Uncertainty-Aware Optimisation for Sustainable Multimedia Event Processing in Big Data Streams

Felipe Arruda Pontes
Insight Centre for Data Analytics
University of Galway
Galway, Ireland
0000-0002-9062-7653

Michael Schukat
School of Computer Science
University of Galway
Galway, Ireland
0000-0002-6908-6100

Edward Curry
Insight Centre for Data Analytics
University of Galway
Galway, Ireland
0000-0001-8236-6433

***Abstract*—Multimedia Event Processing (MEP) systems play a critical role in various Internet of Things (IoT) applications, including Smart Cities and Health and Safety, by processing large amounts of multimedia data streams. These systems often leverage state-of-the-art Deep Neural Network (DNN) models to enhance their capabilities. However, the growth of Cloud and Edge Big Data applications has imposed a significant environmental burden, further intensified by the substantial energy consumption associated with certain DNN model operations.**

This study addresses the environmental impact of growing Big Data stream applications and focuses on optimising MEP systems to mitigate this issue. We tackle uncertainties arising from user-defined Quality of Service (QoS) interpretations and service worker measurement imprecisions by using uncertainty-aware solutions for the service selection problem in order to improve the QoS within MEP systems.

Our results reveal substantial advantages in employing uncertainty-aware strategies. These approaches consistently enhance QoS metrics, outperforming their uncertainty-oblivious counterparts. Specifically, we report improvements in more than 67%, 69%, and 20% of the scenarios, on average, for energy consumption, latency, and accuracy, respectively. These enhancements become evident within just three hours of processing, resulting in energy savings of up to 1.2 kilowatt-hours and latency reductions of 213 seconds, with a 0.29% average loss in query accuracy. These strategies improve system efficiency and ecological sustainability while incurring a small accuracy trade-off. When extrapolated over a year, the environmental benefits become even more noticeable, surpassing the energy requirements for a 1000 Km electric vehicle round-trip from Amsterdam to Paris and back.

***Index Terms*—Sustainability, Big Data, Deep Neural Networks, Multimedia, Streaming**

I. Introduction

Multimedia Event Processing (MEP) systems are commonly used in a wide range of Internet of Things (IoT) applications, such as Smart Cities and Health and Safety, to facilitate the processing of large-scale multimedia data streams through the use of State-of-the-Art Deep Neural Network (DNN) models [1], [2]. However, the increase in Cloud and Edge Big Data applications, such as these, has greatly impacted the environment [3], which is further compounded by the high energy consumption that some DNN model operations can have [4]. In 2016, data centres represented over 1.8% of the energy consumption in the US, and it is estimated that, along with other computing devices, they will represent 14% of the global energy consumption over the next decade [5]. These numbers raise an essential concern for developing more ecologically sustainable MEP systems on the Cloud and Edge. Moreover, it is common for MEP systems to have autonomic (self-adaptive) capabilities in order to constantly adapt to the changes in the deployment environment and select the best service worker (i.e., computing device and DNN model) that meets the Quality of Service(QoS) requirements of each user, not only in terms of sustainability (i.e., energy consumption) but also the accuracy and speed (i.e., latency) of the user query, which is seen as a Multi-Criteria Decision Making (MCDM) problem. However, these adaptations can be significantly affected by uncertainties arising from MEP applications in real-world scenarios, such as different interpretations of the user QoS requirements and imprecision in the performance measurement of the computing devices and DNN models used to process the user queries [6]–[8]. These issues raise the following research question: *What are the effects that uncertainties from real-world MEP applications have on each user's sustainability goals and other QoS requirements of the system (i.e., energy consumption, accuracy and speed)?*

As a response to this question, we developed an uncertainty-aware (UA) solution for the service selection problem that can handle the uncertainties from different QoS interpretations and the imprecise measurements of service worker profiles, which show evident improvements in the QoS of the system when compared to an uncertainty-oblivious (UO) method in a real-world setting. Our system uses the Fuzzy TOPSIS (Technique for Order Performance by Similarity to Ideal Solution) [9] method, an extension of the commonly used TOPSIS MCDM method into the fuzzy domain, which handles some of these uncertainties [10]. In our experiments, after a few hours of processing time, our uncertainty-aware solution already improves the system QoS by reducing the total energy consumption (by up to 1.2 kWh) and the latency of queries (by up to 213 seconds), providing a faster and more ecologically sustainable MEP, with a slight trade-off in

This publication has emanated from research conducted with the financial support of Science Foundation Ireland (SFI) under Grant Number SFI/12/RC/2289_P2, co-funded by the European Regional Development Fund. For the purpose of Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission.

©2023 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. Final published version can be found at: https://doi.org/10.1109/BigData59044.2023.10386088

accuracy (3.9%), when compared to an uncertainty-oblivious method in a real-world setting. Our results show that, on average, the uncertainty-aware solution has better QoS than the uncertainty-oblivious method in more than 67%, 69%, and 20% of the scenarios for the criteria of energy consumption, latency, and accuracy, respectively. Additionally, our solution shows up to 206 kWh of savings in energy in a one-year processing time, or equivalent to 9.8% savings in a Data Centre server, which is more than the energy consumed by an electric vehicle on a round-trip (1000 Km) from Amsterdam (Netherlands) to Paris (France).

The remainder of this paper is organised as follows. Section II presents the motivation scenario of sustainable traffic management for smart cities. Section III describes the related works and the research gap targeted by our study. Next, in Section IV, we describe the types of uncertainty that can affect sustainable MEP applications' adaptation goals, and how much uncertainty-aware and uncertainty-oblivious methods for service selection tend to differ in their results. Section V presents the experiments for comparing the effects on the QoS when the UA and UO solution. Section VI discusses the results of this comparison and our discussion of the findings. Finally, in Section VII, we draw conclusions and present future works.

II. Sustainable and Smart Traffic Management

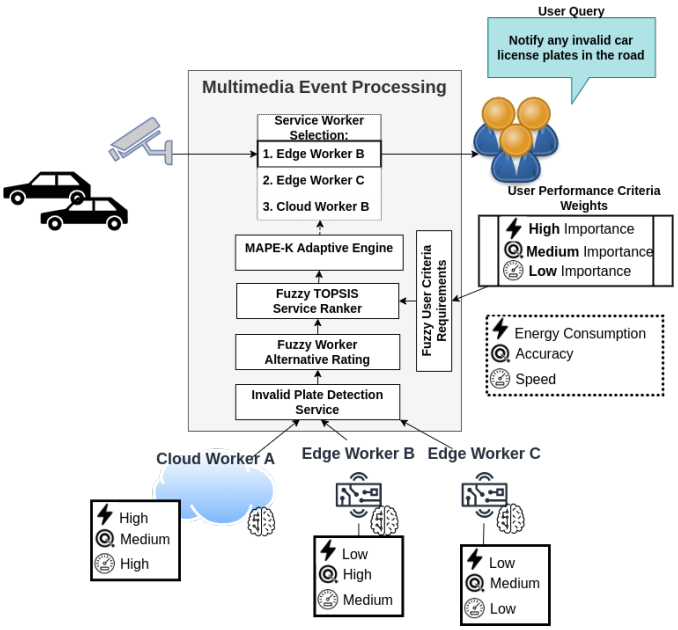

Figure 1: Example of UA service selection in a DNN-based MEP system for sustainable smart-city traffic management deployed on the Cloud and Edge environment. The top-ranked service worker (Edge Worker B) is selected from the available alternatives for processing the user query based on the user's performance criteria requirements (i.e., energy, speed, and accuracy) and available services.

In the context of smart cities, MEP systems are commonly applied to provide a smart traffic management solution [2], because of the large quantity of video data stream from traffic cameras. Figure 1 illustrates a user querying the system for cars with invalid license plates. In this example, the traffic camera images are processed using a specialised service with multiple workers deployed on both Cloud and Edge environments, with pre-trained DNN models to detect the presence of invalid license plates. The Fuzzy TOPSIS service selection is used to rank the top service workers (i.e., Edge Worker B) from the available alternatives according to their values of energy, accuracy and speed and the importance of each criteria defined in the user query (i.e., High, Medium, and Low importance for energy, accuracy, and speed, respectively). Once ranked, the top alternatives are selected during an adaptation cycle. These adaptations are based on the MAPE-K self-adaptive architecture [11]. They are triggered by changes in the deployment environment (e.g., when a worker is no longer available) or current user queries (e.g., new queries with specific quality requirements). Moreover, other queries can be included in this same smart traffic MEP application, such as detecting road accidents or bad traffic conditions, each with its own sustainability and performance requirements.

III. Related Works

This section will describe related works on service selection in uncertainty-aware Sustainable Multimedia Event Processing frameworks for Cloud and Edge, focusing on solutions that use fuzzy logic to handle uncertainty.

An extensive study of uncertainty-aware service selection methods using fuzzy logic, including Fuzzy TOPSIS, Fuzzy AHP, and others, is presented in the review by [12]. Moreover, in the work by [6], a proof-of-concept rule-based fuzzy controller is presented, with a focus on resource-constrained Edge devices based on the system administrator (sys-admin) requirements. Similarly, RobusT2Scale, as described in [13], uses a rule-based fuzzy controller for handling uncertainty in sys-admin policy definitions and system monitoring. They propose a Cloud elastic scaling solution that employs uncertainty-aware self-adaptive load balancing to manage response time and computational resources, also utilising Fuzzy Q-Learning for knowledge evolution. Another rule-based fuzzy solution is presented in [7], where they developed a self-adaptive demand- and uncertainty-aware task management that focuses on reducing energy consumption and maximising performance by improving the process of Edge-Cloud task offloading in the context of web service applications. The study by [8] also addresses energy consumption and the economic impact on self-adaptive, uncertainty-aware schedulers. They analyse both fuzzy and non-fuzzy solutions and how they can affect the interpretability of the scheduler in the context of Cloud-deployable applications. In [14], it presents a use case of a diabetes monitoring system with HTTP-based web services. Similar to our work, they also use the fuzzy TOPSIS method for service selection based on the Quality of Service (QoS) definitions while considering both benefit and cost criteria.

Overall, we see a gap in the current research on:

1) An understanding of the effects that uncertainty plays in the sustainability, speed, and accuracy goals of MEP applications in real-world settings;
2) Most uncertainty-aware self-adaptive applications use sys-admin-defined QoS requirements for the entire application rather than query-based user-defined QoS requirements, which would provide the users with finer control over the quality of their queries;
3) Although there are self-adaptive and uncertainty-aware solutions with energy consumption (i.e., sustainability) as a QoS criterion in the context of HTTP web-based service applications, there are still few to none of these solutions that are developed for the specific requirements of Big Data video streams on Multimedia Event Processing applications.

IV. Real-World Uncertainties in Adaptive Multimedia Event Processing

In a real-world context, there are dynamic and complex environments that depend on variables with vague and ambiguous human definitions and imprecise measurements [10], [15], [16]. As mentioned earlier, to achieve the QoS goals, MEP applications must adapt and select the best service worker for processing each user query based on their QoS requirements. The TOPSIS method [9] is a commonly used solution for this service selection problem. It prioritises the alternatives (i.e., the available service workers for processing a user query) that are closer (in terms of Euclidean distance) to the Positive Ideal Solution (PIS) and farthest from the Negative Ideal Solution (NIS). This distance represents the solution that minimises the cost criteria (i.e., energy consumption) and maximises the benefit criteria (i.e., accuracy). The method also considers the importance weights of each criterion (i.e., the user-defined QoS requirements for energy, speed, and accuracy). The inputs of this methods are a decision matrix D and the weights vector W, with w_j as the weight of criterion j, and X_{ij} as the value of the criterion j of alternative worker i:

$$D = \begin{array}{c} \\ A_1 \\ A_2 \\ A_i \\ \dots \\ A_m \end{array} \begin{array}{c} \begin{array}{ccccc} C_1 & C_2 & C_j & \dots & C_n \end{array} \\ \begin{bmatrix} X_{11} & X_{12} & X_{1j} & \dots & X_{1n} \\ X_{21} & X_{22} & X_{2j} & \dots & X_{2n} \\ X_{i1} & X_{i2} & X_{ij} & \dots & X_{in} \\ \dots & \dots & \dots & \dots & \dots \\ X_{m1} & X_{m2} & X_{mj} & \dots & X_{mn} \end{bmatrix} \end{array} \tag{1}$$

$$W = \begin{bmatrix} w_1 & w_2 & w_j & \dots & w_n \end{bmatrix} \tag{2}$$

The general steps for the TOPSIS method are defined in Algorithm 1.

However, the decision-making process for service selection can be directly impacted by uncertainties arising from different interpretations of the user performance requirements and imprecision in the measurements of the service workers' devices [6]–[8]. Thus, it is essential to understand how these real-world uncertainties can affect MEP systems to ensure that the QoS goals of energy consumption, speed, and accuracy of the user's queries are not inadvertently compromised by selecting a poorly ranked service that misrepresents reality. While the original (i.e., crisp) TOPSIS method remains oblivious to these issues, the Fuzzy TOPSIS, on the other hand, employs fuzzy logic to address these uncertainties in the QoS definitions and service workers' profiles within the system topology [10], which inevitably leads to a high contradiction rate in the ranking results of these two methods when using data from real-world scenarios [17]. The Fuzzy TOPSIS uses fuzzy numbers, such as triangular fuzzy numbers (see Figure 2) instead of crisp (precise) numbers to represent the values of X_{ij} and w_j. In our work we used triangular fuzzy numbers, as seen in the original paper [10], that are defined according to the triangular membership function $\mu(x)$ as follows:

$$\mu(x) = \begin{cases} 0, & \text{for } x < a \\ \frac{x-a}{b-a}, & \text{for } a \leq x < b \\ \frac{c-x}{c-b}, & \text{for } b \leq x < c \\ 0, & \text{for } x \geq c \end{cases} \tag{3}$$

We can see in Figure 3 that the main difference between these two methods is how the uncertainty-aware solution uses linguistic variables to better represent the human-defined QoS and also uses linguistic ratings that can represent multiple values (as triangular fuzzy numbers). In contrast, the uncertainty-oblivious method adds an ambiguous interpretation of the

Algorithm 1 TOPSIS General Steps

Require: D, W {As defined in Eq. 1 and 2}
Ensure: $Rank$, CC {Outputs the rank and Closeness Coefficients}
1: Create a normalised decision matrix R
2: Create the weighted normalised decision matrix V
3: Calculate PIS (A^*) and NIS (A^-)
4: Calculate the distance measures d_i^* and d_i^- for benefit and cost criteria, respectively.
5: Calculate Closeness Coefficient for each alternative (CC_i).
6: Define $Rank$ order according to each CC_i.
7: **return** $Rank$, CC

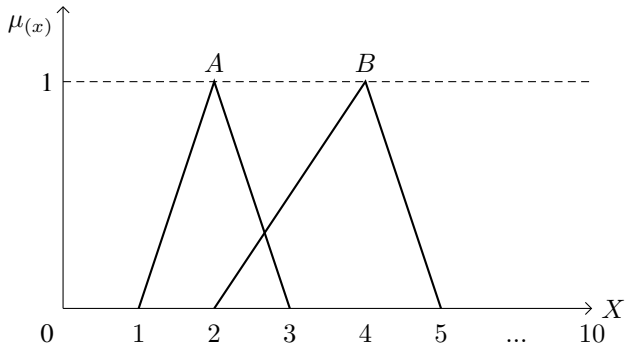

Figure 2: Example of two triangular fuzzy numbers A and B, represented as (1, 2, 3) and (2, 4, 5), respectively.

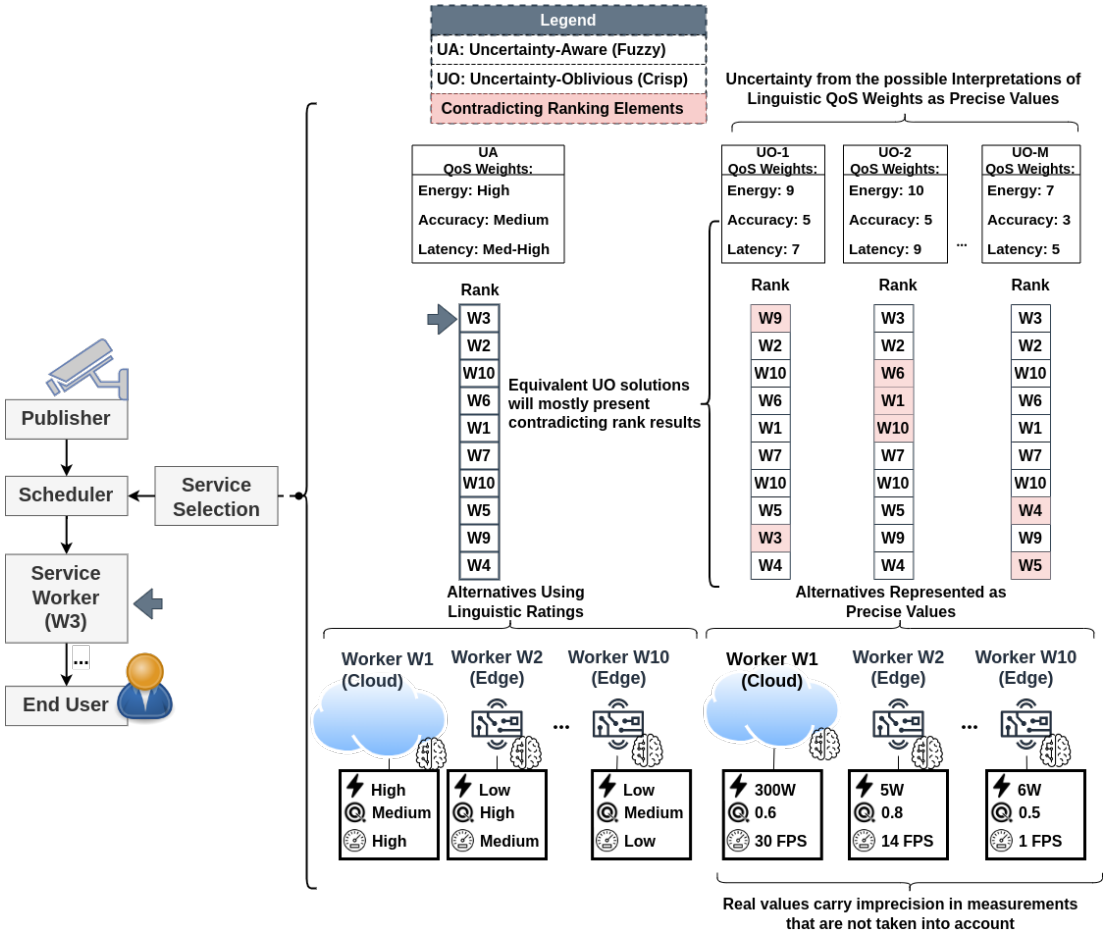

Figure 3: On the right side, we can see how the result of the ranking of the service selection is used during the data processing pipeline. On the left, we can see the comparison of the results obtained from the fuzzy and crisp TOPSIS methods, representing the uncertainty-aware (UA) and oblivious (UO) service selection methods, respectively. This diagram illustrates the uncertainty arising from the multiple interpretations of linguistic QoS weights when treated as precise values. Additionally, it shows how each method treats the alternative ratings, with a linguistic rating variable and the average measurements for the UA and UO methods, respectively. Another source of uncertainty arises from the UO method treating these ratings as precise values, whereas, in reality, they are affected by imprecision. This disparity ultimately results in significant differences in the rankings produced by the UA and UO methods.

QoS as single, precise values (i.e., within the range of 1-10). Furthermore, the oblivious solution neglects the expected variations and imprecision in measuring the worker's profile ratings, treating them as precise values instead. As we have previously identified, these differences ultimately result in significant disparities in the ranking outcomes of these two TOPSIS methods when applied in real-world scenarios, raising the question of how much these differences between the uncertainty-aware and oblivious methods will impact the final QoS of the MEP system.

V. System Evaluation

To assess the impact of real-world uncertainties on the QoS of the MEP application, we designed a series of experiments to account for variations in energy consumption, processing speed, and different interpretations of the QoS requirements, with a total of 1848 experiment executions. In our evaluation, we utilise profiles based on actual measurements of state-of-the-art Object Detection DNN models: SSD-MobilenetV1 (SSD) [18], Faster RCNN-InceptionV2 (Faster RCNN) [19], Faster RCNN-Inception-ResnetV2-Atrous (Faster RCNN-Atrous) [19]. All these models were pre-trained on the COCO 2017 image dataset [20]. These models were deployed on various Cloud and Edge devices, covering multiple deployment environment configurations, considering all possible combinations of 10 out of the 12 available service worker profiles ($12C_{10} = 65$). These we will refer to as *setups*. Each experiment is conducted with a specific setup of workers, using either the UA solution or one of its 27 equivalent UO solutions. Each experiment uses one query to detect "person" on any frames of a simulated video publisher at $\sim$ 58 FPS since this should provide considerable

input load into the system without fully overloading it. The video event data (i.e., simulated video frames) is fed into the system for 90 seconds, and after that, the system continues to run until all events have been processed, a process that typically takes an average of 3.3 hours to complete. As a result, each experiment execution involves approximately 5220 events utilising the UA/UO service selection rankings. Furthermore, in all executions, we assign medium importance to all query QoS criteria (energy, latency, and accuracy). We then compare the QoS results obtained with the UA approach to those of the 27 equivalent QoS interpretations treated as precise values in the UO solutions. This process is repeated for each of the 65 explored setups.

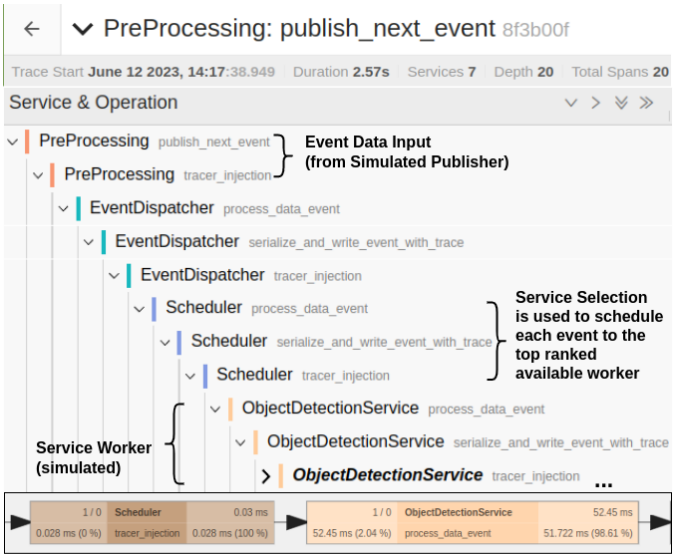

Figure 4: This figure shows some of the information from an event trace in the MEP framework during one of the experiments. The event traces show complete historical data of each step in the data processing pipeline. However, we have cropped out the remaining services after Object Detection due to space constraints. At the bottom, we can see more details of certain service operations (i.e., Scheduler and Object Detection Service). We also indicate that the PreProcessing service ingests each data event from the simulated publisher, which is subsequently scheduled based on the top-ranked Object Detection service worker in the service selection process. Finally, the event is processed by the selected simulated worker.

Additionally, to better capture the variations and imprecision in a realistic application, we execute each experiment on a complete end-to-end Multimedia Event Processing framework (Gnosis MEP) [21], and then plug into the framework a simulated publisher and object detection service workers with realistic behaviour from our 12 service worker profiles. This approach allows us to account for minor variations in event publishing and realistic delays in event communication through the data streams (using Redis Streams). This framework also provides the advantage of leveraging automatic monitoring features, such as the event tracing system (i.e., Jaeger). This feature simplifies our evaluation process and ensures experiments that are closer to reality, as it enables the retrieval of precise event timestamps as they pass through each component of the data processing pipeline in the MEP framework, as depicted in Figure 4.

A. Metrics

To create a more realistic scenario for the evaluation, we introduce random variations into two of the three criteria: energy consumption and speed (i.e., throughput) of each alternative worker during the simulation. We follow the same method from [22] to represent these realistic uncertainties, in which these variations are determined based on the actual measurements of average and standard deviation values of energy and throughput for each worker profile. However, we do not cover the variations in the computation of query accuracy for the workers; instead, we consider the only exact value of accuracy of the model used by each worker.

1) Avg. Query Latency: During the execution of the experiments, each worker retrieves the next event from its processing queue and waits for a specified amount of time (P_e) before marking the event as processed and moving on to the next event in its queue. To simulate a more realistic throughput for the workers, we introduce a random variation(R^{100}_{-100}), ranging from -100% to 100%, based on the standard deviation values (σwt) of throughput in each worker's profile. This adjusted throughput (Δ_t) is then calculated and used as the wait time for each event processed by the workers in the experiments, as shown in the formula below:

$$\Delta_t = \frac{R^{100}_{-100} \times \sigma w_t}{100} \tag{4}$$

After each experiment, the details of the selected worker and the processing time of each event are retrieved from the event traces through the Jaeger API. This API provides us with timestamps for the start of event processing and the duration of the operation, both in microseconds. We sum these values to calculate the timestamp for the end of processing ($EndTS_e$), which is later converted into seconds. Additionally, for the event latency, we gather from the event traces the initial time ($InitTS_e$) in which the event was published into the system; this way, we can also account for the real delays of handling the event in the messaging system, the time for scheduling the event and the waiting time on the worker queue. Finally, the average latency of each experiment ($\bar{L}_i$) is calculated by averaging each processed event's latency, as seen below:

$$\bar{L}_i = \frac{1}{n(Q)} \sum_{e \in Q} EndTS_e - InitTS_e \tag{5}$$

2) Total Energy Consumption: The total energy consumption (E_i) of each experiment execution is calculated using the energy consumption values of each worker (w) configuration and processing speed, and total energy consumption of the workers on standby (when not being used to process any event). Additionally, we apply to each processed event a

random variation (R^{100}_{-100}), from -100% to 100%, to the energy consumption based on the standard variation values (σw_c) of each worker profile, according to the formula:

$$\Delta_c = \frac{R^{100}_{-100} \times \sigma w_c}{100} \tag{6}$$

The energy consumption for processing each event (C_e) is then calculated by retrieving from the event traces (through Jaeger's API) the total processing time (P_e) for the event and multiplying it by the energy consumption of the worker used to process this event, after applying the random variation (Δ_c) to this consumption value:

$$C_e = P_e \times (w_c + \Delta_c) \tag{7}$$

Next, we calculate the total standby time of each worker and multiply it by their profile's standby energy consumption values (S_w). These standby energy values were based on the measured standby energy consumption of the Cloud and Edge workers, which is 72.1 and 2 Watts, respectively. No variation is applied since the standard deviation in these cases was negligible. Finally, we calculate the total energy consumption of each experiment(E_i) by summing the energy consumption of all events and the sum of all worker's energy consumption in standby, as can be seen on the following formula:

$$E_i = \sum_{w \in W} S_w + \sum_{e \in Q} C_e \tag{8}$$

3) Avg. Query Accuracy: The metric we used for the query event's accuracy was the mean Average Precision (mAP), and the values gathered from the profile of the DNN model of the worker used to process each event (according to the model's report in the TensorFlow models repository). Therefore, for each experiment, the average query accuracy ($\bar{A}_i$) is calculated by averaging the accuracy of each processed event (Q) in the experiment, as shown below:

$$\bar{A}_i = \frac{1}{n(Q)} \sum_{e \in Q} mAP_e \tag{9}$$

VI. Results

Looking at each of the 65 setups, we can calculate the percentage of cases where UA produced better QoS than its equivalent UO solutions (27 in total) in that setup. Given all 65 setups explored, we note that UA produces, on average, better QoS results for energy and latency at 67% and 69% of the time, respectively, when compared to equivalent UO solutions. In Figure 5, we can see the distribution of each scenario's percentage of cases where UA was better than UO, which clearly shows us that the UA tends to produce better QoS results for energy consumption and latency criteria than equivalent UO. However, as we will see next, the same cannot be said for the accuracy criterion.

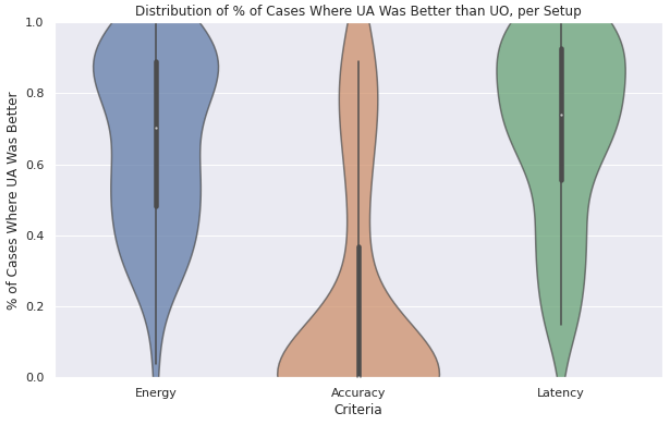

Figure 5: Distribution of percentage of cases in which the Uncertainty-Aware (UA) service selection had better results than an equivalent Uncertainty-Oblivious (UO) solution for the QoS criteria of energy, accuracy and latency. These percentages cover the comparison of each UA against 27 equivalent UO solutions for every one of the 65 setups explored.

A. Accuracy Trade-off

When defining a DNN-based sustainable MEP solution, it is essential to take note of the typical trade-off between energy, accuracy and speed because of the inherent characteristics of the DNN models used in an MEP system [23]–[26]. Therefore, a reduction in the UA's query accuracy is expected, given the improvements in energy and latency. On average, the UA is better only 20% of the time in terms of accuracy, which we can note given how setups are grouped near the 0% mark in Figure 5, which indicates that the UO is not better than most equivalent UOs in these setups. Nonetheless, as we will see next, this highly occurring trade-off in accuracy is not that high.

B. QoS Improvement

Figure 6 shows that the average improvement in QoS of the UA over equivalent UOs is very distinct for the energy and latency, with only a few cases showing some loss in these criteria. On average, the UA reduces the total energy consumption and processing time by 34 Wh and 5.6 seconds, respectively. In the best scenarios, the UA led to a reduction of 1.2 kilowatts-hour and 213 seconds. Furthermore, when using the UA, the average accuracy loss was only 0.29%, reaching a maximum of 3.92% in the worst case. However, it is essential to recognise that in our experiments, we limited the uncertainty in the data to only the energy and speed values, which may cause this small divergence in the accuracy of the UA and UOs.

C. One Year Energy Savings

When extrapolating the experiment results from some hours to one year's worth of processing time, we can better understand the scope of the impact that the uncertainties can have on the QoS of the system, most especially in its ecological

Table I: Savings on UA/UO solutions and reference values of energy consumption

Activity	**Energy (kWh)**
One Data Centre Server (at 100% capacity), 1 year [27]	2100
Uncertainty-Aware (UA) Solution Best Energy Savings, 1 Year	**206**
Electric Vehicle Travel, Amsterdan $\longleftrightarrow$ Paris (1000 Km) † [28]	170
Uncertainty-Oblivious (UO) Solution Best Energy Savings, 1 Year	**139**
Avg. Daily Energy Consumption per Household (adjusted to the climate), EU27+UK (in 2017)‡ [29]	45
Avg. improvement of UA compared to equivalent UO solutions, 1 Year	**11.57**

† Using the energy consumption per Km of a Tesla model 3 (167 Wh/Km), and a route by car traced in Google Maps as a reference.
‡ Daily consumption of the 27 European Union countries and UK, derived from the average yearly consumption value of 16.6 MWh.

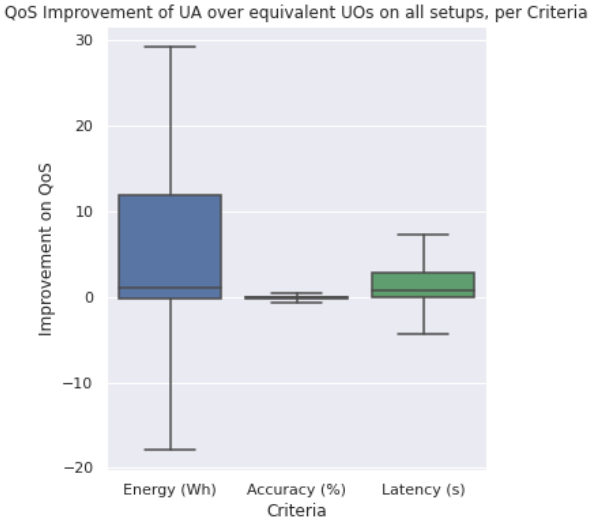

Figure 6: QoS improvement of UA over equivalent UOs on all setups, per criteria. Positive values indicate better QoS in the UA than its 27 equivalent UOs, while negative values indicate cases where a UO had a better QoS.

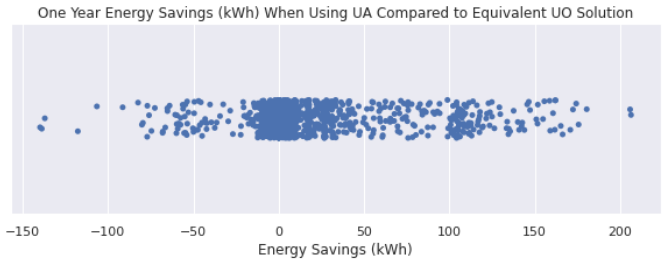

Figure 7: One Year Energy Savings (kWh) When Using UA Compared to Equivalent UO Solution. Each point represents a comparison of the savings of the UA against a UO in the same deployment setup. Negative values are the cases where the UO showed lower energy values than the UA.

sustainability efforts. Figure 7 illustrates the savings in energy (in kilowatt-hours) from using the uncertainty-aware solution compared to the oblivious solutions for all 65 setups. On average, the UA leads to 11.57 kWh of savings, with scenarios with up to 206.4 kWh of savings. In the worst-case scenario, where the uncertainty-oblivious solution outperformed the uncertainty-aware solution, the maximum energy loss was no more than 139.9 kWh.

To give a better perspective, Table I compares the best savings for the UA and UO solutions over one year of processing time against reference values of other energy-consuming activities. We can see that the UA can save more than four times the daily energy consumed per household in Europe in 2017 [29], more energy than that used on a 1000 Km travel with an electric vehicle [28], and up to 9.83% of the annual consumption of a data centre server running at total capacity [27].

VII. Conclusion

In this work, we analysed how the sustainability and other QoS metrics can be significantly affected due to uncertainties arising from the user-defined QoS interpretations and imprecisions in the measurements of service workers in DNN-based Multimedia Event Processing applications when considering real-world settings, such as smart traffic management systems. Our results showed that, on average, using an uncertainty-aware solution for the service selection problem on the adaptation cycle of the MEP system produced better QoS when compared to an uncertainty-oblivious method on 67%, 69% and 20% of the time for energy consumption, latency, and accuracy, respectively. Moreover, after 3 hours of processing, our uncertainty-aware solution already reduces the energy consumption by up to 1.2 kilowatt-hours. It reduces the latency by up to 213 seconds, with a 0.29% average accuracy trade-off in the user's query. Our primary conclusions highlight the advantages of employing an uncertainty-aware service selection approach, especially in real-world applications. This method enhances MEP systems' speed and ecological sustainability while incurring a slight trade-off in accuracy, making it a promising choice. Furthermore, the ecological impact becomes even more pronounced when projecting the extended use of the uncertainty-aware solution over one year, with energy savings exceeding the consumption required for a 1000 Km round-trip from Amsterdam to Paris using an electric vehicle and

almost 10% of the annual consumption of a data centre server at full capacity. As part of our future research, we intend to use public live footage of traffic cameras to produce a realistic dataset, allowing us to also account for the uncertainty in the accuracy of each DNN model when processing each video frame, addressing a current limitation in our work. Additionally, we plan on developing an uncertainty-aware early filtering component to the MEP framework, employing a rule-based fuzzy control system to manage the uncertainty on the real-time monitored data.

REFERENCES

[1] J. Seo, S. Han, S. Lee, and H. Kim, "Computer vision techniques for construction safety and health monitoring," *Advanced Engineering Informatics*, vol. 29, no. 2, 2015.

[2] A. Aslam and E. Curry, "A survey on object detection for the internet of multimedia things (iomt) using deep learning and event-based middleware: Approaches, challenges, and future directions," *Image and Vision Computing*, vol. 106, p. 104095, 2021.

[3] M. Dayarathna, Y. Wen, and R. Fan, "Data center energy consumption modeling: A survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 732–794, 2015.

[4] E. Strubell, A. Ganesh, and A. McCallum, "Energy and policy considerations for deep learning in nlp," *arXiv preprint arXiv:1906.02243*, 2019.

[5] L. Belkhir and A. Elmeligi, "Assessing ict global emissions footprint: Trends to 2040 & recommendations," *Journal of Cleaner Production*, vol. 177, pp. 448–463, 2018.

[6] F. Gand, I. Fronza, N. El Ioini, H. R. Barzegar, S. Azimi, and C. Pahl, "A fuzzy controller for self-adaptive lightweight edge container orchestration," in *Proceedings of the 10th International Conference on Cloud Computing and Services Science-CLOSER*. SciTePress, 2020, pp. 79–90.

[7] A. Karanika, P. Oikonomou, K. Kolomvatsos, and T. Loukopoulos, "A demand-driven, proactive tasks management model at the edge," in *2020 IEEE international conference on fuzzy systems (FUZZ-IEEE)*. IEEE, 2020, pp. 1–8.

[8] S. G. Galán, M. Seddiki, R. J. P. de Prado, E. M. Expósito, A. Marchewka, and N. R. Reyes, "Relevance of using interpretability indexes for the design of schedulers in cloud computing systems," in *2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE, 2020, pp. 1–8.

[9] C.-L. Hwang, K. Yoon, C.-L. Hwang, and K. Yoon, "Methods for multiple attribute decision making," *Multiple attribute decision making: methods and applications a state-of-the-art survey*, pp. 58–191, 1981.

[10] C.-T. Chen, "Extensions of the topsis for group decision-making under fuzzy environment," *Fuzzy sets and systems*, vol. 114, no. 1, pp. 1–9, 2000.

[11] J. O. Kephart and D. M. Chess, "The vision of autonomic computing," *Computer*, vol. 36, no. 1, pp. 41–50, 2003.

[12] M. Masdari and H. Khezri, "Service selection using fuzzy multi-criteria decision making: a comprehensive review," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 2, pp. 2803–2834, 2021.

[13] P. Jamshidi, C. Pahl, and N. C. Mendonça, "Managing uncertainty in autonomic cloud elasticity controllers," *IEEE Cloud Computing*, vol. 3, no. 3, pp. 50–60, 2016.

[14] S. Maheswari and G. Karpagam, "Enhancing fuzzy topsis for web service selection," *International Journal of Computer Applications in Technology*, vol. 51, no. 4, pp. 344–351, 2015.

[15] L. A. Zadeh, "Fuzzy logic, neural networks, and soft computing," in *Fuzzy sets, fuzzy logic, and fuzzy systems: selected papers by Lotfi A Zadeh*. World Scientific, 1996, pp. 775–782.

[16] T. J. Ross, *Fuzzy logic with engineering applications*. John Wiley & Sons, 2005.

[17] F. A. Pontes, M. Schukat, and E. Curry, "Fuzzy vs. crisp in uncertainty-aware service selection: Enabling sustainability on multimedia event processing," in *2023 IEEE International Conference on Fuzzy Systems (FUZZ)*, 2023, pp. 1–7.

[18] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[19] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[20] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[21] E. Curry, D. Salwala, P. Dhingra, F. A. Pontes, and P. Yadav, "Multimodal event processing: A neural-symbolic paradigm for the internet of multimedia things," *IEEE Internet of Things Journal*, 2022.

[22] D. Scheinert, B. S. Z. Aghdam, S. Becker, O. Kao, and L. Thamsen, "Probabilistic time series forecasting for adaptive monitoring in edge computing environments," in *2022 IEEE International Conference on Big Data (Big Data)*. IEEE, 2022, pp. 4583–4588.

[23] N. Kannappan Jayakodi, A. Chatterjee, W. Choi, J. Rao Doppa, and P. Pratim Pande, "Trading-off accuracy and energy of deep inference on embedded systems: A co-design approach," *arXiv e-prints*, pp. arXiv–1901, 2019.

[24] K. Kim, J. Kim, J. Yu, J. Seo, J. Lee, and K. Choi, "Dynamic energy-accuracy trade-off using stochastic computing in deep neural networks," in *2016 53nd ACM/EDAC/IEEE Design Automation Conference (DAC)*. IEEE, 2016, pp. 1–6.

[25] J. Zhang and S. Garg, "Fate: fast and accurate timing error prediction framework for low power dnn accelerator design," in *2018 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. IEEE, 2018, pp. 1–8.

[26] J. Lee, S. Kang, J. Lee, D. Shin, D. Han, and H.-J. Yoo, "The hardware and algorithm co-design for energy-efficient dnn processor on edge/mobile devices," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 67, no. 10, pp. 3458–3470, 2020.

[27] R. I. Dinita, G. Wilson, A. Winckles, M. Cirstea, and A. Jones, "Hardware loads and power consumption in cloud computing environments," in *2013 IEEE International Conference on Industrial Technology (ICIT)*. IEEE, 2013, pp. 1291–1296.

[28] G. Rajendran, C. A. Vaithilingam, N. Misron, K. Naidu, and M. R. Ahmed, "A comprehensive review on system architecture and international standards for electric vehicle charging stations," *Journal of Energy Storage*, vol. 42, p. 103099, 2021.

[29] S. Winchester and S. Oxley, "International comparisons of household energy efficiency of 2017," 2020.