through a sophisticated in-node data fusion process, where every captured value is passed through a dedicated filter to ensure its accuracy and precision ratios. For this purpose, and before sending data, every device in the network checks if the data captured by its embedded sensor falls within the acceptable range determined by the upper and lower bounds. The simulation results show that the

information loss ratio of the proposed two-tier mechanism is negligible as the dropped packets are actual false readings. Apart from this, dropping these useless data values results in reducing the ratio of overall congestion. The ratio of total dropout data values of numerous devices, C_i, in the proposed two-tier aggregation mechanism is depicted in Figure 9, where upper and lower bounds are defined according to the embedded sensors. Figure 9 displays where different dropout ratios are shown for various devices in IoT, which is consistent with real-time deployment.

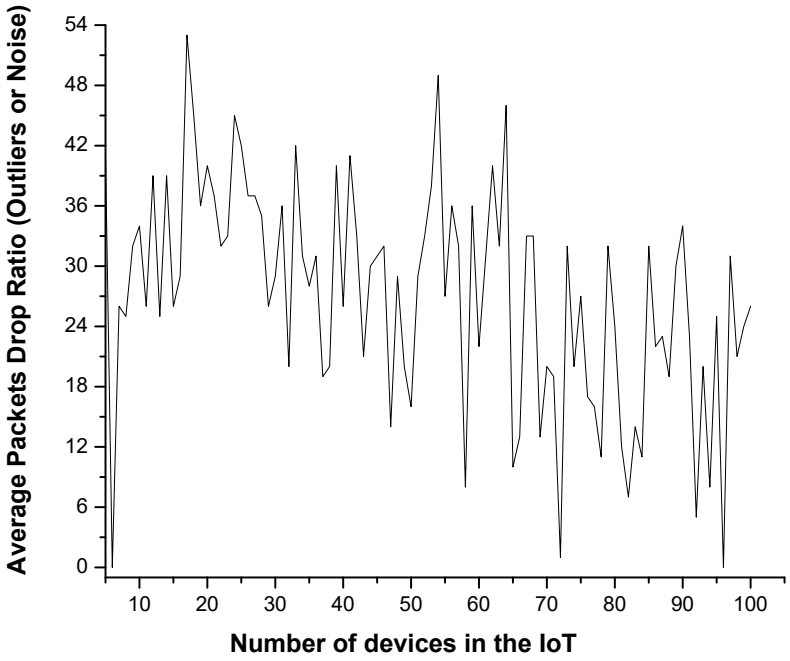

Figure 9. Packet drops by sensor nodes in heterogeneous WSNs to enhance their lifetime.

5. Conclusions and Future Work

The Internet of Things (IoT), due to its significant characteristics, has been widely used across various application domains to increase the productivity and effectiveness of technology-oriented systems with the resources available. However, the accuracy and precision ratios of these systems are compromised if the data values, which are captured through resource-constrained devices and, thus, are highly susceptible to generating noisy data, are not refined prior to the decision-making process. Therefore, in this paper, we present a two-tiered data aggregation approach, namely in-node and server-oriented (dynamic programming-based), to ensure both local and global aggregation of the captured data values. Initially, the in-node data aggregation approach refines the captured data values, including both outliers and duplicates, from the embedded sensors before the transmission process, ensuring that accurate and precise values are transmitted to the respective server module. At the server end, a dynamic programming-enabled data aggregation approach was introduced to minimize (or, if not possible, eliminate) both outliers and duplicates collected throughout the neighboring devices, before their transmission to the edge module in the IoT infrastructure. These approaches not only help in reducing the congestion ratio across the networks but also lead to efficient resource utilization without degrading the performance of the IoT. Finally, the proposed approach can control the congestion ratio as it prohibits the transmission of duplicate and outlier data values. Simulation results have verified the exceptional performance of the proposed two-tier data aggregation technique, especially in terms of the accuracy and precision ratios, processing time, packet drop ratio, and refinement ratio.

The proposed approach could be extended to those IoT networks where either devices C_i or servers S_j, or both, are mobile. Likewise, it could equally be converted to a single robust data aggregation scheme, which applies to both in-node and server-enabled processing. Other future research directions involve the integration of machine learning to refine outlier detection, adapting progressively with the data. Employing edge computing architectures can decentralize computational tasks, diminishing latency and fostering prompt decision-making—key for applications requiring immediacy. Additionally, the drive toward energy conservation is imperative, particularly for battery-dependent IoT devices. Developing algorithms that maximize energy efficiency can prolong device lifespans and promote more sustainable IoT ecosystems. The convergence of machine learning, edge computing, and energy-conscious designs is set to revolutionize IoT networks, rendering them more intelligent, responsive, and long-lasting.

Author Contributions: Conceptualization, S.R.J., B.G. and L.L.; Methodology, S.R.J.; Software, U.U.T. and H.A.; Validation, S.R.J., U.U.T., H.A. and F.S.; Formal analysis, S.R.J.; Investigation, F.S.; Data curation, H.A.; Writing—original draft, S.R.J. and B.G.; Writing—review & editing, S.R.J., B.G., U.U.T., H.A., F.S. and L.L. All authors have read and agreed to the published version of the manuscript

Funding: This research received no external funding.

Data Availability Statement: All required data is either publically available and can be accessed or is given in the paper.

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

References

1. Pramanik, S. An Effective Secured Privacy-Protecting Data Aggregation Method in IoT. In *Achieving Full Realization and Mitigating the Challenges of the Internet of Things*; IGI Global: Hershey, PA, USA, 2022; pp. 186–217.
2. Singh, R.; Dwivedi, A.D.; Srivastava, G.; Chatterjee, P.; Lin, J.C.W. A privacy preserving internet of things smart healthcare financial system. *IEEE Internet Things J.* **2023**, *10*, 18452–18460. [CrossRef]
3. Nabil, Y.; ElSawy, H.; Al-Dharrab, S.; Mostafa, H.; Attia, H. Data aggregation in regular large-scale IoT networks: Granularity, reliability, and delay tradeoffs. *IEEE Internet Things J.* **2022**, *9*, 17767–17784. [CrossRef]
4. Ahmed, A.; Abdullah, S.; Bukhsh, M.; Ahmad, I.; Mushtaq, Z. An energy-efficient data aggregation mechanism for IoT secured by blockchain. *IEEE Access* **2022**, *10*, 11404–11419. [CrossRef]
5. Tariq, U.U.; Ali, H.; Liu, L.; Panneerselvam, J.; Zhai, X. Energy-efficient static task scheduling on VFI-based NoC-HMPSoCs for intelligent edge devices in cyber-physical systems. *ACM Trans. Intell. Syst. Technol. (TIST)* **2019**, *10*, 1–22. [CrossRef]
6. Tariq, U.U.; Ali, H.; Liu, L.; Hardy, J.; Kazim, M.; Ahmed, W. Energy-aware scheduling of streaming applications on edge-devices in IoT-based healthcare. *IEEE Trans. Green Commun. Netw.* **2021**, *5*, 803–815. [CrossRef]
7. Ali, H.; Tariq, U.U.; Hardy, J.; Zhai, X.; Lu, L.; Zheng, Y.; Bensaali, F.; Amira, A.; Fatema, K.; Antonopoulos, N. A survey on system level energy optimisation for MPSoCs in IoT and consumer electronics. *Comput. Sci. Rev.* **2021**, *41*, 100416. [CrossRef]
8. Tariq, U.U.; Ali, H.; Liu, L.; Panneerselvam, J.; Hardy, J. Energy-efficient scheduling of streaming applications in VFI-NoC-HMPSoC based edge devices. *J. Ambient. Intell. Humaniz. Comput.* **2021**, *12*, 9991–10007. [CrossRef]
9. Aujla, G.S.; Jindal, A. A decoupled blockchain approach for edge-envisioned IoT-based healthcare monitoring. *IEEE J. Sel. Areas Commun.* **2020**, *39*, 491–499. [CrossRef]
10. Shirvani, M.H.; Masdari, M. A survey study on trust-based security in Internet of Things: Challenges and issues. *Internet Things* **2022**, *21*, 100640. [CrossRef]
11. Liu, Y.; Wang, H.; Peng, M.; Guan, J.; Xu, J.; Wang, Y. DeePGA: A privacy-preserving data aggregation game in crowdsensing via deep reinforcement learning. *IEEE Internet Things J.* **2019**, *7*, 4113–4127. [CrossRef]
12. Beliakov, G.; James, S.; Kolesárová, A.; Mesiar, R. Cardinality-limiting extended pre-aggregation functions. *Inf. Fusion* **2021**, *76*, 66–74. [CrossRef]
13. Che, C.; Wang, H.; Ni, X.; Lin, R. Hybrid multimodal fusion with deep learning for rolling bearing fault diagnosis. *Measurement* **2021**, *173*, 108655. [CrossRef]
14. Homaei, M.H.; Salwana, E.; Shamshirband, S. An enhanced distributed data aggregation method in the Internet of Things. *Sensors* **2019**, *19*, 3173. [CrossRef] [PubMed]
15. Fitzgerald, E.; Pióro, M.; Tomaszwski, A. Energy-optimal data aggregation and dissemination for the Internet of Things. *IEEE Internet Things J.* **2018**, *5*, 955–969. [CrossRef]
16. Islam, M.M.; Nooruddin, S.; Karray, F.; Muhammad, G. Multi-level feature fusion for multimodal human activity recognition in Internet of Healthcare Things. *Inf. Fusion* **2023**, *94*, 17–31. [CrossRef]

17. Singh, S.; Kumar, D. Energy-efficient secure data fusion scheme for IoT based healthcare system. *Future Gener. Comput. Syst.* **2023**, *143*, 15–29. [CrossRef]
18. Brüser, C.; Kortelainen, J.M.; Winter, S.; Tenhunen, M.; Pärkkä, J.; Leonhardt, S. Improvement of force-sensor-based heart rate estimation using multichannel data fusion. *IEEE J. Biomed. Health Inform.* **2014**, *19*, 227–235. [CrossRef] [PubMed]
19. Abbasian Dehkordi, S.; Farajzadeh, K.; Rezazadeh, J.; Farahbakhsh, R.; Sandrasegaran, K.; Abbasian Dehkordi, M. A survey on data aggregation techniques in IoT sensor networks. *Wirel. Netw.* **2020**, *26*, 1243–1263. [CrossRef]
20. Syed, A.S.; Sierra-Sosa, D.; Kumar, A.; Elmaghraby, A. IoT in smart cities: A survey of technologies, practices and challenges. *Smart Cities* **2021**, *4*, 429–475. [CrossRef]
21. Song, J.; Liu, Y.; Shao, J.; Tang, C. A dynamic membership data aggregation (DMDA) protocol for smart grid. *IEEE Syst. J.* **2019**, *14*, 900–908. [CrossRef]
22. Saleem, A.; Khan, A.; Malik, S.U.R.; Pervaiz, H.; Malik, H.; Alam, M.; Jindal, A. FESDA: Fog-enabled secure data aggregation in smart grid IoT network. *IEEE Internet Things J.* **2019**, *7*, 6132–6142. [CrossRef]
23. Khan, R.; Zakarya, M.; Tan, Z.; Usman, M.; Jan, M.A.; Khan, M. PFARS: Enhancing throughput and lifetime of heterogeneous WSNs through power-aware fusion, aggregation, and routing scheme. *Int. J. Commun. Syst.* **2019**, *32*, e4144. [CrossRef]
24. Fang, J.; Hou, H.; Bi, Z.; Jin, D.; Han, L.; Yang, J.; Dai, S. Data fusion in forecasting medical demands based on spectrum of post-earthquake diseases. *J. Ind. Inf. Integr.* **2021**, *24*, 100235. [CrossRef]
25. Pradhan, S.; Sinha, E.; Sharma, K. Data Fusion by Truncation in Wireless Sensor Network. In *Advanced Computational and Communication Paradigms*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 544–551.
26. Mohseni, M.; Amirghafouri, F.; Pourghebleh, B. CEDAR: A cluster-based energy-aware data aggregation routing protocol in the internet of things using capuchin search algorithm and fuzzy logic. *Peer Peer Netw. Appl.* **2023**, *16*, 189–209. [CrossRef]
27. Zhang, J.; Zhao, Y.; Wu, J.; Chen, B. LVPDA: A lightweight and verifiable privacy-preserving data aggregation scheme for edge-enabled IoT. *IEEE Internet Things J.* **2020**, *7*, 4016–4027. [CrossRef]
28. Haseeb, K.; Islam, N.; Saba, T.; Rehman, A.; Mehmood, Z. LSDAR: A light-weight structure based data aggregation routing protocol with secure internet of things integrated next-generation sensor networks. *Sustain. Cities Soc.* **2020**, *54*, 101995. [CrossRef]