AVOC: History-Aware Data Fusion for Reliable IoT Analytics

Panagiotis Gkikopoulos
University of Neuchâtel
Neuchâtel, Switzerland
panagiotis.gkikopoulos@unine.ch

Peter Kropf
University of Neuchâtel
Neuchâtel, Switzerland
peter.kropf@unine.ch

Valerio Schiavoni
University of Neuchâtel
Neuchâtel, Switzerland
valerio.schiavoni@unine.ch

Josef Spillner
Zurich University of Applied Sciences
Winterthur, Switzerland
josef.spillner@zhaw.ch

ABSTRACT

IoT systems rely on collected data to operate autonomously and generate insights. Such systems commonly produce redundant measurements, which can be insufficient to mitigate complex data disagreements. We believe a well-defined process to achieve internal ground truth through fusion is needed. Leveraging two case studies, we show how *sensor data fusion* with variants of history-aware voting can help to reconcile observations. We contribute a specification scheme with unified format to define the parameters and characteristics of a particular voting scenario, supporting reliable decision-making. Finally, we deploy and evaluate a novel method of bootstrapping historical records of sensor modules using a clustering algorithm. This method boosts the convergence of the measurements by 4×.

CCS CONCEPTS

• **Information systems** → **Data cleaning**; • **Software and its engineering** → *Software reliability*; *Distributed systems organizing principles.*

KEYWORDS

voting algorithms, data quality, data fusion, IoT

ACM Reference Format:
Panagiotis Gkikopoulos, Peter Kropf, Valerio Schiavoni, and Josef Spillner. 2022. AVOC: History-Aware Data Fusion for Reliable IoT Analytics. In *23rd International Middleware Conference (Middleware '22 Industrial Track), November 7–11, 2022, Quebec, QC, Canada.* ACM, New York, NY, USA, 7 pages.

1 INTRODUCTION

Emerging digitalised systems in smart cities, industrial production and other cyber-physical domains involve large amounts of data aggregated from a multitude of sources. Sensor-based measurements are common especially in the Internet of Things (IoT) for triggering data-driven decisions. Quality issues in such measurements [31] have profound adverse effects on systems leverage, and by extension, on humans and society, as shown in *irresponsible AI* community discussions [21]. Consequently, steps to improve input data quality are necessary to achieve better decisions and overall more reliable applications. Data fusion is a technique of merging different inputs [23] in an application-independent middleware to obtain a holistic view of physical objects. Voting is an approach to fuse sensor data for the purposes of reliability and error mitigation [15, 18] in safety-critical environments. For instance, in avionics, three redundant physical sensors are mandated for each logical sensor [15]. In smart shopping scenarios with networked shelf labels, the degree of redundancy rises significantly to dozens of proximity sensors. Thus, in the absence of external ground truth (*i.e.*, a fully trusted and accurate data source, often too expensive in practice), voting is a pragmatic substitute as it leads to internal ground truth upon which critical decision-making can be based.

In this work, we study and observe state-of-the-art voting approaches for sensor data fusion, applying them to two IoT scenarios relying of redundant sensor measurements: light sensors in a smart building setting, and Bluetooth Low Energy (BLE) beacons to track vehicle position in a (simulated) tunnel. We focus on voting algorithms used to reach *data-centric consensus* on numerical values, as these are relevant when merging sensor readings and leverage historical records to factor in the reliability of individual sensors. In §7 we conduct two experiments on such IoT setups for real-time validation and pre-recorded data for the purpose of reproducibility. We exploit our findings to contribute a generic specification format that can be used to define voting schemes for several applications, particularly optimized for IoT and cyber-physical applications. We argue that such a format aids the development of distributed analytics applications by making them more needs-focused and reliable, while shielding software engineers from the voting implementation. Moreover, the increased robustness of the data, induced by the multi-perspective voting, facilitates the input data quality in data-centric artificial intelligence, a recent research direction aimed at overcoming *misprediction* due to lack of input data assurance [28]. Leveraging the outcomes of the practical experiments, we present a novel clustering-based approach to augment the performance of the state-of-the-art history-based voting algorithms.

Our contributions are twofold: *(1)* AVOC (Accurate Voting with Clustering), a novel bootstrapping method for initializing history-based voting systems, which we fully implement and evaluate with

Middleware '22 Industrial Track, November 7–11, 2022, Quebec, QC, Canada

ACM ISBN 978-1-4503-9917-3/22/11…$15.00

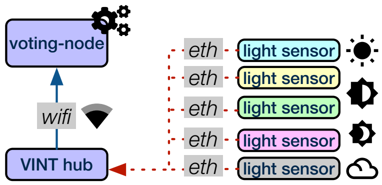

Figure 1: Light sensor use-case: the sensors are wired via ethernet to a hub, streaming data via WiFi to the voting sink-node.

two practical IoT scenarios; *(2)* VDX, a new voting definition specification that precisely defines application requirements and allows users to select appropriate parameters for software voters.

The rest of the paper is structured as follows. §2 surveys state-of-the-art voting algorithms, data fusion and data quality issues relevant to IoT. In §3 we detail our use-case scenarios and how we built our hardware prototypes. §4 investigates voting algorithms used to reconcile redundant data values. §5 presents AVOC, our approach. We describe the proposed format VDX for defining the voting process in §6. §7 presents our experimental evaluation using both a reference scenario dataset and our experimental setup. We conclude in §8 by discussing our findings and prospect future research directions in redundancy-based data quality in IoT.

2 RELATED WORK

Data-driven decisions [13] are key elements of cyber-physical systems and digitalised applications of all scales and domains (*e.g.*, smart cities, mobility, industrial production, home automation, *etc.*). In many such systems, incoming data are subject to real-time analysis and subsequent decision-making. However, while systems research has allowed dealing with the volume, velocity and variety of multi-source data involved in the processing chain [27], data quality, value and veracity issues still emerge. In this work, we focus on the accuracy (*i.e.*, quality) for sensor measurements [25]. Data fusion across multiple homogeneous or heterogeneous sensors has been utilised to tackle the challenge, fixing problematic data and improving analytics reliability with a variety of techniques, such as data association, state estimation, decision fusion, classification, prediction, machine learning and analytics [19]. Initial efforts exist to create standards and frameworks for data management and interoperability, for instance in the smart city space [14, 16]. However, they currently lack a common framework and standardized format. Our work proposes a new interoperable format to define voting-related data fusion alone.

Voting algorithms increase the reliability of measurements [18] in safety-critical domains, *e.g.*, aviation [15] or self-driving cars [9]. We focus on reconciling numeric data using software voters, with either result selection or amalgamation techniques [18]. Specifically, we consider history-based voting algorithms [17] that weigh values based on the historical performance record of the candidate sensor. History-Based Weighted Average [17] weights the historical data to compute an output value. To improve the granularity of historical records, [11] uses a soft dynamic threshold. In [7], authors apply an hybrid approach using module elimination and dynamic threshold. We further detail these approaches in §4.

Figure 2: Portable 'shoe-box' testbed for our light sensor setup (Fig.1). A Raspberry Pi 4 runs the fusion script. An LCD display shows the voting results and weight values.

Some voting-based data fusion frameworks with description languages to define algorithmic details exist [8], but they ignore history-based measurements. We also observe that modern voting algorithms are too complex to be represented in such terms. Where [8] defines voting as three-step process (reaching quorum, excluding outliers and calculating results), modern algorithms often include further steps like weighing and updating historical records, or optimising reliability metrics [20]. We argue that a customisable voting framework serves as encapsulation for sensor-fusion applications if built atop state-of-the-art approaches, as shown next.

3 MULTI-SENSOR APPLICATION SCENARIOS

We devise two use-case scenarios to validate our approach. Both scenarios represent sensor-reliant deployments where redundancy proves valuable. Specifically, we draw on the smart building and self-driving vehicle domains, both showcasing cyber-physical systems relying on sensor measurements to control other critical systems.

Fig. 1 depicts our first use-case, a sunlight detection system in a hypothetical smart building. The hub is connected to a sink node to record 10'000 rounds of concurrent measurements from 5 sensors, polling at 8 samples/s, to create a reference dataset representing 1250 seconds of data collection. Each round produces 5 float values per sensor. The reference dataset consists of the raw readings from all sensors and is used to compare all voting algorithms on the same set of values. We built a portable demonstrator that executes them and our proposed voting algorithm, AVOC (detailed in §5), leveraging a Raspberry Pi 4B unit (see Fig. 2). It uses a Phidget Wifi hub [6] (VINT) connected to a set of 5 LUX1000 Light Phidget sensors [5]. Input, weights and results are shown on an LCD screen [3] connected to the hub. The portable version let us confirm the feasibility to execute on constrained hardware, combining both redundant measurements and voting.

In the second use-case, we mimic a typical smart-city/self-driving vehicle application that relies on indoor-positioning to track the position of a moving unit (*e.g.*, a robot). We simulate the operations to track a cargo vehicle traversing a tunnel, as shown in Fig. 3. Such vehicles use Bluetooth (BLE [1]) beacons as *milestones* to

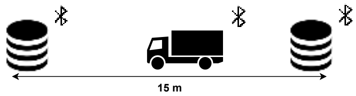

Figure 3: BLE beacon use-case. 2 stacks of beacons 15 meters apart with a robot driving between them, taking signal strength measurements along the way.

Figure 4: Robot driving to the circled beacon stack destination. The laptop acts as bluetooth receiver and edge voter.

locate the position of the truck, as done in emerging country-wide systems (*i.e.*, CST – Cargo Sous Terrain [2, 22]). We deploy two stacks of nine redundant beacons and a cargo vehicle using a Lego Mindstorms EV3 [4] robot. As the Bluetooth receiver on the robot was incompatible with the beacons due to lack of BLE support, we installed a laptop on top of it acting as pragmatic substitute receiver with edge processing capabilities, and without affecting the generality of the findings other than affecting the speed of the robot (which has no effect on what we are measuring). Fig. 4 shows the prototype.

The robot drives slowly in a straight line with no line-of-sight obstacles from one beacon stack to the other, across a distance of 15 meters. The speed of the robot was set to 7% of its specified top speed (0.09 m/s). We collected as many data points as possible along the route, resulting in 297 measurements per beacon, noting that autonomous cargo systems like CST proceed at around 8.3 m/s, thus having 99% less measurement samples available for voting.

4 HISTORY-AWARE VOTING ALGORITHMS

To combine values from uncalibrated redundant sensors, the history-based averaging algorithm (*i.e.*, henceforth referred to as Standard algorithm [17]) either chooses a sensor output value or creates an amalgamation of these values. This approach can be be optimized by temporarily ignoring values produced by modules with below average historical records. This variant, *i.e.*, Module Elimination Weighted Average (Me), assigns zero-weights to the discarded values in the voting until their historical records improve by submitting better values, even if discarded in the voting itself.

The Soft Dynamic Threshold History-Based Weighted Average (Sdt) introduces a finer grain definition of agreement, beyond the binary-only definition [11]. Values between 1 and 0 can be assigned if values are not in agreement based on the accepted error threshold, but are in agreement based on a multiple of it. The magnitude of the multiple is defined by a parameter of the algorithm that can be tuned according to the needs of the specific use case.

We further consider Hybrid History-Based Weighted Average (henceforth Hybrid [7]). It combines Me and Sdt, while utilising agreement-based and not history-based weights. The Hybrid algorithm allows to choose a winning value rather than assigning the resulting average, using the mean nearest neighbour approach. We discuss our findings on the output quality of all algorithms in §7.

5 AVOC: ACCURATE VOTING WITH CLUSTERING

History-based algorithms typically fall back to standard average (or a similar unweighted approach) on the first round until a historical record is established or when the weights become 0 due to severe issues with the data. Weights can drop to 0 after a series of disagreements, which results in notorious disagreers being rated as untrustworthy by the system applying the algorithm. Our approach, named AVOC, builds atop the Hybrid algorithm by applying a simplified clustering algorithm during the first round when the weights are all 0. The clustering step eliminates obvious outliers, improving the accuracy of that round compared to mean average, with little performance overhead and faster convergence speed.

Algorithmic approach. For the clustering step, we leverage a similar logic to the agreement calculation in voting algorithms: we check for values within a given scaling threshold of each other (which is selected to mirror the parameters of the given algorithm), and group the values in agreement. Then, we select as output value the average (or its closest real value) of the largest group (if we are using the mean nearest-neighbour approach to output selection). This grouping logic is similar to DBSCAN [12]; AVOC opts for self-calibration, rather than requiring costly parameters tuning. This is achieved through a majority vote with a soft-dynamic error margin (as the margin depends on a reference value). The clustering step is used for bootstrapping a new set of modules, or as a fallback in cases of issues. To reach this goal, we use the historical record value for each module, and declare that the clustering approach should be used when all records are 1 (indicating a new set) or 0 (indicating a failure of the system or an extreme data spike).

Generalisation. Generalising this approach for multi-dimensional data, an unsupervised clustering algorithm can be used such as Meanshift [10] or X-Means [24]. The logic of choosing an output value would be similar. However, in such scenarios, choosing a single output vector for multiple dimensions is non-trivial as the complexity of data and correlation of errors considerably increases. To mitigate, the voting approach can be applied for each dimension separately, leaving other data fusion techniques to process the multi-dimensional results. In AVOC, we follow the approach of voting on each dimension separately, without incorporating the clustering itself.

6 VDX: VOTING DEFINITION SPECIFICATION

To enable reliable implementations and improve the usability of software voters, we contribute a new voting specification scheme and parsing logic. The scheme can define any of the algorithms described above, as well as simpler ones without history.

Software-defined voting schemes exist, *e.g.*, Voting Definition Language (VDL) [8]. Those predate more complex history-based voting approaches, and extending VDL for finer-grain algorithmic

Listing 1: Vote definition sample in VDX JSON format.

```
1 {
2   "algorithm_name": "AVOC",
3   "quorum": "UNTIL",
4   "quorum_percentage": 100,
5   "exclusion": "NONE",
6   "exclusion_threshold": 0,
7   "history": "HYBRID",
8   "params":{
9       "error": 0.05,
10      "soft_threshold": 2
11  },
12  "collation": "MEAN_NEAREST_NEIGHBOR",
13  "bootstrapping": true,
14 }
```

definitions is challenging. However, our specification VDX supports the relevant parameters of VDL, enabling our definition to describe a superset of VDL-scoped algorithms. The full schema, as well as a sample implementation and usage examples can be found at: https://github.com/EcePanos/vdx. The repository also includes an interactive application that allows users to compare the algorithms presented with the state of the art (Fig. 5).

Capabilities. Listing 1 shows our AVOC algorithm using this scheme as example. VDX allows the specification of several parameters, including quorum (*i.e.*, how many candidates need to submit values for a vote to be triggered), exclusions (to automatically prune outliers) collation techniques similar to VDL, *e.g.*, "mean nearest neighbour" (Listing 1, line 12). It extends VDL by allowing the selection of a history algorithm (Listing 1, line 7), additional parameters (Listing 1, lines 8-11), and whether to enable clustering algorithm as a bootstrap/fallback mechanism (Listing 1, line 13). Another extension VDX adds over VDL is the ability to vote on categorical *i.e.*, non-numeric values, such as character strings and JSON blobs. In such cases however, several features are disabled. Value-based exclusion cannot be applied, as there can be no mean or standard deviation calculation. The 'standard' and 'module-elimination' algorithms for deriving module history are available, however the 'hybrid' algorithm is not, as the fine-grained agreement definition cannot be applied to non-numeric values. Finally, clustering-based bootstrapping cannot be applied to categorical values and the only collation method is the weighted majority vote. Software voting implementers may re-introduce some of these features by supplying a custom distance metric for categorical values.

Limitations and assumptions. VDX currently cannot define algorithms that use parameters for the candidate values, *e.g.*, MLV [20], or genetic voting algorithms [26], but assists already by introducing voting into further IoT software for reliable input data and analytics. It should also be noted that VDX itself has no security features that protect against malicious actors, so this is left up to the client code to implement as needed.

7 EVALUATION

This section presents the experimental evaluation of AVOC. With VDX, we fully implemented the two use-case scenarios from §3, and compare the error-correction performance of the various voting schemes. Our evaluation answers the following questions: *(Q1)* Can AVOC improve the output quality of our use-case scenarios? *(Q2)* Can it mitigate injected errors and reach the same output? *(Q3)* Which algorithm fits which scenario better, and *(Q4)* How can we leverage VDX to customise the voting behavior for each scenario?

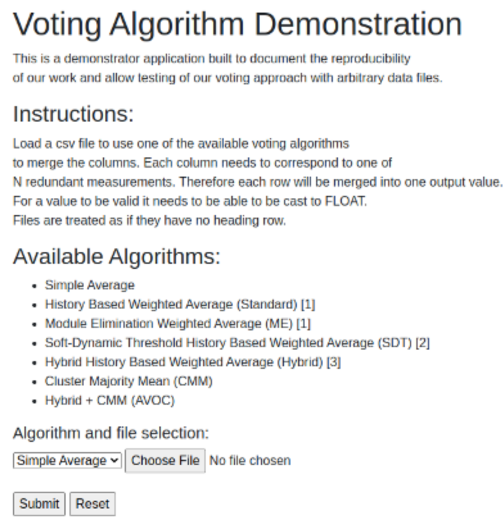

Figure 5: Algorithm comparison application.

Implementation details. We implemented AVOC and the approaches from §4 in Python 3.9, for a total of 490 LOC. We note that though the evaluation was done with pre-recorded data for reproducibility purposes, the system can execute a history-aware voting round in 1 millisecond and a stateless vote in 50 microseconds (datastore reads and writes being the bottleneck). Thus, the system can operate under soft real-time constraints, as in the case of our 'shoe-box' demonstrator in Fig.2 as well as many critical cyber-physical applications in practice.

UC-1: Light sensors. We used the 10'000 value dataset recorded using our light sensor setup (§3) to gather the raw data and reference values for the baseline algorithms (Fig.6-a). Then, we injected an artificial outlier sensor, by adding +6 lumen to one of the sensors. We compare the performance of the different algorithms according to the following metrics: *(a)* voting rounds required to converge back to the baseline, and by extension how quickly outliers are eliminated; and *(b)* how far the new stable value is from the original.

We make a first comparison using the raw reference data. In this scenario (Fig. 6-b), all 6 variants performed equally well, with outputs matching almost completely. The error injection case (shown in Fig. 6-c) exhibits some interesting facts. First, the Standard algorithm exhibits high initial skew, which is then slowly mitigated as the *faulty* sensor (E4) is de-emphasised. However, even after 10000 voting rounds of voting (20 minutes in our experiments), the skew is not eliminated completely. This is where the module elimination feature of Me is beneficial, as the faulty sensor is quickly eliminated in round 2, as performing below average compared to the rest. However, as seen in Fig. 6-c, the gap created by skewing E4 indicates how E3 is also now tagged as outlier, and its result is skewed upwards by 0.2lm.

The Hybrid algorithm also uses a granular definition of agreement score, but combines it with the aggressive elimination of modules from Me. For our experiment, this is the *best of both worlds*

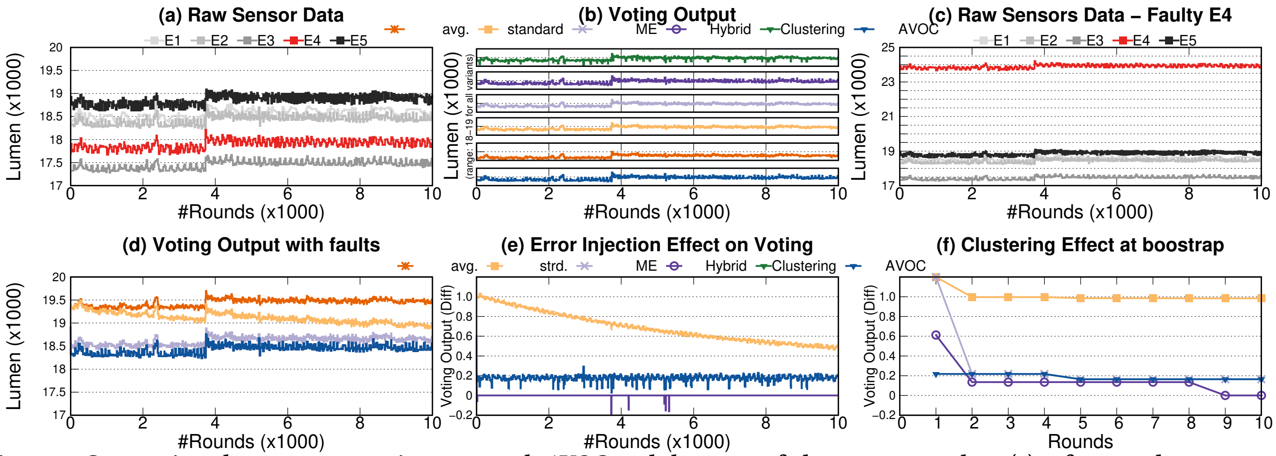

Figure 6: Comparison between our voting approach AVOC and the state-of-the-art approaches. (a) reference data, captured by our light sensor setup for 20min. (b) voting output using AVOC on the reference data. (c) Reference data with injected errors (1 faulty sensor). (d) Output of Hybrid, Clustering and AVOC under these errors. (e) Output difference between voting on the raw values and voting on the error-injected values. (f) Zoom on the first 10 rounds.

result (also shown by the differentials in Fig. 6-e) where, minus few spikes, the value is identical to the pre-error output.

For completeness, we show clustering-based voting on its own without combining it with Hybrid (which remains ideal for this scenario). This can be seen in Fig.6-e. We observe similar behavior to Me, with E4 being excluded from the output immediately. Differently from Me, E4 was also excluded from the first round. In contrast, E3 was not always excluded, due to the lack of history-based elimination, indicating higher variations in the output. Regarding clustering-only voting (COV), it significantly outperforms other stateless approach, *i.e.*, weighted average without history. This result indicates that the COV approach fits well scenarios where maintaining historical result records is impractical: short-lived sensor measurements, one-time comparisons of datasets, *etc.*

One consistent observation in the error injection experiment is that history-based algorithms experience a spike on startup, as the artificially modified value is skewing the output but is not yet mitigated by the history. This is the phase where in principle the clustering step detailed above has higher chances to affect the results. Indeed, although the clustering algorithm alone is not as accurate as Me or Hybrid, it overcomes the initial data spikes. Thus, a system capable of fallback to it when history is not available or suspected unreliable, can benefit from its inclusion.

Next, we run AVOC, which concretely combines the clustering step with Hybrid. We observe how the initial spike is quickly pruned (Fig. 6-f) within the initial rounds. The bootstrap boost can also be noticed: due to the better history adjustment in round 1, the voter already *learns* to exclude E3 from round 2, returning to its pre-error output almost instantly, despite the clustering is only used once. As AVOC converges within 200ms, the experiment confirms its utility for fast accurate voting and provides a positive response to our questions (Q1) and (Q2), as we improve the reliability of the output even in the presence of the injected errors.

UC-2: BLE beacons. We leverage this second use-case with BLE beacons and a Lego Mindstorms EV3 robot to study a scenario with more anomalies and faults. We set up two stacks of 9 beacons each 15 meters apart in an indoor corridor with no obstacles. The robot drives at 7% of its full speed in a straight line, from one stack to the other. The robot takes continuous RSSI (Received Signal Strength Indicator) measurements for each beacon along the way. This experiment examines if the high redundancy and voting actually are beneficial. The scenario simulates the operations to locate a vehicle traversing a tunnel using beacon stacks as *milestones* along the way, determining the closest stack to the vehicle. The state of the art in leveraging RSSI for positioning relies on filtering [30] or collaborative positioning [29] to improve stability and output quality. However, since the scope of our experiment is to evaluate the effects of voting on sensor measurements, we kept the raw RSSI values from the sensors, to keep the values as close to the original measurement as possible for the voting, before applying other techniques to improve positioning performance. The resulting data, which we plan to publicly release, lacks several values as well as mismatched readings in each stack, providing a more challenging fusion scenario. Such missing values allowed us to identify several fault scenarios, which we describe next.

Fault scenario: missing values. Due to some beacons not being reachable from the BLE receiver (*i.e.*, the laptop in our experiment). Missing values can reduce the reliability of the output measurement, since fewer candidate values are being considered, and potentially prevent from reaching a consensus value if most or all values are missing. A small amount of missing values, *i.e.*, less than the majority, does not prevent the system from converging to a common result, though it reduces the redundancy as well as the number of candidates considered, and as consequence the trustworthiness of the outcome. If the majority or all values are missing, the result would no longer be trustworthy, and the system should either revert to the last accepted result, or raise an error. Obviously, these failure scenarios should be accounted for when the voting behavior is defined. Due to the complexity possible and the variability by scenario, these behaviors are currently not modelled by VDX itself, and are instead left up to the client code to define. In our test-bed implementation, the error handling procedure was programmed

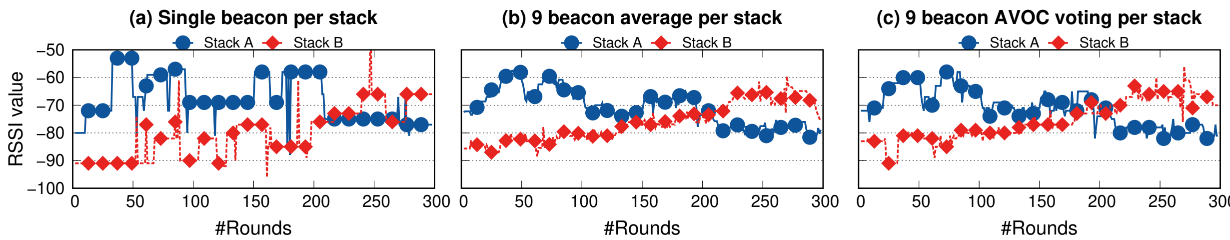

Figure 7: Results of the BLE beacon experiment. (a) shows the RSSI output when only one beacon from each stack is used. (b) shows the average RSSI value of all 9 beacons in the stack for each round. (c) shows the AVOC voting output for all 9 beacons in the stack. Averaging provides visibly less ambiguity in determining which stack is closest to the robot, when compared with the mean nearest neighbour selection used for Hybrid.

into the implementation of each algorithm. It is possible to make this behavior customisable, including the custom error-handling in the voting parameters of the schema definition.

Fault scenario: conflicting results. In case of conflicts, a majority agreement on outputs using automated voting is less likely to be reached. It is possible that a relative majority agrees on an output, but they are an overall minority, and no absolute majority exists. Especially in systems with small number of votes, ties might occur more easily and tie-breaking mechanisms kick in, such as proximity to the previous output. These fault scenarios clearly show that **setting the constraints of a voting system is non-trivial, and that voting algorithm implementations in a generic data fusion platform should be parametric**. In addition, there should be a voting specification declared for the target application, to take the desired error-handling behavior into account. Accounting for such fault scenarios allows to implement more robust versions of the algorithms. It is also possible to extend VDX in a future revision to support high-level descriptions of the desired fault handling policy. Examples of such policies include rejecting a round of measurements if there is no majority quorum or majority agreement.

We then tested our voting algorithms on the BLE experiment data. We used the same recorded values for each algorithm. We run voting between the 9 sensors of each stack to create 2 output values per round (*i.e.*, 1 per stack). We observed that the method for computing the history of each sensor has no effect. The output of all history-based algorithms overlaps completely. (They are not all plotted due to space constraints.) This is because the chaotic nature of the measurements meant the history values were all very low, as there were few agreements between the sensors. We observe however that the value collation method has impact, *e.g.*, averaging the weighted values, a mean-nearest-neighbour selection, *etc.* This created 2 algorithm groups, those averaging and those choosing the mean-nearest neighbour value, with every algorithm in each group performing identically to each other. In order to determine the best results, we study the number of rounds while it is ambiguous which stack of sensors is closest to the robot at any given time. Figure 7 presents these results.

Fig. 7-a is the reference: if each stack only had one sensor, it is not possible to identify the closest stack to the robot for most of the duration of the experiment. Simply averaging the values of the 9 sensors (Fig. 7-b) produces a less ambiguous result. We present the results using AVOC in Fig. 7-c. In spite of the method used to create a historical record for each sensor, what had the most impact on the output was whether or not the last step was to average the values or to select a value (with averaging being the better option in our experiment). In scenarios with high degree of noisy data, such as the BLE one, relying on historical record has no practical value. This confirms our initial assumption that **there is no optimal voting method for all applications**, despite common elements shared by our scenarios (*e.g.*, having to reconcile numerical values from a group of sensors). Thus we conclude that the answer to (Q3) depends of the specifics of the use case, and that the customisation offered by our specification allows us to address (Q4).

8 CONCLUSION AND FUTURE WORK

We have conducted an experimental study of history-aware voting in IoT and smart city/smart building applications. We demonstrated the performance of different algorithms on two different scenarios with different needs in terms of sensor fusion and presented out findings in terms of selecting an optimal algorithm for each scenario. Our findings show that inherently reliable systems can benefit more from history-aware voting as it can more easily root out more nuanced quality issues. On the other end of the spectrum, inherently unstable setups benefit more from smoothing and averaging techniques due to the unpredictability of the values rendering historical records ineffective. We also presented AVOC, our clustering-based voting approach, and demonstrated its effectiveness in bootstrapping a new group of sensors in a voting system. We showed how it can improve the accuracy of the voting result in the early rounds by eliminating outliers in-place, rather than discovering them based on past performance. We then proposed voting definition format VDX that can be used to describe a voting procedure to a compatible voter service running on an edge node. We plan to explore deeper and more realistic scenarios and the applicability of our voting methodology to providing improved data quality, as well as further develop the voting specification in order to field test a voter service prototype with a variety of compute-power-restricted setups.

REFERENCES

[1] 2022. *Bluetooth Technology Overview*. Retrieved June 27, 2022 from https://www.bluetooth.com/learn-about-bluetooth/tech-overview/

[2] 2022. *Cargo Sous Terrain*. Retrieved June 27, 2022 from https://www.cst.ch/

[3] 2022. *Graphic LCD Phidget*. Retrieved June 27, 2022 from https://www.phidgets.com/?tier=3&catid=48&pcid=41&prodid=963

[4] 2022. *Lego Mindstorms EV3*. Retrieved June 27, 2022 from https://education.lego.com/en-us/products/lego-mindstorms-education-ev3-intelligent-brick/45500

[5] 2022. *Light Phidget*. Retrieved June 27, 2022 from https://www.phidgets.com/?tier=3&catid=8&pcid=6&prodid=707

[6] 2022. *Wireless VINT Hub*. Retrieved June 27, 2022 from https://www.phidgets.com/?tier=3&catid=64&pcid=57&prodid=1143

[7] Ahmed Alahmadi and Ben Soh. 2012. A hybrid history based weighted voting algorithm for ultra-critical systems. In *2012 International Symposium on Communications and Information Technologies (ISCIT)*. 1122–1127. https://doi.org/10.1109/ISCIT.2012.6380861

[8] D.E. Bakken, Z. Zhan, C.C. Jones, and D.A. Karr. 2001. Middleware support for voting and data fusion. In *2001 International Conference on Dependable Systems and Networks*. 453–462. https://doi.org/10.1109/DSN.2001.941429

[9] Mohamed Ryad Boukhari, Ahmed Chaibet, Moussa Boukhnifer, and Sébastien Glaser. 2018. Voting algorithm approach for autonomous vehicle safe driving. In *2018 IEEE International Conference on Industrial Technology (ICIT)*. 327–332. https://doi.org/10.1109/ICIT.2018.8352198

[10] D. Comaniciu and P. Meer. 2002. Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24, 5 (2002), 603–619. https://doi.org/10.1109/34.1000236

[11] Manasi Das and Samar Bhattacharya. 2010. A Modified History Based Weighted Average Voting with Soft-Dynamic Threshold. In *2010 International Conference on Advances in Computer Engineering*. 217–222. https://doi.org/10.1109/ACE.2010.45

[12] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining* (Portland, Oregon) *(KDD'96)*. AAAI Press, 226–231.

[13] Yufei Fang, Zhiguang Shan, and Wei Wang. 2021. Modeling and Key Technologies of a Data-Driven Smart City System. *IEEE Access* 9 (2021), 91244–91258. https://doi.org/10.1109/ACCESS.2021.3091716

[14] Seungmyeong Jeong, Seongyun Kim, and Jaeho Kim. 2020. City Data Hub: Implementation of Standard-Based Smart City Data Platform for Interoperability. *Sensors* 20, 23 (2020). https://doi.org/10.3390/s20237000

[15] M. A. Kassab, H. S. Taha, S. A. Shedied, and A. Maher. 2014. A novel voting algorithm for redundant aircraft sensors. In *Proceeding of the 11th World Congress on Intelligent Control and Automation*. 3741–3746. https://doi.org/10.1109/WCICA.2014.7053339

[16] Sefki Kolozali, Maria Bermudez-Edo, Nazli Farajidavar, Payam Barnaghi, Feng Gao, Muhammad Intizar Ali, Alessandra Mileo, Marten Fischer, Thorben Iggena, Daniel Kuemper, and Ralf Tonjes. 2019. Observing the Pulse of a City: A Smart City Framework for Real-Time Discovery, Federation, and Aggregation of Data Streams. *IEEE Internet of Things Journal* 6, 2 (2019), 2651–2668. https://doi.org/10.1109/JIOT.2018.2872606

[17] G. Latif-Shabgahi, J.M. Bass, and S. Bennett. 2001. History-based weighted average voter: a novel software voting algorithm for fault-tolerant computer systems. In *Proceedings Ninth Euromicro Workshop on Parallel and Distributed Processing*. 402–409. https://doi.org/10.1109/EMPDP.2001.905068

[18] G. Latif-Shabgahi, J.M. Bass, and S. Bennett. 2004. A taxonomy for software voting algorithms used in safety-critical systems. *IEEE Transactions on Reliability* 53, 3 (2004), 319–328. https://doi.org/10.1109/TR.2004.832819

[19] Billy Pik Lik Lau, Sumudu Hasala Marakkalage, Yuren Zhou, Naveed Ul Hassan, Chau Yuen, Meng Zhang, and U-Xuan Tan. 2019. A survey of data fusion in smart city applications. *Information Fusion* 52 (2019), 357–374. https://doi.org/10.1016/j.inffus.2019.05.004

[20] Yiu-Wing Leung. 1995. Maximum likelihood voting for fault-tolerant software with finite output-space. *IEEE Transactions on Reliability* 44, 3 (1995), 419–427. https://doi.org/10.1109/24.406576

[21] Deborah Lupton. 2021. 'Flawed', 'Cruel' and 'Irresponsible': The Framing of Automated Decision-Making Technologies in the Australian Press. https://doi.org/10.2139/ssrn.3828952

[22] Negin Minaei. 2022. Future Transport and Logistics in Smart Cities: Safety and Privacy. In *Smart Cities*. CRC Press, 113–142.

[23] Nwamaka U. Okafor, Yahia Alghorani, and Declan T. Delaney. 2020. Improving Data Quality of Low-cost IoT Sensors in Environmental Monitoring Networks Using Data Fusion and Machine Learning Approach. *ICT Express* 6, 3 (2020), 220–228. https://doi.org/10.1016/j.icte.2020.06.004

[24] Dau Pelleg and Andrew Moore. 2000. X-means: Extending K-means with Efficient Estimation of the Number of Clusters. In *In Proceedings of the 17th International Conf. on Machine Learning*. Morgan Kaufmann, 727–734.

[25] Oumaima El Rhazal and Mazri Tomader. 2019. Study of Smart City Data: Categories and Quality Challenges. In *Proceedings of the 4th International Conference on Smart City Applications* (Casablanca, Morocco) *(SCA '19)*. Association for Computing Machinery, New York, NY, USA, Article 4, 7 pages. https://doi.org/10.1145/3368756.3368965

[26] A.C. Torres-Echeverría, S. Martorell, and H.A. Thompson. 2012. Multi-objective optimization of design and testing of safety instrumented systems with MooN voting architectures using a genetic algorithm. *Reliability Engineering & System Safety* 106 (2012), 45–60. https://doi.org/10.1016/j.ress.2012.03.010

[27] Junliang Wang, Chuqiao Xu, Jie Zhang, and Ray Zhong. 2022. Big data analytics for intelligent manufacturing systems: A review. *Journal of Manufacturing Systems* 62 (2022), 738–752. https://doi.org/10.1016/j.jmsy.2021.03.005

[28] Steven Euijong Whang, Yuji Roh, Hwanjun Song, and Jae-Gil Lee. 2021. Data Collection and Quality Challenges in Deep Learning: A Data-Centric AI Perspective. *CoRR* abs/2112.06409 (2021). arXiv:2112.06409 https://arxiv.org/abs/2112.06409

[29] Juthatip Wisanmongkol, Ladawan Klinkusoom, Taweesak Sanpechuda, La-or Kovavisaruch, and Kamol Kaemarungsi. 2019. Multipath Mitigation for RSSI-Based Bluetooth Low Energy Localization. In *2019 19th International Symposium on Communications and Information Technologies (ISCIT)*. 47–51. https://doi.org/10.1109/ISCIT.2019.8905164

[30] Yadi Wu, Senlin Cheng, and Xiaohao Yan. 2020. Study on Improved Algorithm of RSSI Correction and Location in Mine-Well Based on Bluetooth Positioning Information. In *Proceedings of the 4th International Conference on Computer Science and Application Engineering* (Sanya, China) *(CSAE 2020)*. Association for Computing Machinery, New York, NY, USA, Article 146, 6 pages. https://doi.org/10.1145/3424978.3425131

[31] Nashez Zubair, Niranjan A, Kiran Hebbar, and Yogesh Simmhan. 2019. Characterizing IoT Data and its Quality for Use. *CoRR* abs/1906.10497 (2019). arXiv:1906.10497 http://arxiv.org/abs/1906.10497