REVIEW ARTICLE

A Real-Time Semantic Annotation to the Sensor Stream Data for the Water Quality Monitoring

Besmir Sejdiu[1] · Florije Ismaili[1] · Lule Ahmedi[2]

Abstract

Currently, billions of interconnected IoT devices continuously generate a huge amount of various sensed data in the stream called sensor stream data. Their management and processing have become increasingly difficult and important. However, raw sensor stream data are useless unless properly annotated. Therefore, by adding semantic annotations with concept definitions from ontologies, it is possible the interpretation and understanding of sensor stream data. In this paper is presented a sensor stream data management model for real-time IoT monitoring systems, which supports real-time integration and interpretation of semantic annotations into the heterogeneous sensor stream data in the format of Sensor Observation Service (SOS) with context in the IoT. To validate the proposed model, an IoT system for real-time water quality monitoring is developed which applies international regulatory of water quality such as Water Framework Directive (WFD) and United Nations Economic Commission for Europe (UNECE).

Introduction

Nowadays, smart water networks, smart homes, smart cities, smart health, smart grid, intelligent transportation, are infrastructure systems that connect our world more than we ever thought possible. The common vision of such systems is usually associated with one single concept, the *Internet of Things* (IoT) [1]. The idea of IoT was developed in parallel to wireless sensor networks (WSNs). Therefore, the main components that enable IoT are WSNs [2].

A *wireless sensor networks* is a wireless network consisting of spatially distributed autonomous devices using sensors to monitor physical or environmental conditions. A WSN is usually deployed with *static sensor nodes* to perform monitoring in the region of interest, but WSN can be deployed together with *mobile sensor nodes* to perform monitoring in different ad hoc selected locations of interest. WSNs may be homogeneous or heterogeneous. *Homogeneous sensors* send only one type of information (e.g., the water temperature), while *heterogeneous sensors* send more than one type of information (e.g., temperature and dissolved oxygen). All these sensors send observational data referred to as sensor stream data to a remote server. Therefore, sensory data come from multiple sensors of different modalities in distributed locations [3].

Furthermore, sensor stream data are enabled to the web through the *Sensor Web* (SW). SW by incorporating technologies of the Semantic Web creates the *Semantic Sensor Web* (SSW) [4]. As a result, adding semantic annotations to sensor stream data with concept definitions from domain knowledge (e.g., ontologies) allows for the interpretation and understanding of sensor data and metadata. The real-time

✉ Besmir Sejdiu
  bs26916@seeu.edu.mk

  Florije Ismaili
  f.ismaili@seeu.edu.mk

  Lule Ahmedi
  lule.ahmedi@uni-pr.edu

1   Contemporary Sciences and Technologies, South East European University, Tetovo, Macedonia

2   Faculty of Electrical and Computer Engineering, University of Prishtina, Prishtinë, Kosova

integration of sensor data as dynamic data with semantics is defined as *real-time semantic annotation*, while sensor data that are stored in repository (data store) as static data, and then integrated with semantics is defined as *non-real-time semantic annotation* [5]. Organizations such as Open Geospatial Consortium (OGC) and World Wide Web Consortium (W3C) have proposed industry standards such as Sensor Web Enablement (SWE), which are aimed at providing unified standards [6].

The paper is organized as follows: "Literature Review" provides a discussion on related work for semantic annotations to the sensor stream data. "Background" is an overview of the data stream models, annotated sensor data streams, and used technologies. "Proposed Model" represents a proposed management model of integrated semantic annotations to the sensor stream data for the IoT. Prototype implementation of the proposed model is presented in "Implementation of Proposed Model", which includes (A) "Received Sensor Stream Data Format", (B) "Integration and Interpretation of Semantic Annotations into the Sensor Stream Data", and (C) "System Outputs". Finally, "Conclusion and Future Work" concludes the paper and relevels some of the future perspectives of the proposed model.

Literature Review

A sensor stream data is a (potentially unbounded) sequence of tuples. Each tuple consists of a set of attributes, similar to a row in database table. The rate of received sensor stream data is very important for their processing since as bigger the rate it is harder to manage them and process them [7]. For some sensors, the data transmission rate is not very high, such as water quality monitoring sensors, air quality monitoring, and agro-sensors. Therefore, the management and processing of their data can be distinguished by those who have high data transmission rates because they can be stored or even archived. Raw sensor stream data is useless unless properly annotated. Therefore, determining how to integrate semantic annotations to the sensor stream data and make them machine-interpretable is an important issue.

Recently, some researchers have already shown up with several investigations related to semantic enrichment of sensor stream data.

Sejdiu et al. [8] presented a management model, which enables to manage the real-time integration of semantic annotations into heterogeneous sensor stream data with context in the IoT and using OGC SOS standards. Lin et al. [9] proposed a semantic annotation method to annotate IoT sensor data through semantics. They used K-means clustering method for knowledge discovery. The proposed mechanism is semiautomatic, because after clustering the data, there are just clusters without knowledge. The knowledge of the clusters is defined by people. Duy et al. [10] presented a light-weight data model proposal for presenting environment data using semantic middleware based on Semantic Sensor Network (SSN) Ontology for environment events management and query. This model can combine data measurement and other sensor metadata such as geo location, station, and sensor information. Xiaomin et al. [11] proposed an approach for ontology modeling which is used in evaluating the river water quality and its relevant processing knowledge. The presented model consists of the data acquisition layer, diagnosis layer and decision support layer. Rasyid et al. [12] implemented a Semantic Sensor Web (SSW) with existing gas sensor to perform readings from the environment and store into a database. In addition, they deployed API to allow users to monitor and access data gas sensors. Sejdiu el al. [3] presented a system named IoTSAS, which enables real-time integration and interpretation of semantics into observed WSN data in two IoT domains, such as air quality monitoring and weather alert monitoring.

In addition, in this study, are identified different solutions, [13–15], to semantically annotate sensor stream data. Those proposed solutions used *non-real-time semantic annotation* because the sensor stream data are stored in data store or ontology as static data and then integrated with semantics. However, how to advance techniques and models for integration and interpretation of the semantic annotations in real time in the sensor stream data is still an open issue that should be addressed.

The main contributions of our approach are presented as follows:

- A data stream management model of WSNs for real-time IoT monitoring systems has been developed, which support real-time integration and interpretation of data from heterogeneous sensor with semantic annotations.
- To validate the proposed model, an IoT system for real-time water quality monitoring is built, which enables real-time integration and interpretation of semantic annotations to the observational data on water quality coming from wireless sensors.
- By incorporating OGC SOS standards, it hides the heterogeneity of data sensors from arbitrary sources and provides a real-time service interface for published enriched sensor stream data with semantic annotations to display in IoT real-time monitoring systems.
- The developed system applies international regulatory of water quality such as Water Framework Directive (WFD) and United Nations Economic Commission for Europe (UNECE).

Background

The Data Stream Models

Depending on characteristics, source of data transmission and saving of stream data, those can be modeled in various ways: *Real-time data stream, Stream items,* and *Window models.* In the following, each of them is described:

Real-time data stream—data stream in real time is a sequence of data which arrive in order and/or in pre-processed ways, and therefore, creating a possible list of the models [16]: *Unordered cash register* (data from various domains that does not arrive in a specific order or with pre-processing), *Ordered cash register* (individual data from various domains are not pre-processed, but they attain a well-known order in some way), *Unordered aggregate* (individual data from the same domain that has been pre-processed and only one data from the domain comes without being ordered), or *Ordered aggregate* (pre-processed individual data from the same domain, with only one data arriving in a well-known order).

Stream items—since data are received in streams, those can be modeled as a sequence in a list of elements, which can take the form of relational rows or object instances [17].

Window models—in many cases, only a fragment of the streaming data is of interest at any time, and this way encouraging window models, which can be classified according to the three criteria: *Fixed sliding window* (which includes only the most actual data or by showing only the newest data according to the timeframe), *Landmark window* (a time reference point is established and from that time data are included. This criteria is not used so much due to the increase of data quantity within the window), and *Adaptive window* (dynamically the window is changed based on the input data and specified values from users).

Annotated Sensor Data Streams

WSNs consist of small-scale devices that enable observe various physical phenomenon, which provide sensor data in raw format. Typically, classical IoT applications cannot interpret the sensor data and understand its context. This makes it nearly impossible to get the high-level information of the events and infer additional knowledge to gain situational awareness [18].

To provide meaning (semantics) to raw data, annotated sensor data stream is required. The annotated sensor data become more meaningful and understandable, enabling end-users to get high-level details about the real-world situations instead of raw sensor data. It is known as Semantic Sensor Web (SSW). These annotations provide more meaningful descriptions and enhanced access to sensor data than SWE

alone, and they act as a linking mechanism to bridge the gap between the primarily syntactic XML-based metadata standards of the Sensor Web Enablement (SWE) and the RDF/OWL-based metadata standards of the Semantic Web [19].

To encode semantic annotations and data gathered by sensors, in this paper is used SWE, respectively, version 2.0 of the Sensor Observation Service (SOS) standard relies on the OGC Observation and Measurement (O&M).

Technologies

The proposed management model of real-time integrated semantic annotations to the sensor stream data for the IoT utilizes:

- *Spark Streaming*[1]—is an extension of the Apache Spark which enables to build scalable fault-tolerant IoT applications. Major studies conclude that Spark Streaming works best with high throughput when the incoming volume is huge [20]. Data can be ingested from many sources such as Apache Kafka, TCP sockets, Flume, and Twitter, and processed data using complex algorithms expressed with high-level functions such as map, reduce, join and windows. Finally, processed data can be pushed out to file systems, databases, and live dashboards.
- *Apache Kafka*[2]—is a distributed streaming platform which has capabilities to publish and subscribe to streams of records, similar to a message queue or enterprise messaging system.
- *Apache Cassandra database*[3]—is a distributed store for structured data that scale-out on cheap. It is designed to handle large amounts of data.

Proposed Model

In Fig. 1, an overview of the model architecture for processing and managing sensor stream data for IoT real-time monitoring systems, such as water quality monitoring and air quality monitoring, is presented.

The WSNs are deployed in different locations. They produce a continuous stream of data, and transmit to Apache Kafka in various formats (e.g., binary, JSON, and XML). Kafka is utilized to transform them in a specific format that will be processed by Spark Streaming in real-time and parallel. The Spark Streaming enables a real-time integration of semantics into sensor stream data using association rules, mining data streams, WSNs metadata, and archival data

---

[1] http://spark.apache.org

[2] https://kafka.apache.org

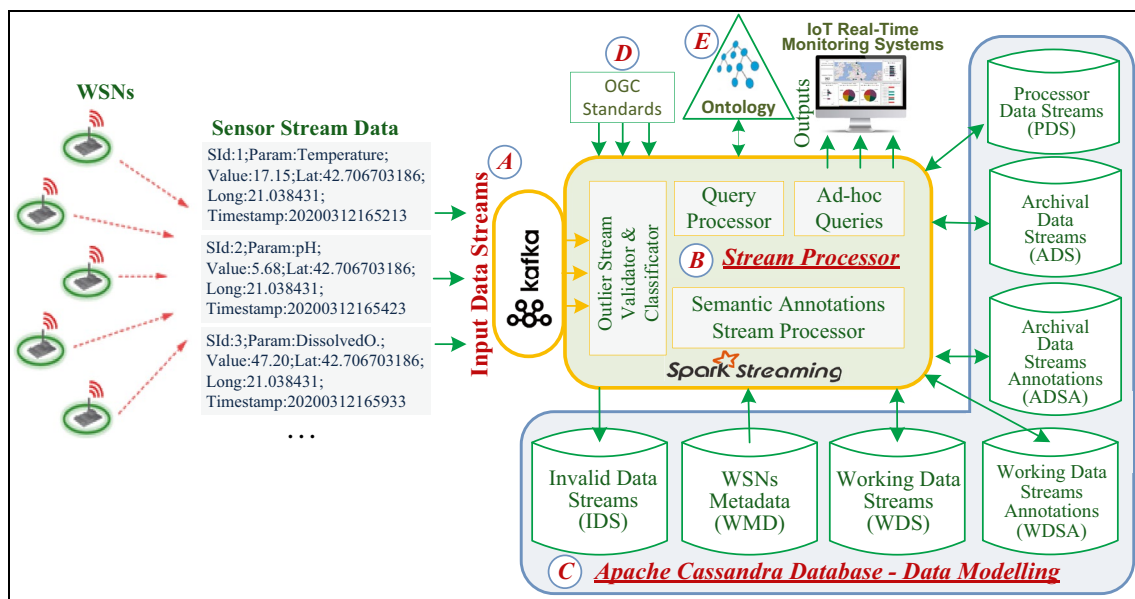[3] http://cassandra.apache.org

Fig. 1 An overview of the model architecture

streams, with concept definitions from ontologies or other semantic sources, which provides the understanding and more meaningful descriptions to enable application areas of IoT to become much more intelligent.

The enriched sensor stream data with semantic annotations results are stored in the Cassandra database, and will be displayed in IoT real-time monitoring systems in format of SOS O&M standard using stakes, such as XLink (without including XPath). A fragment of an output example is presented in Fig. 2.

As shown in Fig. 1, the proposed data stream management model supports real-time data integration of heterogeneous sensors with semantic annotations, continuous queries on streaming data, outlier validation of streaming data, ad hoc queries, and archive stream data with their semantic annotations for applications that need to answer queries from the archival store (persistent data stored). The proposed model consists of the main components: (A) *Input Data Stream*, (B) *Stream Processor*, (C) *Data Modeling*, (D) *OGC standards*, and (E) *Ontology*.

Each component of the model is described in detail as follows:

A. *Input Data Stream*—is implemented in Apache Kafka and accepts in real-time input streaming data sent by the WSNs. Unordered streams come without any kind of pre-processing—as *unordered cash register*. Each stream can provide elements at its own schedule, they do not need to have the same data rates or data types, and the time between elements of one stream need not be uniform.

B. Stream Processor is developed in Spark Streaming and contains Outlier Stream Validator and Classificator, Query Process, Ad hoc Queries, and Semantic Annotations Stream Process:

- *Outlier Stream Validator* and *Classificator*—is part of the stream processor in charge of real-time validating sensor streaming data, which marks stream data with one of the following statuses, '*valid*' or '*outlier*'. Validated data continues to processing forward, while invalid data are stored in *Invalid Data Streams (IDS)*. A data stream object is considered an outlier if it does not conform to expected behavior, which corresponds to either noise or anomaly. For example, the observed value of the pH sensor is '-2' or '*NULL*', in this case this sensor data will be classified as outlier because the range value of pH phenomena is 0 to 14. Outliers can arise due to different reasons such as mechanical faults, other changes in the system, fraudulent behavior, instrument error, human error or natural deviation [21]. Therefore, the *Outlier Stream Validator* and *Classificator* provides data quality for IoT real-time monitoring systems. In addition, this component is advanced to notify system users, when the sensors are sending the outlier data (invalid data).
- *Query Processor*—supports continuous queries for streaming data, and are continuously as data streams continue to arrive. The answer of the continuous query is produced over time, always reflecting the stream data seen so far. The answers of our *Query Processor* can include semantic annotated data in the result.
- Ad hoc *Queries*—queries executed ad hoc from users. A question asked once about the current state of a stream or streams. Users can also specify ad hoc queries that

```xml
<sos:Observation ...="">
<observationData>
  <om:OM_Observation gml:id="o1">
    <om:type xlink:href="http://www.opengis.net/def/observationType/OGC-
OM/2.0/OM_Measurement"/>
    <om:phenomenonTime>
      <gml:TimeInstant gml:id="phenomenonTime_1">
        <gml:timePosition>2020-03-25T20:00:00+09:00</gml:timePosition>
      </gml:TimeInstant>
    </om:phenomenonTime>
    <om:resultTime xlink:href="#phenomenonTime_1"/>
    <om:procedure xlink:href="http://myserver/ontologies/ont-core.owl#SensorId"/>
    <om:observedProperty xlink:href="http://myserver/ontologies/ont-
core.owl#parameterName"/>
    <om:featureOfInterest xlink:href="http://myserver/ontologies/ont-core.owl#Location"/>
    <om:result xsi:type="gml:SemMeasureType" uom="parameterUnit">
      <value>parameterValue</value>
      <sem-annotations>
        <annotation xlink:href="http://myserver/ontologies/ont-
core.owl#semanticAnnotation1"/>
        <annotation embedded:semanticAnnotation2 ="value"/>
        <annotation xlink:href="http://myserver/ontologies/ont-core.owl#
semanticAnnotation3"/>
      </sem-annotations>
      <sem-interpretations>
         Textual interpretatinos..
      </sem-interpretations>
    </om:result>
  </om:OM_Observation>
</observationData>
. . .
</sos:Observation>
```

Fig. 2 A general case of integrated semantic annotations to the sensor stream data

integrate streaming data and persistent data stored on *Working Data Streams*, *WSNs Metadata* or *Archival Data Streams*. In addition, the result of Ad hoc *Queries* can include semantic annotated data in the result.

- *Semantic Annotation Stream Processor*—is part of the Spark Streaming processor which enable a real-time integration of semantic annotations into heterogeneous sensor stream data with context in the Internet of Things. This component can use sensor metadata, archival data streams, mining data streams, association rules (with concept from ontologies), or other sematic sources for adding semantic annotations to the sensor stream data. The semantic annotated of sensor stream data are stored in *Working Data Stream Annotations*.

C. Data Modeling is developed in Apache Cassandra database and contains: Processor Data Streams, Working Data Streams, Working Data Stream Annotations, Archival Data Streams, Archival Data Stream Annotations, Invalid Data Streams, and WSNs Metadata:

- Processor Data Streams (PDS)—contains a summary of streaming data for Stream Processor, which can be used

for answering queries. For each deployed sensor, only a row is saved which includes information described in Table 1.
- ●Working Data Streams (WDS)—contains streaming data for operation of Stream Processor, which are configurable according the quantity and can be used for answering queries. Therefore, it is a Fixed Sliding Window that contains the last sensor streaming data (e.g., 15 last measured values—its configurable number). For each measured values, information are stored as described in Table 2.
- Working Data Stream Annotations (WDSA)—store semantic annotations of sensor streaming data. The Semantic Annotation Stream Processor component is used to tag in real-time integration the sensor data stream with semantic annotations. One measurement that is stored in Working Data Streams can obtain several sematic annotations, which include information described in Table 3.
- Archival Data Streams (ADS)—archives data streams for generating reports and different statistics. The structure of data modeling of ADS is same as WDS.

Table 1 Processor data streams (PDS) model

Field	Description
Sensor Id	An identifier that uniquely identifies a sensor
Sensor parameter	Name of the parameter or phenomena that sensor measure (e.g., temperature, humidity, and pH)
Sensor current value	Current measured value from the sensor
Sensor total rows	The total number of measurements by the sensor since its deployment
Sensor max value	The maximum value measured by the sensor since its deployment
Sensor min value	The minimum value measured by the sensor since its deployment
Sensor sum value	The sum of values measured by the sensor since its deployment
Sensor avg value	The average value measured by the sensor since its deployment, which is derived by finding quotient between sum of values measured by the sensor and the total number of measured by sensor (SensorSumValue/SensorTotalRows)
Sensor window max	The maximum value of sliding window. Sliding window contains last n values, where n is a configurable number (e.g., 15 last measured values) sent by the sensor
Sensor window min	The minimum value of sliding window
Sensor window avg	The average value of sliding window
Sensor current timestamp	Current measured timestamp sent by the sensor
Sensor current latitude & longitude	Current latitude and longitude (current geographical position from where sensor has sent data)

Table 2 Working data streams (WDS) model

Field	Description
Observation Id	An identifier that uniquely identify the observation in the WDS
Sensor Id	An identifier that uniquely identifies a sensor
Sensor parameter	Name of the parameter or phenomena (e.g., temperature) measured by the sensor
Sensed value	Measured value that is sent by the WSN
Timestamp	Time when the sensed value has been generated by WSN
Latitude & longitude	Geo location, geographical position from where the sensor has sent data. It is especially useful when a sensor is attached to a moving object such as a car and airplane or in case of mobile sensor to perform monitoring in different ad hoc selected locations of interest, while in case of static sensor that perform monitoring in the region of interest, the geo location can be NULL because the location of these senor type can be saved as metadata in WSNs Metadata;
Observation Id	A code which identify a single sensor node measurement, example in cases where a sensor node measures three parameters, e.g., temperature, turbidity and conductivity at the same time (as a single measurement), then all of three measurements will get the same transaction code, otherwise the transaction code will be NULL
Entry timestamp	date and time when streaming data has been arrive to Stream Processor

Table 3 Working data stream annotations (WDSA) model

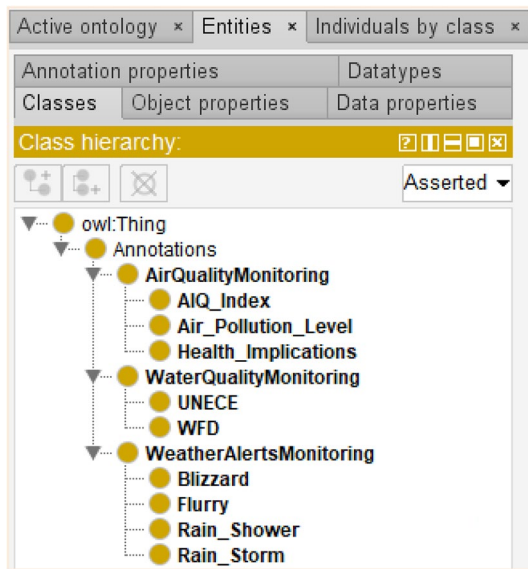| Field	Description
Annotation Id	An identifier that uniquely identifies a sematic annotation
Observation Id	References to working data streams observation Id
Annotated date	Date and time when sensor streaming data has been tagged with sematic annotations
Annotated type	Represents the type of annotation, 'Embedded' (only a single value-scalar of semantic annotation) or 'External' (an external resource linked by 'XLink' that point to our ontology 'ont-core.owl')
Annotated value	Stores the semantic annotated value Example value of 'Embedded' annotation type for water status can be 'Good', 'Moderate ', 'Poor', 'Bad' or 'High': < annotation embedded:WaterStatus = "Bad"/> Example value of 'External' annotation type can be: < annotation xlink:href = "http: //myserver/ontologies/ont-core.owl#WaterStatus_ClassV"/> (for more details see Fig. 2)

Fig. 3 'ont-core.owl' ontology

- Archival Data Stream Annotations (ADSA)—archives semantic annotations of sensor stream data for generating reports and different statistics. The structure of data modeling of ADSA is same as WDSA.
- WSNs Metadata (WMD)—the data describing wireless sensor networks itself, its devices and the corresponding site allocation data. This data is named as static data that describes the wireless sensor networks in the field, its configuration which might involve node types, such as sensing nodes, gateway nodes, central monitoring node, and description of sensors as devices (sensor name, serial number, manufacturer, and type), as well as data about the deployment sites, such as sensors' location, example for water system monitoring, the river basins, and municipalities the rivers belong to.
- Invalid Data Streams (IDS)—archives invalid sensor stream data that are classified as outlier by Outlier Stream Validator and Classificator. The data stored in IDS is optional and it is depending on the system requirements.

D. *OGC Standards*—as mentioned above, the enriched sensor stream data with the semantic annotations results will be published to IoT real-time applications in format of OGC standards, respectively, version 2.0 of the SOS O&M standard.

E. *Ontology*—an ontology named '*ont-core.owl*' is created, shown in Fig. 3. This ontology contains semantic annotations for each IoT domains, example for Water Quality Monitoring domain, Air Quality Monitoring domain, and Weather Alerts Monitoring domain. In our case are used Water Quality Monitoring developed semantic annotations.

More details are presented in "Integration and Interpretation of Semantic Annotations into the Sensor Stream Data".

Details about the working cycle of this model are as follows: streaming data are sent by the wireless sensors networks in *Input Data Streams* (Apache Kafka). Sensor stream data is an array of different types containing sensor id (sid), name of the parameter, measured value of the sensor, geographical position (latitude and longitude) and timestamp, as seen as follows:

'SId: 1; Parameter: Temperature; Value: 17.15; Lat: 42.706703186; Long: 21.038431; Timestamp: 20200312165213'

The stream data elements are then validated using the *Outlier Stream Validator*, in which every sensor stream data takes the status of validity (*true*—data is valid or *false*—data is outlier). When data takes the validation status '*true*', it will be transmitted for further processing in *Semantic Annotated Stream Processor* which makes real-time integration of semantic annotations into those stream data. Then the enriched sensor stream data with the semantic annotations results will be stored in *WDS* and *WDSA* and will be transform in SOS O&M format to displayed in IoT real-time monitoring systems.

It is worth mentioning, when a new value is measured by the sensor, it arrives in *WDS* (and their semantic annotations are stored in *WDSA*) then the oldest value is removed from there and goes to *ADS* (respectively, *ADSA*) for archiving. Therefore, in *ADS* and *ADSA* are archived data which serve for generating reports and statistics for longer time frames.

Implementation of Proposed Model

To validate the proposed model in this study, we implemented a prototype illustrating a water quality monitoring use case, named as Water Quality Monitoring System (WQMS). This system enables the measurement of water quality in real time by applying the latest technology trends, such as wireless sensor networks, which provide continuous monitoring, and consist of nodes referred to as motes that are sensitive to the environment where they are deployed.

Received Sensor Stream Data Format

WQMS enables water parameters monitoring such as temperature, potential of hydrogen (pH), and dissolved oxygen (DO). The type, rank, and unit of these parameters are shown in Table 4. The data produced by the sensor are a real number, e.g., 16.5 °C for temperature measurement. The sensors are configured in such a way that each node will send data every 10 min in XML format. Therefore, the rate of the data stream is not high, therefore, the storing of these

Table 4 Specification of water parameters

Parameter name	Parameter type	Range value	Unit
Temperature	Thermal conditions	− 1 to + 50	°C
pH	Acidification status	0 to 14	pH
Dissolved oxygen	Oxygenation conditions	0 to 300	%

data by applying the proposed model, nowadays, will not be a problem for the new technologies.

The WQMS system architecture is presented in Fig. 4. It mainly consists of the static wireless sensing nodes (which are located in Plemetin (42.70670318, 21.03843116)—Fig. 5.a), mobile wireless sensing nodes (which can observe data from different locations—Fig. 5.b), gateway node and
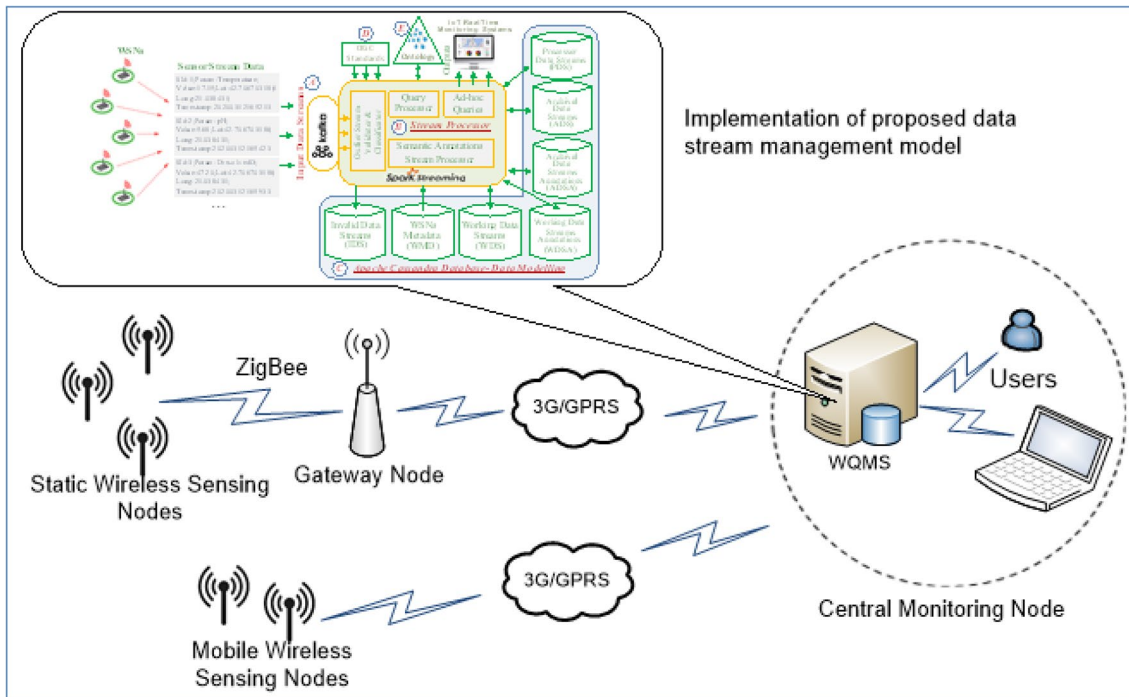

Fig. 4 System architecture: implementation of proposed data stream management model in WQMS

Fig. 5 System implementation in Plemetin: **a** static sensor nodes (left), and **b** mobile sensor nodes (right)
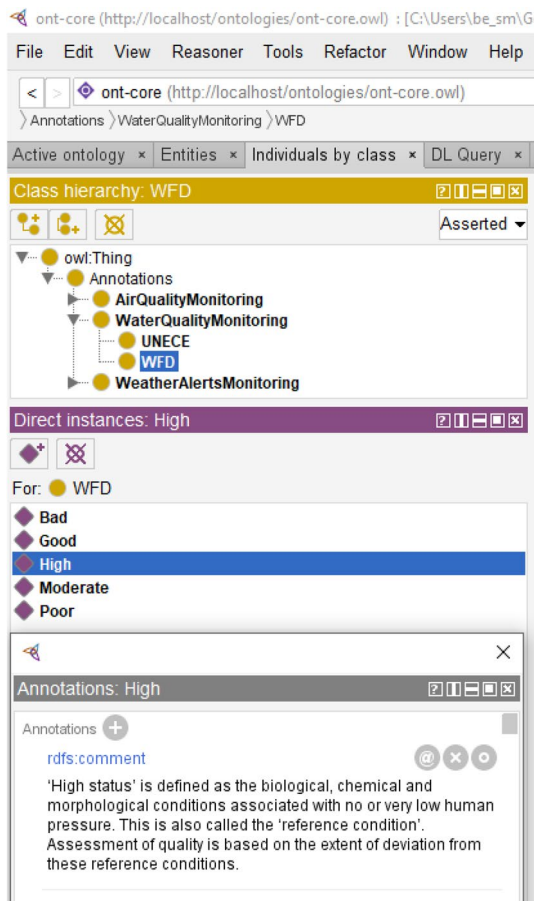
Fig. 6 'ont-core.owl' ontology for Water Quality Monitoring

the monitoring node. Static wireless sensing nodes are located in a given position and through gateway node, continuously transmit sensed data to the central monitoring node, while mobile wireless sensing nodes can move from one position to the other to measure water parameters.

The sensor data in the central monitoring node are transmitted via 3G/GPRS using web services. The WQMS software is installed in the central monitoring node, in which the proposed management model is implemented.

Integration and Interpretation of Semantic Annotations into the Sensor Stream Data

As mentioned above, an ontology named 'ont-core.owl' is created. As shown in Fig. 6, here are developed semantic annotations for international regulatory of water quality, such as:

- Water Framework Directive (WFD) with semantic annotations of water status:
- Good,
- Moderate,
- Poor,
- Bad, and
- High.
- United Nations Economic Commission for Europe (UNECE) with semantic annotations of water status:
- Class I,
- Class II,
- Class III,

Fig. 7 Water status annotation
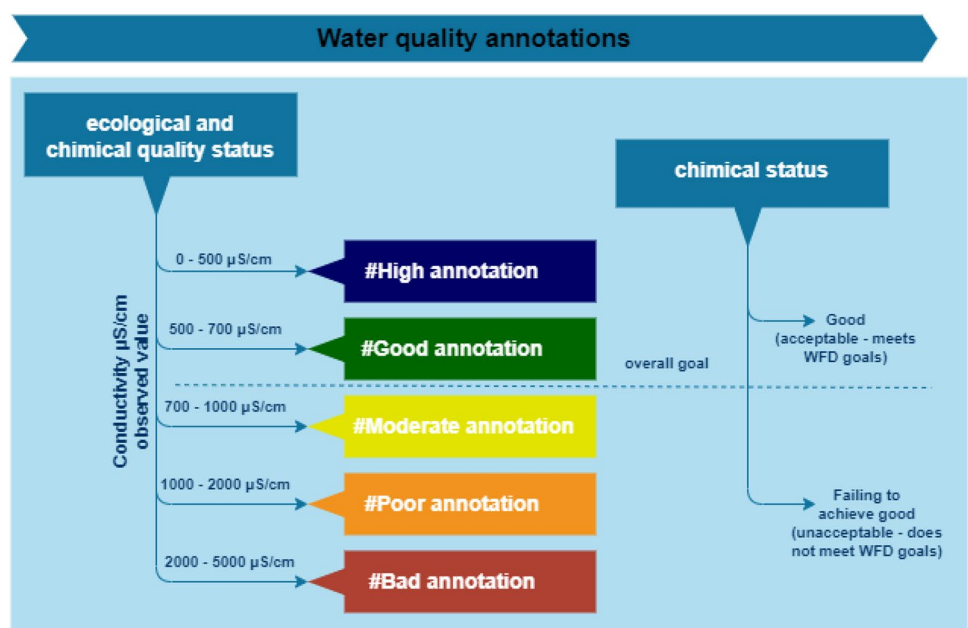
```
1    <sos:Observation ...="">
2        <observationData>
3            <om:OM_Observation gml:id="o45125">
4                <om:type xlink:href=
                 "http://www.opengis.net/def/observationType/OGC-OM/2.0/OM_Measurement"/>
5                <om:phenomenonTime>
6                    <gml:TimeInstant gml:id="phenomenonTime_1">
7                        <gml:timePosition>
8                            2020-04-03T16:40:00+09:00
9                        </gml:timePosition>
10                   </gml:TimeInstant>
11               </om:phenomenonTime>
12               <om:resultTime xlink:href="#phenomenonTime_1"/>
13               <om:procedure xlink:href="http://myserver/ontologies/ont-core.owl#Sensor3"/>
14               <om:observedProperty xlink:href=
                 "http://myserver/ontologies/ont-core.owl#DissolvedOxygen"/>
15               <om:featureOfInterest xlink:href="http://myserver/ontologies/ont-core.owl#Plemetin"/>
16               <om:result xsi:type="gml:SemMeasureType" uom="%">
17                   <value>47.20</value>
18                   <sem-annotations>
19                       <annotation embedded:WaterStatus ="High"/>
20                       <annotation xlink:href="http://myserver/ontologies/ont-core.owl#
                         WaterStatus_Indicates_High"/>
21                   </sem-annotations>
22                   <sem-interpretations>
23                       Now (@2020-04-03 16:40) in location 'Pelemetin (42.70670318, 21.03843116)'
                         is detected '47.20%' dissoleved oxygen, which indicates a 'High' water
                         status with no or very low human pressure and the water is not polluted at
                         all.
24                   </sem-interpretations>
25               </om:result>
26           </om:OM_Observation>
27       </observationData>
28       . . .
29   </sos:Observation>
```

Fig. 8 An example of integrated semantic annotations to the IoT water quality monitoring sensor stream data

- Class IV, and
- Class V.

An example of annotation for the Conductivity parameter (μS/cm) is given in Fig. 7. If the value observed by the conductivity sensor is in the range 0–500 μS/cm, the water status is categorized as high and the semantic annotation result is #High. If the value observed by the sensor is between 500–700 μS/cm, then the system creates the annotation #Good. Both of these types of stators are accepted in terms of water quality by WDF. For other annotations like: #Moderate (700–1000 μS/cm), #Poor (1000–2000 μS/cm), and #Bad (2000–5000 μS/cm), failing to achieve good (unacceptable—does not meet WDF goals).

The sensor stream data may arrive in different formats to Kafka server (XML format—in our case), which will transform them in a specific format that will be processed by Spark Streaming. After that, through the Spark Streaming, based on measuring values, the sensor data stream will semantically be annotated and converted in SOS O&M format. A fragment of an example of integrated semantic annotations to the SOS O&M format using stakes such as XLink and Embedded is presented in Fig. 8.

After the real-time integration of semantics into heterogeneous water sensor stream data, the real-time interpretation of the sensor stream data is done to give a better understanding and infer new knowledge from the water sensor stream data. The following interpretation pattern for water quality IoT domain is developed:

Now (@[#timestamp]) in location [#location(lat, long)] is detected '[#ParamValue] [#ParamUnit]' [#ParamName], '[#WaterStatus]' water status, which [#WaterStatus_Indicates].

As shown in Fig. 8, the interpretation of semantic annotations contains the data about timestamp, location (including latitude and longitude) of sensing node, the phenomenon with the measured value which has caused the water status, and meaning of the current water status:

"Now (@2020-04-03 16:40) in location 'Pelemetin (42.70670318, 21.03843116)' is detected '47.20%' dissolved oxygen, which indicates a 'High' water status with no or very low human pressure and the water is not polluted at all."

System Outputs (UI)

To display the enriched sensor stream data with semantic annotations and interpretation, a web-based IoT real-time application is developed. The interface of the software is
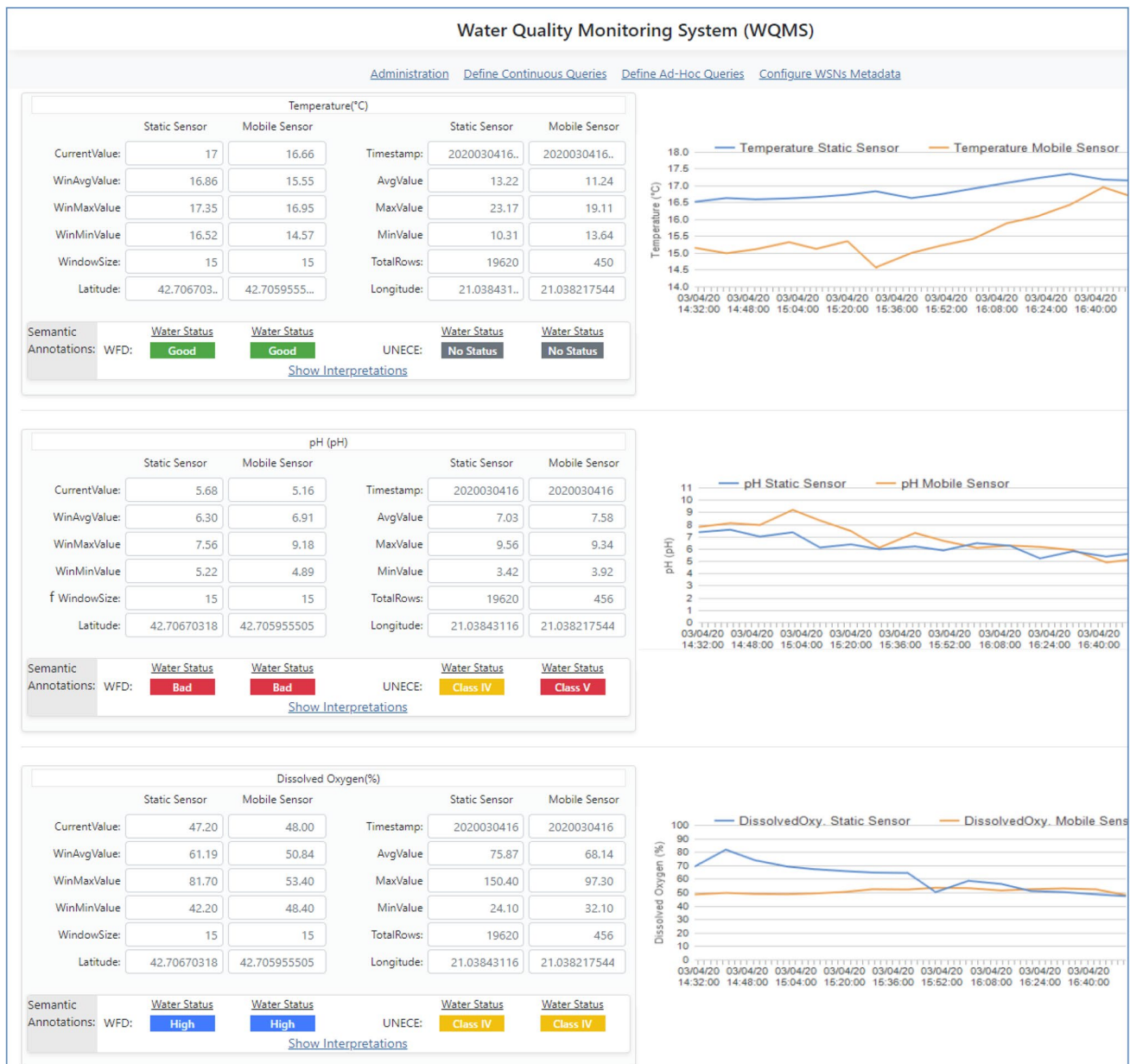
**Fig. 9** Outputs of proposed model in WQMS

shown in Figs. 9 and 10, which enables real-time water quality monitoring through static and mobile sensors. This software includes modules for system administration (to manage users, user groups, rights of users, and change password), definition of the continuous queries, execute of the ad hoc queries, and configuration of the WSNs metadata.

WQMS software executes continuous queries of the proposed model to display information. The displayed information in the textboxes for each parameter are obtained from *Processor Data Streams* through continuous queries. The information displayed in the charts is obtained from *Working Data Streams*, while the semantic annotations data that indicate the water status is obtained from *Working Data Stream Annotations*. As mentioned above, *Working Data Streams* represent a fixed sliding window which is configured in the WQMS by a certain pre-configured size, say 15. This means that the charts show the last 15 measurements for each sensor. As soon as the data reaches the system from the WSNs, the trigger for execution of queries continuously is activated.

The system outputs (UI) is developed in *ASP.NET Core MVC,.NET 5.0 C#*, a cross-platform, high-performance,

Fig. 10 Interpretation of semantic annotations
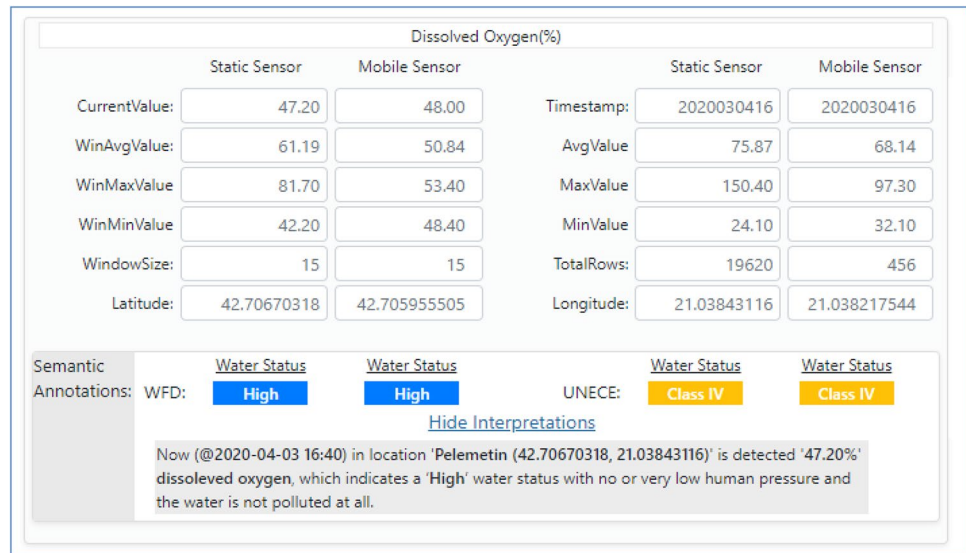
open source for building modern, cloud-based, and internet-connected applications. To read data from Apache Cassandra database, the '*DataStax C# for Apache Cassandra*' is used.

Conclusion and Future Work

In this research, the proposed management model of integrated semantic annotations into the sensor stream data for the Internet of Things is described.

This model supports managing stream data of homogeneous sensors, real-time integration of semantic annotations to the sensor stream data, continuous queries on streaming data, ad hoc queries, outlier validation of streaming data, archive stream data with semantic annotations for applications that need to answer queries form archival store (persistent data stored).

The proposed model support standards like Sensor Web Enablement (SWE), respectively, version 2.0 of the Sensor Observations Service (SOS) standard that relies on the Open Geospatial Consortium (OGC) and Observation & Measurement (O&M), to encode semantic annotations and data observed by sensors.

To validate the conceptual model proposed, a system for water quality monitoring, named Water Quality Monitoring System (WQMS) is implemented by applying advanced technologies of the Internet of Things such as WSNs. The implemented WQM System enables water quality monitoring in real time.

For the future work, among other several extensions under way of the proposed model, the following should be considered:

1.To advance annotation techniques, such as XPath, for integration and interpretation of the semantic annotations in real-time into heterogeneous sensor observation data and metadata with context in the Internet of Things.
2.To implement other international regulatory of water quality.
3.To evaluate the system performance and to compare the proposed model with other existing similar management schemes.

Declarations

Conflict of Interest On behalf of all the authors, the corresponding author states that there is no conflict of interest.

References

1. Yinbiao S, Lee K. Internet of things: wireless sensor networks. Switzerland: International Electrotechnical Commission (IEC); 2014.
2. Lazarescu MT. Wireless sensor networks for the internet of things: barriers and synergies Components and Services for IoT Platforms. Cham: Springer; 2017.
3. Sejdiu B, Ismaili F, Ahmedi L. IoTSAS: an integrated system for real-time semantic annotation and interpretation of IoT sensor stream data. Computers. 2021;2021(10):127. https://doi.org/10.3390/computers10100127.
4. Wang X, Wei H, Chen N, He X, Tian Z. An observational process ontology-based modeling approach for water quality monitoring. Water. 2020;12:715.

5. Sejdiu B, Ismaili F, Ahmedi L. Integration of semantics into sensor data for the IoT—a systematic literature review. Int J Semant Web Inf Syst. 2020;16(4):1.

6. Shi F, Li Q, Zhu T, Ning H. A survey of data semantization in internet of things. Sensors. 2018;18(1):313.

7. Rajaraman A, Leskovec J, Ullman JD. Mining of massive datasets. Cambridge: Cambridge University Press; 2014.

8. B Sejdiu, F Ismaili, L Ahmedi. A management model of integrated semantic annotations to the sensor stream data for the IoT. The 16th International Conference on Web Information Systems and Technologies (WEBIST 2020), November 03—05, 2020, Budapest, Hungary. 2020.

9. Lin SY, Li JB, Yu ChT. Dynamic data driven-based automatic clustering. Sens Mater. 2019;31(6):1789–801.

10. TKDuy, G Quirchmayr, A Tjoa, H Hanh. A semantic data model for the interpretion of environmental streaming data. Seventh International Conference on Information Science and Technology, Da Nang, Vietnam. 2017.

11. Xiaomin Zh, Jianjun Y, Xiaoci H, Shaoli Ch. An ontology-based knowledge modelling approach for river water quality monitoring and assessment. Proc Comput Sci. 2016;96:335–44.

12. MRasyid, A Sayfudin, A Basofi, A Sudarson. Development of semantic sensor web for monitoring environment conditions. International Seminar on Intelligent Technology and Its Application. 2016.

13. Vera D, Izquierdo Á, Vercher J, Gómez L. A ubiquitous sensor network platform for integrating smart devices into the semantic sensor web. Sensors. 2014;2014(14):10725–52.

14. Pradilla J, Palau C, Esteve M. SOSLITE: lightweight sensor observation service (SOS) for the internet of things (IOT). Barcelona: ITU Kaleidoscope: Trust in the Information Society; 2016.

15. Bytyçi E, Sejdiu B, Avdiu A, Ahmedi L. A semantic sensor web architecture in the internet of things. Semantic web science and real-world applications. Pennsylvania: Global IGI; 2019. p. 75–97.

16. Golab L, Tamer Ozsu M. Issues in data stream management. SIGMOD Record. 2003;32(2):5–14.

17. RMotwani, J Widom, A Arasu, B Babcock. Query processing, resource management, and approximation in a data stream management system. Technical Report. Stanford InfoLab. 2003.

18. I Khan, R Jafrin, F Errounda, R Glitho. A data annotation architecture for semantic applications in virtualized wireless sensor networks. In Integrated Network Management, 2015 IFIP/IEEE International Symposium. 2015.

19. BSejdiu, F Ismaili, L Ahmedi. A real-time integration of semantics into heterogeneous sensor stream data with context in the internet of things. The 15th International Conference on Software Technologies (ICSOFT 2020). July 07–09, 2020, Lieusaint, Paris, France. 2020b.

20. DV Gorasiya. Comparison of open-source data stream processing engines: spark streaming, flink and storm. Technical Report. 2019.

21. Yu K, Shi W, Santoro N. Designing a streaming algorithm for outlier detection in data mining—an incremental approach. Sensors. 2020;20(5):1261.