Reducing IoT Data at the Edge: A Comparative Evaluation

Sebastian Leclerc[1], Emma Hansen[1], Alessio Bucaioni[1], Mohammad Ashjaei[1]
[1]Mälardalen University, Västerås, Sweden
{sebastian.leclerc, alessio.bucaioni, mohammad.ashjaei}@mdu.se

***Abstract*—In resource-constrained Internet of Things environments, reducing data transmissions is essential for minimizing energy consumption, network load, and operational costs. Overly aggressive reduction may compromise accuracy, a critical factor in applications such as industrial control. This paper aims to offer practical guidance for selecting suitable data reduction techniques by experimentally evaluating three promising methods from common reduction categories: Data Filtering, Data Aggregation, and Data Prediction. We perform a parameter sweep for each algorithm across three real-world temperature scenarios: stable, rising, and fluctuating. Each configuration is evaluated in terms of data reduction percentage and accuracy, using Total Accumulated Deviation, Mean Absolute Deviation, and Maximum Deviation. Results show that Data Prediction generally achieves the highest accuracy across all scenarios, while Data Filtering tends to yield the greatest reduction at the expense of accuracy. However, all algorithms can be tuned to meet specific scenario demands or accuracy criteria, underscoring that no one-size-fits-all solution exists. We conclude that context-aware algorithm selection and parameter tuning are critical for effective Internet of Things data management.**

***Index Terms*—Internet of Things, Data Reduction, Edge, Fog, Big Data**

I. Introduction

The Internet of Things (IoT) consists of connected heterogeneous devices equipped with sensors and actuators that are distributed across the globe, continuously generating vast amounts of data [1], [2]. Many of these systems rely on cloud computing for centralized processing and storage of the data, which in turn, can provide valuable insights and support automated decision-making across a wide range of domains. As the IoT market continues to grow, the volume of generated data is expected to increase significantly. Real-world applications include smart home systems that monitor and adjust indoor temperature, and agricultural deployments that use environmental data to automate crop irrigation.

A key challenge in IoT systems is the inefficiency, cost, and privacy concerns associated with transmitting large volumes of data to potentially expensive cloud services [1]. Meanwhile, there is also a conflicting need to maintain an acceptable level of accuracy in the transmitted data. The cumulative effect of the challenges presents a significant obstacle to the scalability and sustainability of IoT deployments.

This work is funded by the Swedish Agency for Innovation Systems (Vinnova), project iSecure under Grant 2023-01899, and by the Key Digital Technologies Joint Undertaking (KDT-JU), project MATISSE under Grant 101140216.

To address these challenges, data reduction techniques play a critical role by minimizing the volume of transmitted data while aiming to preserve essential information. Commonly, such techniques are applied in a geographically closer edge or fog computing layer, prior to the cloud.

There exists a wide range of data reduction techniques applicable in edge/fog computing, many of which vary significantly in terms of operational logic, computational cost, and suitability for resource-constrained environments [1], [2]. Kreković et al. [1] identify three broad groups of data reduction methods, while Pioli et al. [2] extend this to 14 distinct categories. Among these, Data Filtering (DF) is the most frequently used, followed by Data Compression (DC), with Data Aggregation (DA) and Data Prediction (DP) occupying third and fourth places, respectively.

In this study, we experimentally evaluate three data reduction techniques: DF, DA, and DP, to investigate which technique most effectively reduces data transmissions *without compromising accuracy*. DC was excluded as it fundamentally reduces payload size rather than transmission frequency.We focus on lightweight, resource-constrained IoT scenarios, as exemplified by temperature monitoring in different scenarios, employing algorithms that embody operationally distinct principles.

An experimental evaluation was conducted by implementing the three data reduction algorithms on a Raspberry Pi (RPi) configured as an edge/fog device. Real sensor data was collected using a temperature sensor across three controlled scenarios: stable, rising, and fluctuating temperature patterns. Each algorithm's accuracy was compared against a baseline that transmits all data, using standard metrics: Total Absolute Deviation (TAD), Mean Absolute Deviation (MAD), and Maximum Deviation (MD). Lower values indicate higher accuracy, each of which may be more critical depending on the application scenario. Additionally, we conducted a parameter sweep for each algorithm across each scenario, showcasing their behavior in terms of data reduction vs. accuracy.

The goal is to provide practical guidance for selecting appropriate reduction methods in resource-constrained IoT deployments. The contributions of this work are:

- Implementation of three representative data reduction techniques based on filtering, aggregation, and prediction principles.
- A comparative benchmark of these techniques using real-world sensor data across multiple conditions.

- Insight into trade-offs between transmission reduction and data fidelity for edge-based IoT applications.

The remainder of this paper is structured as follows. Section II presents background information and reviews related work. Section III details the experimental methodology, including the implementation, algorithms, and comparison of the chosen data reduction techniques. Section IV reports the results of the evaluation. Section V analyzes the findings and discusses potential threats to validity. Section VI concludes the paper and outlines directions for future work. Finally, Section VII addresses the data availability.

II. Background and Related Work

The heterogeneity of IoT systems has led to the development of various reference architectures tailored to specific application domains [3], [4]. These range from simple three-layer and middleware-based models to extended five-layer architectures, often augmented with fog and edge computing components to support low-latency, localized data processing. Placing data reduction closer to the data source improves efficiency and responsiveness, particularly in systems with real-time constraints [2], [5], [6].

Data reduction strategies are typically categorized as single-point or multi-point, depending on whether they are applied at one or multiple system layers [2]. Single-point approaches are more prevalent, especially at the edge or fog layer. Some techniques include a reconstruction phase to recover or approximate the original data, often leveraging predictive models or Machine Learning (ML)-based trend estimation [7].

A wide range of data reduction techniques has been proposed [1]. In this work, we focus on three representative categories of fundamentally different data reduction approaches for resource-constrained IoT: DF, DA, and DP, which are described in the following. DF applies a threshold to filter out minor variations in sensor readings, transmitting only values that differ significantly from the last reported one. This simple yet effective approach is particularly suitable for constrained devices operating on stable data streams. DA combines multiple sensor readings into a single, summarized value to reduce the data volume prior to transmission. Aggregation functions (e.g., average, min, max, sum) are commonly used to eliminate redundancy and reduce communication overhead. DP leverages historical data patterns to forecast future values using analytics or ML models, where the former typically requires fewer computational resources than the latter. This predictive capability enables IoT systems to manage transmission schedules more efficiently by only sending data when predictions diverge significantly from actual readings. It is worth noting that all of the above techniques are inherently lossy, as they discard some data in the reduction process. In contrast, DC techniques may be lossless, allowing exact reconstruction of the original data, but typically require greater computational resources.

Several surveys have examined data reduction strategies in edge and fog contexts. Pioli et al. [2] systematically map techniques such as DF and DC, noting that temperature data is the most commonly used type in evaluations. They identify DF (18.8%) and DC (16.7%) as the most frequently used techniques for reducing data volume at the edge. However, most implementations are simulated or conceptual; our work differs by conducting real-world experiments and comparing methods side by side under controlled conditions. Kreković et al.'s [1] survey categorizes reduction methods into compression, prediction, and aggregation, and evaluate them by energy, accuracy, processing, and data volume. Our implementations of DA and DP align with their general categories, but we deploy them at the edge/fog layer and use consistent datasets and metrics for comparison. In contrast, our DF approach is simpler and targets constrained devices, which their taxonomy does not explicitly include.

Other works propose adaptive or hybrid techniques. Zhang et al. [7] explores a dual-point-based data reduction and reconstruction approach that dynamically adjusts DF thresholds using concept drift detection to balance accuracy and reduction, but does not evaluate deployment feasibility. Papageorgiou et al. [5] propose a switching mechanism among DC strategies to reduce delays, evaluated in a simulated environment only.

While most related works focus on individual techniques, simulations, or high-level surveys, the contributions in this paper lie in a practical, comparative evaluation of DF, DA, and DP under real deployment conditions with consistent performance metrics.

III. Framework for comparing data reduction techniques in the edge

In the following, Section III-A outlines the methodology used to compare the different data reduction algorithms, including the experimental design and evaluation procedure. We then describe the experimental setup and the implementation, covering both hardware and software aspects in Section III-B. Finally, Section III-C explains how each algorithm operates, highlighting their core logic and behavior in the context of the experimental scenarios.

A. Experimental Design and Methodology

The goal of this work is to provide practical guidance for selecting data reduction methods in resource-constrained IoT deployments by experimentally comparing three techniques: DF, DA, and DP, in terms of data reduction and accuracy. To ensure consistency across methods, the same recorded temperature data streams were used for all algorithms, eliminating variability from live sensor input. Data was collected under three representative scenarios:

- Stable: minimal temperature variation in a controlled environment
- Rising: gradual temperature increase, simulating heating conditions
- Fluctuating: irregular changes introduced by alternating heat/cool sources

Each algorithm was tested via a parameter sweep, more thoroughly detailed in Section III-C:

- DF: δ swept from 0.001 to 1.0 (10 logarithmically spaced values)
- DA: N swept from 5 to 50 (10 approximately log-spaced values)
- DP: ε swept from 0.001 to 1.0 (10 logarithmically spaced values)

Filtering and prediction algorithms were constrained to transmit at least once every 30 data points to avoid indefinite silence, while aggregation used fixed intervals of N. The best case corresponds to one transmission every N or 30 points; the worst case is transmitting every N or every point. This ensures comparability across varying configurations.

For each setting and scenario, the reduced stream was compared against the full-resolution baseline. Accuracy was measured using TAD, MAD, and MD, while reduction was expressed as the percentage of the reduced dataset. The final sensor value was not forcibly included to reflect realistic, non-anticipatory conditions.

B. Implementation and Experimental Setup

The physical setup is illustrated in Fig. 1. A BMP280[1] temperature sensor is connected to an ESP32-D0WDQ5-V3 microcontroller running ESPEasy[2], configured as an Message Queuing Telemetry Transport (MQTT) client publishing data at approximately 2 s intervals over default ports without Transport Layer Security (TLS). A RPi running RPi OS serves as the edge/fog node, acting as a local MQTT broker using Eclipse Mosquitto[3]. Communication between the sensor and the RPi occurs over a Wi-Fi Protected Access 2 (WPA2)-secured Wi-Fi (802.11n) network, with the RPi functioning as a hotspot. To isolate and evaluate edge/fog-layer performance, only a single sensor and edge/fog device were used, and data was processed and stored locally on the RPi, excluding any cloud components and spatial/multi-sensor techniques. The data reduction algorithms were implemented in Python ($\geq$ 3.13) to ensure reproducibility and portability across typical edge/fog computing environments.

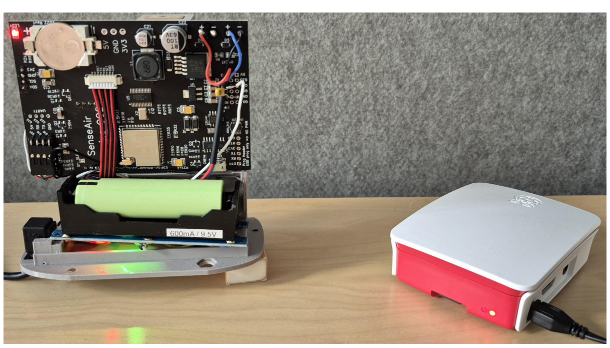


Fig. 1. Physical setup of the experimental system, showing the ESP32 microcontroller with an attached temperature sensor (positioned behind the board) and the Raspberry Pi configured as the edge/fog node.

[1]https://espeasy.readthedocs.io/en/latest/Plugin/P028.html
[2]https://github.com/letscontrolit/ESPEasy
[3]https://github.com/eclipse-mosquitto/mosquitto

C. Overview and Visualization of Data Reduction Algorithms

This section outlines the specific implementations of the three data reduction algorithms evaluated in this study: DF, DA, and DP. The selected algorithms represent operationally distinct principles: selective transmission, data summarization, and trend extrapolation. We illustrate their behavior using a representative temperature dataset to demonstrate how each method reduces the number of transmissions while attempting to preserve data fidelity.

DF: This method uses an adaptive threshold mechanism [7]. The filtering algorithm dynamically adjusts the threshold (e.g., $\delta = \pm 0.05\,^{\circ}\mathrm{C}$) based on recent sensor values. A new data point is transmitted if it either exceeds the threshold deviation from the last sent value, is the first data point, or if a transmission interval limit is reached. Fig. 2 demonstrates how the algorithm suppresses transmissions during stable periods and increases transmission frequency during fluctuations to preserve accuracy.

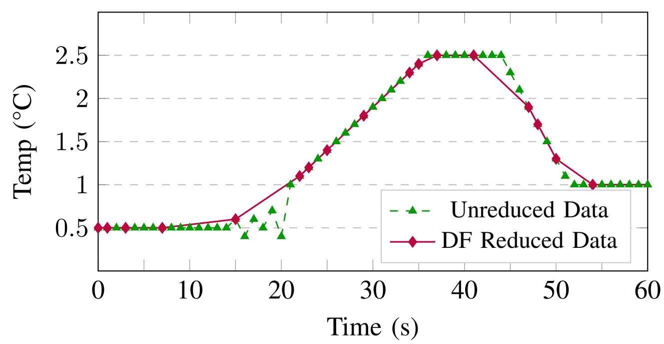


Fig. 2. Transmission behavior of DF algorithm.

DA: This method applies fixed-interval temporal averaging [1]. The aggregation algorithm divides incoming sensor readings into fixed-size groups (e.g., $N = 15$) data points and transmits the arithmetic mean of each complete group. Fig. 3 shows how averaging is performed over fixed intervals (indicated by vertical orange lines). Beyond the first data point, a single value is transmitted per interval. Remaining incomplete intervals are omitted.

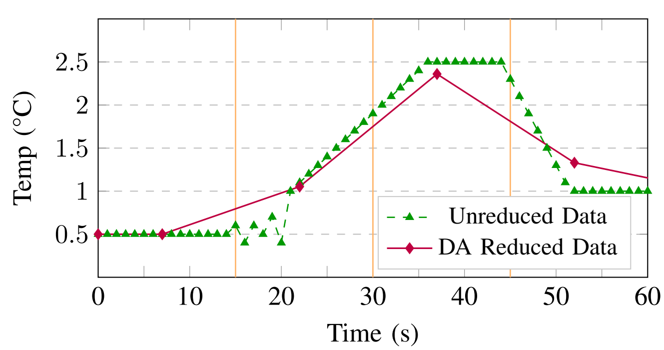


Fig. 3. Transmission behavior of DA algorithm.

DP: This technique uses linear prediction [8]. A predicted value ($\hat{x}_t$) is generated based on a previously transmitted point and the linear slope calculated from the last observed change. If the real-time reading deviates from the prediction beyond a defined error margin (e.g., $\varepsilon = \pm 0.05\,^{\circ}\mathrm{C}$), a new point is transmitted and the prediction line is updated. Fig. 4 illustrates how the algorithm reduces transmissions by sending updates only when the prediction error exceeds the allowed margin. Alternating line colors mark different prediction phases, and shaded regions indicate the tolerated error margin. Color transitions denote recalibrations of the predictive model. For instance, from 0 to 20 s there is one (yellow) prediction phase with a tolerated error margin between 0 and 1 °C.

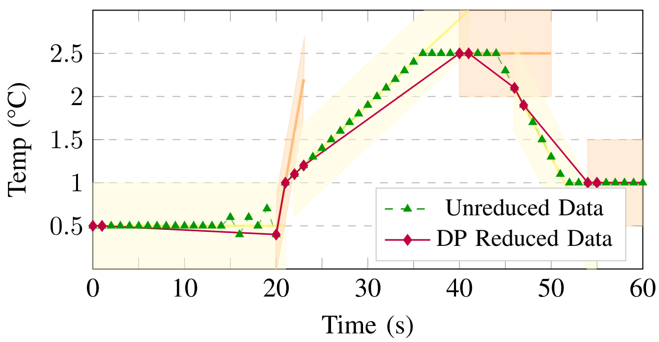


Fig. 4. Transmission behavior of DP algorithm.

IV. Results

The results from all the experiments are presented in Figs. 5, 6, and 7. Each symbol (circle for DP, square for DA, or diamond for DF) represents a distinct algorithm configuration. The leftmost points correspond to the highest tested parameter values (i.e., 1 for DP and DF or 50 for DA), while the rightmost represent the lowest (i.e., 0.001 or 5). Note that some configurations yielded identical results, so not all 10 points may be visible in every case. This typically occurs when the data does not deviate beyond the algorithm's sensitivity threshold, or when extreme parameter settings lead to convergence in accuracy outcomes. The y-axes show the accuracy metrics (TAD, MAD, and MD), where lower is better. Because the experiments involved temperature data, all three metrics are reported in degrees Celsius relative to the baseline to directly reflect the real-world measurement error. The x-axis shows data reduction percentage, where higher is better.

Overall, DP demonstrated the best accuracy (i.e., lowest deviations) across all three scenarios with the tested parameters, achieving an average seen in Table I. This was particularly true when high data retention was acceptable. Its performance was comparable to DF in most cases, with DP outperforming especially in terms of MD. For instance, in the rising scenario (Fig. 6c), a high value of $\varepsilon = 1.000$ resulted in $\approx 93\%$ data reduction with a MD of ≈ 0.14°C. Similar patterns are observed in the fluctuating scenario (Fig. 7c).

DA consistently achieved the highest data reduction across all scenarios, averaging $\geq 91.37\%$, but at the cost of lower accuracy. In Figs. 7a and 7b, both TAD and MAD increased exponentially as N increased. As shown in these figures, with $N = 5$, DA achieved an $\approx 80\%$ reduction, resulting in a TAD of ≈ 60°C and a MAD of ≈ 0.04°C. Although the MD was consistently much higher, e.g., ≥ 5.88°C in Fig. 7c, which is not suitable for sensitive applications.

DF can be considered a low-tier algorithm, slightly better than DP in terms of data reduction when using higher δ values, but trailing in accuracy. Compared to DA, DF with $\delta = 1$ offered similar reduction ($\approx 97\%$) but with significantly better accuracy (e.g., TAD ≈ 84°C in Fig. 6a). Our results, using our temperature data, indicate that $\delta = 0.0215$ is a favorable setting for DF, balancing low MD and respectable data reduction ($\geq$65%) across all scenarios as seen in Figs. 5c, 6c, and 7c.

The overall average data reduction and accuracy metrics across all scenarios are summarized in Table I with the best performance highlighted in green. A clear inverse relationship was observed between data reduction and accuracy for all algorithms.

TABLE I
Overall average data reduction and accuracy.

Algorithm	Reduction (%)	TAD (°C)	MAD (°C)	MD (°C)
DF	56.751%	23.449	0.015	1.462
DA	91.37%	154.070	0.095	3.345
DP	53.99%	17.760	0.012	0.142

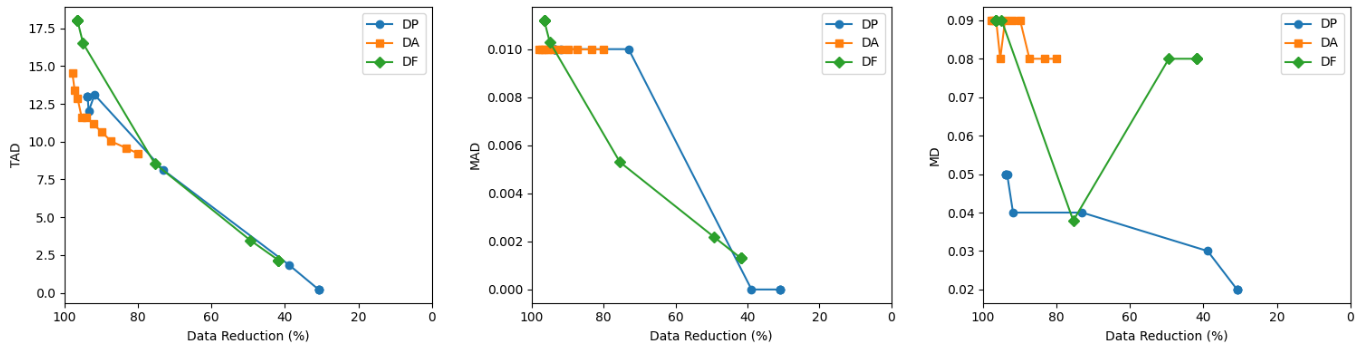
V. Discussion

In this section, we reflect on the experimental results and their implications for selecting data reduction techniques in resource-constrained IoT environments. Section V-A examines the performance of DF, DA, and DP in terms of accuracy and reduction efficiency, highlighting key trends and trade-offs observed across scenarios. Section V-B then considers threats to validity, discussing factors that may have influenced the outcomes and the extent to which the findings can be generalized to other contexts.

A. *Choosing Data Reduction Methods: Accuracy vs. Efficiency*

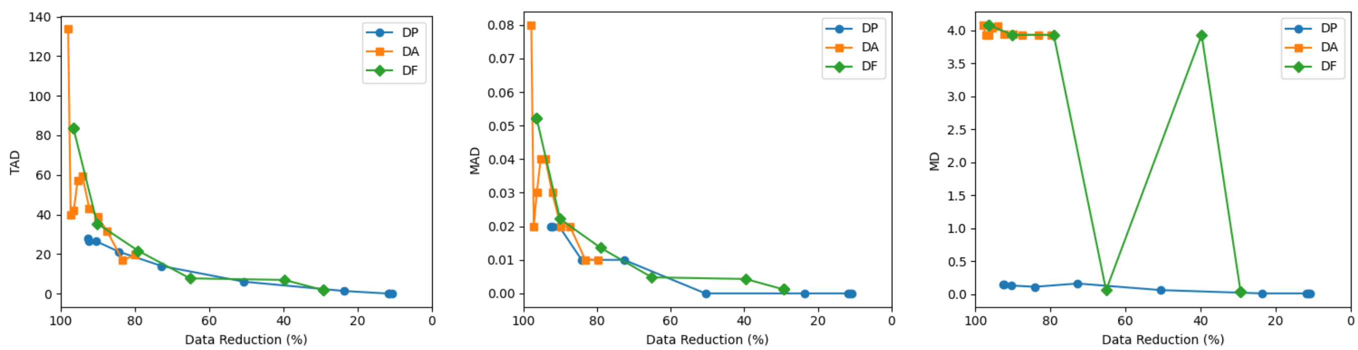
Based on the results presented in Table I, our findings show that DA provides the highest data reduction, albeit at the expense of accuracy. In scenarios where precision is less critical, this trade-off makes DA a suitable choice. Conversely, when accuracy is paramount, DP consistently outperforms the other techniques across all metrics, though it also incurs higher computational cost. DF, by contrast, demonstrated suboptimal overall performance but exhibited competitive results under specific configurations.

The optimal choice of reduction method ultimately depends on the deployment context. IoT developers must make informed, context-aware design decisions based on factors such as data characteristics, expected variability, and the acceptable balance between transmission efficiency and accuracy. Notably, both DA and DF showed cases where, with careful parameter tuning, they outperformed DP in individual metrics, e.g., one DF configuration achieved competitive reduction with improved accuracy (Figs. 5c and 6c). This suggests



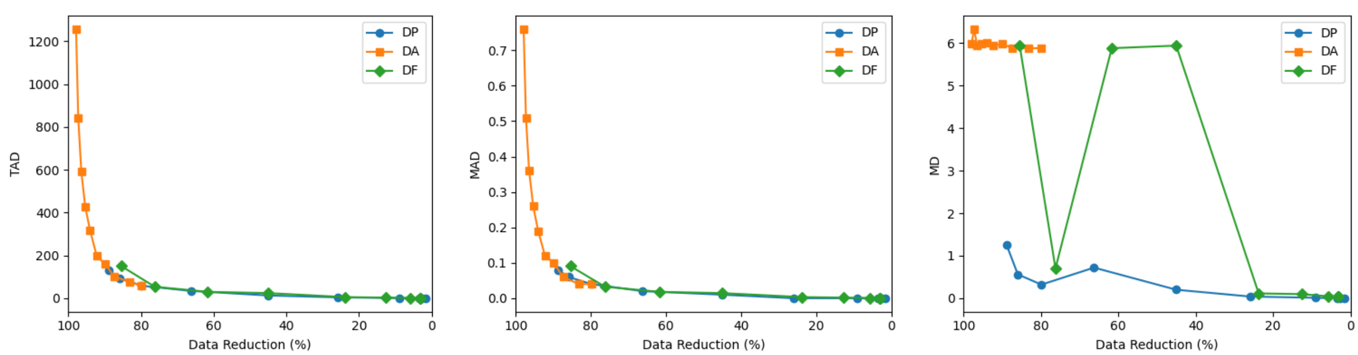
(a) TAD (b) MAD (c) MD

Fig. 5. Stable scenario – Comparison of TAD, MAD, and MD.



(a) TAD (b) MAD (c) MD

Fig. 6. Rising scenario – Comparison of TAD, MAD, and MD.



(a) TAD (b) MAD (c) MD

Fig. 7. Fluctuating scenario – Comparison of TAD, MAD, and MD.

that adaptive or scenario-specific tuning may be advantageous when system resources permit.

The significance of these trade-offs is best illustrated through the lens of the accuracy metrics. A lower TAD is critical in applications like water flow monitoring, where cumulative error affects resource management. MAD is more relevant in stable environments (e.g., indoor climate control), while MD reflects resilience to outliers, vital in safety-critical or industrial systems. In this respect, DP stood out with the lowest maximum deviations.

In summary, our results highlight the need to tailor both algorithm selection and configuration to the application domain, balancing performance with computational constraints and operational goals.

B. Threats to Validity

We categorize the threats to validity according to the framework by Wohlin et al. [9], covering construct, internal, external, and conclusion validity.

Construct validity refers to whether the evaluation metrics accurately capture the phenomena under investigation. In this study, we used point-wise accuracy metrics: TAD, MAD, and MD, to quantify deviations between the reduced and baseline streams. While appropriate for assessing transmission fidelity, these metrics do not account for higher-level temporal structures such as trends or event patterns, which may be critical in domains like anomaly detection or forecasting. Furthermore, our assumption that these metrics fully represent data quality may not hold in all use cases.

Internal validity concerns whether the observed outcomes are caused by the applied techniques and not by confounding factors. To control variability, all reduction algorithms were applied to the same pre-recorded data streams, ensuring a consistent input baseline. However, the specific implementations likely influenced results. Our DF method used an adaptive thresholding mechanism based on recent values, while DA applied fixed-interval temporal averaging, and DP used linear prediction. These were selected for their practicality in resource-constrained environments but do not represent the full range of available strategies. More advanced methods, such as dynamic aggregation windows, higher-order models, or ML-based predictors, may yield different performance profiles. Additionally, while our parameter sweep covered a broad range of values, it did not include extreme edge cases or adaptive tuning.

External validity relates to the generalizability of the results. Our evaluation used only temperature data, which is characterized by low variability and gradual changes. As such, the findings may not transfer directly to domains with higher volatility (e.g., vibration or audio data). Moreover, the system was tested using a single sensor and edge/fog device setup. Real-world IoT systems often involve distributed deployments, heterogeneous hardware, and dynamic network conditions, all of which could impact algorithm behavior.

Conclusion validity addresses whether the data supports the conclusions. Our results showed consistent trends across scenarios and metrics, supporting comparative claims. However, no statistical hypothesis testing was performed, limiting the formal strength of these inferences. Additionally, while reduced transmissions suggest lower energy use, we did not directly measure power consumption. Future work should include such measurements to better quantify efficiency trade-offs considering the power consumption as well.

VI. Conclusions

This work provided a practical comparison of three data reduction techniques: DF, DA, and DP, in resource-constrained IoT settings. We evaluated their performance across three temperature-based scenarios using standard accuracy and reduction metrics. The results show that DP consistently offered the best accuracy, while DA achieved the highest reduction, albeit with greater loss in precision. Although DF underperformed in most cases, it demonstrated competitive performance under specific parameter settings. These outcomes highlight that no single technique is optimal across all metrics, and that context-aware tuning remains essential, aligning algorithm choice with application-specific needs, particularly data variability, tolerance for error, and reduction. The differing performance across TAD, MAD, and MD also illustrates the value of multi-metric evaluation. Future work aims at exploring additional reduction strategies, including adaptive and lightweight learning-based methods. Further evaluation on high-frequency data types, multi-sensor systems, and measurements of computational and energy overhead will strengthen the applicability of these results to real-world deployments.

VII. Data Availability

For transparency and reproducibility, we provide full datasets, source code, and evaluation scripts[4].

Acknowledgements

This work is supported by the Swedish Agency for Innovation Systems through the project "Secure: Developing Predictable and Secure IoT for Autonomous Systems" (2023-01899), and by the Key Digital Technologies Joint Undertaking through the project "MATISSE: Model-based engineering of digital twins for early verification and validation of industrial systems" (101140216).

The authors would like to thank all industrial partners, especially Canarybit and Senseair for their valuable input.

References

[1] D. Kreković, P. Krivić, I. Podnar Žarko, M. Kušek, and D. Le-Phuoc, "Reducing communication overhead in the IoT–edge–cloud continuum: A survey on protocols and data reduction strategies," *Internet of Things*, vol. 31, p. 101553, May. 2025.

[2] L. Pioli, C. F. Dorneles, D. D.J de Macedo, and M. A. R. Dantas, "An overview of data reduction solutions at the edge of IoT systems: a systematic mapping of the literature," *Computing*, vol. 104, pp. 1867–1889, Mar. 2022.

[3] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," *IEEE Commun. Surv. and Tut.*, vol. 17, no. 4, pp. 2347–2376, Jun. 2015.

[4] S. Yi, C. Li, and Q. Li, "A Survey of Fog Computing: Concepts, Applications and Issues," in *Proc. 2015 Workshop Mobile Big Data*, 2015, pp. 37—-42.

[5] A. Papageorgiou, B. Cheng, and E. Kovacs, "Real-time data reduction at the network edge of Internet-of-Things systems," in *Proc. 11th Int. Conf. Netw. Serv. Manag. (CNSM)*, 2015, pp. 284–291.

[6] S. Leclerc, A. Bucaioni, and M. Ashjaei, "Characterizing Time-Critical Internet of Things," *Internet of Things*, pp. 1–38, Oct. 2025.

[7] H. Zhang, J. Na, and B. Zhang, "Autonomous Internet of Things (IoT) Data Reduction Based on Adaptive Threshold," *Sensors*, vol. 23, p. 9427, Nov. 2023.

[8] Y. Rassadin and S. Dushin, "Efficient Wireless Data Collection System Based on LoRaWAN Technology and Distributed Computation Approach," in *Distrib. Comput. Commun. Netw.: Control, Comput., Commun.*, 2020, pp. 510–520.

[9] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in Software Engineering*. Heidelberg, Germany: Springer, 2024.

[4]https://github.com/SebastianLeclerc/edge-data-reduction