An extended IoT framework with semantics, big data, and analytics

Conference Paper · December 2016

DOI: 10.1109/BigData.2016.7840803

4 authors:

Omer Berat Sezer
TOBB University of Economics and Technology

Erdogan Dogdu
Angelo State University

Murat Ozbayoglu
TOBB University of Economics and Technology

Aras Onal
TOBB University of Economics and Technology

An Extended IoT Framework with Semantics, Big Data, and Analytics

Omer Berat Sezer, Erdogan Dogdu, Murat Ozbayoglu, Aras Onal
Department of Computer Engineering
TOBB University of Economics and Technology
Ankara, Turkey 06560
Email: {oberatsezer, edogdu, mozbayoglu, arascanonal}@etu.edu.tr

***Abstract*—Many experts claim that data will be the most valuable commodity in the 21st century. At the same time, two of the most influential components of this era, Big Data and IoT are moving very fast, on a collision course with the methodologies that are associated with conventional data processing and database systems. As a result, new approaches like NoSQL databases, distributed architectures, etc. started appearing on the stage. Meanwhile, another technology, ontology and semantic data processing can be a very convenient catalyzer that might assist in smoothly providing this transformation process. In this paper, we propose a combined framework that brings Big Data, IoT, and semantic web together to build an augmented framework for this new era. We not only list the components of such a system and define the necessary bindings that needs to be integrated together, but also provide a realistic use case that demonstrates how the model can implement the desired functionality and achieve the goals of such a model.**

***Keywords*—Internet of things; semantics; big data analytics; framework; open system**

I. Introduction

Internet of Things (IoT) is a new technological infrastructure that extends wireless sensor technology and enables all kinds of device connectivity all over the world. Internet of things is a paradigm that is defined as "The Internet of Things allows people and things to be connected Anytime, Anyplace, with Anything and Anyone, ideally using Any path/network and Any service" [31]. In the future, it is expected that nearly all electronic devices will connect to the internet and communicate each other via networks.

IoT is widely used in different domains and application areas. Health, environment, traffic, vehicles, aviation, manufacturing, defense, home automation, communication are the examples of the application domains that use IoT technologies. In these areas the number of connected electronic IoT devices are increasing year by year. It is expected that by 2020, the number of connected electronic devices to the internet will be from 50 to 100 billions [31]. As a result of massive usage of IoT connected devices, it is expected that the total amount of generated data will be more than 35 zettabytes [41] [6].

While the number of connected IoT devices are increasing, different IoT platforms, frameworks, services are developed and deployed. There are many IoT platforms, frameworks, and related services that are being used in our daily lives for collecting and analyzing data. These are surveyed and reviewed in several works [29] [30] [25]. Important IoT platforms are listed as follows: AllJoyn, AirVantage, Arkessa, ARMmbed, Brillo Carriots, Devicehub.net, EvryThng, Exosite, GroveStreams, Ericsson IoT-Framework, IFTTT, IoTivity, Intel IoT Platforms, LinkSmart, NinjaBlock, OpenIoT, OpenMTC, OpenRemote, Open.Sen.se, Pentaho, Platform.io, realTime.io, SensorCloud, SkySpark, Statistica (Dell), Tellient, TempoIQ, The Thing System, ThingSpeak, ThingSquare, ThingWorx, Sense Tecnic WoTkit, IBM Watson IoT Platform, Vitria, Weave, Zetta. These platforms mostly support device and data management. However, most of them do not support big data management and big data analytics. Platforms such as AirVantage, Pentaho, Statistica and IBM Watson support big data management. Also, almost all of them do not contain learning tools that support analysing real data which does not fit a certain pattern. Platforms such as IBM Watson IoT Platform, contains learning tools to analyse data, but there are not too many.

Another problem with the existing platforms and frameworks is the interoperability of such platforms. This is usually not possible, and most platforms behave like islands of self-contained tools where specific standards or propriety protocols are employed and the collected data and produced information cannot be communicated or served to the outside world. And, this is totally unacceptable for the upcoming IoT vision where the idea is to communicate with anything, anywhere, anytime. Here, the missing element is the semantics it is all pointed out [2]. Data produced, collected, and stored from devices, as well as all the information generated from this raw data should be conveyed, stored, processed semantically, meaning all agreed upon standards should be utilized to structure and relate data. The Semantic Web vision is brought to the IoT world with this purpose and it seams it is a promising one [2].

In this paper, we analyze the requirements for a novel IoT framework that brings together semantic infrastructure, big data, and learning models. We also present an early design for such a platform and discuss its merits. In Section II we present the related work in semantic and big data IoT. In Section III we discuss the design requirements for a novel IoT. In Section IV, we mention conceptual design of our proposed solution. In Section V, we present a brief initial higher level technical design for such a platform and conclude in Section VI.

II. Related Work

With the development of IoT devices, systems and solutions, massive amounts of heterogeneous and unstructured data are produced continuously. This big data needs to be examined and analysed to extract the hidden and new information from

IoT systems and solutions. The heterogeneity among IoT devices and systems is yet another big problem in terms of processing and analysis of IoT big data. There are upcoming work to resolve these issues and we review some of the related work in this section.

A. Semantics in IoT

With the increasing number of sensors in IoT and sensor world, different types, data formats, measurement specifications emerged in time, which cause interoperability issues now. To overcome this problem, semantic Web solutions are used to represent data in a singular knowledge model. Resource Description Framework (RDF)[1], RDF Schema (RDFS)[2], and Web Ontology Language (OWL)[3] protocols are the main languages for knowledge representation in the semantic Web world. These semantic protocols only present the conceptual data model and rules, but not the specific serialization format. There are specific languages to represent semantic data such as Turtle, N3, and JSON-LD.

Su et al. [36] emphasizes the importance of semantic knowledge representation in the IoT world and evaluate a number of different representation languagues in terms of energy efficiency (in communication and processing). All recent semantic formats are evaluated. These are eXtensible Markup Language (XML)[4], Resource Description Framework (RDF), SenML, Notation3 (N3)[5], Turtle[6], N-Triples[7] and Javascript Object Notation for Linked Data (JSON-LD)[8]. JSON-LD is found to be an efficient and recent solution as a result of experiments.

Semantic data representation alone is not enough to solve the semantificiation and therefore heterogeneity in IoT. There needs to be common vocabularies to represent knowledge in semantic forms. Ontologies are used in the semantic web world to provide that common language to represent "things", their relationships, and more. Semantic Sensor Network (SSN)[9] Ontology developed by W3C Semantic Sensor Network Incubator Group is an initial effort in this path. SSN ontology is used to represent sensors, their properties and observations (data generated), domains, etc. in very simple generic terms and supposed to be used by all types of sensors in the world. SSN can be extended to define new features of IoT devices and sensors. In [4], for example, we extended SSN ontology to create a smart-home ontology.

A number of works attempted to develop semantic ontologies for IoT. Nambi et al. [27] proposed a generic semantic knowledge base for IoT world that supports IoT sensors' semantic definition, sensor discovery, and service infrastructure. It contributes to the area with resource, location, context, domain, policy, and service ontologies. Hachem et al. [15] proposed ontologies for IoT world, consisting of device, domain, and estimation ontologies. And, Wang et al. [39] developed an IoT ontology that covers the following aspects of the IoT domain: IoT Resources, IoT Services, Observation and Measurement, Physical Locations, Deployment Platform, QoS and QoL, and IoT Service Tests. In their ontology design, they reused SSN ontology, GeoNames ontology, and Quantity Kinds and Units[10] ontology.

Also, in literature, semantic IoT frameworks and platforms are proposed. Barbero et al. [3] proposed semantic IoT platform namely The Concept IoT platform that uses semantic languages such as XML and OWL for data format. Their framework consists of context, ontology, service and device manager. Context manager runs rule and query engine and update ontology model. Their platform also contains semantic reasoner that uses Jena Semantic Web Framework. Song et al. [35] proposed a semantic application layer middleware to solve the interoperability problems when using heterogeneous devices in the system. In their implementation, they used Bluetooth and UPnP devices as IoT devices and implement interfaces for their semantic middleware. In addition, Desai et al. [10] proposed an Semantic Gateway as Service architecture between IoT sinks and services to solve interoperability problem. Also, Gyrard et al. [14] proposed a smart IoT system that consists of 3 layers: Sensor Accessing Layer, Deducing New Knowledge Layer and Composition of Services layer. This smart architecture performs these actions respectively as follows: composing, modelling, linking, reasoning, querying, and composition of services.

In literature, generally, all semantic IoT frameworks and platforms reuse existing ontologies such as SSN, GeoNames ontologies, solve interoperability problems between sensors and actuators, and make reasoning to get further information. In our proposed solution, we will use machine learning algorithms especially neural network algorithms to get better results and hidden values. IoT data which is converted to semantic data in the framework, is used as input for learning layer. In 'Conceptual Design' Section, Section IV , it will be explained with details.

B. Big Data and Learning in IoT

Big data is defined as unstructured and mass of data that differs from traditional data with regards to defined terms "Value", "Variability", "Variety", "Veracity", "Volume", "Volecity" [8][16][12]. These "V" terms show the common features of the Big Data.

IoT big data differs from traditional big data. IoT data has its own characteristics in terms of data generation, data interoperability, and data quality [32]. Velocity, scalability, dynamics and heterogeneity are important issues while generating IoT data. Data quality can be measured with uncertainty, redundancy, ambiguity and inconsistency [32]. These IoT data characteristics should be considered while designing new model or framework. In addition to these, data streaming is another big issue that should be considered while designing new IoT frameworks and models. Streaming data has also its own typical characteristics: continuous, disordered arrival of data, unbounded data, and no persistence of data objects [32].

[1]RDF, http://www.w3.org/TR/rdf11-primer/
[2]RDFS, https://www.w3.org/TR/rdf-schema
[3]http://www.w3.org/TR/owl2-overview/
[4]https://www.w3.org/XML/
[5]http://www.w3.org/TeamSubmission/N3
[6]http://www.w3.org/TR/turtle/
[7]http://www.w3.org/TR/n-triples/
[8]https://www.w3.org/TR/json-ld/
[9]SSN Ontology, https://www.w3.org/2005/Incubator/ssn/ssnx/ssn
[10]http://www.w3.org/2005/Incubator/ssn/ssnx/qu/qu

Big data storage methods are different from traditional storage methods. There are critical factors/features that should be considered for storage: consistency (C), availability (A), and partition tolerance (P) [8][13]. When looking from IoT aspects, it can be seen that IoT big data storage frameworks and solutions are developed only recently. Jiang et al. [23] proposed an IoT-oriented data storage framework that works on cloud computing platforms. Li et al. [21] proposed a storage solution for IoT big data based on NoSQL. In addition to these, existing storage solutions are also used while implementing new designs and solutions. Cecchinel et al. [7] developed an IoT architecture that collects and processes IoT sensor data in a university campus (SophiaTech). In their solution, they used MongoDb[11] as a big data storage mechanism. Besides, Tracey et al. [37] use HBase[12] database to store data in their framework.

In literature, there are also different IoT big data management platforms, frameworks, and services for handling IoT data collection, data cleaning, data storage, data processing, and data analyzing processes. In the paper [26], these IoT big data management and knowledge discovery frameworks, models, and systems are reviewed and surveyed. Proposed frameworks, models, and systems have different missions and functions. Jin et al. [17] proposed an information framework that provides data collection, evaluation, knowledge extraction, and decision management. Wu et al. [40] proposed cognitive IoT framework that supports data analysis, semantic derivations, knowledge discovery, and decision management. Besides, Nastic et al. [28] proposed Patricia framework that have following functions: data storage, analysis and management. Tracey et al. [37] proposed an architectural framework that provides data abstraction, aggregetion, and storage. In addition, Ramaswamy et al. [33] proposed a big data framework that supports data quality, storage management, and data analytics. Another proposal is COIB-framework [26] that supports knowledge storage, cognitive decision, and data organization. COIB-framework is another one designed for industrial automation.

In addition to these, in literature, machine learning algorithms are used in IoT related areas such as ubiquitous computing, pervasive computing, ambient intelligence, wireless sensor networks, human activity recognition. Khan et al. [19] and Altun et al. [1] used back propagation artificial neural network (ANN) to recognize human activities such walking, sitting, running, etc. Choi et al. [18] used also back propagation neural network for smart home applications. Lane et al. [20] proposed deep learning networks and convolutional neural networks models in processing of IoT sensor data. In addition, Bhide et al. [5] and Saeedi et al. [34] used Bayesian networks and Naive Base Classifiers in their home automation and navigation systems. Chen et al. [9] used Support Vector Machines, Hidden Markov Model, K-Nearest Neighbor, Bayesian network in context aware search systems for IoT devices. As can be seen from this short list, in literature machine learning algorithms are used in IoT related areas extensively. However, these proposals solve local and limited problems, and they are not suitable for processing big data from heterogeneous IoT sensors and devices.

[11] https://www.mongodb.com/

[12] Apache HBase, https://hbase.apache.org

III. Design Requirements for a Novel IoT Framework

Our review of the existing frameworks shows that a novel IoT framework needs to support the following functional features:

- Semantic data modelling: Semantic Web allows us to describe the domain (IoT) using standard protocols and vocabularies. This is already taken seriously and we see more and more new frameworks supporting semantics-oriented data modelling, storage, and processing. One missing feature is reasoning in most of those frameworks. We will support full-fledge semantic web and rule-based reasoning in our framework.
- Big data analytics and learning: Data generated in IoT will be big unlike the past sensor systems with the limited data storage and processing capabilities. In the age of big data every bit of data is important and needs to be analyzed. We therefore plan to include not just big data analytics, and but also state-of-the-art learning capabilities, such as deep learning features.

We also believe that the above features can be only realized in full if the following non-functional requirements are realized:

- Standards: In the new world of IoT, standards will be more important due to the greater interoperability demands. As more systems, devices, world are connected we will see that this is only possible if all agree on common standards. Therefore, a novel IoT framework should be based on common standards only and refrain from developing its own propriety solutions.
- Open system: Higher interconnectivity and higher interoperability is only possible with open systems. For a system to be open and interconnected, it should conform to the current state-of-the-art interconnectivity requirements. These days it requires a service-oriented approach where the system functionality can be accessed via standard web services or an open API. Web services is the accepted de-facto approach for interconnectivity and openness.

Below we discuss each feature in detail.

A. *Semantics*

Semantic knowledge representation is utilized in many domains, especially in those with complex information management requirements. Domains such as health care, manufacturing, defense industries, commerce, law [11], government, etc. all have very different data modelling requirements and very complex information management issues. Therefore, classical database management approaches do not work after some point due to the enlargement of database schemas with many tables, relationships, etc. Institutions resort to multiple servers, multiple database systems, clusters, and so on, to overcome the data management issues the problem just becomes bigger with the addition of new solutions.

To resolve the data modelling and management issues in complex information systems, innovative solutions around semantic Web technologies are being utilized in recent works[13]. Sensor world also started to use semantic Web earlier [39]. IoT therefore is the next target for the semantic Web where heterogeneity is inherent with many sensor device types, propriety device outputs, many communication protocols, data formats, etc. Semantic data modelling is therefore very appropriate for IoT.

Semantic Web consists of a short list of standard protocols or languages for data modelling. These are Resource Description Framework (RDF), RDF Schema (RDFS), and Web Ontology Language (OWL). Semantic Web also has a standard query language called SPARQL. Therefore for a semantic IoT framework, we need to use these languages as a first rule. Serialization of data protocols (RDF, RDFS, OWL) is not strictly enforced in specific language. XML is used a first choice at the time the Semantic Web was proposed. But over time new languages came to the scene, and recently JSON-LD, an extension of JSON format is preferred in many semantic projects. JSON-LD is also found to be beneficial for energy efficiency in communication and processing [36] due to its compactness in comparison to XML.

B. Big Data Analytics and Learning

Nowadays, big data implementations are closely attached with their data analytics solutions. This is partly driven by the industrial demands to provide the highly anticipated "intelligent" decision support systems for businesses and customers. Companies are racing towards achieving the best decision making capabilities from the available data. This push by the industry forces the big data researchers and data scientists to come up with fast, scalable learning models and analytics solutions for the business and/or customer IoT domain.

One particular stand-out learning model is deep learning that fits well with the highly unstructured and complex nature of IoT data domain. Deep neural networks are basically multi-layered neural networks with a large number of cascaded layers each with hierarchial feature learning capabilities. Deep learning works very well in cases where the data set is huge and there are a large number features (such as individual pixels of images, or individual elements, or components of time series signals, etc). The main reason deep learning is becoming the preferred choice for data scientists is due to the fact that conventional neural network models require feature processing and are mostly not scalable enough to provide solutions for big data. Meanwhile, deep neural networks generally do not require domain expertise and feature engineering, and works well with big data domain, thus feeding the raw data into the deep learning model directly can provide fast, scalable and more accurate data analytics solutions for the users.

C. Standards

One of the main issues in IoT development is the lack of standardization which leads to a large number incompatible systems that cannot interoperate. This, we mentioned above, can be overcome with the semantic data modelling approaches. But, even then, without common terms and definitions, many ontologies will be created in an ad-hoc manner, which is the case now we believe, and ultimately the same situation will be the case, there will be many ontologies with the same interoperability issues. Therefore, it is paramount to develop common ontologies as standards and adhere to these standards. Recent developments show that there are efforts towards this path. For example, Object Management Group (OMG) is developing an Industrial IoT standard with semantics in focus[14]. Another example, OpenADR, develops standards for the electricity demand response systems. It is apparent that we will see more standards in this area in the future for the sake of greater interoperability demands. SSN (Semantic Sensor Network) ontology as we mentioned above is one of the first standards in this area for the basic description of sensors and their data.

In our framework development, we also plan to use the existing standards and adapt the upcoming standards so that the framework provides the best interoperability options as an open system.

D. Open System

As we pointed out above, for a wider interconnected IoT world, any new IoT framework or system should provide utmost possible openness features. Otherwise, the system becomes interlocked and closed, and ultimately disconnected from the outside world. And, this is unwanted for any computer and software system in the internet age. An open system can work (interoperate) with other systems so that data, information, and services can be exchanged for greater value-added benefits for both worlds. Current approach for implementing an open system is to develop the system using a aervice-oriented approach. This means any essential functionality should be implemented as web service and be opened via a web interface. These functionalities could be at least the following in the context of IoT:

- Device management: Adding, deleting, updating IoT devices and their metadata to the system. Device type management, device grouping, and more.
- Device/data management: Controlling the data flow from devices, queueing, data storage, and more.

Of course the list can be extended. But, this does not mean that everything is open to everyone. There could be security and authentication mechanisms to protect the privacy and security of data and services in an open IoT system.

IV. Conceptual Design

As a result of the requirements discussed in the previous sections, we have created a conceptual design of the IoT system we envision. This is a multi-layered framework in order to perform many tasks required independently. The conceptual design is depicted in Figure 1, it consists of five main layers, namely (1) data acquisition, (2) extract-transform-load (ETL), (3) semantic-rule reasoning, (4) learning, and (5) action layers bottom up. In this path, raw data from sensors are parsed, semantics are added, rules are applied, learning is done, and finally actions are taken. In this section, we will describe these layers and their purposes in detail. Furthermore, we will also describe the use of semantics on this platform with an example.

[13] Semantic web use cases, http://www.w3.org/2001/sw/sweo/public/UseCases/

[14] https://www.rtinsights.com/industrial-internet-semantic-standards/

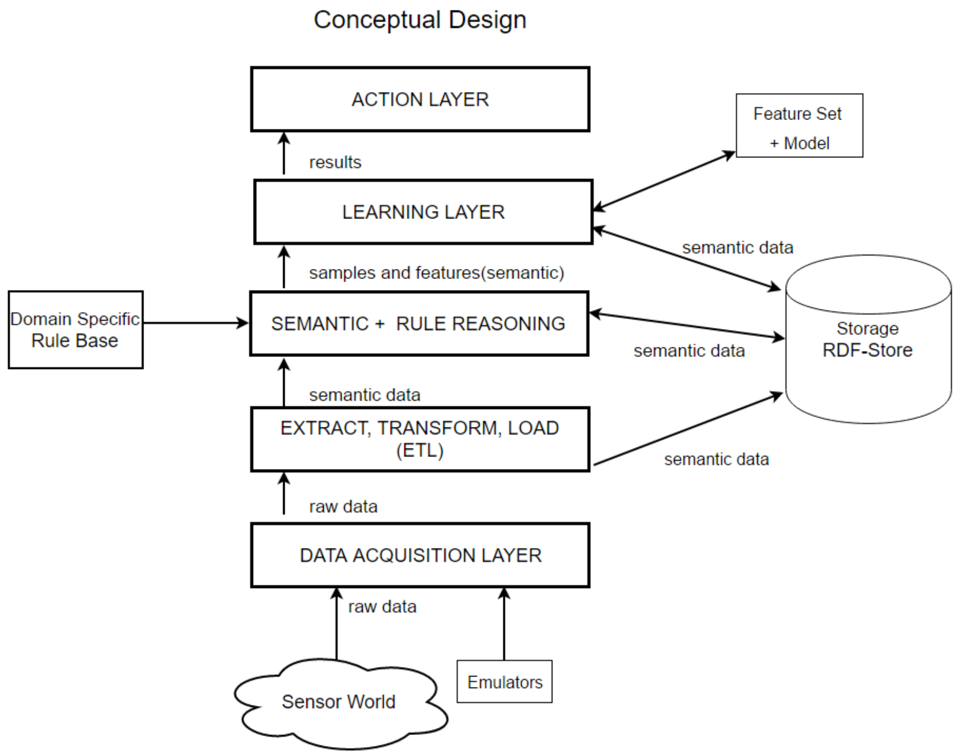

Fig. 1. Conceptual design of our extended IoT framework

The first layer in the framework is data acquisition layer, which is responsible for collecting all kinds of data from resources, more specifically sensors, from the outside world. It can be seen as input layer since the framework uses this layer to interact with sensors. The incoming data is raw data and the only task this layer accomplishes is acquiring and conveying raw data to ETL layer for processing. Assume we have a sensor that sends meteorological data on a regular basis to the framework. Its format can be such as (40, 25,5,39.9334N, 32.8597E). This layer does not touch its contents or parse it in any way. But, it makes sure that the data sent from sensors are not lost. To accomplish this robust data acquisition mechanisms, such as multi-threaded queues, should be employed.

The second layer in the framework is ETL (Extract, Transform, Load) layer. The incoming data from data acquisition layer is received by the ETL layer for parsing purposes. Since different kinds of sensors send different types and formats of data, ETL layer consists of sensor drivers for each sensor type to receive and parse data appropriately. For instance, a humidity sensor and temperature sensor may send data in different formats. Furthermore, each sensor driver is responsible for producing data in the right type, right unit, and format depending on the vendor, type, and version. For example, a temperature sensor from vendor A may produce data in Celcius unit while another sensor from vendor B may produce data in Fahreheit unit. This should be differentiated in the platform in all layers. For this purpose ETL layer is responsible for keeping data in the right type and format regardless of the sensor type via semantic technologies. Data is converted to a semantic format in the form of RDF (Resource Description Framework) protocol, the very basic semantic protocol to describe statements. At this point, artifacts from SSN ontology will be used along with our ontology constructs. Consider the example in the previous paragraph. The data coming from a sensor as (40, 25, 5, 39.9334N, 32.8597E) is parsed by the ETL layer (using a specific driver for the sensor), and the data is split into the right pieces. For example, the first part of the data (40) is taken as temperature (F), (25) as wind speed (km/h), (5) as humidity, and the rest as coordinates. The next step is to store this data in RDF format. And, it goes something like (:obs ssn:hasValue :v1. :v1 :value "40". :v1 :unit "F") describing the data value, type, and unit as triples. An example sensor description can be seen in Figure 2. In Figure 2, :HumiditySensor makes use of ssn:Sensor class and observes humidity and observation value :o1 defined by using ssn:Observation along with its other properties such as observation time and observation result. All sensor data is expressed by using SSN and our ontology as in the example above.

The third layer is semantic-rule reasoning layer. This layer makes use of the data coming from ETL layer that is already in semantic form and properly parsed along with the domain specific parsing rules defined in drivers. The main purpose of reasoning layer is to designate the limits of the domain and make basic inferences from the RDF data using a reasoning engine embedded. Two types of rules are executed on the semantic sensor data. The first is the semantic reasoning rules inherent to the semantic web protocols (RDF, RDFS, OWL). These are protocol or language specific rules[15], inferred automatically using the rule engine. The second is the domain

[15] http://www.w3.org/standards/semanticweb/inference

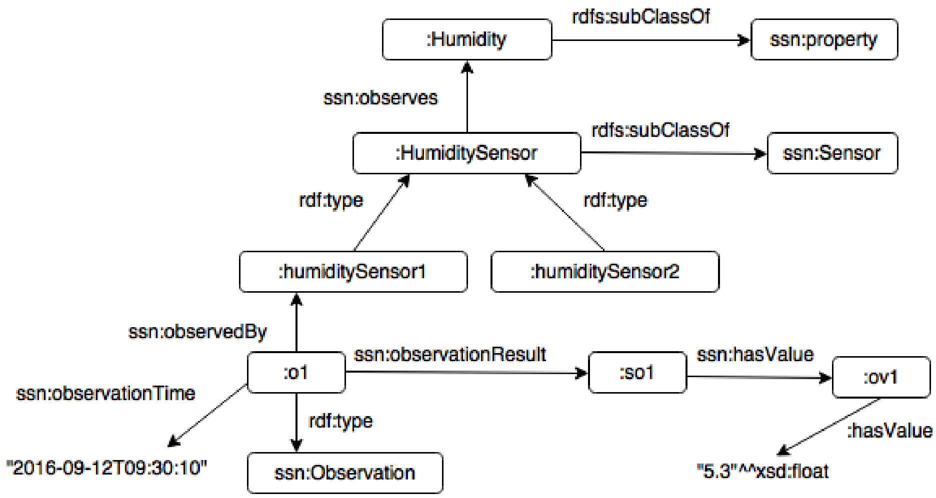

Fig. 2. Sample semantic model of sensors and observation data in our proposed system.

specific or user specific rules. For instance, in a smart home environment you may want to keep the temperature between 25 and 45 celcius. For this purpose a domain rule can be introduced to the framework to state, ”if the temperature is above 45 celcius, then activate the air conditioner or if it is below 25 activate the heater”. Domain specific rules are as important as the semantic rules for a functional system. An example rule in Jena rule syntax[16] can be seen in following code snippet:

```
[(?s rdf:type :HumidtySensor)
(?o ssn:observedBy ?s)
(?o ssn:observationResult ?r)
(?r ssn:hasValue ?v)
(?v :hasQuantityValue ?q)
(greaterThan(?q,5.0))
  -> (:humidtyAlert1 :value "on")
     (:humidtyAlert1 rdf:type :Alert)]
```

This rule basically states that if the humidity value is higher than 5, then set :humidityAlert1 on.

A language-specific semantic reasoning on the hand can be depicted using the following example. If :humiditySensor1 object is a (of type) :HumiditySensor and :HumiditySensor is a subclass of ssn:Sensor, then :humiditySensor1 is also a ssn:Sensor.

```
:humiditySensor1 rdf:type :HumiditySensor
:HumiditySensor rdfs:subClassOf ssn:Sensor
:humiditySensor1 rdf:type ssn:Sensor
```

The last statement is inferred as a result of RDFS reasoning rules (type inferrence due to subclass hierarchy).

The fourth layer is learning layer. This layer basically extracts features from the data and builds learning models by applying machine learning methods. This layer consists of two substeps as preprocessing and learning. The features coming from semantic-rule reasoning layer can be excessive and using all of them without any preprocessing or filtering method can end up with low success rates in learning algorithms. Therefore, feature selection methods should be used to weed out irrelevant features. Principal Component Analysis or subset selection can be used for this task. After selecting the most relevant features, the next step is designating a learning algorithm. Various deep learning algorithms can be used and the most successful one can be selected or the result can be determined by combining various algorithms in a voting classifier style. Also, the learning layer does not need to know about domain specific rules. For example, assume you specified a domain specific rule such as ”if temperature is above 45, activate the air conditioner”. Then, if the temperature is above 45, the air conditioner will be on and the learning layer will acquire this information about whether the air conditioner is on or off rather than the rule itself.

The last layer is action layer. The results that the learning layer produces will be evaluated and necessary actions will be taken in this layer. There will be predetermined actions for learning algorithm’s output values that are defined by user. For instance, assume the learning algorithm produces three different outputs by using meteorological features in order to determine the probability of rain such as ”lowProbability”, ”mediumProbability” and ”highProbability”. User will decide what happens in each case. For instance, the user can define an action for the samples classified as ”highProbability” as ”produce a rain warning”.

V. Technical Design Issues

Implementing a new IoT framework is a very difficult task. There are many design and component choices, but choosing the right technology and method is a challenging task. We are fortunate though that there are many open source components we can choose to construct such a framework.

The new features of our framework will be in two fronts: semantics and big data analytics. Therefore, we will only discuss shortly the technical solutions we can use for these features.

Semantics require, as we mentioned above, the use of standard data modelling protocols RDF, RDFS, OWL, and the query language SPARQL. As we mentioned above in the literature review section, the language of choice for semantic

[16]https://jena.apache.org/documentation/inference/index.html#RULEsyntax

encoding of data these days is JSON-LD. It is much more compact than, for example XML, and there are many open source tools to process JSON data. Therefore, for the persistence of semantic data we will use JSON-LD. For the storage of JSON-LD data, again there are many tools to choose from. But recent developments show that NoSQL databases are frequently used to store and manage JSON data. This design choice will free us from the task of transforming data from JSON to a database format and back, and we will used JSON-LD format all around the framework with no transformation.

We will use NoSQL for storage purposes. And, this choice is based on two reasons. First, the framework should be able to handle many write requests coming from sensors, and at this point the database should handle these requests with ease. Furthermore, the read time should be fast enough to handle many data oriented operations. NoSQL yields better performance in those areas [22]. The second advantage is the scalability of the database. The amount of data has a tendency to grow tremendously considering the fact that there will be many sensors sending data at the same time. Therefore, NoSQL is more convenient for large sensor applications which tend to grow horizontally [38].

There is a drawback for storing semantic JSON-LD data in a NoSQL database. There are not yet any tools, to the best of our knowledge, supporting semantic and rule reasoning on JSON-LD data [24]. There are other tools such as Virtuoso, Jena TDB, Sesame, Oracle Spatial and Graph, AllegroGraph and many more with inherent semantic data storage and reasoning capabilities. As of now, the best choice could be to read JSON-LD data into a reasoner and after executing the reasoning task, the results are stored back in the JSON-LD storage. This is of course not very efficient, but it will be a temporary solution. We belive there will be native reasoners working on JSON-LD data soon.

For big data analytics, again we are faced with a lot of different solutions to choose from. Apache Hadoop has been around quite a while, but new comers such Apache Spark are more robust and efficient in big data analytics jobs. Spark comes with a comprehensive machine learning library, MLlib as well. Therefore, Apache Spark seems to be the technical solution choice for big data analytics.

Another important concern in IoT data processing the streaming data, or in other words, big data coming from devices continuously and in real-time. There are new software and hardware solutions in parallel/distributed platforms and efficient virtualization solutions result in effective distributed analytics solutions that are highly scalable and can address the processing needs for unstructured real-time IoT data flow from a large number of sensors. Apache Spark also has a streaming data processing component called Spark Streaming. That could also free us from attaching a different tool for stream processing.

VI. Conclusion

In this paper we reviewed and discussed the design requirements for a comprehensive IoT framework with novel features in semantics and big data analytics. The framework will combine an all around semantic infrastructure, and big data and learning capabilities to be implemented on the semantic data. It will provide effective support for all kinds of sensors with the purpose of storing data, making semantic rule reasoning on this data, and then employing machine learning methods to get the best results. The framework aims to provide better management features for a sensor network with different types of sensors as a result of using data semantics.

We plan to implement the framework with the aforementioned tools and techniques. We will also employ a real-world use case in a domain such as smart grid, health care, etc. to test and prove the effectiveness of the framework. During implementation we will also try, test, and possibly change the tools and techniques.

References

[1] K. Altun and B. Barshan, “Human activity recognition using inertial/magnetic sensor units,” in *Lecture Notes in Computer Science*, vol. 6219 LNCS, Springer Berlin Heidelberg, 2010, pp. 38–51.

[2] L. Atzori, A. Iera, and G. Morabito, “The internet of things: a survey,” *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[3] C. Barbero, P. D. Zovo, and B. Gobbi, “A flexible context aware reasoning approach for iot applications,” in *IEEE 12th International Conference on Mobile Data Management*, 2011, pp. 266–275.

[4] O. Berat Sezer, S. Z. Can, and E. Dogdu, “Development of a smart home ontology and the implementation of a semantic sensor network simulator: an internet of things approach,” in *International Conference on Collaboration Technologies and Systems (CTS)*, 2015, pp. 12–18.

[5] V. H. Bhide and S. Wagh, “I-learning iot: an intelligent self learning system for home automation using iot,” in *International Conference on Communications and Signal Processing*, 2015, pp. 1763–1767.

[6] J. Borges Neto, T. Silva, R. Assunção, R. Mini, and A. Loureiro, “Sensing in the collaborative internet of things,” *Sensors*, vol. 15, no. 3, pp. 6607–6632, 2015.

[7] C. Cecchinel, M. Jimenez, S. Mosser, and M. Riveill, “An architecture to support the collection of big data in the internet of things,” in *IEEE World Congress on Services*, 2014, pp. 442–449.

[8] M. Chen, S. W. Mao, and Y. H. Liu, “Big data: a survey,” *Mobile Networks & Applications*, vol. 19, no. 2, pp. 171–209, 2014.

[9] Y. Chen, J. Zhou, and M. Guo, “A context-aware search system for internet of things based on hierarchical context model,” *Telecommunication Systems*, vol. 62, no. 1, pp. 77–91, 2016.

[10] P. Desai, A. Sheth, and P. Anantharam, “Semantic gateway as a service architecture for iot interoperability,” in *IEEE International Conference on Mobile Services*, 2015, pp. 313–319.

[11] E. Dogdu, B. Katipoglu, and U. Guney, “Waste management information system,” in *Proc. of The Knowledge Engineering and Ontology Development (KEOD 2014) part of the Int. Joint Conf. on Knowledge Discovery, Knowledge Engineering and Knowledge Management IC3K*, 2014, pp. 312–318.

[12] W. Fan and A. Bifet, “Mining big data,” *ACM SIGKDD Explorations Newsletter*, vol. 14, no. 2, p. 1, 2013.

[13] Gazal and P. D. Kaur, "A survey on big data storage strategies," in *International Conference on Green Computing and Internet of Things*, IEEE, Oct. 2015, pp. 280–284.
[14] A. Gyrard, P. Patel, A. Sheth, and M. Serrano, "Building the web of knowledge with smart iot applications (extended version)," 2016.
[15] S. Hachem, T. Teixeira, and V. Issarny, "Ontologies for the internet of things," in *Proceedings of the 8th Middleware Doctoral Symposium on - MDS '11*, ACM Press, 2011, pp. 1–6.
[16] M. Hilbert, "Big data for development: a review of promises and challenges," *Development Policy Review*, vol. 34, no. 1, pp. 135–174, 2016.
[17] J. Jin, J. Gubbi, S. Marusic, and M. Palaniswami, "An information framework for creating a smart city through internet of things," *IEEE Internet of Things Journal*, vol. 1, no. 2, pp. 112–121, 2014.
[18] J. Jonghwa Choi, D. Dongkyoo Shin, and D. Dongil Shin, "Research and implementation of the context-aware middleware for controlling home appliances," *IEEE Transactions on Consumer Electronics*, vol. 51, no. 1, pp. 301–306, 2005.
[19] A. M. Khan, Y.-K. Young-Koo Lee, S. Y. Lee, and T.-S. Tae-Seong Kim, "A triaxial accelerometer-based physical-activity recognition via augmented-signal features and a hierarchical recognizer," *IEEE Transactions on Information Technology in Biomedicine*, vol. 14, no. 5, pp. 1166–1172, 2010.
[20] N. D. Lane, S. Bhattacharya, P. Georgiev, C. Forlivesi, and F. Kawsar, "An early resource characterization of deep learning on wearables, smartphones and internet-of-things devices," in *Int. Workshop on Internet of Things towards Applications*, ACM, 2015, pp. 7–12.
[21] T. Li, Y. Liu, Y. Tian, S. Shen, and W. Mao, "A storage solution for massive iot data based on nosql," in *IEEE International Conference on Green Computing and Communications*, 2012, pp. 50–57.
[22] Y. Li and S. Manoharan, "A performance comparison of sql and nosql databases," in *2013 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM)*, 2013, pp. 15–19.
[23] Lihong Jiang, Li Da Xu, Hongming Cai, Zuhai Jiang, Fenglin Bu, and Boyi Xu, "An iot-oriented data storage framework in cloud computing platform," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1443–1451, 2014.
[24] A. I. Maarala, X. Su, and J. Riekki, "Semantic data provisioning and reasoning for the internet of things," in *2014 International Conference on the Internet of Things (IOT)*, 2014, pp. 67–72.
[25] J. Mineraud, O. Mazhelis, X. Su, and S. Tarkoma, "A gap analysis of internet-of-things platforms," *Computer Communications*, vol. 89-90, no. 9, pp. 5–16, 2015.
[26] N. Mishra, C.-C. Lin, and H.-T. Chang, "A cognitive adopted framework for iot big-data management and knowledge discovery prospective," *Int. Journal of Distributed Sensor Networks*, vol. 2015, pp. 1–12, 2015.
[27] S. N. A. U. Nambi, C. Sarkar, R. V. Prasad, and A. Rahim, "A unified semantic knowledge base for iot," in *2014 IEEE World Forum on Internet of Things (WF-IoT)*, IEEE, 2014, pp. 575–580.
[28] S. Nastic, S. Sehic, M. Vogler, H.-L. Truong, and S. Dustdar, "Patricia – a novel programming model for iot applications on cloud platforms," in *2013 IEEE 6th International Conference on Service-Oriented Computing and Applications*, IEEE, 2013, pp. 53–60.
[29] C. Perera, C. H. I. H. Liu, S. Jayawardena, and M. Chen, "A survey on internet of things from industrial market perspective," *IEEE Access*, vol. 2, no. 2014, pp. 1660–1679, 2014.
[30] C. Perera, C. H. Liu, and S. Jayawardena, "The emerging internet of things marketplace from an industrial perspective: a survey," *IEEE Transactions on Emerging Topics in Computing*, vol. 3, no. 4, pp. 585–598, 2015.
[31] C. Perera, A. Zaslavsky, M. Compton, P. Christen, and D. Georgakopoulos, "Semantic-driven configuration of internet of things middleware," in *9th International Conference on Semantics, Knowledge and Grids*, IEEE, 2013, pp. 66–73.
[32] Y. Qin, Q. Z. Sheng, N. J. G. Falkner, S. Dustdar, H. Wang, and A. V. Vasilakos, "When things matter: a survey on data-centric internet of things," *Journal of Network and Computer Applications*, vol. 64, no. 4, pp. 137–153, 2016.
[33] L. Ramaswamy, V. Lawson, and S. V. Gogineni, "Towards a quality-centric big data architecture for federated sensor services," in *2013 IEEE International Congress on Big Data*, IEEE, Jun. 2013, pp. 86–93.
[34] S. Saeedi, A. Moussa, and N. El-Sheimy, "Context-aware personal navigation using embedded sensor fusion in smartphones," *Sensors*, vol. 14, no. 4, pp. 5742–5767, 2014.
[35] Z. Song, A. A. Cardenas, and R. Masuoka, "Semantic middleware for the internet of things," in *2010 Internet of Things (IOT)*, IEEE, Nov. 2010, pp. 1–8.
[36] X. Su, J. Riekki, J. K. Nurminen, J. Nieminen, and M. Koskimies, "Adding semantics to internet of things," *Concurrency and Computation: Practice and Experience*, vol. 27, no. 8, pp. 1844–1860, Jun. 2015.
[37] D. Tracey and C. Sreenan, "A holistic architecture for the internet of things, sensing services and big data," in *2013 13th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing*, IEEE, May 2013, pp. 546–553.
[38] J. S. van der Veen, B. van der Waaij, and R. J. Meijer, "Sensor data storage performance: sql or nosql, physical or virtual," in *2012 IEEE Fifth International Conference on Cloud Computing*, IEEE, Jun. 2012, pp. 431–438.
[39] W. Wang, S. De, G. Cassar, and K. Moessner, "Knowledge representation in the internet of things: semantic modelling and its applications," *Automatika Journal for Control, Measurement, Electronics, Computing and Communications*, vol. 54, no. 4, 2013.
[40] Q. Wu, G. Ding, Y. Xu, S. Feng, Z. Du, J. Wang, and K. Long, "Cognitive internet of things: a new paradigm beyond connection," *IEEE Internet of Things Journal*, vol. 1, no. 2, pp. 129–143, 2014.
[41] A. Zaslavsky, C. Perera, and D. Georgakopoulos, "Sensing as a service and big data," in *Proceedings of the International Conference on Advances in Cloud Computing*, 2012, pp. 21–29.