



A Sensor Ranking Approach for Edge–Fog–Cloud Environments

Felipe S. Costa¹ · Silvia M. Nassar¹ · Mario A. R. Dantas^{2,3}

Received: 27 September 2021 / Accepted: 12 April 2023 / Published online: 13 May 2023
© The Author(s), under exclusive licence to Springer Nature Singapore Pte Ltd 2023

Abstract

This research work presents a method for ranking sensors using the data produced by these devices. The method classifies the data, identifying the occurrence of failures in sensors and anomalies in the environments, aiming to maintain a reliability-based sensor ranking list. To generate the ranking list, overcoming the challenges implicit in this activity, the method adopts the theory of active perception as a basis. This approach divides the perception activity into levels that progressively add more sense to the information generated by the sensor, thus providing more reliability to the task of classifying the data generated by the sensors, without spending resources. This step allows you to create the ranking list in which the most reliable sensors will be at the top of the list. This list is managed through a distributed hash table to meet the distributed requirement of the Internet-of-Things (IoT) environment. The proposal was evaluated using four real data sets. The results of this research demonstrate that the proposed approach can provide high reliability in the use of sensor data, using low computational resources and, thus, reducing latency in the processes of selection and use of sensors.

Keywords Sensor ranking · Active perception · Anomaly · Failures · Reliability · Latency

Introduction

In the IoT environment, large-scale data detected in the environments surrounding the physical world can be distributed, processed, and used by various services and applications to enable autonomous and intelligent decision-making [1, 2]. The number of objects added to IoT is approaching 200 billion, and the number of sensors is already over 50 billion [3]. As a result, the data generated by IoT far outweigh the capabilities of existing IT architectures and infrastructures,

and its real-time requirement will also much emphasize the available computing capacity.

The cloud computing model is an efficient alternative for managing large volumes of data, allowing greater use without degrading performance. It is achieved through the implementation of homogeneous and theoretically unlimited computing, storage, and network components. However, these advantages become an issue for latency-sensitive applications, which require nearby network nodes to meet their delay requirements [4].

To meet this demand, CISCO proposed the creation of a platform that it called Fog Computing, or, briefly, Fog, simply because fog (Fog) is a cloud close to the ground (network edge) [4]. In this computational architecture, each smart object is connected to one of the Fog devices. The devices can be interconnected, and each one is linked to the cloud [5]. The architecture allows sensors from the IoT environment (Edge) to produce useful data about the environment around them, aiming to make this environment more intelligent. This task is made possible through access to the necessary information related to the environments, the collection, and analysis of past and present data. These data allow optimal decisions about people and their environments, preferably in real time [6].

This article is part of the topical collection “Cloud Computing and Services Science” guest edited by Donald Ferguson, Markus Helfert and Claus Pahl.

✉ Felipe S. Costa
felipekosta@gmail.com

Silvia M. Nassar
silvia.nassar@ufsc.br

Mario A. R. Dantas
mario.dantas@ice.ufjf.br

- ¹ Department of Informatics and Statistics (INE), Federal University of Santa Catarina (UFSC), Florianopolis, Brazil
- ² Department of Computer Science (DCC), Federal University of Juiz de Fora (UFJF), Juiz de Fora, Brazil
- ³ INESC P&D Brazil, Santos, Brazil

To allow data sharing, using these concepts, IoT middleware solutions have been adopted [7]. IoT middleware is used to collect data from sensors to be used by different applications, acting as an interface between the user/application and the IoT network [8].

An IoT middleware usually offers resources for acquisition, the discovery of new sensors, indexing, ranking, and query [9]. The indexing process involves storing and indexing the data collected from the resources, considering that the resources include sensors, devices, and services, to allow a fast and efficient search. The IoT resources ranking is performed by prioritizing criteria as quality of data, device availability, energy efficiency, and network latency. This task can be performed based on the generated observation (content) and the measurement data (context). Examples of use are applications such as industry and health that require reliability associated with the processing and provision of data with low latency [10]. Only these two resources, indexing, and ranking will be considered in this research work.

The complexity and heterogeneity of objects connected to the IoT environment are also present in the data generated by them. Thus, a set of challenges arises from the vast, changeable, and diversified mass of data (data-intensive), on how to handle and use these data [11]. According to the National Science Foundation (NSF) [12], some of the significant challenges of this environment include supporting real-time applications, from enabling artificial intelligence to controlling feedback in cyber-physical systems, by reducing latency.

Therefore, IoT systems with intensive use of data are challenged by the dynamics and uncertainty of the circumstances in which the system is located and by the recognition and search quickly in an ocean of noisy data coming from heterogeneous devices, meeting requirements that evolve rapidly [11].

To overcome these challenges, this research paper presents a sensor ranking approach. The method proposes to reduce latency and increase the reliability in the use of the information generated by the sensor. The construction of the ordered list of sensors is carried out based on the quality of the collected data. The work also intends to help in identifying extreme events or failures in the devices, serving as a support tool in the control of the monitored environments. For this, the method uses the content (data generated by the sensor) and context information related to the state of the sensor at the exact moment of capture. To evaluate the data, two evaluation techniques are used. The first one is applied to most of the data and the second one is applied only to uncertain data, which considers the dynamic aspect and uses a time slice of data to define the sensor state. Finally, using context parameters, the method updates the ranking list.

This article describes the evolution of the research work presented in [13]. Unlike the previous work, the second level

of the proposal now uses the nearest neighbor algorithm combined with the Interquartile Range (IQR) metric. The IQR metric was also used to generate labels for the data in the training step. Fuzzy logic is now only used in the third and last level of the proposal, to generate the final label related to the current state of each sensor. In addition, the proposal now brings an implementation of the computational architecture with a distributed approach, through the maintenance of ranking lists in distributed hash tables. Different datasets were also used, aiming to work with datasets used in more recent works. Additionally, the new datasets were also studied in works that identify the points of anomaly and failure present in each set, thus providing a way to assess the accuracy when using these data.

The remainder of the paper is organized as follows: the section "Related Works" section presents the related works with this proposal. The section "Background" presents the concepts related to the proposal. In the section "Sensor Ranking", the details of this proposal are discussed. The materials and methods used in this work are presented in the section "Evaluation". In the section "Experimental Results", the results obtained are presented and discussed in the section "Discussion". The section "Conclusions and Future Works" brings the conclusions and future works.

Related Works

Applications, such as industry, agriculture, and healthcare, require reliability associated with computing in services that require low latency requirements [10]. In this sense, many works have been dedicated to the discovery, indexing, and ranking of sensors to provide a fast and safe way to recover data from IoT devices. In this section, the most relevant works related to the ranking of sensors are presented.

In Table 1, it is possible to see a list of the works focusing on the ranking of sensors. The second column identifies whether the architecture proposed by the work can be implemented with a distributed approach. The third column identifies which tier (Cloud, Fog, or Edge) the data are processed in the proposal. The fourth column presents the computational complexity of the proposal (computational complexity less than $O(n^2)$ is considered low). The fifth column, Ranking Calculation, identifies how the values used in the ranking calculation are generated: Before (a user query), After, or Diverse (when part of the values is previously calculated, and another part is calculated after the user query). The sixth column identifies how many steps or techniques the work uses to evaluate the data and generate the ranking of sensors. The seventh shows which works consider using historical sensor data (time) and the eighth column if the work considers fault handling (an anomaly in the environment or failure in capture).

Table 1 Related works

Work	Features characteristics						
	Distributed	Tier	Complexity	Ranking	Evaluation	Time	Anomaly/failure
Costa et al. [13]	No	Fog	Low	Before	Two	Yes	Yes
Nesa and Banerjee [15]	Yes	Cloud	High	Before	One	No	No
Dautov and Distefano [16]	Yes	Edge	Low	Diverse	One	No	No
Ruta et al. [17]	Yes	Fog/cloud	Low	Diverse	One	No	No
Kakunsi and Candra [18]	No	Cloud	Low	Diverse	One	No	No
Kertiou et al. [8]	Yes	Fog/cloud	High	Before	One	No	No
Dilli et al. [19]	No	Cloud	Low	Before	One	No	No
Hussain et al. [20]	No	Cloud	Low	Before	One	No	No
Nunes et al. [21]	No	Cloud	Low	Before	One	No	No
Kang et al. [22]	Yes	Edge	Low	Before	One	No	No
Zhang et al. [23]	Yes	Fog/edge	High	Diverse	One	Yes	No
Saxena [24]	No	Cloud	Low	Before	One	No	No
Wang et al. [25]	Yes	Fog/cloud	Low	Diverse	One	No	No
Yuen and Wang [1]	No	Cloud	Low	Diverse	One	No	No
Cabral et al. [26]	Yes	Cloud	Low	Before	One	Yes	No
Niu et al. [27]	No	Cloud	Low	Diverse	Two	No	No
Perera et al. [28]	Yes	Cloud	Low	Before	Two	No	No
Truong et al. [29]	Yes	Fog/Cloud	Low	Before	One	Yes	No
Walters et al. [30]	Yes	Fog	Low	After	One	No	No
Wang et al. [31]	Yes	Fog	Low	Before	One	No	No
Ostermaier et al. [32]	No	Cloud	High	Before	Two	Yes	No

The previous version of this research [13], named Greatest of Actual Time (GoAT), covers all aspects present in related works, except the feature of maintaining the classification list in a distributed way, which is extremely important to meet the latency requirements present in this scenario. Comments below on related work do not include the previous version of this research. This means that statements like "none of the work", for example, disregard GoAT.

According to the bibliographic research carried out, only one of the articles [14] considers aspects related to identifying failures or anomalies. Despite this, the work does not describe how it is done and does not show results, so it was not included in the table of related works. None of the related works considers the fact that sensor failures can be transient or persistent, given the dynamic nature of the IoT environment, so the generated ranking may not reflect the actual state of the sensors. The researched works also do not create the ranking previously (make the ordering available before receiving queries), making it impossible to immediately respond to a request for a list of trusted sensors. The use of the theory of active perception for ranking sensors was not used in any related works. The design of the proposal associated with the theory of perception aims to create a method that adds knowledge to the reasoning at each level of the theory, improving the data analysis process.

Regarding this aspect, only three studies use more than one technique to evaluate the data.

Thus, considering the use of two evaluation techniques, one of them considering the dynamic aspect using the evaluation of the data referring to a time slice; regarding the classification of the state of the sensors in the normal, anomalous, and failure categories and, finally, considering the previous generation of the ranking list, thus reducing the latency in queries; and finally considering the ability to manage the ranking list of sensors in a distributed way, make evident the important scientific contribution of this proposal.

The literature review carried out presents the various approaches used in the sensor ranking activity, demonstrating the importance of this activity for the development of the IoT. However, despite the related proposals showing promising results, there are still some challenges and opportunities:

- Properly treat the data generated by sensors in the IoT environment, differentiate data that represent real measurements of the environment around the sensor from data generated by failures or interference.
- Improve the sensor selection processes to reduce the amount of data to be analyzed in decision-making, allowing the time between analysis and action to be as short as possible, meeting low latency requirements.

- Improve the reliability of data obtained from sensors in the IoT environment, to reduce uncertainty in decision-making.
- Provide a fully distributed solution capable of meeting low latency requirements.

This work aims to contribute to overcoming these challenges.

Background

This section describes the basic concepts related to the techniques and algorithms employed. Concepts related to the Gradient Tree Boosting, used in the first data evaluation, are described. Nearest Neighbors, used to identify anomalous segments in time series, are also presented. To compare these segments, a similarity measure is used to define an anomaly index for each segment. Using the generated anomaly index, the Interquartile Range metric is used to divide the data into normal, anomaly, and failure types. This division uses the anomaly index to evaluate each data segment and thus define which category it belongs to. The Fuzzy Inference System (FIS), used to evaluate the sensor state, is also presented. The Quality of Context (QoC) theory, which one of the parameters is used, and the Distributed Hash Table (DHT), used to manage the distributed sensor rank lists, are also presented.

Gradient Tree Boosting

The first evaluation of the sensor data in this work uses an algorithm belonging to the Gradient Tree Boosting (GDB) class, as shown in Fig. 1. Figure 1 demonstrates the application of the GDB algorithm to the Intel Lab dataset [33], which contains information about data collected from sensors, mounted on a Mica2Dot¹ board, deployed at the Intel Berkeley Research Lab (IntelLab). The algorithms of the GDB class are algorithms that start by creating only one decision tree. After training, the values for which the tree makes incorrect predictions are recorded. Then, a new tree is created to see how much the other tree makes mistakes. Then, the cycle is repeated up to a certain limit, always trying to reduce the error rate [34]. The XGBoost implementation improves the performance of the conventional gradient augmentation tree by introducing two techniques: weighted quantile sketching (a data structure that supports merging and removing operations) and dispersion recognition split location (default direction at each node in the tree). XGBoost

¹ Mica2Dot is a board used to enable low-power wireless sensor networks that allows connecting various types of sensors.

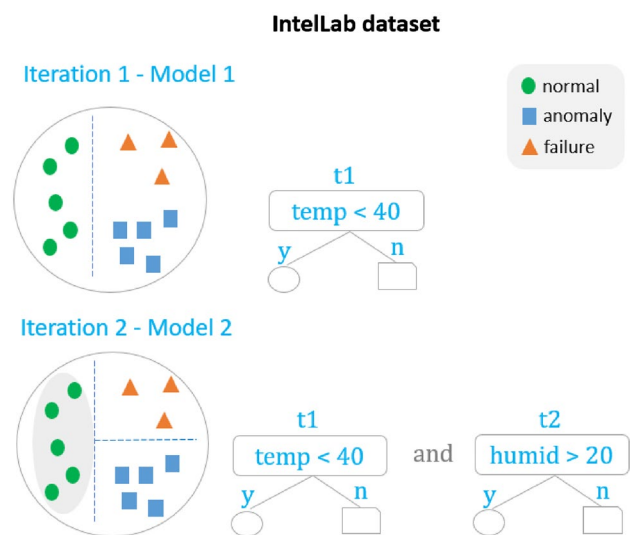


Fig. 1 Classification of a single sensor data [13]

has been applied to several machine learning problems and has obtained better results than the other algorithms [35]. This algorithm has linear time complexity.

Nearest Neighbors Algorithm

Nearest neighbor techniques are widely applied to detect anomalous segments [36]. The technique uses a measure to generate an index that represents the similarity between each segment in the time series and all other segments. Segments with similar sequences of samples represent closest neighbors. The similarity of a segment to its nearest neighbors is the basis for its anomaly index. In particular, if a segment is anomalous, its similarity to any other segment will be low [37].

Similarity Measure

To measure the similarity between segments, the Euclidean distance metric was used. The motivation for using the Euclidean distance is that a zero distance, in other words, maximum similarity, occurs only when two segments are very similar in all n samples [37]. The Euclidean metric is presented in Eq. 1

$$d(u, v) = \sqrt{\sum_{i=1}^n (u_i - v_i)^2}. \quad (1)$$

Interquartile Range

The Interquartile Range (IQR) metric is useful to indicate whether the values of a dataset can be considered outliers

[38]. Quartiles are the separators that divide a numerical dataset into four equal parts. The first quartile (Q1) is the value of the set that delimits the 25% smaller values (25% of the values are less than Q1 and 75% are greater than Q1). The second quartile (Q2) is the median itself, which separates the 50% smaller from the 50% larger values. The third quartile (Q3) is the value that delimits the 25% higher values (75% of the values are lower than Q3 and 25% are higher than Q3). Thus, the first range of data (first 25% of data) starts at the smallest value and goes to Q1. The second range between 25 and 50% are values greater than Q1 and less than Q2. The third range corresponds to values between 50 and 75%, that is, values greater than Q2 and smaller than Q3. The fourth and last range corresponds to values between 75 and 100% (or the maximum value) of the data [39]. The interquartile range is the difference between Q3 and Q1.

The idea is to create thresholds using the value of the IQR index to facilitate the identification of values that are far from the most frequent values, and in some cases, even extrapolating the valid range of domain values.

Fuzzy Inference System (FIS)

The fuzzy logic is inspired by real-world phenomena in which events can hardly be considered entirely false or completely true [40]. Fuzzy logic offers a method, based on rules and mathematical sets, for the treatment of inaccuracy, thus being able to deal with imprecise or qualitative terms, such as "Low", "Medium", and "High" which cannot be expressed using binary logic.

These characteristics of fuzzy logic will be used to manage the current state of each sensor, with more qualitative information than just the value of the *trustworthiness* parameter.

Quality of Context (QoC)

One of the main problems in the way of intelligent IoT is to understand the "context" or to understand the environment, situation, or state using data from sensors and then act autonomously. This is called "context-aware computing". However, context information of the same entity, of the same type, and obtained simultaneously may differ in terms of accuracy, probability of correction, reliability, etc. Therefore, it can be said that the information has a quality, and this quality is called "Quality of Context" (QoC) [41]. QoC can help to resolve uncertain and conflicting situations about context information. Therefore, context-aware applications can take advantage of QoC if they are provided with usable QoC metrics that are evaluated considering their requirements regarding the collection, processing, and provision of context information [42]. It is important to highlight the difference between data and contextual information: "data"

are raw data, taken directly from its source (sensors), and "context information" is processed data obtained by processing raw data and usually with added metadata [43].

Distributed Hash Table (DHT)

This work considers the use of gateways, in which the data of each sensor will be processed and evaluated. The gateways exchange information in a ranking list management model using the Distributed Hash Table (DHT) technique. The choice of this technique is due to its high degree of scalability and the flexible support for query and update operations. DHT is a system with a decentralized approach that provides fast mechanisms for storage, queries, and updates. DHT is built on overlapping networks in which network objects are spread and identified with unique keys [44].

The Apache Cassandra implementation was chosen to enable exchanging information between the distributed lists [45]. Cassandra achieves horizontal scalability by partitioning all data stored in the system using a hash function called consistent hashing. Consistent hashing is similar to a Chord model [46], which is based on DHT. This model provides a scalable and efficient protocol for dynamic research in P2P systems with frequent node inputs and outputs. It indicates how to locate a resource, how a new node enters the system, and how to deal with failures. DHT is used for both file storage and resource tracking. The nodes are placed in ascending order according to their IDs in a virtual ring. To improve scalability, the Chord node does not need routing information about all other nodes, but only a number of $O(\log N)$ nodes, and therefore, the search needs a maximum of $O(\log N)$ messages.

Active Perception Theory

The active perception uses models to inform the context-dependent adjustments in sensors, aiming to direct the sensors toward phenomena of greater interest [47].

Perception is not passive but active. Perceptual activity is exploratory, probing, searching. Perceptions do not merely fall on sensors as rain falls on the ground. During the act of looking, the pupils adjust to the level of illumination, the eyes bring the world into sharp focus, converge or diverge, the heads move to improve the vision of something [48]. This adaptability is crucial for survival in an uncertain world. Thus, the problem of active perception can be defined as the use of control strategies applied to the data acquisition process, which, in turn, will depend on the interpretation of the data and the objective of the task [48].

Considering that perception is often dependent on context, other senses, and time, the active perception approach is divided into four levels: sensation, perception, perception over time, and active perception. Sensation is the process

by which the captured data enter the system. After sensation, perception interprets these data and gives it meaning. By observing a process for a particular time, it is possible to improve the interpretation of these data. This process is called perception over time, in which data are classified according to time, indicating that it is possible to evaluate perception concerning another perception. Finally, active perception includes reasoning, decisions, and actions based on the information received [49, 50].

Sensor Ranking

This section describes the method proposed in this research paper. Before describing the process, itself, some important concepts used in the proposal are defined. The datasets used in evaluating the method, as well as the experimental computing platform used in the tests, are also described.

Time Series

The proposed method uses time series data. A time series is a finite sequence of samples, taken at increasing instants of time. In the proposal, the data of a measurement are evaluated first individually and then in small subsets that refer to the data generated during a given time slice, or segments. A segment refers to a sequence of time series samples ordered in time.

Normal and Abnormal Data

In this proposal, if a given sample follows the trend of the rest of the data, then it is considered normal and, otherwise, it is considered abnormal. If considered normal, no further assessment is necessary. On the other hand, if it is considered abnormal, a second evaluation is made.

Persistent and Transient Failures

Sensor failures can be persistent or transient. As it is common for failures to occur in this type of communication (in capture and transmission), if the interference occurs in only one measurement, generating a single incorrect value, there is a high probability that these data will be misinterpreted. The way to get around this problem is through the analysis of the data produced in a time slice. The reason is that disturbances related to energy sources and means of transmission are mainly transient, caused by imbalances that lead to momentary instabilities [51].

Failures and Anomalies

To verify the behavior of each sensor, the data generated by them are classified with the labels normal, abnormal, anomaly, and failure. Data marked as normal are those that indicate that both the sensor and the environment in which the measurement is being made are stable (no anomalies) and the reading process was not affected by interference during data capture and transmission. Data in this range of values represent the data most frequently generated by the sensor.

Data considered abnormal are evaluated again, but in this case, considering the temporal aspect (slice of time). In this second evaluation, a piece of data can also be considered normal, even though it was considered abnormal in the first evaluation. This occurs in cases where the data were considered abnormal due to a transient failure. If a measurement affected by a fault is analyzed as a time series, the sequence or sample segment corresponding to the fault will be different from the normal trend of the time series. However, this will only be true if the deviation from normality is large enough to influence the measurement of the entire segment, which occurs in persistent failures. In the case of transient failures, a single abnormal data point, the influence on series measurement is less significant. Thus, when analyzed in a data segment, the disturbance caused is mitigated due to the presence of several other normal values.

In this second evaluation, the data can also be marked as anomalies, indicating the occurrence of anomalies in the environment. This occurs when several of the values in the data segment are relatively far from the normal range, but within a range of values that may exist in the environment's domain. This kind of data does not happen that often.

Finally, data considered as failure are data that occur due to persistent failures in devices, affecting the capture or transmission of data. In this case, the range of values is much further away from the other values, extrapolating the limits of valid values in the domain

$$Q3 + FF * IQR < ai > Q3 + AF * IQR \quad (2)$$

$$ai > Q3 + FF * IQR. \quad (3)$$

Equations 2 and 3 use the anomaly index (ai) and IQR metric to label samples. Equation 2 is used to verify if a value is considered anomaly and Eq. 3 is used to label values considered failures. The factors anomaly factor (AF) and failure factor (FF) used in the formulas were defined in the data set training stage to better identify the discrepant data in each set.

Figure 2 shows an example of the application of this metric: the points located below the gray line (close to 0.035) represent the data considered normal; the points between the gray and the green line (close to 0.05) represent the

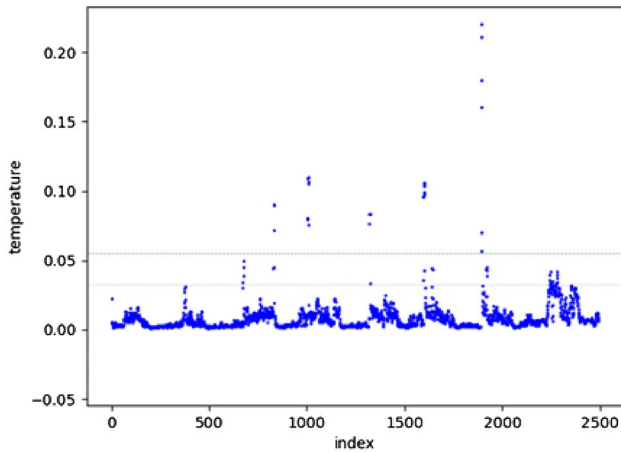


Fig. 2 Division of data into normal, anomaly, and failure values using the IQR metric

anomalous data, and the points above the green line represent probably failure data. No data exist below zero, because the values represent the temperature in an environment that does not have negative temperatures.

The samples marked as failures refer to noise in the transmission or capture of sensor data, and the data considered as anomalies represent the presence of a phenomenon in the observed environment surrounding the sensor.

Three-Level Sensor Ranking

The ranking method proposed in this work is based on the theory of active perception [49, 50]. Similarly, in this proposal, the task of ranking sensors is performed through an analysis that allows different levels of reasoning according to the context found.

Figure 3 shows an overview of all steps involved in this proposal and Fig. 4 presents an overview of the proposed algorithm only.

In the pre-processing step shown in Fig. 3, the samples are labeled, the variable data are discretized, and the training phase of the algorithms used is performed. Then, in the processing step shown in the figure, it is possible to identify the layer (Edge, Fog, and Cloud) the processes (acquisition, modeling, reasoning, and distribution) of the middleware that are executed (boxes with a gray background). Below the identification of each process, in orange, the functions performed in the standard middlewares are represented. In the boxes with a green background, the processes present in the architecture of the developed experimental environment are described, represented by the text in black. In these boxes, the orange text represents the process performed or the input data sent to the tools used, identified by the gray text.

Finally, the ranking process is detailed in the last section (indicated by the up arrow), on the top right (Active

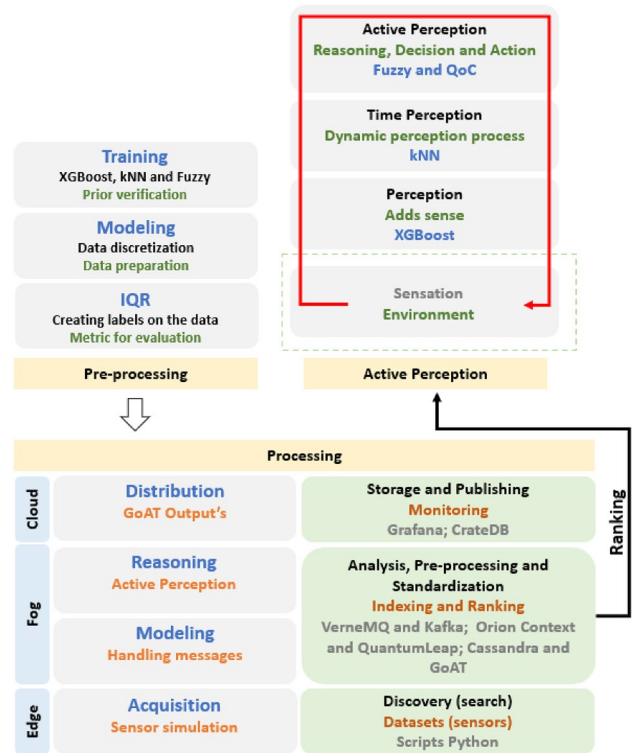


Fig. 3 Overview of all steps involved in this proposal: tiers of processing, processes, and tools for each tier and active perception (proposed)

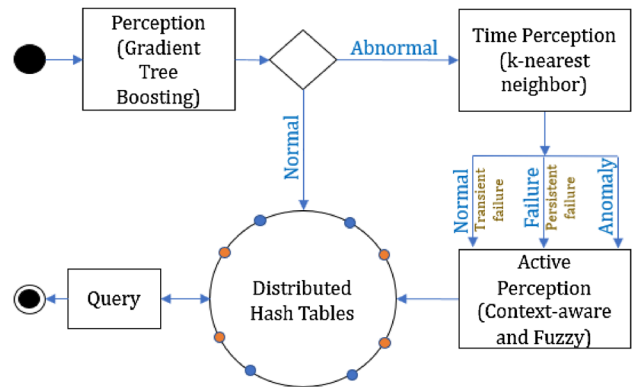


Fig. 4 Algorithm for ranking sensors

Perception), which shows the information flow of the proposal. The image displays the levels of active perception starting with the activity of collecting data in the environment (following the direction of the red arrow), perception, perception over time, and active perception. The first level of the active perception, sensation (wrapped in a dashed line), is carried out by the sensors and, therefore, is outside the scope of this work. This activity is emulated by scripts in the evaluation tests performed. In this part of the image, the

flow returns to the environment, indicating the execution of the actions taken at the level of active perception. Below the identification of each level, characteristics of the respective activity, and the techniques used to implement them are presented. Each step is identified according to the theory of active perception (black text), as well as the function of each one (text in green) and the algorithm or technique used (in blue).

Figure 4 presents a general flowchart of the algorithm used in the decision process during information processing. The flow shows the levels of the proposal and the decisions taken at the exit of the first and second level. As can be seen, the second and third levels are only executed if the information on the first level (perception) is classified as abnormal. Otherwise, the flow goes directly to updating the distributed hash tables. Considering the case of the data being classified as abnormal, it is evaluated again at the second level (time perception) to see if it is a normal, a fault or if it is a change in the environment, that is, an anomaly. The flow continues to the third level, where the amount of each type of data generated by the sensor is used to define its current state as normal, fault, or anomaly. Finally, the process ends by updating the distributed hash tables. The image also displays a query module through which the distributed ranking list can be queried. In the current state of the proposal, this module has not yet been implemented and queries are made manually. In the following subsections, the implementations of each level will be detailed.

First Level

At this level, which was implemented using XGBoost, the model created in the training phase is used to classify data into normal and abnormal. The training of the algorithm was carried out with a sample of 25,000 cases from each dataset. In this proposal, if the segment that does not follow the trend of normal data will be considered abnormal, and after that, a second evaluation will be made to decide whether the data are an anomaly or a failure.

Second Level

This level uses time to evaluate the data provided by the sensor. The adoption of this approach (time) allows distinguishing transient from persistent failures. If a measurement affected by a failure is analyzed as a time series, then the sequence of samples or segments corresponding to the failure will be distinct from the normal trend of the time series. In other words, this segment will be considered anomalous [36].

To implement this concept, the nearest neighbor's algorithm was chosen. The basic idea is to use a similarity measure to assess the similarity between each segment in the

time series and all other segments. Similar segments, that is, with similar sequences of samples, are known as nearest neighbors. These must be differentiated from neighbors almost in time, which are adjacent segments with overlapping samples [52].

The anomaly index, a measure of similarity between each segment, calculated using the Euclidean distance metric, is used to generate the thresholds values that divide data into normal, anomaly, and failure categories. These thresholds are generated by applying the IQR metric on the calculated anomaly indices. As a result, the data are divided into three categories according to how far each value is from most of the data (those furthest to the center of the normal distribution), as shown in Fig. 2. The limits are generated in the training step of the algorithm and then used in the tests. This allows the computational cost to be very small when executing the algorithm.

Third Level

In the final level, the sensor index is updated with this new information. To update the index, the data are processed using the concepts of Quality of Context (QoC) theory. The *trustworthiness* parameter of QoC (Eq. 4) is used as an indicator to generate the ranking of sensors. In the *trustworthiness* parameter, the value 0 (zero) means that this context source is not reliable, and 1 (one) represents total reliability in the context source [53]. In Eq. 4, the parameter T is the reliability, the ctx_i is the set of trusted context elements for a sensor (labeled as "normal"), and W is the total number of context elements and must be greater than zero. The terms reliability and trustworthiness have the same meaning, and the second was used to name the QoC parameter

$$T(ctx_i) = \frac{\text{Number of Reliable Samples}(ctx_i)}{W}. \quad (4)$$

The value of the *trustworthiness* context parameter is then evaluated using a FIS. The modeled FIS has an input variable and an output variable. Figure 5 presents the fuzzy pertinence functions of the input variable (*trustworthiness*) and the output variable (*sensor state*). The *sensor state* variable shows the functions with the output value already set.

As a final step, regardless of the level that was last performed, the sensor ranking is updated with the new values of the *trustworthiness* parameter.

Evaluation

In this section, the computational architecture and the data sets used in the tests of the proposal are presented.

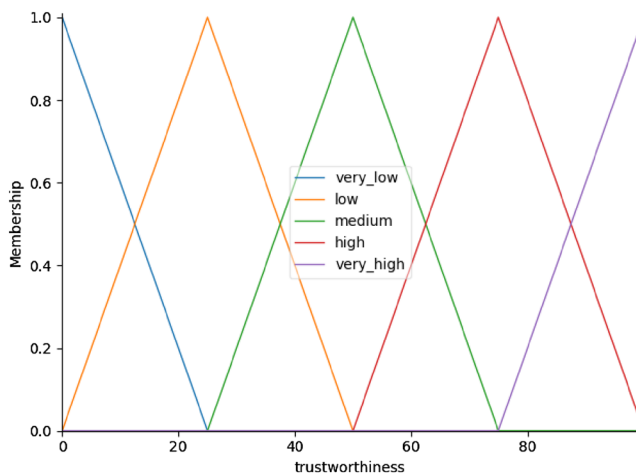


Fig. 5 Fuzzy membership functions

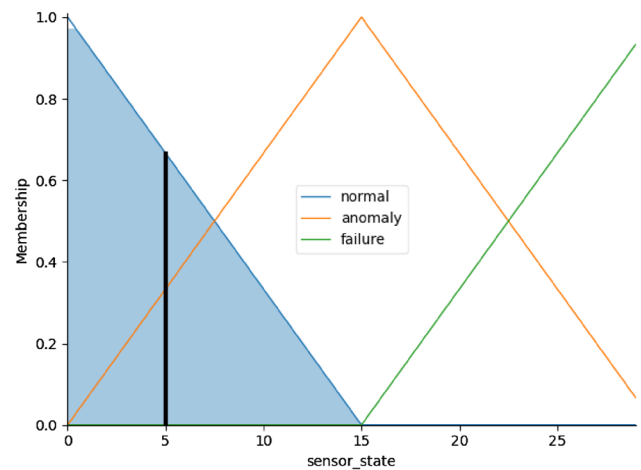
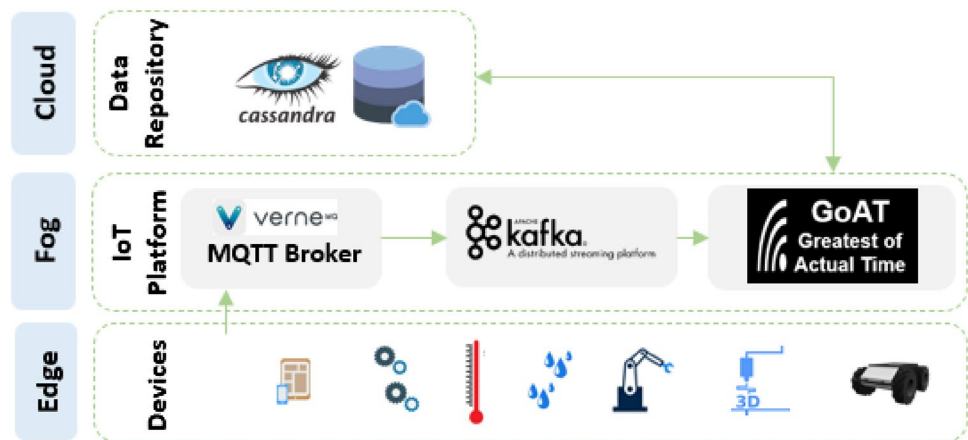


Fig. 6 Proposed experimental environment—IoT platform



IoT Platform

Given the distributed nature of IoT devices, it is necessary to adopt a distributed architecture capable of ensuring the scalability of the proposal. The exponential growth in the number of sensors in the IoT environment justifies the requirement for this type of approach. Thus, the proposed method will use a distributed approach using a cooperation model between Cloud and Fog Computing. The Apache Cassandra framework was used to manage the distributed ranking lists.

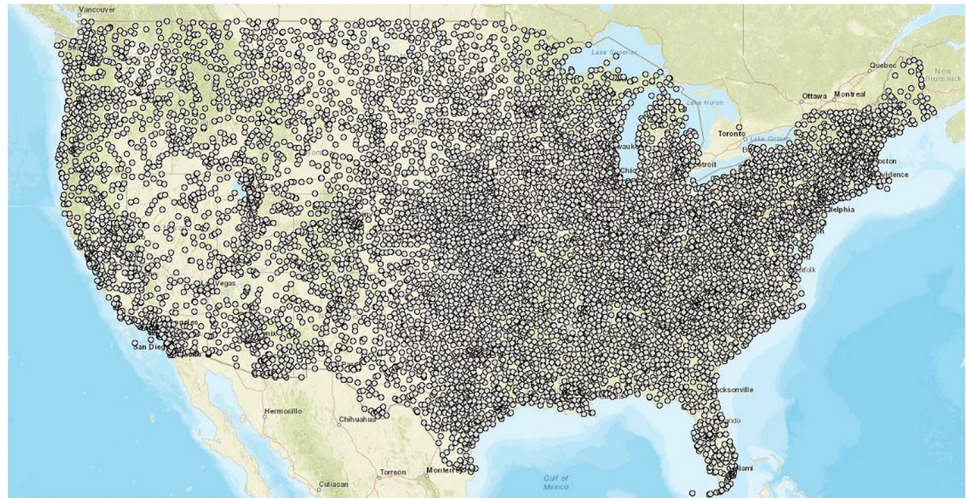
The computational architecture used in the tests is an adaptation of the FASTEN project [54]. Flexible and Autonomous Manufacturing Systems for Custom-Designed Products (FASTEN) is a project funded by the EU Horizon 2020 program. FASTEN's Industrial IoT Platform aims to manipulate data from devices (robots, sensors, etc.) in industrial environments and serves as an intelligent data repository for the optimization and forecasting layer, allowing to improve the quality of the services offered and while meeting Industry 4.0 [55] requirements.

Figure 6 presents architecture used. The changes about the FASTEN project consist in the use of the sensor ranking solution (GoAT [13]) and the use of distributed rank lists (Apache Cassandra). The bottom box in Fig. 6 represents the sensors (devices). The data from these sensors are sent to the platform using the Message Queuing Telemetry Transport (MQTT) protocol. To perform this function, the VerneMQ tool [56] was chosen. A connector continuously monitors topics (such as a message queue) in MQTT, and, as soon as a message arrives, it is automatically transferred to the Apache Kafka streaming platform [57]. Following the flow, the next box, GoAT, represents the proposal of this research work for ranking sensors.

After processing by the GoAT broker, messages are maintained in the Apache Cassandra tool. The persisted data can be used for monitoring the system. For the tests, three containers were configured in the Docker tool, representing a node in the DHT network. In this way, this architecture allows parallel processing, also reinforcing the scalability of the proposal.

Table 2 Datasets

Dataset	Cases (millions)	Stations
Intel	2.3	54
NAMOS	0.24	9
NOAA	127	14,181
SensorScope	0.58	23

Fig. 7 NOAA hourly dataset [59]

Datasets Used

Four real data sets were selected to evaluate the proposed method, as shown in Table 2.

The Intel Lab dataset [33] contains information about data collected from 54 sensors with weatherboards that collected timestamped topology information, along with humidity, temperature, light, and voltage values once every 31 s. The sensors, deployed in the Intel Berkeley Research Lab (Intel), were arranged in the lab. The data from sensor 52 were used in the training phase, and data from sensors 2, 3, 20, and 39 were used for testing the proposal.

Another dataset referred to environmental data and was collected from 9 buoys of the Networked Aquatic Microbial Observing System (NAMOS), which is widely used in anomaly detection schemes [58]. In this dataset, data from sensor 107 were used in training, and that from sensors 101, 102, and 106 were used for tests.

Figure 7 shows an overview of 14,181 stations of the National Oceanic and Atmospheric Administration (NOAA) [59]. In the NOAA dataset, data from station 010080 were used in training, and data from stations 010100, 010230, and 010490 were used in the tests.

The last of the four datasets used have data collected from 23 stations of the SensorScope [60], an outdoor temperature. For training, data from station 19 were used, and data from stations 2, 6, 7, 9, and 17 were used in the tests.

The subsets of data for each sensor (stations) were chosen, because they contain data that are far from the average of each set (possible anomalies or failures). The combination of these datasets, in which the data were generated under different conditions, contributed to the evaluation of our proposal. To realize the experiments, a computer with an Intel Core i7 processor with 16 GB of RAM and a 250 GB SSD is used.

Experimental Results

Since the cases of datasets do not have labels, and to provide some metrics capable of demonstrating the quality of our method, the IQR metric is used to generate labels for the data in the training step. This approach allows generating indicators of the accuracy of the proposal, enabling the comparison of the outputs generated in the tests with the information obtained in the training step.

As an example of the application of the IQR metric to generate labels in the data, a graphical representation of the division of the data from the dataset SensorScope into the normal and abnormal values is showed in Fig. 8.

In the SensorScope dataset image shown in Fig. 8, the dots in blue (normal values) represent the natural temperature variations during a day (between 15 and -10 °C) and the red dots represent the values considered to be abnormal (anomalies in the environment or failures in capture or transmission). The data shown in the image refer to station 19 of the dataset.

The thresholds of the IQR metric (Fig. 9) defined in the training step of the k-nearest neighbor algorithm were used in the tests of the algorithm.

Table 3 presents the overall error rates (in parentheses next to the name of each dataset) and the confusion matrices (the letters *N* and *A* represent the *normal* and *abnormal*

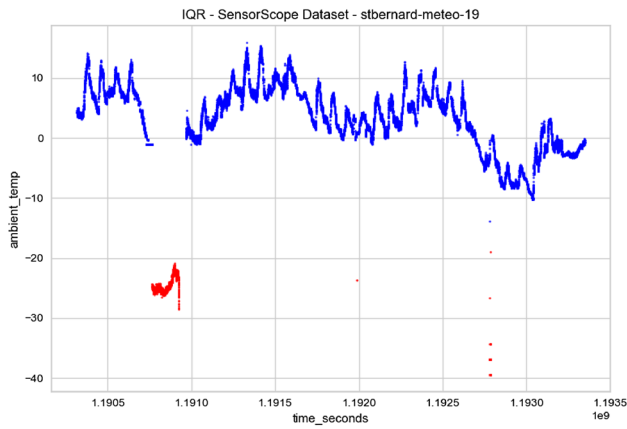
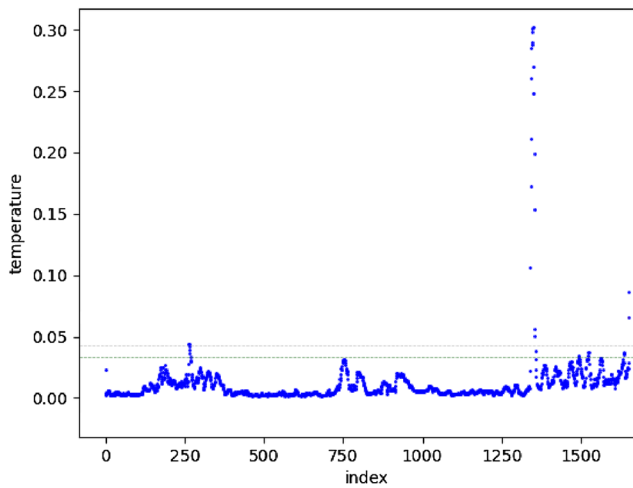


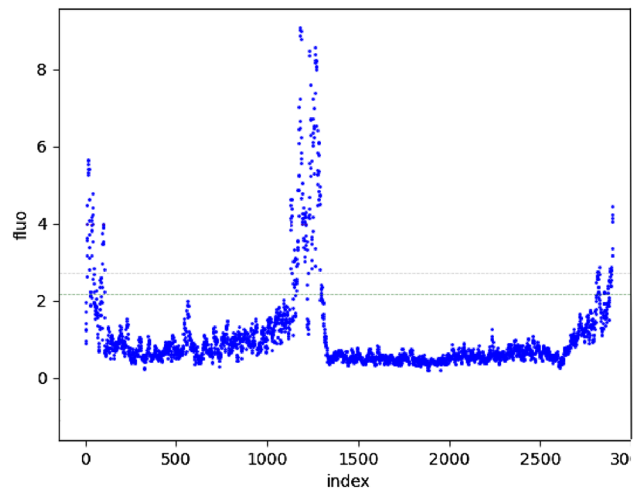
Fig. 8 SensorScope dataset classified using the IQR metric

labels, respectively) for all datasets. Both overall error rates and confusion table values are percentage values. Note that the values are presented in percentages, but values such as the total normal values and the total abnormal values were considered to inform the percentages, so the values add up to 100% for each column of normal values and for each column of values abnormal. As the dataset cases did not have labels, the responses of the proposed algorithm were compared with the labels generated in the pre-processing step.

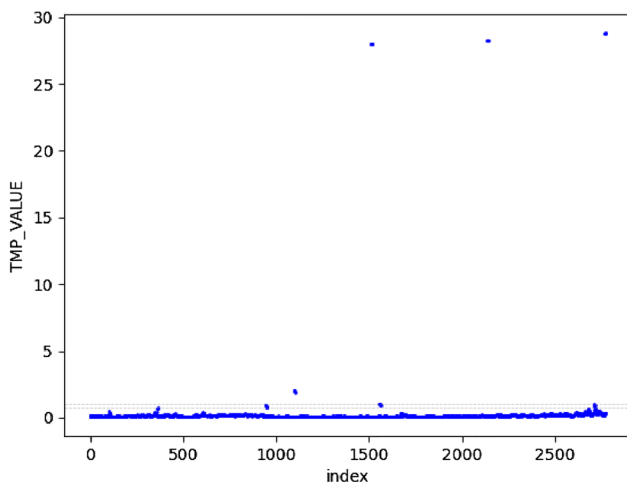
Table 4 shows the differences, for abnormal values, between the classification obtained in the pre-processing step (Abnormal-True) and in the tests (Abnormal-Predicted). The values referring to the pre-processing step are expressed in quantity and the other values are expressed in percentage. The table also displays the percentage of values for each data label. The Abnormal-True and Abnormal-Predicted columns



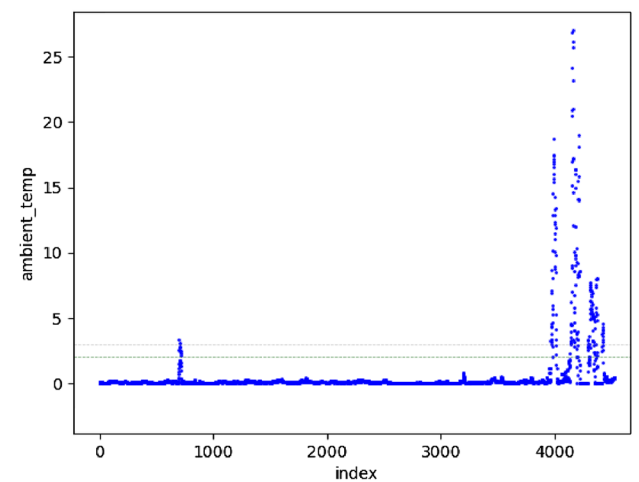
(a)



(b)



(c)



(d)

Fig. 9 The IQR metric is applied to values of the datasets: a Intel, b NAMOS, c NOAA, and d SensorScope

Table 3 Normalized confusion matrix for all datasets (%)

Predicted Values (%)		Actual Values (%)									
		Intel (0,04)		NAMOS (1.92)		NOAA (0.40)		SensorScope (0.09)			
		N	A	N	A	N	A	N	A		
N	99,998	3,600	N	98,333	11,455	N	99,998	33,885	N	99,913	2,597
A	0,002	96,400	A	1,667	88,545	A	0,002	66,115	A	0,087	97,403

N=Normal; A = Abnormal

Table 4 Classification of abnormal values (%)

Dataset	Label first level		Label second level		
	Abnormal (True)	Abnormal (Predicted)	Anomaly	Failure	Normal
Intel	500	96%	0%	100%	4%
NAMOS	1711	89%	44%	56%	13%
NOAA	785	66%	16%	84%	0%
SensorScope	77	97%	1%	99%	3%

framework was disregarded. In Fig. 10a, the times spent in the first step of the algorithm (first level of active perception) are shown, and in Fig. 10b, the times referring to the execution of the three levels of the algorithm are shown.

Discussion

In Figs. 8 and 9, it is possible to see that the use of the IQR metric is efficient to separate the data into *normal*, *anomalous*, and *fault* data. The images clearly show the separation

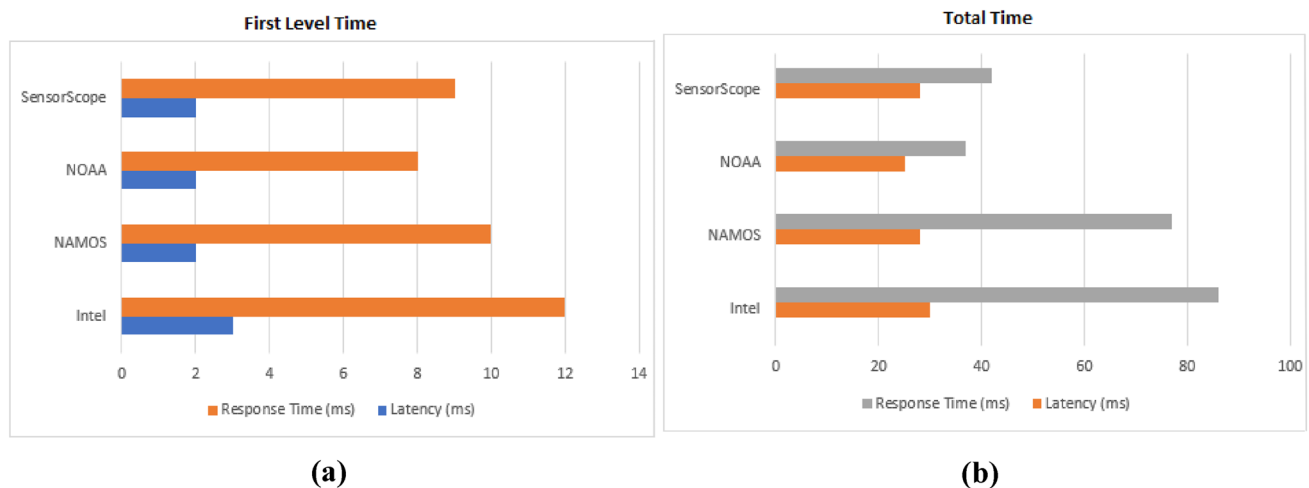


Fig. 10 Response time and latency costs

have values defined at the first level of the algorithm (Perception), and the *anomaly*, *failure*, and *normal* columns have values defined at the second level (Perception Time) of the algorithm. Note that in the values referring to the second level, the values of anomaly and failure add up to 100%, as they correspond to the parcels of abnormal values.

Figure 10 shows the times spent per dataset for processing a request. The item "Response Time" refers to the total time from the submission of the request to the receipt of the response. The item "Latency" refers to the processing time of the ranking algorithm. The difference between latency and response time is because, in the calculation of latency, the time spent on data manipulation by the python Pandas

of the data close to the average of the most distant data. The justification lies in the fact that the occurrence of anomalies and failures in the sensors should generally be rare events when compared to data considered normal.

The general error percentages shown in Table 3 demonstrate the efficacy of our proposal. However, in the NOAA data set, it is possible to verify higher values by analyzing only the error rates of the values classified as *abnormal* by the algorithm. In the other data sets, this rate is minimal. It was possible to verify, after analysis, that the threshold values of this set, for the *failure* and *anomaly* categories, differ drastically from the others. As can be seen in Fig. 9,

the limits between the *normal* and the other values are very close, which probably reveals the cause of the errors made by the algorithm.

Besides that, only a small part of the data are analyzed at all levels of the proposal. This is because the data considered *normal*, which consists of most of the data, are not processed in the second level of the proposal (Perception Time). This idea is clear when checking the graphs shown in Fig. 10, which shows considerably shorter processing times relative to the first level, Fig. 10a, concerning to the total processing time of the three levels shown in Fig. 10b.

Finally, Table 4 presents the data referring to the division of the data considered *abnormal* into data that represent *anomalies* in the environment and in data that represent *faults* in the sensors. This type of approach can be extremely useful not only for controlled environments, assisting in disaster prevention and maintenance of problematic sensors, but mainly to allow the safe use of information from sensors in the Normal state.

Conclusions and Future Works

This paper describes the efforts and results of a differentiated research work focusing on the ranking of sensors for the Edge–Fog–Cloud environments. The main contributions of the work are to improve the reliability in the use of sensor data and to reduce the latency of this process, offering a method based on artificial intelligence to meet the requirements of intelligent environments, such as Smart Cities and Industry 4.0.

The immense volume of complex and heterogeneous data generated in the IoT environment poses challenges that must be overcome. One of the main challenges is related to computational resources, essential to process these data, to meet the real-time requirements inherent to the IoT environment. Also, the uncertainty added to the process, generated precisely by the complexity of the data, also represents an obstacle to be overcome.

The presented proposal allows to manipulate this data and using software components capable of processing data in a parallel, distributed way and, thus, meeting time restriction requirements.

Besides, through the implemented mechanisms, it allows to drastically reduce the computational effort involved in data processing, since almost all the data are analyzed with a very low computational cost (only the first level of the proposal is used in this subset of the data).

Another characteristic present in the proposal is related to the ability to deal with the imprecision inherent to

devices in the IoT environment. The artificial intelligence algorithms applied incrementally provide more reliability in the use of the generated data.

Finally, due to the treatment of the uncertainty present in the data, the proposal also offers resources to facilitate monitoring in controlled environments through the classification of sensors in anomalous or in a state of failure.

The presented method makes use of several sets of real data, already used in other proposals, aiming to help to overcome the present challenges, as well as to facilitate the improvement of future proposals. Test results show error rates below 3% in two sets of data, 11% in one of them, and in only one of the four sets used, the rate cannot be considered low. Considering together the times obtained with regard to latency, which were close to 2 ms in the first processing level and 30 ms in the total time, it can be said that the proposed method is promising, as it combines reliability and latency reduction.

Finally, to the best of our knowledge, this is the first sensor ranking proposal that considers the fact that sensor failures can be transient or persistent, given the dynamic nature of the IoT environment. This work, as it was implemented, also previously generates the list with the classification of the sensors, which allows responding immediately to a request for a list of reliable sensors. The use of the theory of active perception as the basis for developing the proposal allowed the creation of a method that adds knowledge to the reasoning at each level of the theory, thus improving the reliability of the data analysis process, using low computational resources.

As future work, tests are planned in a computational grid environment. This will considerably increase the amount of data processed by the method, since the volume of data generated by sensors is quite large. This procedure aims to demonstrate and evaluate the computational capacity of the proposal. In this test, we also intend to use data captured online to reflect the characteristics of real environments further. In the proposal presented, only univariate data were considered, and the missing data were disregarded. In future work, we also intend to address these two aspects. Finally, to optimize the use of the proposal, we want to use artificial intelligence techniques to define the hyperparameters, currently defined manually.

Funding This research is funded by the Federal University of Santa Catarina and by the Federal Institute of Santa Catarina.

Data availability All datasets are available at the following link: https://github.com/felipekosta/SNCS_datasets.

Declarations

Conflict of Interest The authors declare no conflict of interest.

References

1. Yuen KKF, Wang W. Towards a ranking approach for sensor services using primitive cognitive network process. In: The 4th annual IEEE international conference on cyber technology in automation, control and intelligent, Hong Kong, China: 2014, p. 5. <https://doi.org/10.1109/CYBER.2014.6917487>.
2. Sezer OB, Dogdu E, Ozbayoglu AM. Context-aware computing, learning, and big data in internet of things: a survey. *IEEE Internet Things J.* 2018;5:27. <https://doi.org/10.1109/JIOT.2017.2773600>.
3. Turner V. The digital universe of opportunities: rich data and the increasing value of the internet of things 2014. <https://www.emc.com/leadership/digital-universe/2014iview/executive-summary.htm>. Accessed 12 Sept 2018.
4. Bonomi F, Milito R, Zhu J, Addepalli S. Fog computing and its role in the internet of things. In: Proceedings of the first edition of the MCC workshop on mobile cloud computing, Helsinki, Finland: ACM; 2012, p. 13–6. <https://doi.org/10.1145/2342509.2342513>.
5. Stojmenovic I, Wen S. The fog computing paradigm: scenarios and security issues. In: Proceedings of the 2014 federated conference on computer science and information systems, Warsaw, Poland: 2014, p. 1–8.
6. Mitchell HB. Multi-sensor data fusion: an introduction. Berlin: Springer; 2007.
7. Costa FS, Nassar SM, Dantas MAR. A three level sensor ranking method based on active perception. In: IECON 2019 - 45th annual conference of the IEEE industrial electronics society, vol. 1, Lisbon, Portugal: IEEE; 2019, p. 2889–95. <https://doi.org/10.1109/IECON.2019.8927612>.
8. Kertiou I, Benharzallah S, Kahloul B, Beggas M, Euler R, Laouid A, et al. A dynamic skyline technique for a context-aware selection of the best sensors in an IoT architecture. *Ad Hoc Netw.* 2018;81:14. <https://doi.org/10.1016/j.adhoc.2018.08.011>.
9. Pattar S, Buyya R, Venugopal KR, Iyengar SS, Patnaik LM. Searching for the IoT resources: fundamentals, requirements, comprehensive review, and future directions. *IEEE Commun Surv Tutor.* 2018;20:31. <https://doi.org/10.1109/COMST.2018.2825231>.
10. Fathy Y, Barnaghi P, Tafazolli R. Large-scale indexing, discovery, and ranking for the internet of things (IoT). *ACM Comput Surv.* 2018;51:53. <https://doi.org/10.1145/3154525>.
11. Xiao B, Rahmani R, Li Y, Gillblad D, Kanter T. Intelligent data-intensive IoT: a survey. In: Proceedings of 2nd IEEE international conference on computer and communications (ICCC), Chengdu, China: 2016, p. 2362–8. <https://doi.org/10.1109/CompComm.2016.7925122>.
12. NSF. NSF Edge Workshop Report. Washington, DC, USA: NATIONAL SCIENCE FOUNDATION; 2016.
13. Costa F, Nassar S, Dantas M. GoAT: A sensor ranking approach for IoT environments. In: Proceedings of the 11th international conference on cloud computing and services science - CLOSER, Florida, USA: 2021, p. 169–77.
14. Skarmeta AF, Santa J, Martínez JA, Parreira JX, Barnaghi P, Enshaeifar S, et al. IoTcrawler: Browsing the internet of things. In: Proceedings of the 2018 global IoT summit (GIoTS), Bilbao, Spain: Institute of Electrical and Electronics Engineers (IEEE); 2018, p. 6.
15. Nesa N, Banerjee I. SensorRank: An energy efficient sensor activation algorithm for sensor data fusion in wireless networks. *IEEE Internet Things J.* 2019;6:2532–9. <https://doi.org/10.1109/JIOT.2018.2871469>.
16. Dautov R, Distefano S. Automating IoT data-intensive application allocation in clustered edge computing. *IEEE Trans Knowl Data Eng.* 2019. <https://doi.org/10.1109/TKDE.2019.2923638>.
17. Ruta M, Scioscia F, Pinto A, Gramegna F, Ieva S, Loseto G, et al. CoAP-based collaborative sensor networks in the semantic web of things. *J Ambient Intell Hum Comput.* 2019;10:18. <https://doi.org/10.1007/s12652-018-0732-4>.
18. Kakunsi DC, Candra MZC. SDRank: An Adaptable Service Selection for IoT Based on Ranking. In: IEEE Mataram Indonesia: 2018, p. 6. <https://doi.org/10.1109/ICODSE.2018.8705811>.
19. Dilli R, Argou A, Pilla M, Pernas AM, Reiser R, Yamin A. Fuzzy logic and MCDA in IoT resources classification. In: Proceedings of the 33rd annual ACM symposium on applied computing, New York, NY, USA: ACM; 2018, p. 6. <https://doi.org/10.1145/3167132.3167216>.
20. Hussain N, Anees T, Azeem U. Development of a novel approach to rank resources in IoT. *Int J Adv Comput Sci Appl IJACSA.* 2018;9:14. <https://doi.org/10.14569/IJACSA.2018.090949>.
21. Nunes LH, Estrella JC, Perera C, Reiff-Marganiec S, Delbem ACB. The elimination-selection based algorithm for efficient resource discovery in Internet of Things environments. In: 2018 15th IEEE annual consumer communications networking conference (CCNC), 2018, p. 7. <https://doi.org/10.1109/CCNC.2018.8319280>.
22. Kang H, Kim M, Bae M, Bang H-C, Yoe H. A conceptual device-rank based resource sharing and collaboration of smart things. *Multimed Tools Appl.* 2016;75:13.
23. Zhang P, Liu Y, Wu F, Liu S, Tang B. Low-overhead and high-precision prediction model for content-based sensor search in the Internet of Things. *IEEE Commun Lett.* 2016;20:4. <https://doi.org/10.1109/LCOMM.2016.2521735>.
24. Saxena S. Vector method for ranking of sensors in IoT. *IEEE, vol. 3, Coimbatore, India: 2016, p. 5.* <https://doi.org/10.1109/INVENTIVE.2016.7830231>.
25. Wang W, Yao F, De S, Moessner K, Sun Z. A ranking method for sensor services based on estimation of service access cost. *Inf Sci.* 2015;319:17. <https://doi.org/10.1016/j.ins.2015.05.029>.
26. Cabral L, Compton M, Müller H, et al. A use case in semantic modelling and ranking for the sensor web. In: Mika P, Tudorache T, Bernstein A, Welty C, Knoblock C, Vrandečić D, et al., editors. 13th International semantic web conference. Riva Del Garda, Italy: Springer International Publishing; 2014. p. 15.
27. Niu W, Lei J, Tong E, Li G, Chang L, Shi Z, et al. Context-aware service ranking in wireless sensor networks. *J Netw Syst Manag.* 2014;22:25. <https://doi.org/10.1007/s10922-012-9259-8>.
28. Perera C, Zaslavsky A, Liu CH, Compton M, Christen P, Georgakopoulos D. Sensor search techniques for sensing as a service architecture for the Internet of Things. *IEEE Sens J.* 2014;14:406–20. <https://doi.org/10.1109/JSEN.2013.2282292>.
29. Truong C, Römer K, Chen K. Fuzzy-based sensor search in the Web of Things. *IEEE, Wuxi, China: 2012, p. 8.*
30. Walters J, Österberg P, Kanter T. The MediaSense framework: ranking sensors in a distributed architecture. In: Proceedings of fifth international conference on digital telecommunications, the updated MediaSense Framework, Athens, Greece: IEEE conference proceedings; 2011, p. 5.
31. Wang H, Tan CC, Li Q. Snoogle: a search engine for pervasive environments. *IEEE Trans Parallel Distrib Syst.* 2010;21:15. <https://doi.org/10.1109/TPDS.2009.145>.
32. Elahi BM, Romer K, Ostermaier B, Fahrmaier M, Kellerer W. Sensor ranking: a primitive for efficient content-based sensor search. *IEEE, San Francisco, CA, USA: 2009, p. 12.*
33. Intel Lab Data n.d. <http://db.csail.mit.edu/labdata/labdata.html>. Accessed 20 Dec 2018.
34. Chen T, Guestrin C. XGBoost: A scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining - KDD '16, San Francisco, California, USA: ACM Press; 2016, p. 785–94.

35. Shi H, Wang H, Huang Y, Zhao L, Qin C, Liu C. A hierarchical method based on weighted extreme gradient boosting in ECG heartbeat classification. *Comput Methods Programs Biomed.* 2019;171:1–10. <https://doi.org/10.1016/j.cmpb.2019.02.005>.
36. Chandola V, Banerjee A, Kumar V. Anomaly detection: a survey. *ACM Comput Surv.* 2009;41:1–58. <https://doi.org/10.1145/1541880.1541882>.
37. Cecilio IM, Ottewill JR, Pretlove J, Thornhill NF. Nearest neighbors method for detecting transient disturbances in process and electromechanical systems. *J Process Control.* 2014;24:1382–93.
38. Witten IH, Frank E, Hall MA. *Data mining: practical machine learning tools and techniques.* 3rd ed. Burlington: Morgan Kaufmann; 2011.
39. Barbetta PA, Bornia AC, Reis MM. *Estatística Para Cursos De Engenharia E Informática.* 3rd ed. São Paulo: Atlas; 2010.
40. Zadeh LA. Fuzzy sets. *Inf Control.* 1965;8:338–53. [https://doi.org/10.1016/S0019-9958\(65\)90241-X](https://doi.org/10.1016/S0019-9958(65)90241-X).
41. Buchholz T, Schiffers M. Quality of context: what it is and why we need it. In: *Proceedings of the 10th workshop of the OpenView University Association: OVUA'03*, 2003.
42. Manzoor A, Truong H-L, Dustdar S. Quality of context: models and applications for context-aware systems in pervasive environments. *Knowl Eng Rev.* 2014;29:16. <https://doi.org/10.1017/S0269888914000034>.
43. Perera C, Zaslavsky A, Christen P, Georgakopoulos D. Context aware computing for the internet of things: a survey. *IEEE Commun Surv Tutor.* 2014;16:40. <https://doi.org/10.1109/SURV.2013.042313.00197>.
44. Cherbal S, Boukerram A, Boubetra A. A survey of DHT solutions in fixed and mobile networks. *IJCNDS.* 2016;17:14. <https://doi.org/10.1504/IJCNDS.2016.077938>.
45. Apache Foundation. *Apache Cassandra Project* n.d. <http://cassandra.apache.org/>. Accessed 21 Sept 2020.
46. Stoica I, Morris R, Karger D, Kaashoek MF, Balakrishnan H. Chord: A scalable peer-to-peer lookup service for internet applications. In: *Proceedings of the 2001 conference on applications, technologies, architectures, and protocols for computer communications - SIGCOMM '01*, vol. 31, New York, NY, USA: ACM Press; 2001, p. 149–60. <https://doi.org/10.1145/383059.383071>.
47. Hofmann A, Robertson P. Active perception: improving perception robustness by reasoning about context. vol. 2, 2015, p. 9.
48. Bajcsy R. Active perception. *Proc IEEE.* 1988;76:10. <https://doi.org/10.1109/5.5968>.
49. Biel L, Wide P. Active perception for autonomous sensor systems. *IMM.* 2000;3:4. <https://doi.org/10.1109/5289.887457>.
50. Schiffman HR. *Sensation and Perception: An Integrated Approach.* *Sensation and Perception: An Integrated Approach.* Edição: 5th, New York: John Wiley & Sons; 2001, p. 12.
51. Bevrani H. *Robust power system frequency control.* New York: Springer; 2009.
52. Thornhill NF. Finding the source of nonlinearity in a process with plant-wide oscillation. *IEEE Trans Control Syst Technol.* 2005;13:434–43.
53. Bringel Filho J, Agoulmine N. A quality-aware approach for resolving context conflicts in context-aware systems. In: *9th international conference on embedded and ubiquitous computing, Melbourne, VIC, Australia: 2011*, p. 229–36. <https://doi.org/10.1109/EUC.2011.9>.
54. Costa FS, Nassar SM, Gusmeroli S, Schultz R, Conceição AGS, Xavier M, et al. FASTEN IIoT: an open real-time platform for vertical, horizontal and end-to-end integration. *Sensors.* 2020;20:5499.
55. Drath R, Horch A. Industrie 4.0: hit or hype? [industry forum]. *IEEE Ind Electron Mag.* 2014;8:56–8. <https://doi.org/10.1109/MIE.2014.2312079>.
56. Octavo LA. VerneMQ - A MQTT broker that is scalable, enterprise ready, and open source 2019. <https://vernemq.com/>. Accessed 1 Aug 2019.
57. Apache. *Apache Kafka.* Apache Kafka 2019. <https://kafka.apache.org/>. Accessed 1 Aug 2019.
58. Zamry NM, Zainal A, Rassam M. Unsupervised anomaly detection for unlabelled wireless sensor networks data. *Int J Adv Soft Comput Its Appl.* 2018;10:172–91.
59. NOAA n.d. <https://tidesandcurrents.noaa.gov/gmap3/>. Accessed 20 Dec 2018.
60. Ingelrest F, Barrenetxea G, Schaefer G, Vetterli M, Couach O, Parlange M. SensorScope: Application-specific sensor network for environmental monitoring. *ACM Trans Sen Netw.* 2010;6:17:1-17:32.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.