

Article

# Spatio-Temporal Semantic Data Model for Precision Agriculture IoT Networks

Mario San Emeterio de la Parte , Sara Lana Serrano , Marta Muriel Elduayen   
and José-Fernán Martínez-Ortega 

Departamento de Ingeniería Telemática y Electrónica (DTE), Escuela Técnica Superior de Ingeniería y Sistemas de Telecomunicación (ETSIST), Universidad Politécnica de Madrid (UPM), C/Nikola Tesla, s/n, 28031 Madrid, Spain

\* Correspondence: mario.sanemeterio@upm.es

**Abstract:** In crop and livestock management within the framework of precision agriculture, scenarios full of sensors and devices are deployed, involving the generation of a large volume of data. Some solutions require rapid data exchange for action or anomaly detection. However, the administration of this large amount of data, which in turn evolves over time, is highly complicated. Management systems add long-time delays to the spatio-temporal data injection and gathering. This paper proposes a novel spatio-temporal semantic data model for agriculture. To validate the model, data from real livestock and crop scenarios, retrieved from the AFarCloud smart farming platform, are modeled according to the proposal. Time-series Database (TSDB) engine InfluxDB is used to evaluate the model against data management. In addition, an architecture for the management of spatio-temporal semantic agricultural data in real-time is proposed. This architecture results in the DAM&DQ system responsible for data management as semantic middleware on the AFarCloud platform. The approach of this proposal is in line with the EU data-driven strategy.

**Keywords:** precision agriculture; real-time systems; data engineering; middleware; database systems; spatio-temporal databases (TSDB); big data; Internet of Things (IoT)



**Citation:** San Emeterio de la Parte, M.; Lana Serrano, S.; Muriel Elduayen, M.; Martínez-Ortega, J.-F. Spatio-Temporal Semantic Data Model for Precision Agriculture IoT Networks. *Agriculture* **2023**, *13*, 360. <https://doi.org/10.3390/agriculture13020360>

Academic Editor: Yanbo Huang

Received: 14 November 2022

Revised: 23 January 2023

Accepted: 31 January 2023

Published: 1 February 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

The agricultural sector is being driven by innovation and scientific research. Among the projects developed in the field of farming and livestock management, the inclusion of numerous solutions in the IoT framework stands out. In innovation projects, monitoring and decision-making systems, process automation, and even analysis or planning tools are implemented, among others. The deployment of new technological solutions in the agricultural sector aims to meet the Sustainable Development Goals (SDG) [1] and animal welfare [2]. With this aim, the Food and Agricultural Organization [3] and the International Fund for Agricultural Development (IFAD) [4] arise to promote research, investment, and sustainability in agriculture.

Since 2015, the number of people suffering from hunger has increased, reaching a percentage of almost 9% of the world's population. If the current trend continues, the Zero Hunger Goal [5] will not be achieved by 2030. Targets 2.3 and 2.a of the Zero Hunger goal aim to double agricultural productivity and increase agricultural research. To achieve this, precision agriculture solutions can offer increased productivity and sustainable food production.

In the field of farming, the deployment of sensors allows continuous monitoring of the state of crops. Some actuators allow the automation of processes, as in the case of intelligent greenhouses [6]. Data collection and post-processing allow inclusion in the flow of decision-making algorithms and artificial intelligence (AI) [7,8] to recognize the best stage for harvesting, or the different stages of irrigation and drying of crops. In addition, these

scenarios include different types of robots, autonomous and semi-autonomous vehicles (such as tractors), as well as unmanned aerial vehicles (UAVs). For monitoring, mission management systems are implemented for these devices or vehicles. Missions enable the automation of tasks such as harvesting, crop irrigation, or data collection in scenarios with transmission difficulties.

In the field of livestock management, the installation of devices, such as collars, on cattle allows a large amount of information to be extracted. In many fields, cattle movement is restricted by delimiting plots of land or even by electrical stimulation. These types of techniques do not allow for precise control of the animal and cause discomfort to the animal itself. However, equipping cattle with devices in the form of collars allows them to be constantly monitored. Some of the developments and research on cattle collars are presented in [9,10].

The use of collars provides geolocation, temperature, acceleration, and certain semantic data on cow identification. Geolocation makes it possible to extract a large amount of information, and tracking applications can be developed. It allows for observing whether an animal has been lost or has wandered away from the group. It is possible to observe the route described or the geographical conditions sought by the herd according to the annual season. In addition, thanks to sensors such as temperature, the animal's state of health can be monitored. An accelerometer will make it possible to detect sudden movements repeated with a certain frequency in the animal's head, which offers the detection of stress or discomfort of the animal.

However, these scenarios present a common problem: collecting and analyzing huge data volumes. It implies high delays and the need for computationally powerful equipment. Real-time data gathering and injection becomes a complex scenario to achieve.

When analyzing the factors that characterize data in precision agriculture, working with customers, devices, sensors, and even vehicles or robots, almost all these data can be characterized according to three fundamental dimensions: spatial, temporal, and semantic. To characterize a datum and extract its meaning, or aggregate data and generate high-level knowledge, three fundamental questions are asked: When were the data taken? Where were the data taken? What are the semantics of the data?

As in precision agriculture, spatio-temporal data management has become one of the fundamentals in many growing technological fields such as the Internet of Things (IoT) or Big Data, because the raw products of the 21st century are data. There are many appliances for temporal, ge positioning, and semantic information stored in databases. It should be highlighted that the necessity of managing large amounts of data in a real-time environment is one of the biggest challenges of technology nowadays. Different open-source and market systems provide features to manage time series or even APIs with different methods for ge positioning of measurements or stored documents.

The collection of huge volumes of data by all kinds of systems developed in precision agriculture or even industry and technological sectors has reached its peak. It is due to the development of multiple high-precision sensors and the rapid data transmission with technologies such as four and five-generation networks, or a variety of lightweight or long-range protocols such as MQTT, LoRa, Zigbee, etc. However, the goal of this uncontrolled data collection is the generation of knowledge or wisdom from such raw data.

One of the main challenges of this environment is the reduction in processing times, delivery latency between components of the distributed system, information management in the repositories, and the availability of the device itself to collect, send information, and receive possible orders in real-time. However, when analyzing each of the weak points, a common element is detected, the repositories, and their management through semantic middleware architectures, APIs, etc. In addition, the management of repositories together with their respective data extraction and injection operations generate one of the largest latency additions in the communication chains between components and platforms of an IoT system for precision agriculture [11–14]. This impedes the implementation of solutions

that require real-time data, such as herding livestock using UAVs or Robots, or acting via mobile devices or actuators when detecting anomalous observations or alarms.

This article proposes a model for the management of precision agriculture data. The objective is to provide injection and retrieval of information in real-time, with the least possible use of resources, enabling the management of large volumes of data. Spatial, temporal, and semantic dimensions are defined for the precise characterization of data through the model. The characteristics of one or more of the defined dimensions are studied, as well as different loading and configuration situations. This allows the selection of the most appropriate techniques for data analysis and retrieval, reducing delays, and allowing data characterization according to the three defined dimensions. The proposal is carried out in a real environment, with data extracted from the AFarCloud project. This article includes a validation of the proposed Agricultural Data Model, through the InfluxDB engine [15], oriented to time series management.

Agricultural diversification is based on the evaluation of weather and land conditions to study the adaptability of new crops. Thanks to the information collected by sensors, it is possible to estimate the profitability and yield of a new crop under certain conditions. The proposed spatio-temporal semantic model and the data management architecture resulting from this paper are key elements for an accurate assessment of the physiological, environmental, and yield conditions of a given land. Thanks to the management of historical and real-time updated data, the proposal offers a solution adaptable to any farm, allowing to evaluate the precise moment and the type of crop that can be grown on a specific plot, enabling agricultural diversification.

Due to the accurate characterization offered by the proposed model, it is possible to collect the growth and yield history of a specific crop. The spatial and temporal values of pH, humidity, soil temperature, leaf color, or stem thickness of a given crop allow for estimation of its adaptability. Thus, the proposed model allows us to determine if a specific crop can be harvested in other farms or plots, supporting agricultural diversification.

There are several similarities between the data generated by devices and sensors deployed in different IoT application domains. The spatio-temporal nature of the data, the use of a multitude of devices with low computational resources, the real-time operating requirements, and the needs presented by data management architectures in the industrial, healthcare, and energy sectors, in smart cities, smart homes, and agriculture represent common characteristics [16]. Therefore, the spatio-temporal semantic model and architecture proposed in this paper can be easily adapted to any other IoT environment. The description of measurements, devices, and state vectors enables its use for any environment described by the use of sensors and devices, with minimal development effort.

Section 2 presents some of the most innovative research and technological solutions in precision agriculture. It also shows how the collection of large datasets is one of the common agents in this framework. Section 3 describes the proposed data model, the syntax and structure of the data, and the query functions. The experimental framework is described in Section 4, as well as the equipment used and the evaluation of the model against the defined indicators. Section 5 presents the results obtained from the proposal and a data management system resulting from the implementation of the proposed model, structuring, and query functions. Finally, Section 6 presents the conclusions drawn from the results of the work.

## 2. Related Work

Agriculture plays an important role in the global economy and sustainability. Due to the need to increase food production for a growing population, there has been a focus on improving, automating, and increasing livestock and crop management activity. This has resulted in a negative impact on the environment. However, smart farming technologies aim to optimize productivity by reducing costs and waste production, minimizing the impact on crops and livestock, and improving their quality.

There is a huge variety of sensors and types of measurements captured in a precision agriculture scenario. In the book “Sensing Approaches for Precision Agriculture” [17],

a wide variety of sensors currently on the market and even under research or development are presented. Among the sensors presented are those with capabilities for soil sensing, crop health, vigor, and disease detection, and even detection from unmanned aerial vehicles.

There are interesting proposals in the literature that, based on sensor data, can optimize efforts and resources. In [18], a system capable of optimizing water consumption in crop irrigation is proposed, thanks to the decision making generated by a machine learning algorithm (KNN), nourished with a specific ontology and the values generated by humidity, temperature, and light sensors.

In the field of precision crop and livestock farming, some technological and scientific solutions can be found, described in [19]. Applications presented include the use of devices to monitor livestock and their geolocation tracking with GPS sensors. At the same time, machine learning and artificial intelligence solutions for detecting animal discomfort have been exposed to reduce mortality and increase the welfare of livestock. In terms of crop management, some systems based on the use of UAVs or robots for crop fertilization are presented. In addition, certain automation systems based on monitoring using sensors and the installation of actuators governed by artificial intelligence systems are also presented.

There are a variety of specific solutions for livestock monitoring and surveillance. One of the most striking solutions is the use of multi-sensor collars fitted around the neck of livestock. This solution allows individual monitoring of the animal and the collection of multiple data on its activity. A specific solution is described in [20], which together with a cloud-based software environment can manage livestock data and alert the farmer proactively. In the study, the importance of real-time data collection through collars, subsequent extraction, and processing of the data is exposed.

Through the measurement of crop data by sensors, numerous activities are developed for crop harvesting. Examples include sorting and counting fruits or monitoring the health status of plants, including the detection of weeds, insects, and diseases. In [21], some studies of different agricultural activities framed in such harvesting solutions are reviewed. In addition, some of the research on vehicle and robot guidance systems for the automation of harvesting tasks in agriculture is presented.

Pesticide spraying is used to prevent crop diseases and pests, and is one of the main causes of a reduction in productivity. However, the manual spraying of these chemicals presents a risk to humans, such as through the contraction of diseases. This is why some precision farming solutions propose automating such tasks. In [22], the application of UAVs for crop monitoring and pesticide spraying is reviewed. In this way, direct human contact is avoided. However, for autonomous control of UAVs by a management system or platform, real-time data exchange will be necessary.

Precision agriculture presents an environment rich in spatial and temporal data. To correctly extract information from the spatio-temporal data generated by sensors deployed in crops and livestock, prior processing is necessary to make management decisions. The open-source GeoFIS software (version 1.2), designed to cover the entire process, from spatial data to spatial information and decision making, is presented in [23]. The experiment is applied to three contrasting crops, bananas, wheat, and vineyards. The article evaluates GeoFIS software together with its integrated algorithms to address the needs of farmers, advisors, or spatial analysts when dealing with precision agriculture data. The GeoFIS framework focuses on the analysis and management of spatial data, offering decision support. However, unlike the data model proposed in this article, GeoFIS lacks the temporal and semantic characterization and management inherent in the nature of agricultural data.

Modeling and management of a large amount of spatio-temporal data are some of the main goals in research and development due to the increasing implementation and use of devices, sensors, and IoT terminals. Yongjiun Ren et al. in [24] present a novel secure storage mechanism for large amounts of spatio-temporal data. However, the implementation of a blockchain-based mechanism to guarantee the integrity and security of spatio-temporal data storage requires high computational capabilities, and makes its implementation impossible for real-time management scenarios.

A collaborative platform based on linked data and machine principles for the viticulture domain is presented in [25]. This proposal includes the automatic enrichment of metadata and services, with detailed workflow and user participation. The platform aims to improve smart viticulture/agriculture services, and their efficient management, involving all stakeholders.

Irya Wisnubhadra et al. in [26] present an open spatio-temporal data warehouse for agriculture. This new concept aims to fill the gap in spatial and temporal attributes in open agricultural data scenarios that reside in sources such as Linked Open Data, Linked Open Statistical Data, and Open Government Data. The spatio-temporal data warehouse is implemented based on MobilityDB, an extension of Postgresql. The offered data model, based on RDF syntax (triples), could be extended to consider data extracted by devices in the context of precision agriculture. This open data warehousing approach is oriented to the analysis of agricultural production, whose precision of measurement of the temporal dimension is focused on the date. Furthermore, the response time offered by the queries presented in the warehouse evaluation exceeds 12 s, so the data modeling and the warehouse presented do not offer a valid response for real-time scenarios.

The management of large spatio-temporal datasets in real-time is increasingly required in different sectors. Atsushi Isomura et al. present a novel technology for spatio-temporal data management called Axispot [27], applied to obstacle detection and lane-specific congestion in the automotive domain. The paper describes the need to store spatio-temporal data sent simultaneously by a large number of moving objects. The study presents spatial data search and aggregation capabilities by reducing the complexity of polygonal shape lines to deal with data management in a real-time environment. The proposed technology does not consider the height or altitude dimension, but the authors consider it as an objective for future study. The orientation of the proposal of this article towards the automotive world, and the high computational capabilities offered by the agents involved, make its application in an agricultural scenario difficult.

Studies on the management of large spatio-temporal datasets offer different approaches to improve performance. Some approaches include proposals such as that of Dong Wang et al. who propose an extension of the SPARQL query language with spatio-temporal assertions for RDF data collection [28], plus a corresponding index and query algorithm. An alternative approach is the generation of new spatio-temporal data indexing engines, such as the spatio-temporal data engine called JUST, designed to handle large amounts of data with a query language similar to SQL, which is presented in [29].

The application of spatio-temporal semantic data management systems in precision agriculture still has a long way to go in terms of research and improvement. However, in the literature, there are already some systems, such as SEMAP, presented by Henning Deeken et al. [30], mainly developed for spatio-semantic management for robot planning, which later, in [31,32], was successfully applied to the precision farming domain. The SEMAP framework is characterized by a powerful spatial description of the agents and the environment, offering 2D and 3D representations. However, this complex characterization is not necessary for most use cases in agriculture. Moreover, as the authors argue in the article, the temporal dimension is a missing piece in the system. In contrast, in this paper, a data model specifically designed for precision agriculture is proposed and validated against real data, captured by devices and sensors deployed on farms. The proposed data model reduces the complexity in spatial characterization offered by the SEMAP framework, which makes real-time management impossible, and adds the temporal and semantic dimension for the complete characterization and management of agricultural data.

Most precision agriculture solutions are based on data retribution through various types of sensors, devices, vehicles, or robots. To enable interaction between devices, orchestration of tasks, or provide an agile response from monitoring or supervisory systems, data must be managed in a real-time environment. Numerous approaches to the description of ontologies and agricultural models have been proposed in recent decades, which prioritize the semantic power of data. A current example is the Agricultural Information

Model (AIM) [33] based on the SOSA (Sensor, Observation, Sample, and Actuator) [34] and SSN (Semantic Sensor Network) [35,36] ontologies. However, these approaches are characterized by complex syntax, hampering management in real-time and generation and delivery by devices with limited resources. Most of the sensors and devices deployed in agricultural scenarios have low computational power and resources; in some scenarios, availability and connectivity constitute a problem. The model proposed in this paper offers a simple and light syntax that enables its generation by sensors and devices with lower computational power and resources, without sacrificing the power of spatial, temporal, and semantic characterization of the data.

Thanks to the implementation of numerous smart environments, IoT, and monitoring projects, hundreds of repositories with large volumes of data have been nurtured. In turn, data present a much more accurate characterization, adding, to the paradigm, the spatial and temporal dimensions as generic characteristics in data.

Relational databases (SQLs) are known for their high flexibility, ease of use, and maturity of the technology itself. However, relational databases are not particularly known for their scalability and ability to handle large volumes of data, which drives the creation of non-relational (NoSQL) databases. In addition, the data entered by IoT systems, the interconnection of devices, and the exploitation of data are characterized by their time-series nature. In [37], a study is oriented to time series databases (TSDBs) where InfluxDB, Kdb+, Graphite, Prometheus, and RRDTOOL are exposed.

For the spatio-temporal semantic agricultural data model proposed in this article, InfluxDB has been chosen as the NoSQL database engine for validation in the management system environment. However, there are multiple open-source solutions for the management of non-relational time series oriented databases. A comparison of several open-source TSDBs is presented in [38]. Among the solutions compared are InfluxDB, Graphite, RRDTOOL, Prometheus, OpenTSDB, and TimescaleDB. Through the definition of a set of quantitative and qualitative attributes with different scales and units of measurement, it is intended to select or classify the different TSDBs selected in the study. To solve the analysis, evaluation, and selection of the TSDBs, a multi-attribute TSDB maturity model is proposed, consisting of 10 quantitative and 8 qualitative attributes. As a result of the analysis of the different rankings defined during the experimentation, the article concludes with InfluxDB as the leading solution among the TSDBs analyzed.

This paper proposes a novel spatio-temporal semantic agricultural data model and management architecture that allow real-time performance. The proposed model enables data generation by devices with low computational resources deployed in the field, without sacrificing the power of characterization under the three defined dimensions. In Table 1, a comparison between the most relevant proposals related to our work is presented, based on the key features that our proposal presents: (i) big spatio-temporal (ST) data management, (ii) real-time operation, (iii) simple syntax, (iv) low use of computational resources by the data management architecture and the repository, (v) specific application and design for agriculture, and (vi) adaptability to other domains. We consider that our proposal can be applied to the contributions exposed in the literature, completing the shortcomings exposed by them in the specific validation scenarios. For example, to fill the gap in the management of the temporal dimension exposed in [32], and enable real-time data management, or to support data generation by low-resource devices and enable real-time management of [33].

**Table 1.** A comparison between the most relevant related proposals.

Proposal Ref. #	Authors	Big Data Management	ST	Real-Time Operation	Simple Syntax	Low Computational Resources	Agricultural Application	Adaptable to New Domains
[23]	Leroux et al.	N		N	N	N	Y	N
[24]	Ren et al.	Y		N	N/A	N	N	Y
[26]	Wisnubhadra et al.	Y		N/A	N	N/A	Y	N/A

Table 1. Cont.

Proposal Ref. #	Authors	Big Data Management	ST	Real-Time Operation	Simple Syntax	Low Computational Resources	Agricultural Application	Adaptable to New Domains
[27]	Isomura et al.	Y		Y	N/A	N	N	Y
[28]	Wang et al.	Y		N	N	N/A	N	Y
[29]	Ruiyuan Li et al.	Y		Y	N/A	Y	N	Y
[30]	Deeken et al.	N		Y	N	N	N	Y
[32]	Deeken et al.	N		Y	N	N	Y	Y
[33]	Palma et al.	Y		N	N	Y	Y	Y
[34]	Janowicz et al.	Y		N	N	Y	N	Y
[35]	Compton et al.	N		N/A	N	Y	N	Y
	Our proposal	Y		Y	Y	Y	Y	Y

### 3. Spatio-Temporal Semantic Data Model for Agriculture

Most innovative technological solutions for precision agriculture are driven by the IoT and the collection of information through the deployment of a multitude of devices in the field. These devices are divided into two main groups: (i) static devices and (ii) mobile devices.

Implementing monitoring, decision support, or task automation solutions is based on data collected in the field. An accurate characterization of the data collected by devices and sensors is essential. Repositories and data management systems constitute the heart or cornerstone of architecture and must guarantee availability, efficiency, and high performance.

Due to the lack of data model standards for precision agriculture and the heterogeneity of data captured by devices, establishing a correct characterization and modeling of data in precision agriculture is a complex but essential task. The generation of large volumes of data by agricultural devices and sensors results in an increase in the space required for storage, increasing *digitization footprint* [39]. The design of a lightweight syntax and efficient structuring reduces the cost of storage and increases the efficiency of management. Therefore, the design of data models for precision agriculture presents a common problem:

1. Defining common dimensions in data characterization empowering semantic information.
2. Lightweight syntax design in modelling for transmission, enabling generation and delivery for devices with limited resources.
3. Data modelling and structuring to reduce processing times and increase performance.
4. Design of data queries directly linked to the characterized dimensions of data, allowing real-time gathering.

In this paper, a novel data model for agriculture is proposed which aims to address this problem. The interest of this proposal is particularly encouraged by the European data strategy [40], which aims to turn Europe into a leader in the data-driven society.

#### 3.1. Data Model Definition

Precision agriculture is characterized by fully evolving scenarios. Observations captured by on-farm devices present three main dimensions, common in all types of measurement: (i) temporal evolution of conditions, where observations are associated with a time stamp; (ii) ge positioning stamp, which presents a static or dynamical nature, depending on the type of device performing the capture; (iii) semantic characterization of data and its relationships with devices or the environment.

The temporal dimension is of vital importance in the structuring of data in agricultural IoT repositories, so the proposal focuses on non-relational TSDBs oriented modelling.

NoSQL databases offer greater versatility and performance in terms of data volume scalability compared to traditional SQL databases. TSDBs consider the time stamp of the data directly as the index for indexing, making the gathering, aggregation, and sorting of time series much faster and more efficient. Writes on frozen (disk-compressed) shards (A shard is a horizontal partition of data in a database or search engine; each shard remains on a separate instance of the database server to spread the load) of historical data will have longer delays. However, this is anomalous in IoT scenarios for agriculture, where writes are performed at the current time instant.

Due to the different characteristics presented by the devices in an agricultural scenario, a subdivision of measurements into three main sets is proposed: (Dataset-1) observations captured by the devices, (Dataset-2) information collected by collars installed on the neck of livestock, and (Dataset-3) information from the state vector of vehicles.

(Dataset-1) The set of observations will contain all those measurements captured by sensors of devices deployed on the field, on robots, on autonomous or semi-autonomous vehicles, or Unmanned Aerial Vehicles (UAVs).

(Dataset-2) The collar set will contain all the information extracted by sensors embedded in collars fitted on the neck of the cattle. This includes information on temperature, accelerometers, or anomaly detection.

(Dataset-3) The set of state vectors will include information about the battery, the inclination, and of course, the geolocation of vehicles. This information is especially useful when solutions integrate the use of UAVs.

In the definition of the model for a database, there are two types of attributes, “Tags” and “Fields”. It must be considered that the attributes defined as “tags” will be stored in the memory of the server until its disappearance from the current shard. On the contrary, the information of attributes defined as Fields will be stored directly on the disk. In this way, queries whose predicate includes filters on attributes defined as Tags will be much faster, as they access memory, as opposed to disk access for Fields. For the correct definition of the type of attribute, two factors must be addressed: (i) The frequency of use for the attribute in the query predicate. (ii) The cardinality presented by the attribute.

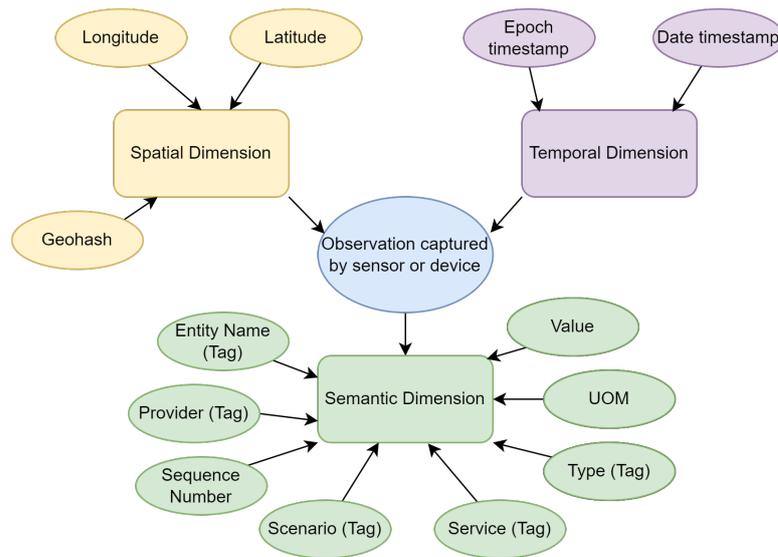
For instance, being a spatio-temporal semantic model, it is understandable to think that all those attributes that present spatial information should be defined as Tags, and their value should be stored in memory (will be subject to frequent consultation). However, as it is an attribute that presents a high variability in its content, and therefore a high cardinality, this decision would lead to an exponential use of memory and resources, even blocking the system.

Figure 1 shows the attribute definition of the proposed model for the set of observations captured by devices (Dataset-1).

IoT devices have limited resources and low computational power, which limits the generation of data with complex syntaxes, such as those in the literature that work with JSON-LD or Turtle (TTL) for RDF graphs. Therefore, for the proposed data model, the JSON syntax is chosen for data modelling, which is much lighter for devices.

The set of **JSON schemas** that defines the spatio-temporal semantic data model is given in [41]. This approach allows for more efficient and lightweight data management, without sacrificing the full characterization of measurements captured by devices under the spatial, temporal, and semantic dimensions. In turn, the modeling of sensor data embedded in collared devices adds livestock management to the paradigm, without the need to use external models.

The model includes schemas for simplified data generation. In this way, the generation and delivery of information by devices with limited resources are enabled. Semantic information about the device must be collected in registry systems, so that the message can be completed in the data management system upon reception, facilitating the transmission from these devices.



**Figure 1.** Data model of Observations.

### 3.2. Performance Indicators

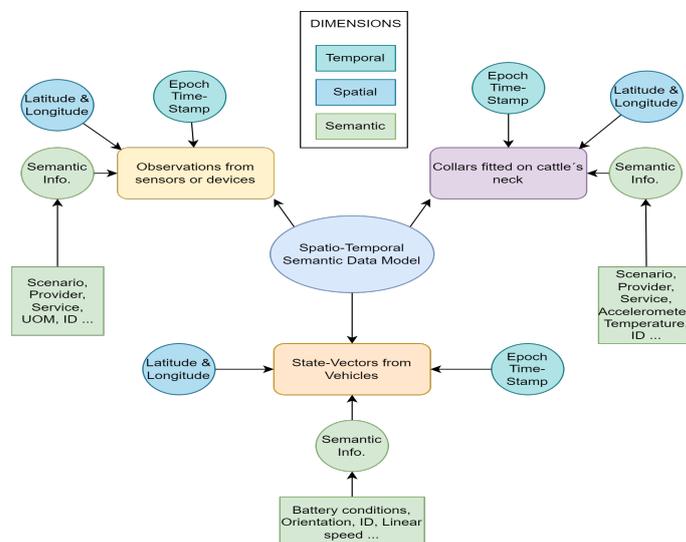
The following performance indicators are described as part of the presented spatio-temporal semantic model:

1. Availability (Ind-I).
2. Resources Consumption (Ind-II). Memory, disk, and server CPU consumption.
3. Response time (Ind-III).
4. Scalability (Ind-IV).

For validation, each of the three datasets defined in the model will be fed with different volumes and structuring in tables, offering multiple experimentation scenarios. In this way, the model will be evaluated under a complete experimentation scenario, taking into account each of the defined indicators.

### 3.3. Data Modeling for Generation and Delivery

This section describes the modeling of information from different agricultural devices, for subsequent submission and injection into the repository. The information modeled by the schemas for each dataset against spatial, temporal, and semantic dimensions is represented in Figure 2.



**Figure 2.** Information associated with each dimension in the modeling schemas.

The model includes three types of schema for modeling the (Dataset-1) set of observations captured by a device (and their simplified versions). (i) The first schema models the observation captured by one sensor or device. (ii) The second schema models the information for several observations captured by a single sensor or device. (iii) The third schema models the information for several observations captured by a device with several sensors installed. An example of modelling a message on the set of observations captured by several sensors installed in a single device (iii) is given below (Listing 1):

**Listing 1.** Observations captured by several sensors installed in a single device.

```

1 {
2   "resourceId": "urn:stsd:AS01:environmentalObservations:
      UPM:soil:KotipeltoFarmWeatherStation",
3   "location": {
4     "latitude": 64.05029,
5     "longitude": 24.72468,
6     "altitude": 450.75
7   },
8   "observations": [
9     {
10    "observedProperty": "wind_speed",
11    "resultTime": 1666453901,
12    "result": {
13      "value": 3.2,
14      "uom": "http://qudt.org/vocab/unit/M-PER-SEC"
15    }
16  },
17  {
18    "observedProperty": "wind_direction",
19    "resultTime": 1666453901,
20    "result": {
21      "value": 295,
22      "uom": "http://qudt.org/vocab/unit/DEG"
23    }
24  },
25  {
26    "observedProperty": "solar_radiation",
27    "resultTime": 1666453901,
28    "result": {
29      "value": 308.96,
30      "uom": "http://qudt.org/vocab/unit/W-PER-M2"
31    }
32  }
33 ],
34 "sequenceNumber": 7324
35 }

```

In real-life scenarios, connectivity is not always available to send the measurements captured by the sensors embedded in the collar fitted on the cattle neck (Dataset-2). Normally, these measurements are sent through a gateway. In the proposed model, two types of schemas are defined for generating and injecting information captured by the collars: (i) The first schema models the information captured by the collar sensors in a single instant of time. (ii) The second schema models the information captured by the collar sensors in several instants of time (accumulated information).

An example of the information collected by the sensors installed in the cattle collar in a single instant of time (i) is shown below (Listing 2):

**Listing 2.** Observations captured by sensors on a collar.

```
1 {
2   "collar": {
3     "resourceId": "7F4F9",
4     "location": {
5       "latitude": 40.698695354,
6       "longitude": -4.532722972,
7       "altitude": 2.10789
8     },
9     "resultTime": 1666453135,
10    "resourceAlarm": false,
11    "anomalies": {
12      "locationAnomaly": false,
13      "temperatureAnomaly": false,
14      "distanceAnomaly": false,
15      "activityAnomaly": false,
16      "positionAnomaly": false
17    },
18    "acceleration": {
19      "accX": 0.2331,
20      "accY": 0.898,
21      "accZ": 0.998
22    },
23    "temperature": 22.5
24  },
25  "sequenceNumber": 903
26 }
```

The modeling of vehicle state vector information (Dataset-3) includes a description of geolocation, orientation, linear speed, and battery of a specific vehicle in a time instant. An example of its modelling for submission is shown below (Listing 3):

**Listing 3.** Vehicle state vector information.

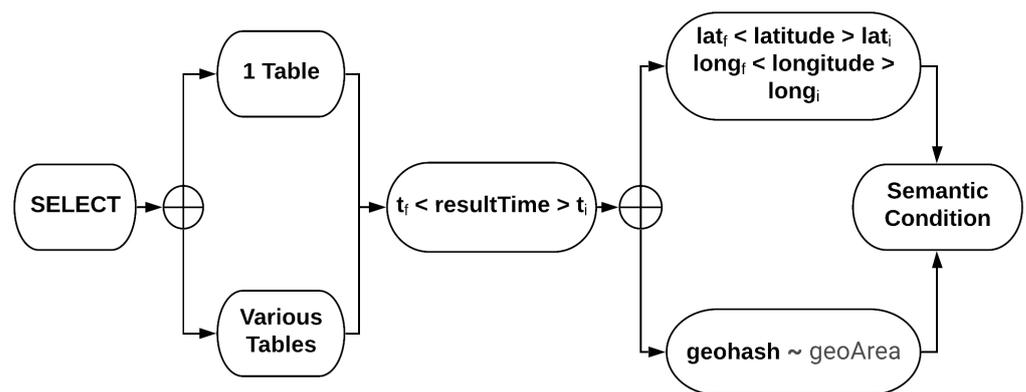
```
1 {
2   "vehicleId": 42,
3   "sequenceNumber": 802,
4   "lastUpdate": 1666452395,
5   "location": {
6     "latitude": 32.34534454345703,
7     "longitude": 12.563429832458496,
8     "altitude": 999.75
9   },
10  "orientation": {
11    "roll": 82.30000305175781,
12    "pitch": 5.300000190734863,
13    "yaw": 2.299999952316284
14  },
15  "battery": {
16    "batteryCapacity": 49,
17    "batteryPercentage": 0.24
18  },
19  "linearSpeed": 11
20 }
```

The timestamp is included in "Epoch"–Unix Timestamp format, generally with precision in seconds (10-digit integer); this will facilitate its generation and simplify the message syntax.

### 3.4. Data Query Functions

The proposed model aims to reduce data gathering times, so query techniques and function syntax constitute a key element to increase performance. Querying precision agriculture data provides the feed for decision-making algorithms, automation, monitoring, and tracking applications, among other use cases. In this way, a wide range of queries must be described. Queries must allow the extraction of the evolution of data over time, geopositioning, and some semantics such as the identification, type, or provider of the device delivering the desired data.

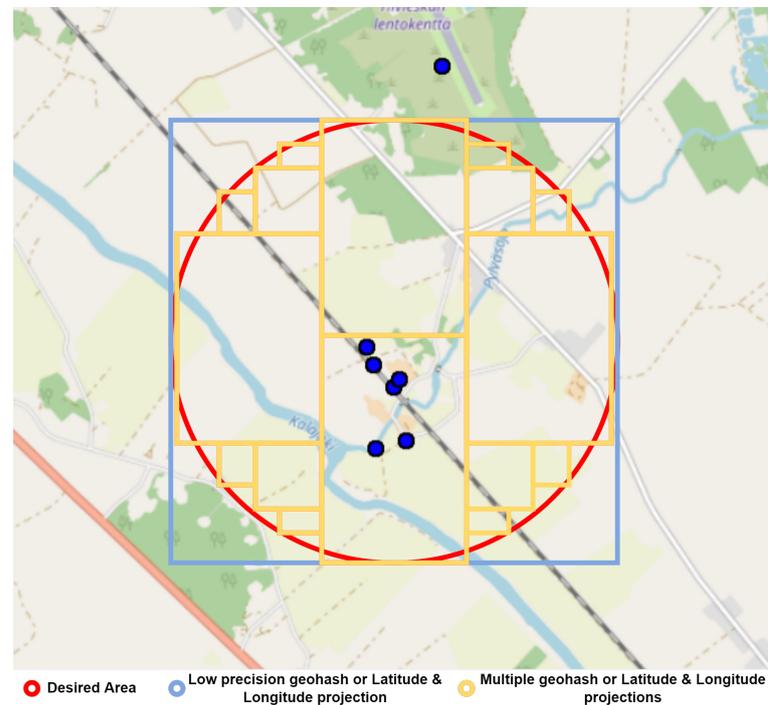
To cover the range of queries required, a complex query with different requirements for the retrieval of data will be described, as well as a set of queries made by splitting the original complex query. The original query should present a minimum of one attribute per dimension; this means at least a time interval, an area geolocated, and an attribute to describe some semantics about the desired data to be extracted. In addition, the complex original query is going to be adapted to three different configurations, varying the size of the time and geolocation windows of the search. The original query is described in Figure 3.



**Figure 3.** Spatio-Temporal Semantic Query definition. Query parameters ( $t_i$ ,  $t_f$ ,  $lat_i$ ,  $lat_f$ ,  $long_i$ ,  $long_f$ ,  $geoArea$ ) ( $t_i$  = Initial time,  $t_f$  = Final time,  $lat_i$  = minimum latitude,  $lat_f$  = maximum latitude,  $long_i$  = minimum longitude,  $long_f$  = maximum longitude,  $geoArea$  = Circumscribed bounding box defined through low precision geohash projection).

During the evaluation stage of the spatial clause construction for data gathering, three types of filtering have been described (see Figure 4). (i) Circumscribed bounding box defined through low precision geohash projection; (ii) multiple geohash or latitude and longitude projections of different precision to adjust to the search area; (iii) circumscribed bounding box defined by two points (latitude and longitude) determining the diameter of the search area. After repeating the set of tests defined in the experimentation section, it has experienced delays that are too high for the first two configurations, even reaching the timeout configured on the server, making an agile query unfeasible, close to real-time. This was the expected result, as the TSDBs are designed to optimize retrieval time under temporal filter conditions, increasing its response time when the number of semantic predicates is increased or in response to "like" type queries (*Like* type queries are queries based on matching patterns. In the case of the spatial predicate of a query, it could be used to filter by those geohashes that start with a certain pattern. This allows the accuracy of the search geohash projection to be adapted). However, agile responses were obtained for the delimitation of areas by means of rectangles made up of two latitude and two longitude coordinates. Therefore, the options of processing with geohash or with more than one delimitation area for the composition of a superior area have been discarded from the

experimentation. This is due to the lack of sufficient performance for a production scenario, and because it exposes the server to a load too high for concurrent accesses.



**Figure 4.** Spatial clause.

For a precise study of each possible query to be performed, several aspects must be considered when conducting experiments. To measure and evaluate the performance of the proposed data model, queries will be performed according to the following cases:

- **Isolated vs. Concurrent injections** (Section 4.5). The response of the injection engine to injection requests from one or more client-threads is analyzed. It allows drawing a performance curve according to the type of database attack (Ind-I and IV).
- **Isolated Cached queries** (Section 4.6). Execute each query for a number 'n' of iterations without altering the semantics of the query (response retrieved from cache) (Ind-III).
- **Isolated no-Cached queries** (Section 4.7). A measurement of the different time responses for queries that do not remain in cache (Ind-III).
- **Concurrent Injection-Query** (Section 4.8). Execution of a set of queries, each one on a thread-client concurrently and in parallel to the injection of data through several injector clients (Ind-II, III, and IV).
- **Sharding** (Section 4.9). A direct relationship is established between the delay times for the extraction of data stored on disk or in RAM. The same test battery is run for data in an active and non-active shard (Ind-III).

#### 4. Data Model Validation through the TSDB Engine InfluxDB

Due to the high performance exposed in the literature and its previous use in the implementation of the AFarCloud project repositories, the InfluxDB database engine will be used for the validation of the proposed data model. However, the objective of the article is not the measurement of performance or comparison between the existing TSDBs, but the validation of the proposed data model, checking that it can offer support in a real-time scenario.

Through various tests, the objective is to measure performance against the different indicators described in the definition of the proposed model. The validation of the proposal

is performed through experimentation and measurement with data from real devices and sensors, deployed in the AFarCloud European project scenarios.

#### 4.1. Real Data Modelling from the AFarCloud Framework

The AFarCloud (Aggregate FARMing in the CLOUD) ECSEL JU project [42] presents a distributed platform capable of offering integration and cooperation of agriculture cyber physical systems to increase efficiency, productivity, food quality, animal welfare, and to reduce costs in agricultural labors. Focused on real-time data exchange, it offers services for mission management, task automation, decision support, or even data processing and analysis, among others.

For the evaluation and validation of the proposed model, real data are extracted from the AFarCloud project scenarios, structured in each of the defined sets, and modelled according to the spatio-temporal semantic proposal. A set of measurements generated by a multitude of sensors of different nature deployed on farms has been chosen (soil sensors, atmospheric sensors, crop management sensors, agricultural machinery sensors, etc.), contemplating the broad agricultural spectrum. AFarCloud comprises a total of 11 scenarios in Europe. In this way, the evaluation of the model offers a generic view that can be applied to any type of device and sensor used in an agricultural scenario. In Table 2, the volume of each dataset, its modelling against the three defined dimensions, and its structuring according to the type of information provided by each dataset is presented.

**Table 2.** Data model for datasets 1, 2, and 3.

		Single Database					
		Dataset-1 (Observations)		Dataset-2 (Collars)		Dataset-3 (Vehicles)	
<b>Number of points</b>		7,652,221		743,295		34,602	
<b>Dimensions</b>	<b>Time</b>	Timestamp (Epochtime)					
	<b>Geoposition</b>	1-Latitude and Longitude (Float) 2-Geohash (String)					
	<b>Semantic</b>	Semantic information regarding data nature, identifiers, and service provide					
<b>Structure</b>	<b>Tags</b>	6 attributes		5 attributes		1 attribute	
	<b>Fields</b>	6 attributes		15 attributes		10 attributes	
		Float:	5	Float:	8	Float:	9
	String:	1	Boolean:	6	String:	1	
			String:	1			

Time series are stored in the form of tables within the database. This has a direct bearing on the planning and execution time of queries by TSDBs engine. The structure defined for this work replicates two different cases to analyze queries in a given dataset that resides in one or several tables (“measurements” in the TSDB notation):

- Dataset-1 (Observations): 42 tables;
- Dataset-2 (Collars): 1 table;
- Dataset-3 (Vehicles): 1 table.

#### 4.2. Equipment Used

The experiments are performed with stable laboratory equipment. Kernel updates and any possible software updates have been blocked during testing so that no delay or alteration of the measurements can be experienced by any kind of process outside of the experimentation.

The hardware requirements of the InfluxDB host server are determined according to the official documentation provided in [43]. Taking into account the high volume of queries and injections in the agricultural IoT scenario under study, a server is built based on the “high performance” specification.

For the tests carried out in this paper, machines with similar characteristics were used. One is configured as a server, and the others work as clients, responsible for submissions to the database hosted on the server. Its characteristics are shown in Table 3.

**Table 3.** Equipment Used.

	Client Host	Server Host
<b>OS</b>	<b>Microsoft Windows 10 Enterprise</b>	<b>Ubuntu server 18.04.3 LTS</b>
<b>Mother Board</b>	MSI MS-7A72	MSI MS-7A72
<b>Architecture</b>	64-bit	64-bit
<b>Mic</b>	i7-7700 3.6 GHz	i7-7700 3.6 GHz
<b>N° Cores</b>	4 (8 threads)	4 (8 threads)
<b>RAM</b>	<b>8 GB</b>	<b>64 GB</b>
<b>Disk</b>	Samsung SSD 850 EVO 500GB	Samsung SSD 850 EVO 500GB

The version of InfluxDB used for the experimentation is 1.8.0.

#### 4.3. InfluxDB

InfluxDB is an open-source time series database written in Go; InfluxDB databases are NoSQL. The query language implemented by InfluxDB is an SQL-like query language defined as InfluxQL. InfluxDB is not oriented to geospatial queries; therefore, it is an optimal experimentation environment for testing the query functions defined in the model.

InfluxDB works with in-memory indexing and the time-structured merge tree (TSM). The TSM has a write ahead log (WAL) and a set of read-only data files, which represent a similar concept to the sorted strings table (SSTables) in an LSM Tree. SSTables are a format for storing key-value pairs in which the keys remain in sorted order. An SSTable will consist of multiple sorted files called segments. These segments are immutable once they are written to disk.

The WAL is a temporary cache for recently injected points (writes). To reduce the frequency with which permanent disk storage files are accessed, InfluxDB stores the new points in the WAL and groups them in batches until their total size or age triggers a flush into permanent storage. This allows an efficient grouping of writings in the TSM.

To speed up the InfluxDB engine searches, a definition must be made of each of the attributes associated with a temporary series, such as the aforementioned “Tags” or “Fields”. To evaluate the proposed model, the definition of the type of attribute shall be as described in the model definition.

#### 4.4. Experiments

To analyze the performance indicators defined in the proposal against the TSDB system, such as the one provided by the InfluxDB engine, the different characteristics presented in its operation must be considered. In Table 4, the time consumed to perform data injection is referring to datasets-1, 2, and 3.

The total disk space allocated for the storage of the 8430118 data points regarding the three datasets is 165 MB, a low disk space consumption referring to Ind-II.

The main objective of the proposed agricultural data model is to enable real-time data gathering. To evaluate the query functions defined in the model, a complete set of queries

is exposed, addressing each of the defined dimensions (spatial, temporal, and semantic). In this way, the model is intended to be evaluated against different configurations of table structuring and database loading.

**Table 4.** Data Injection through InfluxDB.

InfluxDB					
Dataset-1		Dataset-2		Dataset-3	
N° Data Points	Injection Time (ms)	N° Data Points	Injection Time (ms)	N° Data Points	Injection Time (ms)
7,652,221	271,471	743,295	37,413	34,602	2054
N° points/s	28,188	N° points/s	19,867	N° points/s	16,848

The set of defined queries complies with the following filtering conditions for the different experiments performed (Tables 5 and 6).

**Table 5.** Cached query sets.

Clauses-Cached Queries							
Query Batteries	Time	Geopositioning	Semantics	Grouping	Ordering	Limit	
1	24 h	Square of 1 km	Service type	By entity name	Ascendent time	10	
2	24 h					10	
3	2 h		Service type				
4							
5			Service type				10
6	24 h		Service type				10

**Table 6.** Sequential query sets.

Clauses-Sequential Queries							
Query Batteries	Time	Geopositioning	Semantics	Grouping	Ordering	Limit	
1	30 m * Half-hourly intervals	Sequential squares of 1 km * Always within the margins of the scenario under study.	Service type	By entity name	Ascendent time	10	
2						10	
3			Service type				
4							
5			Service type				10
6			Service type				10

The following subsections detail each of the experiments performed, and present the response of the system to the proposed model.

#### 4.5. Isolated vs. Concurrent Injections

Precision farming scenarios are characterized by a large volume of devices constantly and concurrently measuring and injecting data into the system. Therefore, evaluating the performance of the model against writes is vital for its validation in a real-time environment.

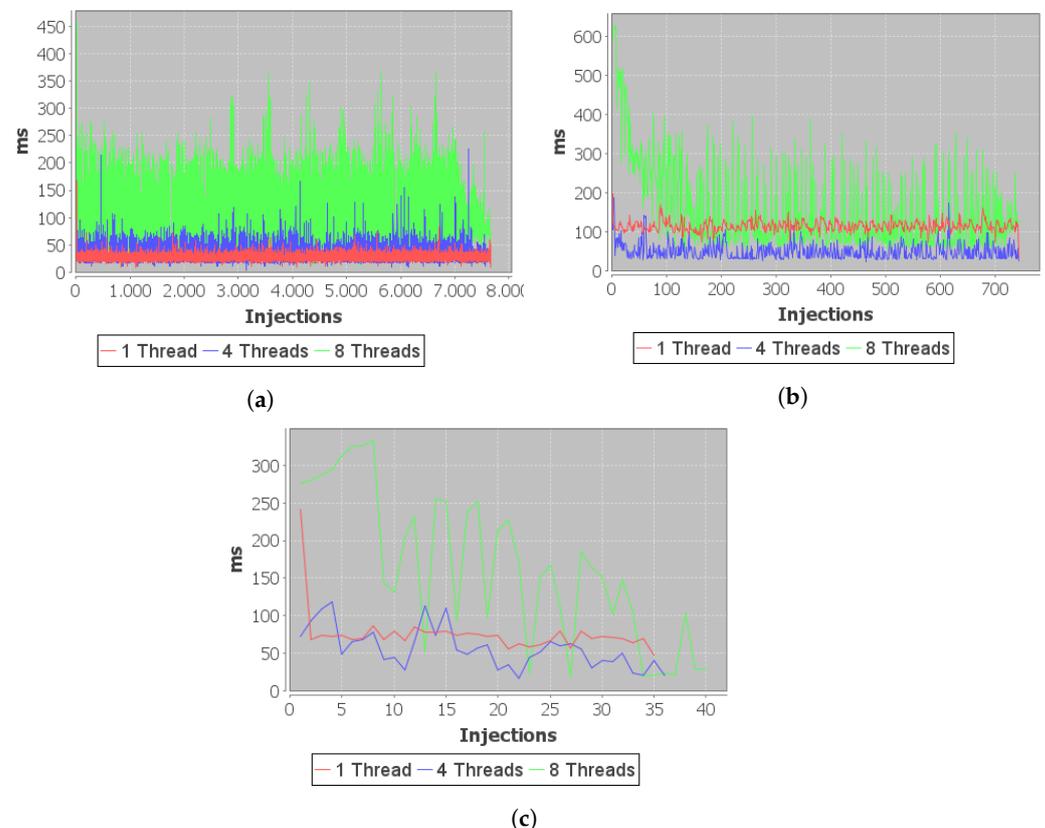
In the first set of tests, it is going to proceed to the evaluation of data injection into the repository. For this purpose, the tests are divided according to the nature of the data to be injected and the differences between the injection times for the structures in one or several tables.

Similarly, a comparison is made between sequential injections and concurrent injection requests by several clients.

Concurrent data injection into the repository is performed according to the capabilities of the host server. For this purpose and considering the total number of core threads (A thread is a virtual component that manages the tasks of the core. Usually, each core is composed of one or two threads, depending on the architecture) in the experimental equipment, concurrent injection requests will be made between four and eight clients (In the experimentation described in this article, a client is a person or program that performs data injection or query requests. A given client request will be attended by a single thread on the server hosting the repository. Accordingly, the number of clients making simultaneous requests corresponds to the number of threads that serve the request in the repository. Four clients making simultaneous injection requests = four threads in charge of the injection process in the repository), and will be compared to injection requests from a single client. Each injection represented on the x-axis of the graphs corresponds to the writing of 1000 data points. Figure 5a shows the injection times for the total dataset of observations (dataset-1—7,652,221 data points).

To evaluate the behavior and performance against the dataset referred to the measurements extracted by the collars, the process is repeated for dataset-2. This dataset is more structurally complex, but will be stored in a single table (see Figure 5b).

Finally, the response obtained against a smaller dataset, composed of vehicle state vectors, is evaluated. Figure 5c shows data injections concerning dataset-3.



**Figure 5.** Sequential and concurrent dataset injections. Injection time in ms (milliseconds). (a) Dataset-1 Injections. (b) Dataset-2 Injections. (c) Dataset-3 Injections.

After the set of experiments is performed for sample injection into the repository, a correlation is observed between the number of clients attacking the database and the total sample injection time. The server is overloaded for a large number of clients with simultaneous requests, affecting the scalability of the system and the availability of data gathering (Ind-I and IV).

In addition, the structure of the data to be injected directly influences the results obtained. To visualize the influence of each of the factors studied, the number of samples injected per second is shown for the different cases under study (Table 7).

The results reflect a clear influence of an overload on the system. Sample injection will be more efficient when using a larger number of clients attacking the database. However, when attacking the host server with several clients close to or equal to the total number of threads presented, the injection will slow down and will be reflected in less efficiency. This is the expected behavior.

**Table 7.** Injections (samples/second).

	Number of Injections per second (Samples/s)		
	Dataset 1	Dataset 2	Dataset 3
<b>1 Thread</b>	34,905	8698	13,007
<b>4 Threads</b>	28,188	19,867	16,848
<b>8 Threads</b>	11,099	6453	5250

In contrast, complexity, that is, the number of fields and tags defined for a dataset, will directly influence the sample injection process. Thus, it is observed that for less complex data structures, such as those presented in the first dataset samples, the injection efficiency increases when such injections are performed from a single client sequentially. On the contrary, when injecting time series with high structural complexity (a higher number of fields and tags), the injection performance peaks when using a larger number of clients or injector threads, without reaching the maximum number of host threads.

We observe the evolution of injection time (Ind-III) for datasets 2 and 3, which are structured in a single table within the database. The first injections will be delayed by the creation of the table structure and the indexing of tags in the server's memory. Subsequent injections show a decrease in the injection time used. On the contrary, dataset-1 represents a constant injection time, due to the constant creation of tables during the injection of new data, due to the defined structure, showing some delay spikes in the creation of new tables.

The proposed model presents very fast injection times. The model presents a better injection performance than that required in a typical agricultural scenario. The writing and storing of data in the repository will not be a problem for the needs of real-time environments in agriculture. However, before concluding, the query time offered under the different scenarios should be evaluated.

#### 4.6. Isolated Cached Queries

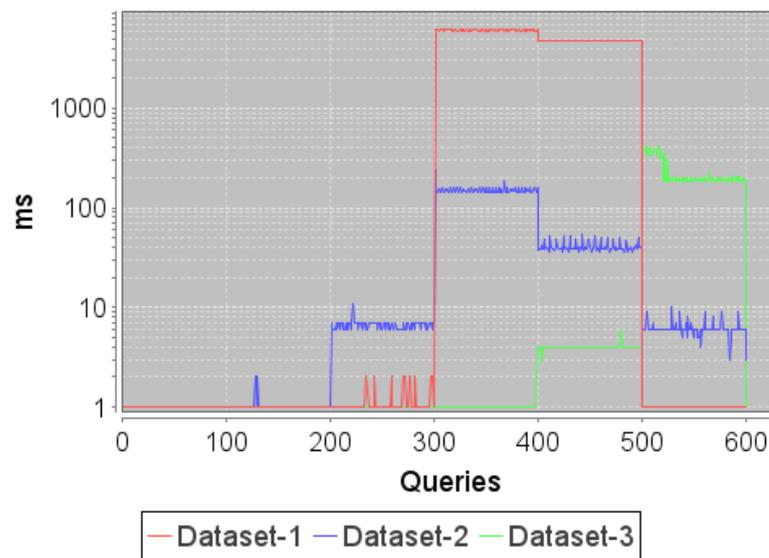
In a real scenario, when querying data are residing in the repository, updates or new data are not always extracted. For instance, this may be to monitor the temperature and humidity of a certain crop field. Therefore, it is important to know the response to the extraction of data residing in tables that have not been altered (no new data have been injected between consecutive queries). Implementing a query cache would speed up the response to such queries and reduce the use of server resources.

To evaluate the proposed agricultural model against data gathering of cached queries, the previously defined set of queries is used (Table 5). These queries have been executed by a unique client attacking the server (see Figure 6a). To measure and compare performance with a concurrent attack, the process will be repeated in the following subsections. The experiment is executed for each of the defined datasets. The response of the system allows us

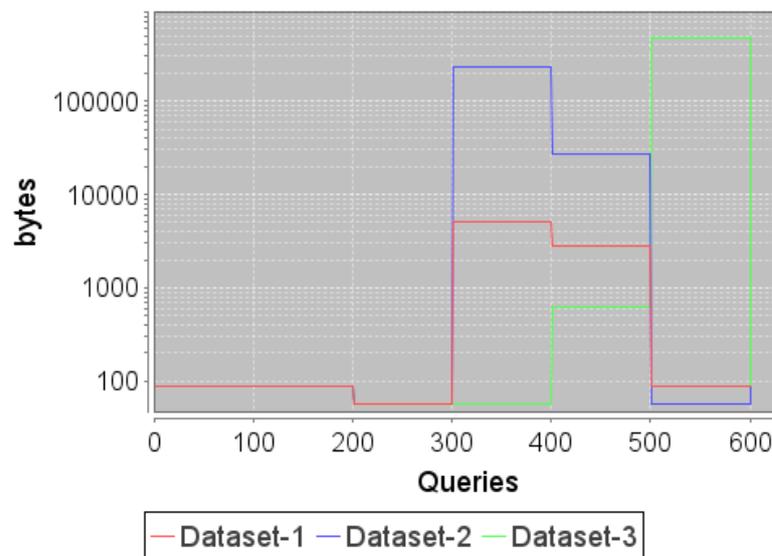
to evaluate the behavior for the different structures and volume presented by each set. The requests made for each query have been studied to establish a relationship between query time, serialization, and volume of extracted data. A representation of the volume in bytes of the response to each query can be seen in Figure 6b.

The processing performed for queries by the InfluxDB engine can be observed natively in the tool, thanks to the Explain analyze API [44]. By decomposing each of the actions performed by the InfluxDB engine for data gathering through the query sets performed in this experiment, a direct relationship is found between the volume of data and the number of tables to be searched, with the measured response time.

Analyzing the results obtained in the experiment, it is observed that the Influx Engine (as well as other TSDBs) does not use query response caching. Therefore, the response times are constant for a given query, without exposing a maximum delay for the first query performed, and a decrease in response time for the following identical queries (Ind-III).



(a)



(b)

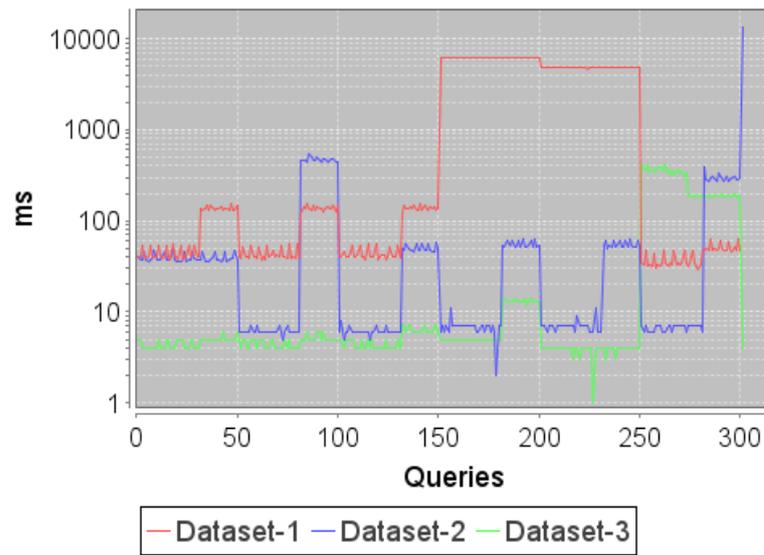
**Figure 6.** Cached query sets. (a) Cached query sets. Response time in ms (milliseconds). (b) Cached query sets. Response volume in bytes.

Query caching should be implemented on top of the database system to reduce the response time and resource use for identical queries performed consecutively. The implementation of caching for query responses should only be applied to historical queries. Data residing in active shards (current data) are targets of continuous writes, and the use of caching for responses to these queries would prevent the collection of the most recent writes.

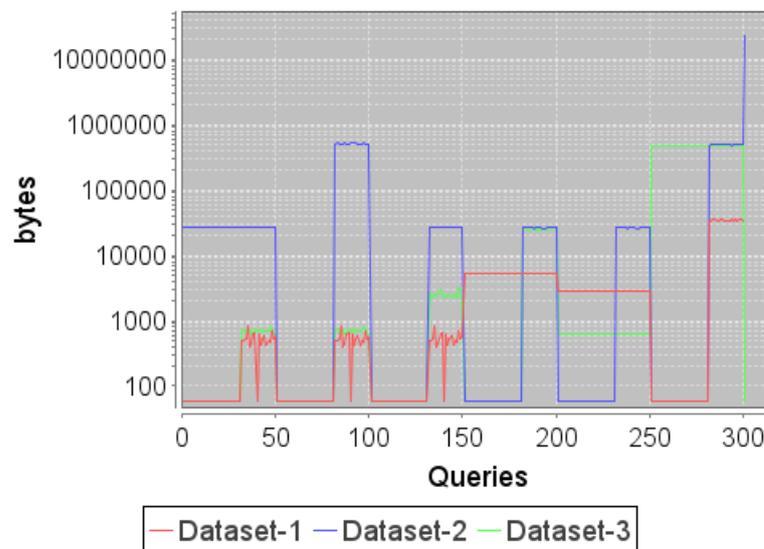
4.7. Isolated No-Cached Queries

In the cached test set, a stable response time has been experienced between repetitive queries, demonstrating that the caching of the query does not have an impact. However, to analyze the model response to sets of queries with variant filters, the process defined in the previous section has been repeated for the sequential queries defined in Table 6.

The response time to the execution of the set of sequential queries is presented below (see Figure 7a).



(a)



(b)

Figure 7. Sequential query sets. (a) Sequential query sets. Response time in ms (milliseconds). (b) Sequential query sets. Response volume in bytes.

By studying the volume of data extracted for each query (see Figure 7b) and breaking down the plan executed by the search engine for data retribution, the impact on the delay times added to each query can be observed.

With respect to Ind-III, the time costs of each of the operations performed in data gathering have been divided (the planning and the execution of the defined plan). One of the main sources of delay in the data gathering process is the extraction and serialization of the data to compose the response to the query. For this reason, there is a direct relationship between the volume in bytes extracted by the query and the total delay time for query execution. However, it can be observed that the response times captured for the requests from dataset-1 are notably higher, even though the volume of data extracted is smaller. This high delay is introduced by two different stages in the data extraction operation: development of the extraction plan itself, which will need to define a simultaneous attack on different tables for data collection; and the delay introduced by the concurrent search in the various tables and their subsequent serialization for the composition of the response. This process is simpler for dataset-2 and 3, where data are structured in a single table.

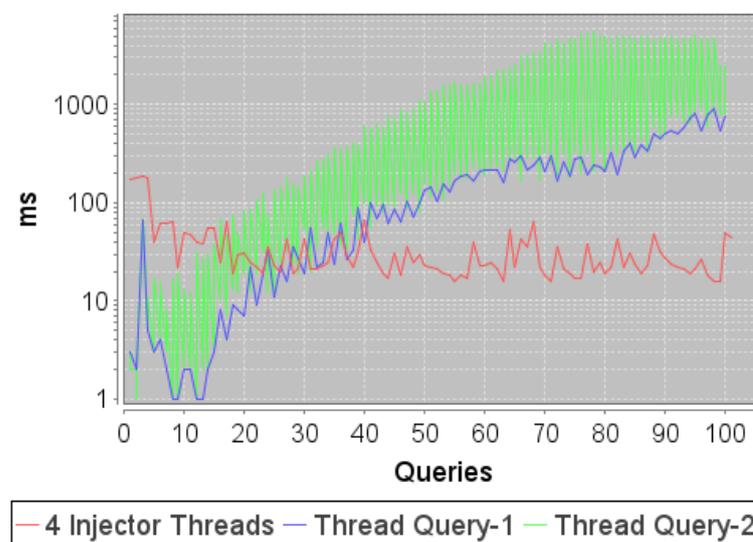
#### 4.8. Concurrent Injection-Query

In a real precision agriculture scenario, access to repositories is subject to several types of simultaneous requests. Therefore, this section will evaluate the impact of concurrent database accesses (Ind-III and IV). To represent a database load state, it will be excited through various concurrent injections and queries, causing different loads on the server resources where the database is located.

Figure 8a shows the response time to concurrent injections and queries by several client threads. Figure 8b shows the volume of data retrieved by each of the query threads. There is a direct relationship between the delay time experienced in queries and the volume of data to be extracted. In addition, the queries executed by the consultant threads retrieved data from dataset-1 (dataset injected by the injector thread), so that as injections are performed, the number of tables on which data are extracted increases.

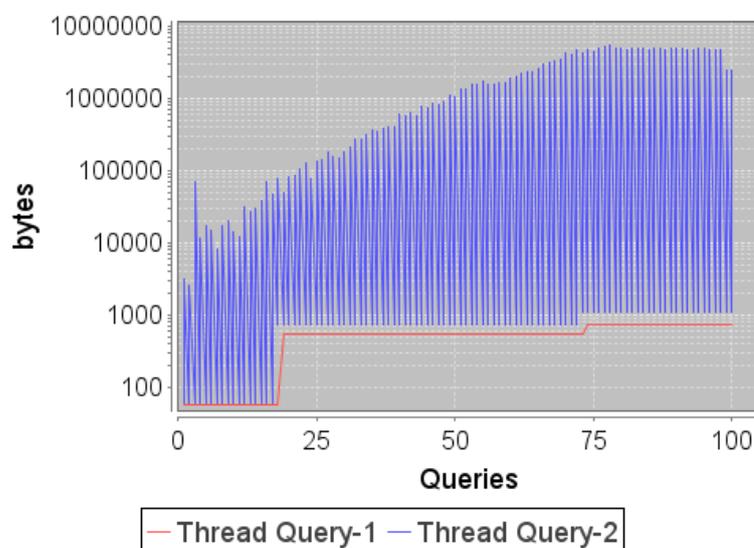
These two factors, the increase in data volume and in the number of tables on which the data is distributed, result in an exponential increase in the retrieval time experienced.

By using a larger number of tables for the organization of data subject to recurring queries, response times (Ind-III) and the use of server memory and disk resources are increased (Ind-II), so its scalability will be lower, since a large number of simultaneous queries with long delay times would block the system (Ind-IV).



(a)

Figure 8. Cont.



(b)

**Figure 8.** Simultaneous requests. Concurrent injections and data gathering. (a) Concurrent injection and data gathering. Injection and response time in ms (milliseconds). (b) Volume of data extracted by the consultant threads.

#### 4.9. Sharding

To analyze the delay introduced by the model for gathering data residing in an uncompact or compacted shard, two different scenarios are considered: querying or extracting data from an active shard or from an inactive shard (Ind-III).

The data writing performed by a TSDB engine on an injection request is affected by the type of shard it attacks. In most cases, a write is performed on the active shard, as this is an agricultural scenario where a write is performed about current data. When receiving a write request on a cold or inactive (disk-compacted) shard, the write requires prior decompaction and subsequent compaction of the shard. Therefore, writes of historical data, on cold shards, are slowed down by the process of decompaction and subsequent compaction.

In addition, to measure the impact on delay when querying data residing in hot (active) shards versus querying the same data residing in cold (inactive) shards, a comparison has been made for the three datasets defined in the paper. Figure 9a shows the total time taken to extract data related to dataset-1.

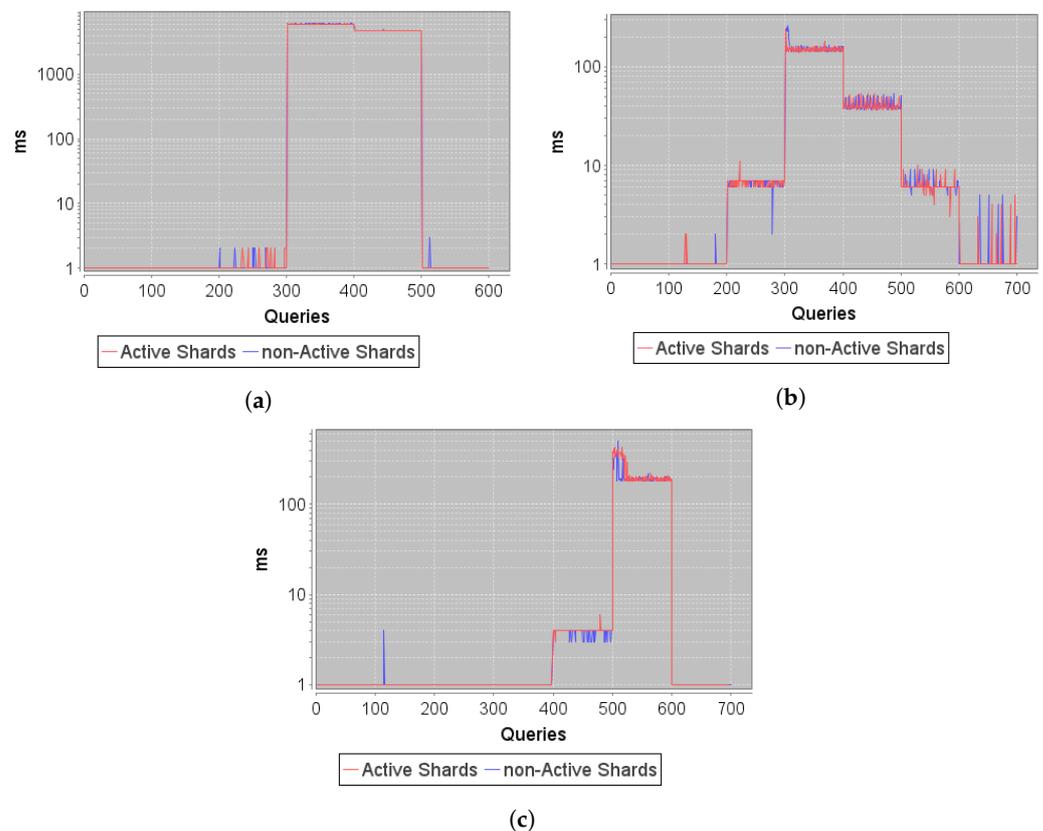
The process is repeated with the battery of queries for dataset-2 and 3, and a similar behavior to that described by the experiment carried out with dataset-1 is observed (see Figure 9b,c).

It is observed that data query times do not increase whether the data reside in an active or inactive shard. On the contrary, a significant delay would be experienced if the data were distributed in different shards, and the operation of data retribution would have to be decomposed in the attack on the different shards containing the measurements to be searched.

For the proposed data model, as long as queries are performed on a single shard, the gathering times will be really low, regardless of whether the objective of the query is historical or current data. Therefore, a positive model evaluation is assumed for data gathering in a precision agriculture scenario requiring real-time performance.

The model proposed in this paper provides a more efficient use of computational resources, enabling the management of large spatio-temporal semantic datasets generated by sensors and agricultural devices. The model enables the generation of telemetry by devices with lower resources, thanks to a lightweight syntax, without reducing the semantic spatio-temporal information provided by the device. The response time to data injection and query, presented in the evaluation of the model, corroborates its performance in a real-time environment, improving the response for implementation in monitoring systems

and algorithms or models for automation or decision support. Finally, the proposed model offers a reduction in the data management effort which offers more efficient management. This allows for the use of resource-limited equipment without sacrificing performance, for data management and repository implementation.



**Figure 9.** Comparison between access to active and inactive shards. Response time in ms (milliseconds). (a) Dataset-1 queries. Access from non-active shards or from active shards. (b) Dataset-2 queries. Access from non-active shards or from active shards. (c) Dataset-3 queries. Access from non-active shards or from active shards.

## 5. Results

The implementation of a data management architecture for agriculture is divided according to two main cases: (i) the extraction of the latest captured measurements, which should be in real-time, allowing the rapid detection of anomalies, enabling the acting or the representation of the current status of crops and livestock; (ii) historical data retribution, which needs to perform searches over large volumes of data, for cases such as feeding decision making or machine learning algorithms.

Due to the number of attributes in the point structure of the agriculture datasets described in the proposal, TSDBs perform batching of up to 1000 points in the WAL (the maximum size of the WAL segment is 10 MB). This reduces the number of accesses to permanent disk storage for writes. WAL is also used as protection against the loss of recently added data on power loss. However, the WAL storage format is not easily queryable.

As a result of experimentation, additional delays on querying the last injected data have been observed due to this use of the WAL by TSDBs and, in particular, by InfluxDB. Therefore, a data management system is proposed in which the latest measurements captured by the sensors are also stored in a small SQL (relational) database. By overwriting these values as new values, they are captured by the sensors. The latest measurements can be retrieved in real-time from the SQL database. This avoids querying TSDBs data stored in the WAL.

Furthermore, to reduce the network load, it is recommended to send the injection request to the TSDB, directly in a batch of points, tuned with the WAL segment. For complex structures, between 10 to 20 attributes per point, the most efficient grouping of points in a client will be between 1000 to 2000 points per batch. In an agricultural IoT scenario, it is recommended to collect write requests to form a batch at the edge. The time window for the collection of points in a batch at the edge should not exceed 100 ms for a real-time scenario.

The InfluxDB TSM engine and that in general TSDBs are not recommended for scenarios with constant writes or alterations to historical data. To inject historical data into a cold shard, it must have been previously decompressed, written, and then compressed again. This slows down the writing of historical data.

To increase the efficiency in injecting or querying agricultural data from the repository, several factors must be considered.

Firstly, a server hosting the repository must be chosen, with a number of cores thought in relation to the number of clients or devices that can perform queries or data injections concurrently at a given instant of time to reduce delays.

Secondly, we look at the structure of the data to be injected. It must be attempted to reduce the definition of tags to those attributes that are going to be filtered in the queries more frequently, always making sure that these attributes do not have a cardinality that is too high; that is, the fewer values these attributes can take, the greater the efficiency due to a lower memory load on the server.

In precision agriculture, the use of tags is only recommended in attributes with information concerning the type of sensor and service offered, the scenario, provider, unit of measurement (uom), and device identifier. Measurement values, geopositioning, sequence identifiers, and attributes with a high range of possible values should be avoided.

Unlike tags, fields are not indexed, which implies a sequential scan of the field column values. Field-based queries, which increase response time directly proportional to the volume of data in the query target, should be avoided. Due to the structure of the data model, queries will slow down when extracting data are residing in multiple tables; however, a response time suitable for a real-time performance scenario is still observed.

The division of data into tables is recommended according to the main query targets. The aim is that queries only need to attack one table at a time. In this way, query times are reduced due to the simplification of the plan design for the extraction and subsequent serialization of data by the TSDBs engine. Therefore, structuring in tables is defined according to the type of device and the nature of the measurement (Table 8). This structure offers a maximum reduction in the number of tables without obtaining a volume of data per table that is too high.

**Table 8.** Structure of the database in tables.

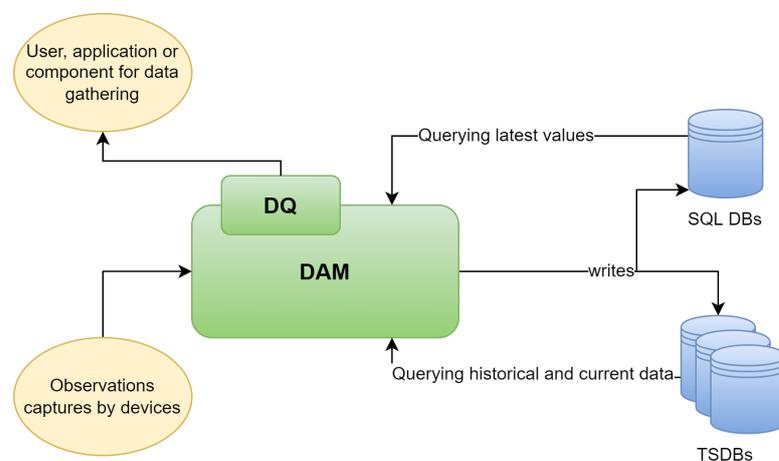
	<b>Structuring by type of device</b>	<b>Structuring by nature of measurement</b>
<b>Definition of tables in the database according to:</b>	Livestock collar or device	Division is not recommended
	Vehicle or robot	Division is not recommended
	Field device	Climatic information Device status Specific crop information

Regarding the composition of spatial clauses defining a search area, as query arguments, it will be more efficient to establish a bounding box type search. The bounding box will be conformed by latitudes and longitudes, describing a square. “Like” type queries (as the use of geohash) or the definition of complex areas through a multitude of geolocated

points will slow down queries. By defining spatial filtering in the form of a square, only four spatial filtering parameters (two latitudes and two longitudes) will be presented.

This type of query structure allows the definition of three dimensions of filtering on the agricultural data repository, spatial, temporal, and semantic, without adding high delays in data extraction and allowing efficient exploitation with a minimum amount of resources on the host server.

The architecture, the spatio-temporal semantic data model, and the configurations and structures implemented in this article result in the Data Access Manager & Data Query (DAM & DQ) data management system. This system provides a distributed repository between TSDBs and relational databases to overcome the delays added by the WAL management in the TSDBs for querying the latest data Figure 10. This system has been implemented as the core of the semantic middleware of the AFarCloud intelligent platform, offering real-time performance. In this way, the platform can offer real-time actuation or detection solutions.



**Figure 10.** Data Access Manager & Data Query (DAM & DQ).

The source code of the architecture, the description of the components, and the license scheme are available in [45].

Thanks to the Afarcloud project, with multiple live scenarios, it has been possible to apply the solution described in this article as a data management system to feed various decision-making, AI, and monitoring systems. One of the use cases using this data management system is the work described [46]. It presents a smartphone application that allows the monitoring and management of assets from the field in real-time. In addition, the application allows the creation of observations, evaluations, and alarms associated with the devices, including the user in the flow.

To ensure the interoperability of the data model and the data management system proposed in this article, it is intended to continue with research to offer interoperability through the preparation and integration of data with other existing models. Similarly, it is intended to distribute the data management system between edge, fog, and cloud, offering high availability of data management in real-time, from the farm itself or from the cloud.

## 6. Conclusions

The projects developed in the field of precision agriculture present a wide range of solutions for automation, monitoring, and increased efficiency in agricultural activities. These projects are made up of a large number of sensors and devices that make these tasks possible. The article discusses the characterization of device data across three dimensions: spatial, temporal, and semantic.

The deployment of numerous agents for the digitization of agricultural processes involves the generation of large volumes of data [39], increasing the *digitization footprint* generated by the agricultural sector. The model and architecture proposed in this article

offer an improvement in resource management, reducing the storage space required and increasing management performance, thus reducing the digitization effort.

Many of the solutions presented in the field of precision agriculture include the operation of autonomous vehicles or the response to emergency situations in crops or livestock. To provide such solutions, a system that offers real-time data management is required. To reduce the latency times in data retribution, an agricultural data management system based on the proposed model and a distributed repository has been exposed. Several measures have to be taken for the structuring of data and the specification of techniques in the construction of the query clauses to enable real-time gathering, and reduce resource usage.

The paper exposes how the delay times for data gathering are directly affected by the structuring of data and the attack on various tables for a given query. It explains how to reduce the number of clauses to improve query speed and describes the impact. Even the implementation of a manager to handle the cache or the access to the latest measurements or historical data is explained.

Thanks to the precise spatio-temporal semantic characterization offered by the proposed model, the risks and uncertainties involved in agricultural diversification are minimized. The management system allows us to evaluate the historical and current conditions of a given piece of land to assess the feasibility of planting a new crop. Furthermore, the data management system offers an entry point for feeding machine learning models, allowing the comparison of historical conditions offered by the terrain and those necessary for a high yield of a variety of new crops. In this way, the farmer will be limited to choosing among the proposed crops, for which a correct yield is guaranteed.

The structuring, data model, and data management system implemented and described in this article have been applied to the AFarCloud project, which develops a platform for precision agriculture, applied to multiple live scenarios. This framework allows the testing of the solution against real data. Thanks to the use of the system and techniques presented in the paper, the use of resources has been reduced, and data gathering by the semantic middleware of the AFarCloud project has been accelerated. This has sped up and improved the responsiveness of various solutions and applications developed within the project framework.

**Author Contributions:** Conceptualization, M.S.E.d.l.P. and S.L.S.; data curation, M.S.E.d.l.P. and J.-F.M.-O.; formal analysis, M.S.E.d.l.P. and M.M.E.; funding acquisition, J.-F.M.-O.; investigation, M.S.E.d.l.P., S.L.S. and M.M.E.; methodology, M.S.E.d.l.P., S.L.S. and J.-F.M.-O.; project administration, M.S.E.d.l.P. and J.-F.M.-O.; software, M.S.E.d.l.P., S.L.S. and J.-F.M.-O.; supervision, M.S.E.d.l.P. and J.-F.M.-O.; validation, M.S.E.d.l.P., S.L.S., M.M.E. and J.-F.M.-O.; visualization, M.M.E.; writing—original draft, M.S.E.d.l.P.; writing—review and editing, M.S.E.d.l.P., S.L.S., M.M.E. and J.-F.M.-O. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by AFarCloud project, which has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No. 783221. The JU receives support from the European Union's Horizon 2020 research and innovation programme, and Austria, Belgium, Czech Republic, Finland, Germany, Greece, Italy, Latvia, Norway, Poland, Portugal, Spain, and Sweden. This publication is part of the project PCI2018-092965 funded by MCIN/AEI/ 10.13039/501100011033 and by the "European Union".

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Dpicampaigns. Take Action for the Sustainable Development Goals. Available online: <https://www.un.org/sustainabledevelopment/sustainable-development-goals/> (accessed on 19 October 2022).
2. Animal Welfare. Available online: [https://food.ec.europa.eu/animals/animal-welfare\\_en](https://food.ec.europa.eu/animals/animal-welfare_en) (accessed on 21 October 2022).
3. Home | Food and Agriculture Organization of the United Nations. Available online: <https://www.fao.org/home/en> (accessed on 11 October 2022).
4. International Fund for Agricultural Development. Available online: <https://www.ifad.org/en/> (accessed on 11 October 2022).
5. Martin. Goal 2: Zero Hunger. Available online: <https://www.un.org/sustainabledevelopment/hunger/> (accessed on 21 October 2022).
6. Achour, Y.; Ouammi, A.; Zejli, D. Technological progresses in modern sustainable greenhouses cultivation as the path towards precision agriculture. *Renew. Sustain. Energy Rev.* **2021**, *147*, 111251. [[CrossRef](#)]
7. Shadrin, D.; Menshchikov, A.; Somov, A.; Bornemann, G.; Hauslage, J.; Fedorov, M. Enabling Precision Agriculture Through Embedded Sensing With Artificial Intelligence. *IEEE Trans. Instrum. Meas.* **2020**, *69*, 4103–4113. [[CrossRef](#)]
8. Patrício, D.I.; Rieder, R. Computer vision and artificial intelligence in precision agriculture for grain crops: A systematic review. *Comput. Electron. Agric.* **2018**, *153*, 69–81. [[CrossRef](#)]
9. Trotter, M.; Lamb, D. GPS tracking for monitoring animal, plant and soil interactions in livestock systems. In Proceedings of the 9th International Conference on Precision Agriculture, Denver, CO, USA, 20–23 July 2008; International Society of Precision Agriculture (ISPA): Monticello IL, USA, 2008.
10. Bailey, D.W.; Trotter, M.G.; Knight, C.W.; Thomas, M.G. Use of GPS tracking collars and accelerometers for rangeland livestock production research1. *Transl. Anim. Sci.* **2018**, *2*, 81–88. [[CrossRef](#)] [[PubMed](#)]
11. Kumar, H.A.; Rakshith, J.; Shetty, R.; Roy, S.; Sitaram, D. Comparison Of IoT Architectures Using A Smart City Benchmark. *Procedia Comput. Sci.* **2020**, *171*, 1507–1516. [[CrossRef](#)]
12. Cerbulescu, C.C.; Cerbulescu, C.M. Large data management in IOT applications. In Proceedings of the 2016 17th International Carpathian Control Conference (ICCC), High Tatras, Slovakia, 29 May–1 June 2016; pp. 111–115. [[CrossRef](#)]
13. Jung, M.G.; Youn, S.A.; Bae, J.; Choi, Y.L. A Study on Data Input and Output Performance Comparison of MongoDB and PostgreSQL in the Big Data Environment. In Proceedings of the 2015 8th International Conference on Database Theory and Application (DTA), Jeju, Republic of Korea, 25–28 November 2015; pp. 14–17. [[CrossRef](#)]
14. Wang, C.; Huang, X.; Qiao, J.; Jiang, T.; Rui, L.; Zhang, J.; Kang, R.; Feinauer, J.; McGrail, K.A.; Wang, P.; et al. Apache IoTDB: Time-series database for internet of things. *Proc. VLDB Endow.* **2020**, *13*, 2901–2904. [[CrossRef](#)]
15. InfluxDB. Available online: <https://www.influxdata.com/products/influxdb/> (accessed on 16 October 2022).
16. Perwej, D.Y.; Haq, K.; Parwej, D.F.; Mumdouh, M.; Hassan, M. The Internet of Things (IoT) and its Application Domains. *Int. J. Comput. Appl.* **2019**, *182*, 36–49. [[CrossRef](#)]
17. Kerry, R.; Escolà, A. (Eds.) *Sensing Approaches for Precision Agriculture*; Progress in Precision Agriculture; Springer International Publishing: Cham, Switzerland, 2021. [[CrossRef](#)]
18. Munir, M.S.; Bajwa, I.S.; Ashraf, A.; Anwar, W.; Rashid, R. Intelligent and Smart Irrigation System Using Edge Computing and IoT. *Complexity* **2021**, *2021*, e6691571. [[CrossRef](#)]
19. Monteiro, A.; Santos, S.; Gonçalves, P. Precision Agriculture for Crop and Livestock Farming—Brief Review. *Animals* **2021**, *11*, 2345. [[CrossRef](#)]
20. Andonovic, I.; Michie, C.; Cousin, P.; Janati, A.; Pham, C.; Diop, M. Precision Livestock Farming Technologies. In Proceedings of the 2018 Global Internet of Things Summit (GIoTS), Bilbao, Spain, 4–7 June 2018; pp. 1–6. [[CrossRef](#)]
21. Mavridou, E.; Vrochidou, E.; Papakostas, G.A.; Pachidis, T.; Kaburlasos, V.G. Machine Vision Systems in Precision Agriculture for Crop Farming. *J. Imaging* **2019**, *5*, 89. [[CrossRef](#)]
22. Mogili, U.R.; Deepak, B.B.V.L. Review on Application of Drone Systems in Precision Agriculture. *Procedia Comput. Sci.* **2018**, *133*, 502–509. [[CrossRef](#)]
23. Leroux, C.; Jones, H.; Pichon, L.; Guillaume, S.; Lamour, J.; Taylor, J.; Naud, O.; Crestey, T.; Lablee, J.L.; Tisseyre, B. GeoFIS: An Open Source, Decision-Support Tool for Precision Agriculture Data. *Agriculture* **2018**, *8*, 73. [[CrossRef](#)]
24. Ren, Y.; Huang, D.; Wang, W.; Yu, X. BSMD:A blockchain-based secure storage mechanism for big spatio-temporal data. *Future Gener. Comput. Syst.* **2023**, *138*, 328–338. [[CrossRef](#)]
25. Mylonas, P.; Voutos, Y.; Sofou, A. A Collaborative Pilot Platform for Data Annotation and Enrichment in Viticulture. *Information* **2019**, *10*, 149. [[CrossRef](#)]
26. Wisnubhadra, I.; Baharin, S.S.K.; Herman, N.S. Open Spatiotemporal Data Warehouse for Agriculture Production Analytics. *Int. J. Intell. Eng. Syst.* **2020**, *13*, 419–431. [[CrossRef](#)]
27. Isomura, A.; Shigematsu, N.; Ueno, I.; Oki, N.; Arakawa, Y. Real-time Spatiotemporal Data-management Technology (Axispot™). *NTT Tech. Rev.* **2022**, *20*, 54–60. [[CrossRef](#)]
28. Wang, D.; Zou, L.; Zhao, D. gst-store: Querying Large Spatiotemporal RDF Graphs. *Data Inf. Manag.* **2017**, *1*, 84–103. [[CrossRef](#)]
29. Li, R.; He, H.; Wang, R.; Huang, Y.; Liu, J.; Ruan, S.; He, T.; Bao, J.; Zheng, Y. JUST: JD Urban Spatio-Temporal Data Engine. In Proceedings of the 2020 IEEE 36th International Conference on Data Engineering (ICDE), Dallas, TX, USA, 20–24 April 2020; pp. 1558–1569. [[CrossRef](#)]

30. Deeken, H.; Wiemann, T.; Lingemann, K.; Hertzberg, J. SEMAP—A semantic environment mapping framework. In Proceedings of the 2015 European Conference on Mobile Robots (ECMR), Lincoln, UK, 2–4 September 2015; pp. 1–6. [[CrossRef](#)]
31. Deeken, H.; Wiemann, T.; Hertzberg, J. A Spatio-Semantic Model for Agricultural Environments and Machines. In *Lecture Notes in Computer Science, Proceedings of the Recent Trends and Future Technology in Applied Intelligence, Montreal, QC, Canada, 25–28 June 2018*; Mouhoub, M., Sadaoui, S., Ait Mohamed, O., Ali, M., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 589–600. [[CrossRef](#)]
32. Deeken, H.; Wiemann, T.; Hertzberg, J. A spatio-semantic approach to reasoning about agricultural processes. *Appl. Intell.* **2019**, *49*, 3821–3833. [[CrossRef](#)]
33. Palma, R.; Roussaki, I.; Döhmen, T.; Atkinson, R.; Brahma, S.; Lange, C.; Routis, G.; Plociennik, M.; Mueller, S. Agricultural Information Model. In *Information and Communication Technologies for Agriculture—Theme III: Decision*; Bochtis, D.D., Sørensen, C.G., Fountas, S., Moysiadis, V., Pardalos, P.M., Eds.; Springer Optimization and Its Applications; Springer International Publishing: Cham, Switzerland, 2022; pp. 3–36. [[CrossRef](#)]
34. Janowicz, K.; Haller, A.; Cox, S.J.D.; Le Phuoc, D.; Lefrançois, M. SOSA: A lightweight ontology for sensors, observations, samples, and actuators. *J. Web Semant.* **2019**, *56*, 1–10. [[CrossRef](#)]
35. Compton, M.; Barnaghi, P.; Bermudez, L.; García-Castro, R.; Corcho, O.; Cox, S.; Graybeal, J.; Hauswirth, M.; Henson, C.; Herzog, A.; et al. The SSN ontology of the W3C semantic sensor network incubator group. *J. Web Semant.* **2012**, *17*, 25–32. [[CrossRef](#)]
36. Taylor, K.; Haller, A.; Lefrançois, M.; Cox, S.D.; Janowicz, K.; García-Castro, R.; Le Phuoc, D.; Lieberman, J.; Atkinson, R.A.; Stadler, C. The Semantic Sensor Network Ontology, Revamped. In Proceedings of the 18th International Semantic Web Conference, Auckland, New Zealand, 26–30 October 2019.
37. Nasar, M.; Abu Kausar, M. Suitability Of Influxdb Database For IoT Applications. *Int. J. Innov. Technol. Explor. Eng.* **2019**, *8*, 1850–1857. [[CrossRef](#)]
38. Petre, I.; Boncea, R.; Alin, Z.; Radulescu, C. A Time-Series Database Analysis Based on a Multi-attribute Maturity Model. *Stud. Informatics Control* **2019**, *28*, 177–188. [[CrossRef](#)]
39. Kayad, A.; Sozzi, M.; Paraforos, D.S.; Rodrigues, F.A.; Cohen, Y.; Fountas, S.; Francisco, M.J.; Pezzuolo, A.; Grigolato, S.; Marinello, F. How many gigabytes per hectare are available in the digital agriculture era? A digitization footprint estimation. *Comput. Electron. Agric.* **2022**, *198*, 107080. [[CrossRef](#)]
40. Communication from the Commission to the European Parliament, the Council, the European Economic and Social Committee and the Committee of the Regions a European Strategy for Data. 2020. Available online: <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A52020DC0066> (accessed on 21 October 2022).
41. Parte, M.S.E.d.l.; Serrano, S.L.; Díaz, V.H.; Martínez-Ortega, J.F. *grys-upm/Spatio-Temporal-Semantic Data Model for Precision Agriculture*; Zenodo: Genève, Switzerland, 2022. [[CrossRef](#)]
42. Castillejo, P.; Johansen, G.; Cürüklü, B.; Bilbao-Arechabala, S.; Fresco, R.; Martínez-Rodríguez, B.; Pomante, L.; Rusu, C.; Martínez-Ortega, J.F.; Centofanti, C.; et al. Aggregate Farming in the Cloud: The AFarCloud ECSEL project. *Microprocess. Microsystems* **2020**, *78*, 103218. [[CrossRef](#)]
43. InfluxData. *Hardware Sizing Guidelines | InfluxDB OSS 1.8 Documentation*; InfluxData: San Francisco, CA, USA. Available online: [https://docs.influxdata.com/influxdb/v1.8/guides/hardware\\_sizing/](https://docs.influxdata.com/influxdb/v1.8/guides/hardware_sizing/) (accessed on 10 November 2022).
44. Betts, R. *InfluxDB 1.4 | InfluxQL Enhancements, Prometheus Read/Write & More*; InfluxData: San Francisco, CA, USA, 2017.
45. de la Parte, M.S.E.; Serrano, S.L.; Díaz, V.H.; Martínez-Ortega, J.F. *grys-upm/Data-Access-Manager\_Data-Query: Final Version of DAM&DQ Semantic Middleware*; Zenodo: Genève, Switzerland, 2022. [[CrossRef](#)]
46. Bastos, J.; Shepherd, P.M.; Castillejo, P.; Emeterio, M.S.; Díaz, V.H.; Rodriguez, J. Location-Based Data Auditing for Precision Farming IoT Networks. In Proceedings of the 2021 IEEE 26th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), Porto, Portugal, 25–27 October 2021; pp. 1–6, ISSN 2378-4873. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.