technologies

Review

A Review of Efficient Real-Time Decision Making in the Internet of Things

Kyoung-Don Kang

Department of Computer Science, State University of New York at Binghamton, Binghamton, NY 13902, USA; kang@binghamton.edu

Abstract: Emerging applications of IoT (the Internet of Things), such as smart transportation, health, and energy, are envisioned to greatly enhance the societal infrastructure and quality of life of individuals. In such innovative IoT applications, cost-efficient real-time decision-making is critical to facilitate, for example, effective transportation management and healthcare. In this paper, we formally define real-time decision tasks in IoT, review cutting-edge approaches that aim to efficiently schedule real-time decision tasks to meet their timing and data freshness constraints, review state-of-the-art approaches for efficient sensor data analytics in IoT, and discuss future research directions.

Keywords: Internet of Things; real-time decision making; timing and data freshness constraints; predicate evaluation; real-time scheduling; sensor data analytics

Citation: Kang, K.-D. A Review of Efficient Real-Time Decision Making in the Internet of Things. *Technologies* **2022**, *10*, 12. https://doi.org/10.3390/technologies10010012

Academic Editor: Vijayakumar Varadarajan

Received: 21 December 2021
Accepted: 12 January 2022
Published: 19 January 2022

1. Introduction

IoT envisions to enable many innovative applications, such as smart transportation, healthcare, and emergency response [1–4]. In IoT, timely decision-making using real-time sensor data is essential. For example, drivers in New York, Chicago, and Philadelphia lost 102, 104, and 90 h on average in 2021 despite a −27% to −37% drop since 2019 due to the reduced traffic during the COVID-19 pandemic [5]. Real-time decision-making for efficient traffic routing based on sensor data streams from roadside sensors (if any) or dashboard-mounted smartphones can greatly alleviate traffic congestion [6,7]. Also, an agent for real-time decision-making needs to find an available route among several alternative routes to send an ambulance to a patient when some of them are unavailable because of construction, social/political event, or disaster [8]. As another example, patients in an emergency department or intensive care unit with abnormal shock index values have much higher mortality rates [9] and higher risks to suffer from hyperlactatemia [10] and cardiac arrest [11]. Thus, making real-time triage decisions based on the analysis of physiological sensor data from wearable devices within decision-making deadlines is desirable.

In the presence of alternative actions, a real-time decision-maker needs to select one of them that is currently feasible within decision-making deadlines using fresh sensor data that represent the current real-world status to minimize, for example, traffic congestion or mortality in an emergency department. Furthermore, a real-time decision-maker should require IoT devices to provide minimal sensor data necessary for decision-making only to avoid possible network congestion and significant energy consumption in IoT devices for transmitting redundant sensor data wirelessly. Logic predicates, also called Boolean queries, can effectively evaluate alternative courses of action in IoT [8,12,13]. For example, an ambulance may try to find an available route among several alternative routes to a patient where some of them are unavailable due to construction, a social/political event, or disaster. Let us suppose that there are two alternative routes, A-B-C and D-E-F, which are expressed as $(A \wedge B \wedge C) \vee (D \wedge E \wedge F)$ where $\wedge$ and $\vee$ represent the logical `AND` and `OR` operator, respectively. If road segment A of the route A-B-C is unavailable, the data

indicating the status of the road segment B or C does not have to be retrieved from the sensors and analyzed for real-time decision making, but can be short-circuited to reduce the latency and resource consumption [8,12,13]. Similarly, effective treatment can be selected among alternative treatments by efficiently analyzing the logic predicate in a timely manner using fresh data that represent the current status of the patients in an emergency department or intensive care unit. In the rest of this paper, we use emergency vehicle routing and triage/treatment as our running examples for real-time decision support.

There are many reviews of the general area of IoT, including (but not limited to) [1–4]. In this paper, we review cutting-edge research on efficient real-time decision support by analyzing logic predicates in a timely fashion using fresh sensor data in IoT. Instead of being exhaustive, we focus on systematic approaches for efficient processing of real-time decision tasks that aim to meet stringent timing constraints (i.e., deadlines) and data freshness requirements of the tasks for real-time decision support in IoT. We take this approach because it is essential to meet stringent timing and data freshness constraints for real-time decision support in IoT. For example, the real-time decision task to route a vehicle should complete within the deadline before the vehicle passes the next exit using fresh data that represent the current traffic status. Otherwise, the vehicle may miss the exit or make an ineffective decision using stale data. In the same vein, we do not review techniques for *near* real-time decision support or data analytics in IoT, such as [14–21], that do not consider explicit timing and data freshness constraints. They are agnostic to timing and data freshness constraints and only aim to decrease the average latency or increase the average throughput without providing any timing or freshness assurance critical in real-time decision making in IoT [8,12,13,22–24]. Therefore, we have chosen papers published in top-tier real-time conferences and journals that aim to schedule real-time decision tasks to efficiently meet the stringent timing and freshness constraints for real-time decision support in IoT. We have avoided search via keyword-based queries, such as "real-time decision making" and "IoT" because most papers returned by such queries are near real-time at best.

Recently, a set of pioneering works, such as [8,12,13,22–24], has been done to efficiently support real-time decision-making in IoT using fresh sensor data. More specifically, they aim to efficiently evaluate logic predicates that model alternative courses of action [8,12,13] and to effectively schedule decision making tasks [22–24]. The field of research on real-time decision making in IoT, however, is in an early stage and relatively little work has been done to review the area [25], even though closely related areas that form a basis for real-time decision making, such as wireless networking for IoT [6,26–29] and sensor data analytics via machine learning [30–32], have been reviewed extensively.

To bridge the gap, in this paper, we review state-of-the-art approaches for real-time decision-making in IoT and other closely related topics in terms of their strengths and limitations. A summary of our key contributions follows.

- We define real-time decision tasks in IoT that intend to evaluate logic predicates within their deadlines using fresh sensor data. In this way, we clearly distinguish them from near real-time approaches agnostic to timing and data freshness constraints.
- We review leading-edge scheduling methodologies for efficient processing of real-time decision tasks in IoT by thoroughly analyzing their advantages and disadvantages while reviewing effective machine learning techniques that can be leveraged by real-time decision tasks.
- Furthermore, we propose future research directions to meet the timing and data freshness constraints of real-time decision tasks in IoT more cost-efficiently.

In Section 2, we give background for real-time decision making in IoT and define real-time decision tasks in IoT. In Section 3, we review state-of-the-art approaches for efficient predicate evaluation, freshness management of the sensor data, and real-time analytics of sensor data in IoT. In Section 4, we discuss future research directions. Finally, Section 5 concludes the paper.

2. Background

In this paper, we focus on event-driven sensing and data analysis for efficient real-time decision making based on the ECA (Event-Condition-Action) model depicted in Figure 1 using IoT devices equipped with sensors and a wireless communication module (In this paper, we mainly discuss real-time decision making using wireless IoT devices that are easier to deploy in a distributed area. However, the techniques for efficient real-time decision making reviewed in this paper are applicable to decision support using wired sensors too, without loss of generality.). Sensors normally do *not* transfer data to the real-time decision-maker. Instead, a sensor streams data into the real-time decision-maker only when an event of interest occurs to avoid wireless data transfer that consumes precious bandwidth and energy unnecessary for real-time decision making. In this paper, we employ a *comprehensive definition of events*: an event is anything noteworthy in terms of enhancing the quality of life or safety and efficiency of societal infrastructure. An event can be triggered by any sensor, software agent, or user that forms a real-time decision-making system in IoT. For example, a surveillance camera begins to send a video stream to the real-time decision-maker when the motion sensor detects a motion. For cost-effective triage, a patient at an emergency department or an intensive care unit can be continuously monitored using a wearable device. The wearable device triggers an event and reports to the real-time decision-maker when the shock index, SI $= \frac{\text{heart rate}}{\text{systolic blood pressure}}$, of the patient exceeds the threshold, such as 0.9 [9]. Also, certain cameras send images of the specific road segments that they monitor when the decision-maker begins route planning and requests data from them. Given fresh sensor data, the real-time decision-maker needs to make decisions within the specified deadlines to support the IoT application, such as real-time traffic control or triage.

Figure 1. Real-time decision making in IoT based on the Event-Condition-Action model.

In this paper, we focus on the problem of efficiently evaluating logic predicates where a predicate represents the availabilities or feasibilities of alternative courses of action in IoT, such as alternative routes or treatments. In this paper, we assume that predicates for decision-making are in disjunctive normal form (DNF), which is a canonical normal form, without loss of general applicability. Because any predicate can be converted to an equivalent predicate in DNF [33], our discussions in this paper apply to any logic formula for real-time decision making. Let us use $\wedge$ and $\vee$ to represent the logical *and* and *or* operator, respectively. Given that, a DNF predicate P is a disjunction of one or more conjunctions of literals (Boolean variables):

$$P = C_1 \vee \ldots \vee C_n = (C_{1,1} \wedge \ldots \wedge C_{1,m_1}) \vee \ldots \vee (C_{n,1} \wedge \ldots \wedge C_{n,m_n}) \quad (1)$$

where C_i $(1 \leq i \leq n)$ is a conjunction of $m_i \geq 1$ literals; that is, $C_i = (C_{i,1} \wedge \ldots \wedge C_{i,m_i})$ and C_{ij} is a Boolean variable whose value is either true or false. For example, a DNF predicate $P = (A \wedge B \wedge C) \vee (A \wedge D \wedge E) \vee (F \wedge G \wedge H)$ may represent the availabilities of alternative routes A-B-C, A-D-E, or F-G-H or feasibilities of alternative medical treatments.

A DNF predicate is useful when a decision-maker aims to find one of the alternative solutions that are currently available/feasible in a timely manner. For example, if the conjunction $(A \wedge B \wedge C)$ in the DNF predicate P above evaluates to true before the other conjunctions in P, the route A-B-C is returned as a solution without further evaluating P, if necessary, to meet the deadline for real-time decision making. On the other hand, if the condition A is false, $(A \wedge B \wedge C)$ and $(A \wedge D \wedge E)$ in P immediately become false, via short-circuiting. In IoT, a real-time decision-maker, such as the traffic controller in a city or triage agent in an emergency department, can leverage short-circuiting to avoid unnecessary transfer and analysis of sensor data. When A is false in the previous example,

IoT devices do not need to transfer sensor data for the road segments B, C, D, and E to the real-time decision-maker, because the first two conjunctions in the predicate are already false. The decision-maker does not have to analyze them to evaluate the entire predicate P, either. Instead, it can focus on evaluating the remaining conjunction of P, i.e., $(F \wedge G \wedge H)$.

Furthermore, machine learning is an important building block for real-time decision making in IoT. For example, a real-time decision-maker can analyze the images of road segments A, B, and C, via deep learning, to tell if the route A-B-C is available, which is represented by the conjunction $(A \wedge B \wedge C)$ in P.

For the clarity of the presentation, we formally define real-time decision tasks in IoT as follows.

Definition 1 (Real-Time Decision Tasks). *A real-time decision maker in IoT has a set of n ($\geq$1) real-time decision tasks, $\tau = \{\tau_1, \ldots, \tau_n\}$ that are triggered on demand. A real-time decision task $\tau_i \in \tau$ is associated with the relative deadline D_i: if it is triggered at time t, it must complete by the absolute deadline, $t + D_i$, to meet its timing constraint. When triggered at time t, τ_i must retrieve and analyze a set of sensor data objects $S_i = \{O_{i,1}, \ldots, O_{i,n_i}\}$, where $n_i = |S_i|$ (the cardinality of the set S_i), to evaluate its predicate P_i and choose one of the alternative solutions expressed in P_i for decision making by $t + D_i$. To manage the freshness of the sensor data, each data object $O_{i,j} \in S_i$ $(1 \leq j \leq n_i)$ is associated with the absolute validity interval $avi(O_{i,j})$. $O_{i,j}$ is considered fresh at time $t' (\geq t)$ if it was updated (retrieved) at time t and $t' \leq t + avi(O_{i,j})$ [34,35]; that is, its validity interval has not expired yet. Otherwise, it is considered stale.*

3. A Review of Techniques for Cost-Efficient Real-Time Decision Support in IoT

In this section, we discuss state-of-the-art approaches that aim to efficiently process real-time decision tasks, while meeting their timing and data freshness constraints as per Definition 1. More specifically, we review state-of-the-art approaches to efficient processing of real-time decision tasks via short-circuiting, scheduling of real-time decision tasks to meet timing and data freshness constraints, and sensor data analytics via machine learning for real-time decision support in IoT. In addition, the outline of our review is shown in Figure 2.

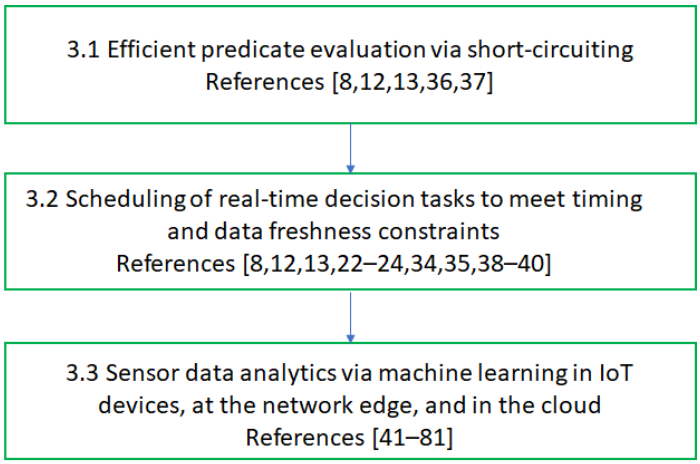

Figure 2. Flow of our review of state-of-the-art methods for real-time decision support in IoT.

3.1. Efficient Evaluation of a Single Conjunction via Short-Circuiting

Efficiently evaluating one conjunction in a computer program via short-circuiting is a well-established technique [36,37]. Previous studies [36,37] prove that short-circuiting is optimal in terms of the computational cost for evaluating a single conjunction, e.g., $A \wedge B \wedge C$. Based on the theoretic results [36,37], a series of novel works [8,12,13] have recently explored how to efficiently evaluate predicates for real-time decision-making in IoT via short-circuiting. Given an arbitrary conjunction $C_i = C_{i1} \wedge C_{i2} \wedge \ldots$ that represents *a single action*, the common approach presented in [8,12,13] evaluates the condition in C_i with

the highest short-circuit probability per unit cost first, where the cost is the consumption of the bottleneck resource, such as the communication bandwidth. If $C_{ij} \prec C_{ik}$ denotes that the condition C_{ij} precedes the condition C_{ik} in the order of evaluating the literals in the conjunction C_i, the heuristic presented in [8,12,13] requires the following:

$$C_{ij} \prec C_{ik} \Leftrightarrow \frac{1 - P_{ij}}{cost_{ij}} \prec \frac{1 - P_{ik}}{cost_{ik}} \quad (2)$$

where P_{ij} is the probability of the jth literal in C_i to be true. In addition, $cost_{ij}$ is the cost for retrieving the sensor data needed to evaluate the jth literal, such as the latency to retrieve the corresponding sensor data over a wireless connection. By doing this, this heuristic can considerably reduce the cost for evaluating one conjunction in a DNF predicate via short-circuiting; however, it has several limitations:

- In [8,12,13], the authors only consider how to evaluate a single conjunction via short-circuiting [8,12,13] without investigating how to efficiently evaluate the entire DNF predicate that is a conjunction of one or more conjunctions representing alternative courses of action.
- They implicitly assume that the conjunction evaluation scheme has a priori knowledge of short-circuit probabilities for efficient evaluation of the conjunction based on history [8,12,13]. However, they do not discuss how to derive the short-circuit probabilities. Estimating the probabilities may incur additional sensor data retrievals. If the accurate short-circuit probabilities are unavailable *a priori* or the cost for probability estimation is not negligible, the greedy heuristic that orders the literals in a conjunction via Equation (2) for efficient real-time decision making via short-circuiting [8,12,13] may become ineffective.

3.2. Pull Model and Data Freshness

In [8,12,13], the real-time decision-maker employs the pull model, in which it pulls (retrieves) data from sensors over a single wireless connection upon an event of interest to analyze, for example, the availabilities of alternative routes. To make decisions based on fresh data representing the current real-world status, the real-time decision-maker in [8,12,13] periodically retrieves sensor data based on their validity intervals—the notion originated in real-time databases (RTDBs) [34,35]. A sensor data object is fresh within its predefined validity interval; however, the real-time decision-making system considers it stale after the validity interval expires. By doing this, the system ensures that it makes real-time decisions based on fresh data representing the current real-world status.

Although managing the data freshness (data temporal consistency) via validity intervals could be effective in RTDBs with its own sensors, it can be too strict and expensive in IoT. First, sensor data, such as indoor temperature readings, may not normally change significantly in a short time period. Thus, the data could be still valid even after its validity interval expires. Periodic updates even in the absence of any noteworthy change may incur unnecessary consumption of the precious wireless bandwidth and energy in IoT devices without enhancing real-time decision making.

Moreover, if a decision-making task uses several sensor data with different validity intervals, the real-time decision-maker may have to retrieve the data repeatedly to ensure that all of them are fresh until the decision task completes. The system also should undo and redo any analysis performed using stale data. Hu et al. [8] investigate this problem for a single decision task that uses sensor data pulled over a wireless connection. Their algorithm, called the LVF (Least Volatile First), pulls the data with the longest validity interval first. By doing this, LVF minimizes repeated data retrievals for one decision task that pulls sensor data with different validity intervals.

Kim et al. [22,23] extend LVF to schedule multiple real-time decision-making tasks with potentially different deadlines using fresh data. Their algorithm, called EDEF-LVF (Earliest Deadline or Expiration First-Least Volatile First), schedules the real-time task with

the earliest deadline or the shortest time to the expiration of the validity interval first. Within each task, the least volatile sensor data is retrieved first, similar to [8]. They assume that there is a single bottleneck resource, such as a wireless connection, and real-time tasks do not share any data. Under the assumptions, EDEF-LVF is optimal in the sense that it can schedule real-time decision-making tasks to meet their deadlines and data validity constraints if such a schedule exists. In addition, Kim et al. [24] devise several suboptimal heuristics to efficiently schedule real-time decision-making tasks that share sensor data with each other.

However, none of these approaches [8,12,13,22–24] is free of repeated sensor data retrievals and re-executions of data analytics upon expiration of any validity interval. As a result, the precious wireless bandwidth and energy of IoT devices can be wasted and many deadlines for real-time decision making can be missed. In an extreme case, it may become impossible to run a task using fresh data as per the strict notion of validity intervals. For the sake of simplicity, let us suppose that there is only one real-time task that needs to pull data A and B from sensors deployed in a wide area over a wireless connection with relatively low bandwidth. Using LVF, the task pulls A with the longer validity interval first. When it tries to pull B, however, the wireless connection may become unstable. As a result, the sensor should retransmit B several times. Meanwhile, the validity interval of A expires. By the time a new version of A arrives, the validity interval of B may expire, and the whole process may repeat indefinitely. Finally, the system misses the deadline of the real-time decision-making task, wasting the bandwidth and energy. If there are multiple real-time decision-making tasks in the system, the problem may become worse. In addition to the situations described above, a real-time task can be preempted by a higher priority task, such as a task with an earlier deadline under the EDF (Earliest Deadline First) scheduling algorithm. When all higher priority tasks are completed, the preempted task may have to pull certain sensor data again, if their validity intervals have expired already.

The root cause of the problem is using the rigid freshness requirements based on data validity intervals. Surprisingly little work has been done to address this critical issue for cost-efficient real-time decision-making in IoT. A viable way to address the problem is the adaptive updated policy based on flexible validity intervals [38–40]. Instead of using fixed validity intervals, the validity intervals of sensor data are dynamically adapted based on their access to update ratio in RTDBs such that the validity intervals of the data updated frequently but accessed infrequently are extended, if necessary, to reduce update workloads under overload [38–40]. The notion of flexible validity intervals can be extended to efficiently manage the data freshness for real-time decision-making in IoT. Instead of requiring the real-time decision-maker to pull data from IoT devices, *sensors start to push data into the decision-maker when they detect an event of interest*, e.g., a moving object in surveillance or traffic congestion in transportation management. After sending the first sensor readings to the decision-maker upon an event, the sensors only send new data if they differ from the previous version by more than the specified threshold. They periodically send a heartbeat message to the real-time decision-maker to indicate that they are still alive and monitoring the event of interest, even though they have not transferred new data to the decision-maker due to little changes. When the decision-maker receives a heartbeat message from a device, it extends the flexible validity interval to the next heartbeat period. On the other hand, when the sensor data changes by more than the threshold, the device sends new data to the decision-maker. By doing this, the decision-maker can avoid significantly wasting the network bandwidth, computational resources, and energy to repeatedly pull sensor data from IoT devices due to the expirations of strict validity intervals even when the actual data values hardly change.

3.3. Sensor Data Analytics via Machine Learning for Real-Time Decision Making

Machine learning is effective to analyze sensor data. For example, the availability of a bridge or a road segment can be analyzed by a CNN (Convolutional Neural Network) [41], which is very effective for image processing and computer vision. Thus, machine learning

is useful to evaluate the literals of a DNF predicate for real-time decision support. Sequence models are also useful for real-time decision support in IoT. For example, Markov decision processes [42] and partially observable Markov decision processes [42] are leveraged for near real-time health monitoring, treatments, and interventions in various medical applications [43]. More recently, long-short term memory (LSTM), which is an artificial recurrent neural network (RNN) architecture effective for sequence modeling, has been applied to detect emotion [44], to predict cardiovascular disease risk factors [45], and to predict healthcare trajectories [46]. Machine learning is applied to smart homes [47–49]. Guo et al. have designed a graph CNN optimized for traffic predictions [50]. In [51–54], GRNN (General Regression Neural Network) and GRNN-SGTM (GRNN-Successive Geometric Transformation Model) are used to recover missing IoT data, respectively. Wang et al. [55] devise a GRNN and a multivariate polynomial regression model to estimate unmeasurable water quality parameters from measurable parameters. In addition, Tien [56] gives a high-level view of IoT, (near) real-time decision making, and artificial intelligence instead of focusing on technical approaches for real-time decision support in IoT, unlike this review.

Although it is effective for data analytics, machine learning is resource hungry. A complex machine learning model often consumes a significant amount of memory and computational resources, such as CPU cycles and GPU (Graphics Processing Unit) thread blocks, that may not be available in IoT devices with relatively little resources. Thus, in IoT devices, it is hard to run sophisticated prediction models in a timely manner to meet stringent timing constraints. A naive approach to address this challenge is transferring all sensor data from IoT devices to the cloud with virtually infinite resources. However, this approach is unsustainable, as described before. Therefore, the question of "where to analyze sensor data?" is as important as the question of "how to analyze them efficiently?". Ultimately, it is desirable to optimize the tradeoff between the timeliness and bandwidth conservation of real-time data analytics near IoT devices vs. the scalability of data analytics in the cloud. In this regard, we summarize the relative advantages and disadvantages of sensor data analytics in IoT devices, at the network edge, and the cloud in Table 1, and discuss them in the following.

Table 1. Comparisons of real-time decision-making at different places.

	Cloud	Edge	IoT End-Devices
Resources	High	Medium	Low
Latency	High	Medium	Low
Bandwidth consumption	High	Medium	Low
Energy consumption	High	Medium	Low
Geographic coverage	High	Medium	Low

The first category is centralized analytics of sensor data in the cloud. A cloud has abundant computational resources and provides rich functionalities, such as very deep learning with many layers and training complex machine learning models using big datasets. Another advantage of real-time analytics in the cloud is that it can support real-time data analytics in a more global geographic area. However, centralized data analytics for real-time decision making in the cloud has several serious drawbacks:

- It requires IoT devices to transmit all sensor data to the cloud for analytics, incurring long, unpredictable latency, and many deadlines miss in real-time decision making. (The Internet backbone latency is relatively long and varies significantly from tens to hundreds of milliseconds [57].) Tardy decisions may lead to undesirable results, such as severe traffic congestion or chaos in an emergency department.
- Such a naive approach may saturate the core network with the limited bandwidth as the number of sensors and IoT devices is increasing rapidly [58,59]. It may substantially impair the performance, scalability, and availability of the Internet. Thus, centralized analytics of sensor data in IoT is unsustainable.

- In addition, IoT devices may consume a lot of precious energy and wireless bandwidth to transfer all their sensor data to the cloud for centralized data analytics in the cloud. Typically, IoT devices communicate wirelessly for the ease of deployment in a distributed area. Wireless networking consumes a significant fraction of the energy in an IoT device [60,61]. Wireless IoT networks, such as LPWAN (Low-Power Wide-Area Network) [62,63], often have stringent bandwidth constraints.

To address these problems, a system designer can consider another extreme—on-device analytics, where all data analytics occur in IoT end-devices. By supporting distributed analytics of sensor data, this approach can significantly reduce the latency and bandwidth consumption compared to the centralized analytics in the cloud. However, this approach also has several challenges:

- It is challenging to meet stringent timing constraints for real-time data analytics and decision support due to the stringent resource and energy constraints of IoT devices.
- IoT devices with limited resources may not be able to support sophisticated machine learning models or extensive model training. Instead, they typically use simplified models trained in the cloud to analyze local sensor data in a timely fashion [64,65]; however, the stripped-down model may suffer from lower predictive performance.
- Each IoT device is likely to have a relatively myopic view of the specific area it is monitoring only without a global view necessary to optimize, for example, the overall traffic flow in a city.

By analyzing sensor data at the network edge near IoT devices and sensors, edge analytics [66–69] aims to integrate the advantages of cloud and on-device analytics, while mitigating their shortcomings. Edge computing brings more computational resources at the network edge near data sources. It can be supported at different places. First, IoT end devices can preprocess sensor data and perform lightweight analytics [70–72]. Second, an edge node, such as an IoT gateway, access point, cellular base station, or software-defined routers/switches, can collect and analyze data from IoT devices [71,73–75]. Edge servers deployed at the network edge can be leveraged for more sophisticated data analytics [68].

Thus, edge analytics for real-time decision support can be performed in a *hierarchical and event-driven* manner. An IoT device preprocesses sensor data and performs a lightweight analysis of them to detect any event of interest while filtering irrelevant data out. An IoT gateway, if any, further analyzes data received from the devices connected to the gateway. It forwards important information, if any, to one or more relevant edge servers. For example, traffic cameras can send images to the edge server in charge of monitoring traffic flows in a specific area of a big city. Li et al. [76], on-camera filtering is performed for efficient real-time video analytics. In [77], an IoT camera analyzes the traffic flow using a low-resolution image and the edge server also analyzes the image, identifies an important part of the image (if any) in terms of data analytics, and requests an important part in high resolution from the device. IoT devices in a smart building can transfer their sensor readings to the IoT gateway on the same floor for efficient HVAC (Heating, Ventilation, and Air Conditioning). In these examples, IoT devices can do relatively simple data analytics to drop redundant or low-quality data, such as blurry images [76]. Edge servers analyze real-time sensor data from multiple IoT devices/gateways to derive a more comprehensive view of the real-world status essential for real-time decision making. They can also communicate with each other to exchange information for a global view of real-world situations, such as the overall traffic flow in a city or hurricane paths in a nation. Edge computing and analytics are a booming area of research and industrial adoption due to their significant potential. Leveraging emerging edge computing for cost-efficient real-time decision support is in an early stage of research with ample room to grow.

Overall, efficient evaluations of predicates are important across IoT devices, gateways, and edge/cloud servers to significantly reduce latency as well as energy and bandwidth consumption. The efficiency of real-time decision-making can also be further enhanced by effectively exploiting cloud, on-device, and edge analytics frameworks and synthesizing them to optimize timing, predictive performance, bandwidth, and other resource consump-

tion. Relatively little work, however, has been done for real-time decision-making in IoT from this holistic, overarching perspective.

Another promising direction for real-time analytics of sensor data on IoT devices is model compression [78–81]. The key idea of model compression is to compact a machine learning model to minimize the resource requirements without significantly reducing the predictive performance of the compressed model. Especially, deep learning has been very successful and outperformed other machine learning techniques in killer applications, such as computer vision and natural language processing. DNNs (Deep Neural Networks) with many hidden layers and parameters, however, consume a lot of memory, computation time, and energy. They are too big and too expensive to learn on low-end IoT devices. The motivation for model compression is to significantly reduce the memory consumption and computational complexity of DNNs without significantly comprising their accuracy. Effective approaches for model compression include (1) compact models, (2) tensor decomposition, (3) data quantization, and (4) network sparsification [78]:

- Compact CNNs (Convolutional Neural Networks) are created by leveraging the spatial correlation within a convolutional layer to convolve feature maps with multiple weight kernels (Compact RNNs (Recurrent Neural Networks) for sequence data analysis has also received significant attention from researchers [78].). They also leverage the intra-layer and inter-layer channel correlations to aggregate feature maps with different topologies. In addition, network architecture search (NAS) aims to automatically optimize the DNN architecture.
- Tensor/matrix operations are the basic computation in neural networks. Thus, compressing tensors, typically via matrix decomposition, is an effective way to shrink and accelerate DNNs.
- Data quantization decreases the bit width of the data that flow through a DNN model to reduce the model size and save memory while simplifying the operations for computational acceleration.
- Network sparsification attempts to make neural networks sparse, via weight pruning and neuron pruning, instead of simplifying the arithmetic via data quantization.

Model compression in hardware, as well as hardware and algorithm co-design, is also effective. Good surveys are given in [78,79].

4. Future Research Directions

In this section, we discuss research issues for significantly enhancing the cost-effectiveness of real-time decision-making in IoT in the future.

4.1. Efficient Analysis of an Entire DNF Predicate Requiring No Knowledge of Short-Circuit Probabilities

In this subsection, we discuss how to efficiently evaluate an entire DNF predicate that is a disjunction of one or more conjunctions without requiring a priori knowledge of short-circuit probabilities.It is important to analyze the entire DNF predicate to find one of the alternative solutions, e.g., one of the alternative routes, as fast as possible to meet more decision-making deadlines, while short-circuiting infeasible options. However, refs. [8,12,13] only consider the efficient processing of one conjunction, without considering the efficient processing of the entire predicate. Another drawback is that they assume that the short-circuit probabilities, which may not be available, are known *a priori* as discussed before. Therefore, research on the efficient analysis of a complete DNF predicate without requiring a priori knowledge of short-circuit probabilities is necessary.

To this end, we outline a fundamental approach in Algorithm 1. We propose to build a hash table H offline to efficiently look up which literals in the predicate P in Equation (1) are associated with a specific sensor data X. Thus, when sensor data X arrives at runtime, a hash table lookup $H(X)$ returns a set of literals $S = H(X)$ that depend on X as shown in lines 2–4 of Algorithm 1. For example, if sensor data X (e.g., the image of a road segment) is used to evaluate two Boolean literals $C_{1,1}$ and $C_{2,3}$, $H(X) = \{C_{1,1}, C_{2,3}\}$. In lines 5–18, we

retrieve C_{ij}, which is the first literal in the set S. If C_{ij} evaluates to true, we replace C_{ij} with true in the conjunction of $C_i \in P$ that includes C_{ij}.

Algorithm 1: Efficient DNF Predicate Evaluation for Real-Time Decision Making in IoT

input : Predicate $P = (C_1 \vee \ldots \vee C_n)$, Hash table H of sensor data
output: C_i that is the first conjunction that evaluates to true (null if P is unstasfiable)

```
1  S = ∅
2  if sensor data X arrives then
3      /* Hash table lookup */
4      S = H(X) where S = {C_ij} and X = C_ij ∈ P ∀i, j
5  for k = 1; k ≤ |S|; k++ do
6      C_ij = delete_first_element(S)
7      if C_ij == true then
8          /* Find one of possible alternative solutions */
9          Replace C_ij ∈ C_i with true
10         if C_i == true then
11             /* return a solution */
12             return C_i
13     else
14         /* Short-circuiting: eliminate any invalid conjunctions */
15         P = P − C_i
16         if P = ∅ then
17             /* no solution */
18             return null
```

If C_i becomes true after the replacement, Algorithm 1 returns C_i and terminates. On the other hand, if C_{ij} evaluates to false, we short-circuit the conjunction C_i, which has just become false, and repeat lines 5–18. Overall, Algorithm 1 finds one of the alternative solutions (if any) in the predicate and short-circuits invalid conjunctions quickly requiring no knowledge of short-circuit probabilities.Based on this fundamental method, further research in the future is necessary to support this approach efficiently in a timely, decentralized fashion. For example, distributed hash tables can be leveraged to support efficient lookups of sensor data and predicate evaluations in a distributed IoT application.

4.2. Predicting Probabilities of Satisfying Conjunctions and Two-Level Scheduling for Efficient Evaluation of an Entire Predicate

Recent work [8,12,13] based on the theoretic results [36,37] assumes that the short-circuit probabilities are known based on history without discussing how to derive/estimate them. Neither do they consider which conjunction in a DNF predicate should be evaluated first to further enhance the efficiency of real-time decision support.

To address these issues, an effective technique for query optimization in databases, called Eddies [82], can be applied. An Eddy gives incoming data to a query operator randomly picked from a set of compatible operators in the query. It gives a credit to an operator when it gives input data to the operator but takes the credit back from the operator if it returns any data as a result. Thus, more selective operators accumulate more credits over time. When a new tuple arrives, the Eddy assigns it to the operator with the highest credit. In this way, Eddies favor more selective operators to accelerate query processing. To apply Eddies to real-time decision support in IoT, when sensor data X arrives, the real-time decision-maker randomly picks any $C_{ij} \in S$ in Algorithm 1. By repeatedly doing this

over time, it learns which conjunction C_i in the predicate P can be met with the highest probability and which literal C_{ij} in C_i may have the highest short-circuit probability.

Given that, we can efficiently evaluate conjunctions in the predicate via two-level scheduling. The real-time decision maker evaluates the conjunction with the highest probability to be met first. As Algorithm 1 immediately returns the conjunction once it evaluates to true without further evaluating the other conjunction, this approach can significantly reduce the latency and resource consumption for real-time decision making. In turn, the decision-maker evaluates the literal in the selected conjunction that has the highest short-circuit probability to further reduce the latency and resource consumption. For example, let us consider a DNF predicate $P = (A \wedge B \wedge C) \vee (D \wedge E \wedge F)$ that is a disjunction of the two conjunctions that represent the route A-B-C and D-E-F, respectively. If the first conjunction $A \wedge B \wedge C$ has a higher probability to be met, $P(A) \times P(B) \times P(C)$, the real-time decision-maker will evaluate it first to minimize the latency and resource consumption for real-time decision support in IoT. The decision-maker then evaluates B in $A \wedge B \wedge C$ first, if it has the highest short-circuit probability among A, B, and C; that is, $1 - P(B) = max[1 - P(A), 1 - P(B), 1 - P(C)]$. Thus, it is important to investigate a cost-effective design and implementation of this approach and explore more advanced techniques in the future.

4.3. Efficient Management of Sensor Data Freshness

Managing the freshness of sensor data based on the strict notion of validity intervals [34,35] may force the real-time decision-maker to repeatedly pull (retrieve) data from sensors and restart the same decision masking tasks from the beginning, which may result in many deadline misses and waste of resources, as discussed in Section 3. There are several directions for future research to address these issues, including (but not limited to) the following ones. First, it is necessary to explore a more efficient update model, such as the push model, where data sources take control for data updates and transfer, or a hybrid of the push and pull model to minimize unnecessary data transfer and processing in terms of real-time decision support. Second, it is important to investigate more flexible metrics used to measure and manage the freshness of sensor data, such as flexible validity intervals [38–40], and extend them for efficient real-time decision support in IoT with a formal assurance of data freshness. It is also necessary to investigate cost-effective methods to ensure that a set of sensor data used by a real-time analytics task is fresh simultaneously to avoid updating them and re-executing the task again whenever one of them becomes stale before the task completes.

4.4. Scheduling Real-Time Analytics Tasks

Tasks for real-time analytics may need to be scheduled and processed in a geographically distributed manner. For example, to monitor traffic flows and detect any incidents, real-time data analytics tasks can be distributed across roadside IoT devices, dash-mounted smartphones, edge servers, and cloud. A critical research question is how to place the operations/functions for distributed real-time analytics to minimize the latency and resource consumption, while providing informative decision support based on fresh data reflecting the current real-world status. It is also important to investigate how to schedule analytics tasks within each device or server to reduce the deadline miss ratio and resource consumption, while collaborating with the global scheduling scheme discussed above to enhance the overall timeliness, scalability, and cost-effectiveness of real-time decision support in IoT. Although the underlying mechanisms, such as edge computing and model compression for on-device analytics, have recently received a lot of attention, much more research is necessary for holistic optimization of distributed real-time decision support.

5. Conclusions

Efficient real-time decision support is essential in IoT emerging applications, such as smart transportation and health, with significant societal impact. Especially, it is important

to minimize the latency and resource consumption for effective real-time decision support, via machine learning and logic predicate evaluation, using fresh sensor data reflecting the current real-world status. In this paper, we formally define real-time decision tasks in IoT in terms of their predicates, timing constraints, and data freshness requirements. Based on the definition of real-time decision tasks, we review leading-edge approaches that schedule real-time decision tasks to efficiently meet their timing and data freshness constraints, state-of-the-art approaches for sensor data analytics via machine learning, and advanced techniques to support efficient real-time data analytics in IoT devices, at the network edge, and in the cloud. Moreover, we propose future research directions to meet timing and freshness constraints of real-time decision tasks cost-efficiently. Despite the importance, research on real-time decision support in IoT considering explicit timing and data freshness constraints is still in an early stage with many open research issues, including the issues discussed in this paper.

Funding: This research was funded by National Science Foundation: CNS-2007854.

Acknowledgments: This work is supported, in part, by NSF grant CNS-2007854.

Conflicts of Interest: The author declares no conflict of interest. The funding sponsor had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, and in the decision to publish the results.

References

1. Lee, I.; Lee, K. The Internet of Things (IoT): Applications, investments, and challenges for enterprises. *Bus. Horiz.* **2015**, *58*, 431–440. [CrossRef]
2. Sisinni, E.; Saifullah, A.; Han, S.; Jennehag, U.; Gidlund, M. Industrial internet of things: Challenges, opportunities, and directions. *IEEE Trans. Ind. Inform.* **2018**, *14*, 4724–4734. [CrossRef]
3. Stoyanova, M.; Nikoloudakis, Y.; Panagiotakis, S.; Pallis, E.; Markakis, E.K. A survey on the internet of things (IoT) forensics: Challenges, approaches, and open issues. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 1191–1221. [CrossRef]
4. Fortino, G.; Savaglio, C.; Spezzano, G.; Zhou, M. Internet of Things as System of Systems: A Review of Methodologies, Frameworks, Platforms, and Tools. *IEEE Trans. Syst. Man Cybern. Syst.* **2020**, *51*, 223–236. [CrossRef]
5. INRIX 2021 Global Traffic Scorecard. Available online: https://inrix.com/scorecard/ (accessed on 10 January 2022).
6. Ji, B.; Zhang, X.; Mumtaz, S.; Han, C.; Li, C.; Wen, H.; Wang, D. Survey on the internet of vehicles: Network architectures and applications. *IEEE Commun. Stand. Mag.* **2020**, *4*, 34–41. [CrossRef]
7. Yang, F.; Wang, S.; Li, J.; Liu, Z.; Sun, Q. An overview of Internet of vehicles. *China Commun.* **2014**, *11*, 1–15. [CrossRef]
8. Hu, S.; Yao, S.; Jin, H.; Zhao, Y.; Hu, Y.; Liu, X.; Naghibolhosseini, N.; Li, S.; Kapoor, A.; Dron, W.; et al. Data Acquisition for Real-Time Decision-Making under Freshness Constraints. In Proceedings of the IEEE Real-Time Systems Symposium, San Antonio, TX, USA, 1–4 December 2015.
9. Kim, S.Y.; Hong, K.J.; Shin, S.D.; Ro, Y.S.; Ahn, K.O.; Kim, Y.J.; Lee, E.J. Validation of the Shock Index, Modified Shock Index, and Age Shock Index for Predicting Mortality of Geriatric Trauma Patients in Emergency Departments. *J. Korean Med. Sci.* **2016**, *31*, 2026–2032. [CrossRef]
10. Berger, T.; Green, J.; Horeczko, T.; Hagar, Y.; Garg, N.; Suarez, A.; Panacek, E.; Shapiro, N. Shock Index and Early Recognition of Sepsis in the Emergency Department: Pilot Study. *West. J. Emerg. Med.* **2013**, *XIV*, 168–174. [CrossRef] [PubMed]
11. Kennedy, C.E.; Turley, J.P. Time series analysis as input for clinical predictive modeling: Modeling cardiac arrest in a pediatric ICU. *Theor. Biol. Med. Model.* **2011**, *8*. [CrossRef]
12. Abdelzaher, T.F.; Amin, M.T.A.; Bar-Noy, A.; Dron, W.; Govindan, R.; Hobbs, R.L.; Hu, S.; Kim, J.; Lee, J.; Marcus, K.; et al. Decision-Driven Execution: A Distributed Resource Management Paradigm for the Age of IoT. In Proceedings of the IEEE International Conference on Distributed Computing Systems, Atlanta, GA, USA, 5–8 June 2017.
13. Lee, J.; Marcus, K.; Abdelzaher, T.; Amin, M.T.A.; Bar-Noy, A.; Dron, W.; Govindan, R.; Hobbs, R.; Hu, S.; Kim, J.-E.; et al. Athena: Towards Decision-centric Anticipatory Sensor Information Delivery. *J. Sens. Actuator Netw.* **2018**, *7*, 5. [CrossRef]
14. Miles, A.; Zaslavsky, A.; Browne, C. IoT-based decision support system for monitoring and mitigating atmospheric pollution in smart cities. *J. Decis. Syst.* **2018**, *27*, 56–67. [CrossRef]
15. Morfino, V.; Rampone, S. Towards near-real-time intrusion detection for IoT devices using supervised learning and Apache Spark. *Electronics* **2020**, *9*, 444. [CrossRef]
16. Valliappan, S.; Sivakumar, P.B.; Ananthanarayanan, V. Efficient real-time decision making using streaming data analytics in IoT environment. In Proceedings of the International Conference on Advanced Computing Networking and Informatics, West Bengal, India, 20–21 December 2019; Springer: Berlin/Heidelberg, Germany, 2019; pp. 165–173.
17. Turner, C.J.; Emmanouilidis, C.; Tomiyama, T.; Tiwari, A.; Roy, R. Intelligent decision support for maintenance: An overview and future trends. *Int. J. Comput. Integr. Manuf.* **2019**, *32*, 936–959. [CrossRef]

18. Hassani, A.; Medvedev, A.; Zaslavsky, A.; Delir Haghighi, P.; Jayaraman, P.P.; Ling, S. Efficient execution of complex context queries to enable near real-time smart IoT applications. *Sensors* **2019**, *19*, 5457. [CrossRef]
19. Shahinmoghadam, M.; Natephra, W.; Motamedi, A. BIM-and IoT-based virtual reality tool for real-time thermal comfort assessment in building enclosures. *Build. Environ.* **2021**, *199*, 107905. [CrossRef]
20. Puiu, D.; Barnaghi, P.; Tönjes, R.; Kümper, D.; Ali, M.I.; Mileo, A.; Parreira, J.X.; Fischer, M.; Kolozali, S.; Farajidavar, N.; et al. Citypulse: Large scale data analytics framework for smart cities. *IEEE Access* **2016**, *4*, 1086–1108. [CrossRef]
21. de Assis, M.V.; Carvalho, L.F.; Rodrigues, J.J.; Lloret, J.; Proença, M.L., Jr. Near real-time security system applied to SDN environments in IoT networks using convolutional neural network. *Comput. Electr. Eng.* **2020**, *86*, 106738. [CrossRef]
22. Kim, J.E.; Abdelzaher, T.F.; Sha, L.; Bar-Noy, A.; Hobbs, R.L.; Dron, W. On Maximizing Quality of Information for the Internet of Things: A Real-Time Scheduling Perspective (Invited Paper). In Proceedings of the IEEE International Conference on Embedded and Real-Time Computing Systems and Applications, Daegu, Korea, 17–19 August 2016; pp. 202–211.
23. Kim, J.E.; Abdelzaher, T.F.; Sha, L.; Bar-Noy, A.; Hobbs, R. Sporadic Decision-centric Data Scheduling with Normally-off Sensors. In Proceedings of the IEEE Real-Time Systems Symposium, Porto, Portugal, 29 November–2 December 2016.
24. Kim, J.; Abdelzaher, T.F.; Sha, L.; Bar-Noy, A.; Hobbs, R.L.; Dron, W. Decision-driven scheduling. *Real-Time Syst.* **2019**, *55*, 514–551. [CrossRef]
25. Kang, K.D. Towards Efficient Real-Time Decision Support at the Edge. In Proceedings of the ACM/IEEE Workshop on Hot Topics on Web of Things (in Conjunction with ACM/IEEE Symposium on Edge Computing, New York, NY, USA, 7–9 November 2019.
26. Alsulami, M.M.; Akkari, N. The role of 5G wireless networks in the internet-of-things (IoT). In Proceedings of the 2018 1st International Conference on Computer Applications & Information Security (ICCAIS), Riyadh, Saudi Arabia, 4–6 April 2018; pp. 1–8.
27. Morin, E.; Maman, M.; Guizzetti, R.; Duda, A. Comparison of the device lifetime in wireless networks for the internet of things. *IEEE Access* **2017**, *5*, 7097–7114. [CrossRef]
28. Mohammed, A.H.; Khaleefah, R.M.; Hussein, M.K.; Abdulateef, I.A. A review software defined networking for internet of things. In Proceedings of the 2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), Ankara, Turkey, 26–28 June 2020; pp. 1–8.
29. Wijethilaka, S.; Liyanage, M. Survey on network slicing for Internet of Things realization in 5G networks. *IEEE Commun. Surv. Tutor.* **2021**, *23*, 957–994. [CrossRef]
30. Das, H.; Dey, N.; Balas, V.E. *Real-Time Data Analytics for Large Scale Sensor Data*; Academic Press: Cambridge, MA, USA, 2019.
31. Raafat, H.M.; Hossain, M.S.; Essa, E.; Elmougy, S.; Tolba, A.S.; Muhammad, G.; Ghoneim, A. Fog intelligence for real-time IoT sensor data analytics. *IEEE Access* **2017**, *5*, 24062–24069. [CrossRef]
32. Elijah, O.; Rahman, T.A.; Orikumhi, I.; Leow, C.Y.; Hindia, M.N. An overview of Internet of Things (IoT) and data analytics in agriculture: Benefits and challenges. *IEEE Internet Things J.* **2018**, *5*, 3758–3773. [CrossRef]
33. Cormen, T.H.; Leiserson, C.E.; Rivest, R.L.; Stein, C. *Introduction to Algorithms*, 3rd ed.; The MIT Press: Cambridge, MA, USA, 2009.
34. Ramamritham, K.; Son, S.H.; DiPippo, L. Real-Time Databases and Data Services. *Real-Time Syst.* **2004**, *28*, 179–215. [CrossRef]
35. Ramamritham, K. Real-Time Databases. *Int. J. Distrib. Parallel Databases* **1993**, *1*. Available online: https://link.springer.com/article/10.1007/BF01264051 (accessed on 10 January 2022). [CrossRef]
36. Greiner, R.; Hayward, R.; Jankowska, M.; Molloy, M. Finding optimal satisficing strategies for and-or trees. *Artif. Intell.* **2006**, *170*, 19–58. [CrossRef]
37. Casanova, H.; Lim, L.; Robert, Y.; Vivien, F.; Zaidouni, D. Cost-Optimal Execution of Boolean Query Trees with Shared Streams. In Proceedings of the IEEE International Parallel and Distributed Processing Symposium, Phoenix, AZ, USA, 19–23 May 2014; pp. 7–16.
38. Kang, K.D.; Son, S.H.; Stankovic, J.A.; Abdelzaher, T.F. A QoS-Sensitive Approach for Timeliness and Freshness Guarantees in Real-Time Databases. In Proceedings of the Euromicro Conference on Real-Time Systems, Vienna, Austria, 19–21 June 2002.
39. Kang, K.D.; Son, S.; Stankovic, J.A. Managing Deadline Miss Ratio and Sensor Data Freshness in Real-Time Databases. *IEEE Trans. Knowl. Data Eng.* **2004**, *16*, 1200–1216. [CrossRef]
40. Kang, K.D.; Oh, J.; Son, S.H. Chronos: Feedback Control of a Real Database System Performance. In Proceedings of the IEEE Real-Time Systems Symposium, Tucson, AZ, USA, 3–6 December 2007.
41. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016. Available online: http://www.deeplearningbook.org (accessed on 10 January 2022).
42. Bertsekas, D.P. *Dynamic Programming and Optimal Control*, 3rd ed.; Athena Scientific: Belmont, MA, USA, 2005; Volume I.
43. Zois, D.S. Sequential decision-making in healthcare IoT: Real-time health monitoring, treatments and interventions. In Proceedings of the 2016 IEEE 3rd World Forum on Internet of Things (WF-IoT), Reston, VA, USA, 12–14 December 2016; pp. 24–29.
44. Awais, M.; Raza, M.; Singh, N.; Bashir, K.; Manzoor, U.; ul Islam, S.; Rodrigues, J.J. LSTM based Emotion Detection using Physiological Signals: IoT framework for Healthcare and Distance Learning in COVID-19. *IEEE Internet Things J.* **2020**, *8*, 16863–16871. [CrossRef]
45. Islam, M.S.; Umran, H.M.; Umran, S.M.; Karim, M. Intelligent Healthcare Platform: Cardiovascular Disease Risk Factors Prediction Using Attention Module Based LSTM. In Proceedings of the International Conference on Artificial Intelligence and Big Data (ICAIBD), Chengdu, China, 25–28 May 2019; pp. 167–175.
46. Pham, T.; Tran, T.; Phung, D.; Venkatesh, S. Predicting healthcare trajectories from medical records: A deep learning approach. *J. Biomed. Inform.* **2017**, *69*, 218–229. [CrossRef]

47. Khan, N.S.; Ghani, S.; Haider, S. Real-Time Analysis of a Sensor's Data for Automated Decision Making in an IoT-Based Smart Home. *Sensors* **2018**, *18*, 1711. [CrossRef] [PubMed]
48. Machorro-Cano, I.; Alor-Hernández, G.; Paredes-Valverde, M.A.; Rodríguez-Mazahua, L.; Sánchez-Cervantes, J.L.; Olmedo-Aguirre, J.O. HEMS-IoT: A big data and machine learning-based smart home system for energy saving. *Energies* **2020**, *13*, 1097. [CrossRef]
49. Rashidi, P.; Cook, D.J. Keeping the resident in the loop: Adapting the smart home to the user. *IEEE Trans. Syst. Man Cybern.-Part Syst. Hum.* **2009**, *39*, 949–959. [CrossRef]
50. Guo, K.; Hu, Y.; Qian, Z.; Liu, H.; Zhang, K.; Sun, Y.; Gao, J.; Yin, B. Optimized graph convolution recurrent neural network for traffic prediction. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 1138–1149. [CrossRef]
51. Izonin, I.; Kryvinska, N.; Vitynskyi, P.; Tkachenko, R.; Zub, K. GRNN approach towards missing data recovery between IoT systems. In Proceedings of the International Conference on Intelligent Networking and Collaborative Systems, Oita, Japan, 5–7 September 2019; Springer: Berlin/Heidelberg, Germany, 2019; pp. 445–453.
52. Tkachenko, R.; Izonin, I.; Dronyuk, I.; Logoyda, M.; Tkachenko, P. Recovery of missing sensor data with GRNN-based cascade scheme. *Int. J. Sens. Wirel. Commun. Control* **2021**, *11*, 531–541. [CrossRef]
53. Tkachenko, R.; Izonin, I.; Kryvinska, N.; Dronyuk, I.; Zub, K. An approach towards increasing prediction accuracy for the recovery of missing IoT data based on the GRNN-SGTM ensemble. *Sensors* **2020**, *20*, 2625. [CrossRef] [PubMed]
54. Izonin, I.; Tkachenko, R.; Verhun, V.; Zub, K. An approach towards missing data management using improved GRNN-SGTM ensemble method. *Eng. Sci. Technol. Int. J.* **2021**, *24*, 749–759. [CrossRef]
55. Wang, Y.; Ho, I.W.H.; Chen, Y.; Wang, Y.; Lin, Y. Real-time Water Quality Monitoring and Estimation in AIoT for Freshwater Biodiversity Conservation. *IEEE Internet Things J.* **2021**. [CrossRef]
56. Tien, J.M. Internet of Things, Real-Time Decision Making, and Artificial Intelligence. *Ann. Data Sci.* **2017**, *4*, 149–178. [CrossRef]
57. Latency Across Cloud Backbones Varies Significantly. Available online: https://www.sd-wan-experts.com/blog/latency-across-cloud-backbones-varies-significantly/ (accessed on 10 January 2022).
58. State of IoT 2021: Number of Connected IoT Devices Growing 9% to 12.3 Billion Globally, Cellular IoT Now Surpassing 2 Billion. Available online: https://iot-analytics.com/number-connected-iot-devices/ (accessed on 10 January 2022).
59. Al-Garadi, M.A.; Mohamed, A.; Al-Ali, A.K.; Du, X.; Ali, I.; Guizani, M. A Survey of Machine and Deep Learning Methods for Internet of Things (IoT) Security. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 1646–1685. [CrossRef]
60. Martinez, B.; Monton, M.; Vilajosana, I.; Prades, J.D. The power of models: Modeling power consumption for IoT devices. *IEEE Sens. J.* **2015**, *15*, 5777–5789. [CrossRef]
61. Min, M.; Xiao, L.; Chen, Y.; Cheng, P.; Wu, D.; Zhuang, W. Learning-based computation offloading for IoT devices with energy harvesting. *IEEE Trans. Veh. Technol.* **2019**, *68*, 1930–1941. [CrossRef]
62. Mekki, K.; Bajic, E.; Chaxel, F.; Meyer, F. A comparative study of LPWAN technologies for large-scale IoT deployment. *ICT Express* **2019**, *5*, 1–7. [CrossRef]
63. Lavric, A.; Popa, V. Internet of things and LoRa™ low-power wide-area networks: A survey. In Proceedings of the 2017 International Symposium on Signals, Circuits and Systems (ISSCS), Iasi, Romania, 13–14 July 2017; pp. 1–5.
64. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. ImageNet Large Scale Visual Recognition Challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [CrossRef]
65. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef] [PubMed]
66. Satyanarayanan, M.; Simoens, P.; Xiao, Y.; Pillai, P.; Chen, Z.; Ha, K.; Hu, W.; Amos, B. Edge Analytics in the Internet of Things. *IEEE Pervasive Comput.* **2015**, *14*, 24–31. [CrossRef]
67. Jedari, B.; Premsankar, G.; Illahi, G.; Di Francesco, M.; Mehrabi, A.; Ylä-Jääski, A. Video Caching, Analytics, and Delivery at the Wireless Edge: A Survey and Future Directions. *IEEE Commun. Surv. Tutor.* **2020**, *23*, 431–471. [CrossRef]
68. Liu, F.; Tang, G.; Li, Y.; Cai, Z.; Zhang, X.; Zhou, T. A survey on edge computing systems and tools. *Proc. IEEE* **2019**, *107*, 1537–1562. [CrossRef]
69. Liu, D.; Yan, Z.; Ding, W.; Atiquzzaman, M. A survey on secure data analytics in edge computing. *IEEE Internet Things J.* **2019**, *6*, 4946–4967. [CrossRef]
70. Mazumder, A.N.; Meng, J.; Rashid, H.A.; Kallakuri, U.; Zhang, X.; Seo, J.S.; Mohsenin, T. A Survey on the Optimization of Neural Network Accelerators for Micro-AI On-Device Inference. *IEEE J. Emerg. Sel. Top. Circuits Syst.* **2021**, *11*, 532–547. [CrossRef]
71. Sanabria-Russo, L.; Pubill, D.; Serra, J.; Verikoukis, C. IoT data analytics as a network edge service. In Proceedings of the IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Paris, France, 29 April–2 May 2019; pp. 969–970.
72. Hanyao, M.; Jin, Y.; Qian, Z.; Zhang, S.; Lu, S. Edge-assisted online on-device object detection for real-time video analytics. In Proceedings of the IEEE INFOCOM 2021-IEEE Conference on Computer Communications, Vancouver, BC, Canada, 10–13 May 2021; pp. 1–10.
73. Marjani, M.; Nasaruddin, F.; Gani, A.; Karim, A.; Hashem, I.A.T.; Siddiqa, A.; Yaqoob, I. Big IoT data analytics: Architecture, opportunities, and open research challenges. *IEEE Access* **2017**, *5*, 5247–5261.
74. Sharma, S.K.; Wang, X. Live data analytics with collaborative edge and cloud processing in wireless IoT networks. *IEEE Access* **2017**, *5*, 4621–4635. [CrossRef]

75. Dayalan, U.K.; Fezeu, R.A.; Varyani, N.; Salo, T.J.; Zhang, Z.L. VeerEdge: Towards an Edge-Centric IoT Gateway. In Proceedings of the 2021 IEEE/ACM 21st International Symposium on Cluster, Cloud and Internet Computing (CCGrid), Melbourne, Australia, 10–13 May 2021; pp. 690–695.
76. Li, Y.; Padmanabhan, A.; Zhao, P.; Wang, Y.; Xu, G.H.; Netravali, R. Reducto: On-camera filtering for resource-efficient real-time video analytics. In Proceedings of the Annual Conference of the ACM Special Interest Group on Data Communication on the Applications, Technologies, Architectures, and Protocols for Computer Communication, (Virtual Conference), New York, NY, USA, 10–14 August 2020; pp. 359–376.
77. Du, K.; Pervaiz, A.; Yuan, X.; Chowdhery, A.; Zhang, Q.; Hoffmann, H.; Jiang, J. Server-driven video streaming for deep learning inference. In Proceedings of the Annual conference of the ACM Special Interest Group on Data Communication on the Applications, Technologies, Architectures, and Protocols for Computer Communication, (Virtual Conference), New York, NY, USA, 10–14 August 2020; pp. 557–570.
78. Deng, L.; Li, G.; Han, S.; Shi, L.; Xie, Y. Model compression and hardware acceleration for neural networks: A comprehensive survey. *Proc. IEEE* **2020**, *108*, 485–532. [CrossRef]
79. Cheng, Y.; Wang, D.; Zhou, P.; Zhang, T. A survey of model compression and acceleration for deep neural networks. *arXiv* **2017**, arXiv:1710.09282.
80. He, Y.; Lin, J.; Liu, Z.; Wang, H.; Li, L.J.; Han, S. AMC: AutoML for model compression and acceleration on mobile devices. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 784–800.
81. Polino, A.; Pascanu, R.; Alistarh, D. Model compression via distillation and quantization. *arXiv* **2018**, arXiv:1802.05668.
82. Avnur, R.; Hellerstein, J.M. Eddies: Continuously Adaptive Query Processing. In Proceedings of the ACM SIGMOD Conference, Dallas, TX, USA, 16–18 May 2000.