Providing Sensor Services by Data Correlation: The #SmartME Approach

Nidhi Kushwaha[1(✉)], Giovanni Merlino[1], Longo Francesco[1], Bruneo Dario[1], Antonio Puliafito[1], and O.P. Vyas[2]

[1] Department of Engineering, University of Messina, Messina, Italy
kushwaha.nidhi12@gmail.com,
{gmerlino,flongo,dbruneo,apuliafito}@unime.it
[2] Department of Information Technology, IIIT-Naya Raipur, Raipur, India
opv@iiitnr.edu.in

Abstract. In the current era Internet is the most used medium for sharing and retrieving the information for building applications which are commonly developed for enhancing the user experience in terms of comfort, communication. For this, the need of real-time sensor data gains importance. The data collected from the physical objects should be easily available for different applications. Semantic representation of the sensor data directly addresses the problem of storing it in logical, easily accessible and extensible manner. Our paper works towards converting the already collected sensor data of the #SmartME project into semantic format and also proposes real-time storage of semantically enriched sensor data. To build applications using these sensor data the authors consider mainly three kinds of sensors, i.e., Temperature, Humidity, Pressure. Predicting the observed value of any sensor data is the main aim of this work. The analysis leverages other sensors & environmental parameters such as Date, Time, Longitude, Latitude, Altitude etc. Correlation among these parameters and the accuracy of the predicted results showed the suitability of our proposed idea.

Keywords: SmartME · IoT · Sensor network · Stack4Things · Semantic web · Data mining · Correlations

1 Introduction

A Smart City concept incorporates advanced Information and Communication Technologies for providing the services for the betterment of the citizens. These key services depend not only on the needs of the citizens but also on the region/area where the city is located. As per the recent prediction of Cisco Systems and World Health Organization in 2015, it was noticed that more than 60% of the world's population will live in cities by 2050 [6]. This shifts certainly raise challenges for the basic needs of a person such as water, electricity, fuel supply, building cost-effective infrastructures, proper drainage, waste, air-pollution, parking, traffic, transportation, street lighting management system, healthcare,

L. Barolli and O. Terzo (eds.), *Complex, Intelligent, and Software Intensive Systems*,
Advances in Intelligent Systems and Computing 611, DOI 10.1007/978-3-319-61566-0_82

education system, safety and security services. For building such type of applications, the domain of the Internet of Things (IoT), Semantic Web, Social Web and Machine Learning have been exploited in recent years. The main components of IoT are sensors, smart phones & other embedded portable, devices and their internet connectivity. Connectivity helps in leveraging the raw data for building smart applications. Semantic and Social Web provide supporting knowledge for making the applications even smarter. Furthermore, Semantic Web technologies have the ability to annotate sensor's raw data & observation by specific domain ontology. This annotations enhance the interpretability of the sensor data. It also allows to build an application without the overhead of heterogeneous sources of data. Semantic annotation of sensor data provides an opportunity to do deep queries. For example, IBM proposed a model [9,10] to diagnose the faulty sensors deployed in a building by expressing the rules through Semantic Web. The correlation logic among different sensors has been explored for diagnosing the faulty sensor. However, Machine Learning acts as a complement for smart applications by analysing the previously collected data and acts making predictions based on it. For example, by doing predictive analysis, Chicago is controlling the rodent population. The application is able to determine which trash dumps are most likely to be full and attract more rats in the near future.

In this paper the authors summarize the recent applications developed in semantic sensor areas. The paper first outlines the conversion of raw data into Semantic Web for the running #SmartME project. Secondly, the authors proposes an idea on how correlation between sensors can be taken into consideration for predicting unspecified/destroyed sensors deployed in a specific location.

2 Related Work

Sensors are becoming popular these days to collect huge amount of specific information about real world surroundings. However, these sensors cannot work alone for providing the services smart cities need. To provide the systematic description of sensor networks various attempts were made in the past such as, Sensor Web Enablement (SWE) developed by the Open Geospatial Consortium (OGC) that are widely being adopted by industry, government organizations and academia. SWE provides XML representation for sensor description, thus limited to only syntactical representations [12], rather than semantics. Moreover, semantics of data provide the interpretability of sensor data & observations in terms of machine-readability. This interpretability helps to provide high-level information to the application's user and devises the application more user-friendly as well as easy to use.

Semantic Sensor Web [11] and Semantic Sensor Network (SSN) Ontology [7] have been proposed to define sensor data model for describing the sensor and their services. They also provide uniform descriptions and high-level interfaces for sensors and actuators. The semantic description of sensors mitigates the complexities involved in the heterogeneity of data collection and in the underlying technologies being used for sensors and actuator devices. It has the capability to provides low-level information about sensor data, as well as high-level reasoning for human understandability. For example, information about the sensor type, observation, respective board (where the sensor is connected), location name and accuracy of measurement are defined as low-level information about sensors. On the other hand, analysing the relation between sensors and recommend the user with personalized weather prediction is called as high-level information. Furthermore, the collection of sensor observations for building applications requires Cloud real-time storage to analyse the sensors data in various combinations for smart applications. Predicting thermal comfort of employees was proposed by Kojima in 2008 [4]. To achieve it, 21 sensor were deployed inside a room. The paper utilizes thermal comfort data from employees in parallel with other physically located sensors. On the other hand, [8] proposed the service infrastructure for semantic sensor data storage and querying. The paper considered the parking spots and empty room examples and described the information related in semantic formats. Also, the authors investigated about the automatic registration of a new sensors by automatically annotating the sensors with appropriate descrition. Different ontologies were merged such as Dolce Ultralite, the W3C Semantic Sensor Network (SSN-XG), Event model ontology to support cross-domain descriptions. Ploennings et al. in 2014 [10] proposed a diagnosis model for smart building application. This diagnosis model uses semantic information or cause-effect-relationships between sensors. Mainly the approach used two type of sensors i.e. Temperature and Occupancy sensors. The defect of temperature sensors were deduced by the occupancy and cooling sensors. The authors extended the SSN ontology by modifying it specifically for the building infrastructure. The approach was used as a common model for all the buildings create automatically the similar physical model for them. Consoli et al. [2] gathered Catania municipality data from different sources/formats and made them available online as semantic knowledge base. The authors introduced different techniques to convert JSON, XML, SQL Server database and Excel files into RDF format. The goal of this approach is to boost the semantic data based smart city applications for more relevant information retrieval and processing, without the overhead of complexities involved in it.

Recently, authors in [1,5] proposed a project named #SmartME which aims to create a Cloud based infrastructure to model IoT and to control the smart objects remotely. This Smart City infrastructure provides a unique view for all the smart objects located in the different parts of the smart cities. In this infrastructure, the IoT nodes comprise different types of sensors (Temperature, Humidity, Brightness, Noise, Pressure) that are attached to Arduino YUN boards. These boards are able to send the information to the CKAN repository through the Stack4Things cloud infrastructure. The Stack4Things framework was developed by Mobile and Distributed System (MDLSLab) at the University of Messina, Italy. Lightning-rod on the client side and Iotronic at the server side are the main building blocks of the Stack4things technology. Stack4Things Lightning-rod runs on the micro-processor unit of the smart boards (e.g., Arduino YUN). It interacts with the OS tools and services of the board as well as sensing and actuating resources through I/O pins. On the other hand, Stack4Things Iotronic provides an OpenStack service to the end user for managing the board resources remotely. The users can manage the boards via REST APIs or using the Stack4Things command line client. The logic deployed by Stack4Things pushing the data from Arduino YUN boards to store them in CKAN repository, is written under the guise of Node.js plug-ins at the client side, i.e. in Lightning-rod. The sensors data are currently in standalone (raw) format and also have less interpretability for the application developer. To enhance the experience of sensor data acquisition and to link them logically a semantic layer should be there. Furthermore, siloed information from unrelated sensors would not help to build a smart applications. Thus, there is a need to develop a prediction layer for analysing the appropriate correlated sensor to build a smart application. So, this work focuses on how to provide semantic interoperability to the Stack4Things infrastructure and to investigate the correlation between sensors, for building simple but effective smart applications.

3 Semantic Integration

In this paper, we described the work done to incorporate the semantic layer within Stack4Things and also to provide sensor correlations. In our case, to store the information into Virtuoso RDF storage a Node.js plugin file has been modified so that it can send the RDF data at the same time when it is being sent to the CKAN repository for storage. Thus, it enables us to store real-time data in the RDF format.

Furthermore, to store previous data into the storage, the RDF4J 2.2 Java framework has been exploited. Using this framework the CKAN data has been converted directly into Turtle format and later on inserted into the same storage. We stored the converted file on a monthly basis. For example, “2016-04.ttl” file contains all the sensor information of April 2016. All active board’s information has been collected from the CKAN repository. To access these data using the same vocabularies we modified the SSN ontology by re-using three basic ontologies i.e., DateTime.owl, SSN.owl, wgs84pos.rdf, in the Protege tool. We used URI design specifications for choosing the appropriate URIs for “SmartMe.owl” ontology. After storing this information we are able to extract specific information using SPARQL queries. For example, for a particular board, we can retrieve label of the board, geolocation (latitude, longitude, altitude), Manufacturer name, Model of the board, Time and Date of the deployment. Furthermore, for a particular property of a sensor, the SPARQL query can retrieve boardID, observed value, day of the week, date of observation(YYYY-MM-DD), time of observation (hh:mm:ss), unit of the observation type, geo-location, maximum-minimum observation values for a certain time period, and also, group the sensor data by date, week and month. For example, Fig. 2 retrieved the values of temperature sensor (connected to a specific board) gathered over a certain time period grouped by month. The results of this query has been shown in Fig. 3. Moreover, Figs. 4 and 5 provide the list of the specific boards (i.e. manufacturer

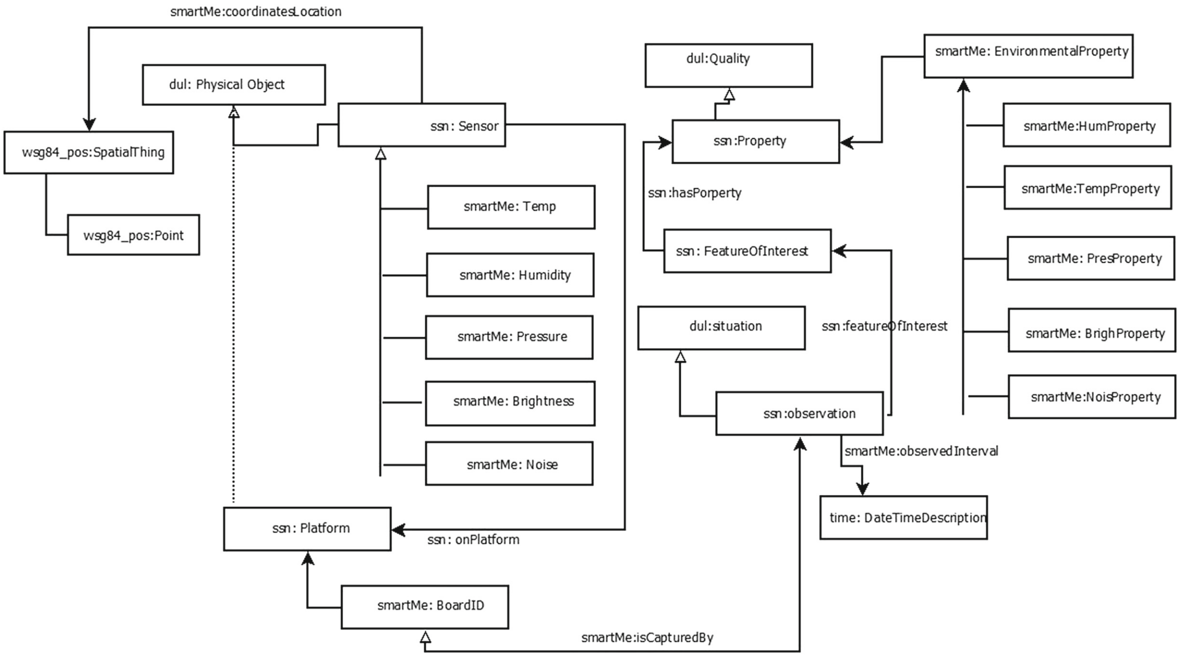


Fig. 1. SmartMe ontology description

= "Arduino" & model = "Yun") hosting a specific kind (i.e. manufacturer = "Honeywell" & model = "HIH-4030131 Series") of sensors. As shown in Fig. 1, the ontology has been developed in the Protege tool by integrating DateTime, Geographical ontologies with SSN ontology. The URIs have been selected with the consideration of pattern based ontology design techniques. All the five sensors of #SmartME project are defined as sub-class of the sensor class. The observation property of each sensors is created as a subclass of the Environmental Property of sensor. Sensor observation is related to the time and Date description, for which the vocabulary of DateTime ontology has been exploited. Our SmartME ontology is closely related to the idea described in by Atemezing et al. This modified ontology with the converted data (i.e., inside from CKAN) gets stored in virtuoso RDF storage for querying.

We collected data from 20 Arduino YUN boards, which have been active since May 2016. The data have been converted from Comma Separated Values (CSV) to RDF and stored into the Virtuoso RDF storage. For CSV data generation, Python code is used to automatically fetches data[1] from the CKAN repository and store it into CSV files. These files are then used to generate TTL (Turtle, a RDF serialization) files using RDF4J library. The fields of the CSV file are as follows:

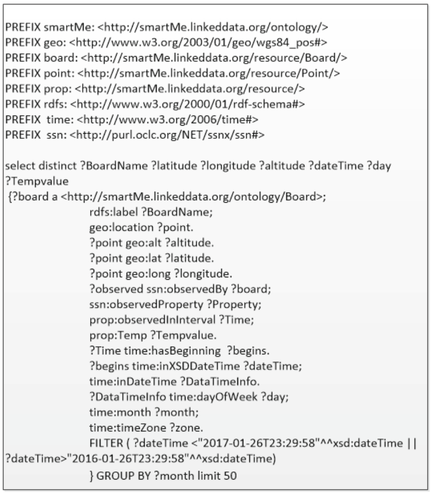
- BoardID
- SensorID
- Observation Type
- Observation Value
- Day of Week
- Date
- Month
- Year
- Hour
- Minute
- Second
- Zone
- Latitude
- Longitude
- Altitude

All the .ttl files and SmartMe ontology have been stored in Virtuoso RDF storage. To insert the real-time data captured by the sensors, we used HTTP POST code[2] inside Node.js plugin file. This Node.js plugin is being used by the Lightning-rod component of the Stack4Things. We stored all the real-time as well previously collected data in "https://smartMe.linkeddata.org" graph of Virtuoso RDF Storage. Note that, this graph also contains "smartMe.owl" as well as other required ontologies.

The overall view of the project has been shown in Fig. 6. The figure depicts real-time sensor data being stored in both CKAN and Virtuoso RDF Storage. We converted the already stored CKAN data into RDF format using RDF4J 2.2. It helps us to analyse previously stored data as well. In future we are planning to link data with GeoNames Ontology for further data enhancement. Also, provide an interface with virtuoso global server for user interaction and analysis.

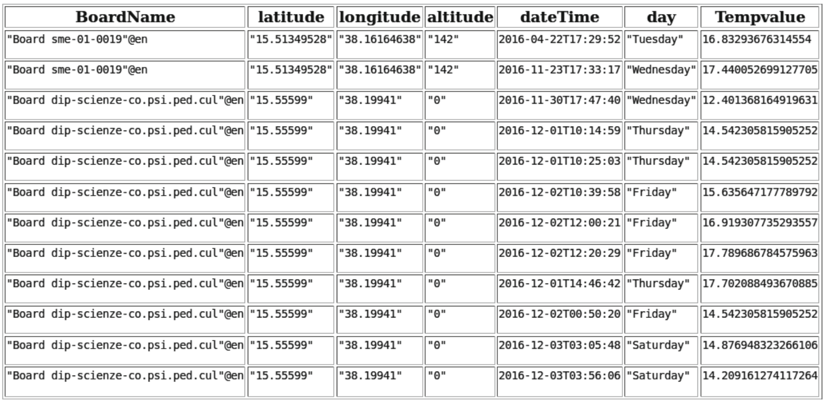
[1] http://smartme-data.unime.it/organization/smartme.

[2] https://virtuoso.openlinksw.com/dataspace/doc/dav/wiki/Main/VirtGraphProtocolCURLExamples.



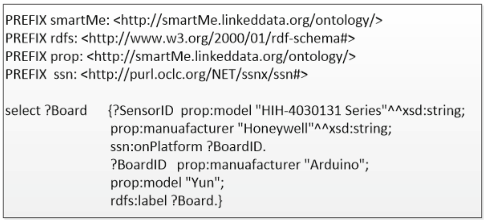
```
PREFIX smartMe: <http://smartMe.linkeddata.org/ontology/>
PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>
PREFIX board: <http://smartMe.linkeddata.org/resource/Board/>
PREFIX point: <http://smartMe.linkeddata.org/resource/Point/>
PREFIX prop: <http://smartMe.linkeddata.org/resource/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX  time: <http://www.w3.org/2006/time#>
PREFIX  ssn: <http://purl.oclc.org/NET/ssnx/ssn#>

select distinct ?BoardName ?latitude ?longitude ?altitude ?dateTime ?day
?Tempvalue
 {?board a <http://smartMe.linkeddata.org/ontology/Board>;
                  rdfs:label ?BoardName;
                  geo:location ?point.
                  ?point geo:alt ?altitude.
                  ?point geo:lat ?latitude.
                  ?point geo:long ?longitude.
                  ?observed ssn:observedBy ?board;
                  ssn:observedProperty ?Property;
                  prop:observedInInterval ?Time;
                  prop:Temp ?Tempvalue.
                  ?Time time:hasBeginning  ?begins.
                  ?begins time:inXSDDateTime ?dateTime;
                  time:inDateTime ?DataTimeInfo.
                  ?DataTimeInfo time:dayOfWeek ?day;
                  time:month ?month;
                  time:timeZone ?zone.
                  FILTER ( ?dateTime <"2017-01-26T23:29:58"^^xsd:dateTime ||
?dateTime>"2016-01-26T23:29:58"^^xsd:dateTime)
                  } GROUP BY ?month limit 50
```

Fig. 2. Query to fetch Temperature Sensor data over a specific interval



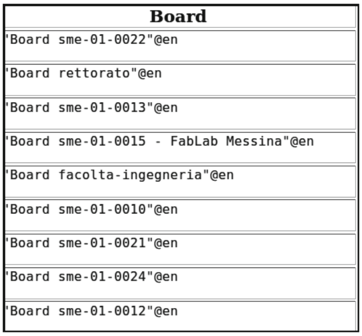
BoardName	latitude	longitude	altitude	dateTime	day	Tempvalue
"Board sme-01-0019"@en	"15.51349528"	"38.16164638"	"142"	2016-04-22T17:29:52	"Tuesday"	16.83293676314554
"Board sme-01-0019"@en	"15.51349528"	"38.16164638"	"142"	2016-11-23T17:33:17	"Wednesday"	17.440052699127705
"Board dip-scienze-co.psi.ped.cul"@en	"15.55599"	"38.19941"	"0"	2016-11-30T17:47:40	"Wednesday"	12.401368164919631
"Board dip-scienze-co.psi.ped.cul"@en	"15.55599"	"38.19941"	"0"	2016-12-01T10:14:59	"Thursday"	14.542305815905252
"Board dip-scienze-co.psi.ped.cul"@en	"15.55599"	"38.19941"	"0"	2016-12-01T10:25:03	"Thursday"	14.542305815905252
"Board dip-scienze-co.psi.ped.cul"@en	"15.55599"	"38.19941"	"0"	2016-12-02T10:39:58	"Friday"	15.635647177789792
"Board dip-scienze-co.psi.ped.cul"@en	"15.55599"	"38.19941"	"0"	2016-12-02T12:00:21	"Friday"	16.919307735293557
"Board dip-scienze-co.psi.ped.cul"@en	"15.55599"	"38.19941"	"0"	2016-12-02T12:20:29	"Friday"	17.789686784575963
"Board dip-scienze-co.psi.ped.cul"@en	"15.55599"	"38.19941"	"0"	2016-12-01T14:46:42	"Thursday"	17.702088493670885
"Board dip-scienze-co.psi.ped.cul"@en	"15.55599"	"38.19941"	"0"	2016-12-02T00:50:20	"Friday"	14.542305815905252
"Board dip-scienze-co.psi.ped.cul"@en	"15.55599"	"38.19941"	"0"	2016-12-03T03:05:48	"Saturday"	14.876948323266106
"Board dip-scienze-co.psi.ped.cul"@en	"15.55599"	"38.19941"	"0"	2016-12-03T03:56:06	"Saturday"	14.209161274117264

Fig. 3. Result of query shown in Fig. 2



```
PREFIX smartMe: <http://smartMe.linkeddata.org/ontology/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX prop: <http://smartMe.linkeddata.org/ontology/>
PREFIX  ssn: <http://purl.oclc.org/NET/ssnx/ssn#>

select ?Board      {?SensorID  prop:model "HIH-4030131 Series"^^xsd:string;
                    prop:manuafacturer "Honeywell"^^xsd:string;
                    ssn:onPlatform ?BoardID.
                    ?BoardID   prop:manuafacturer "Arduino";
                    prop:model "Yun";
                    rdfs:label ?Board.}
```

Fig. 4. Query to list of specific boards hosting specific kind of sensors



Board
'Board sme-01-0022"@en
'Board rettorato"@en
'Board sme-01-0013"@en
'Board sme-01-0015 - FabLab Messina"@en
'Board facolta-ingegneria"@en
'Board sme-01-0010"@en
'Board sme-01-0021"@en
'Board sme-01-0024"@en
'Board sme-01-0012"@en

Fig. 5. Result of query shown in Fig. 4

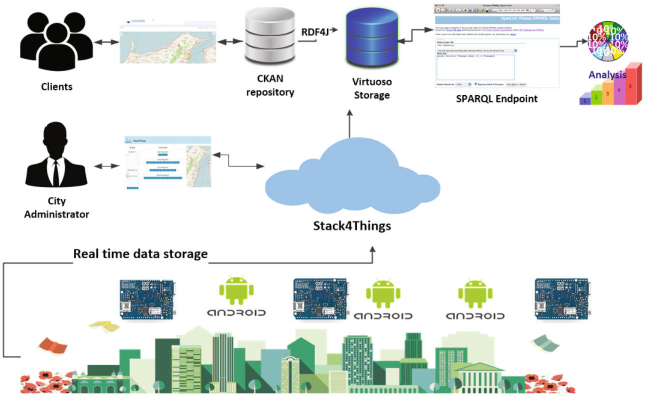


Fig. 6. SmartMe workflow description

4 Sensors Correlation Analysis

In this section we describe the correlation analysis among the sensors deployed in the various areas of Messina under the #SmartME project. Our main aim for this analysis is to check, whether the correlation between sensors can assist for predicting the value of unavailable sensor. More specifically, this phenomena can help for building a cost effective Smart city, where huge number of sensors need to be deployed.

The sensors that are important for our analysis are Temperature, Humidity, Pressure. To check the correlation of Temperature sensor with the environmental features captured parallel, we define "Temperature Value" as the predicted attribute (i.e., class label). The attributes for mining are as follows:

- Temperature Value
- Humidity Value
- Pressure Value
- Day of Week
- Date
- Month
- Year
- Hour
- Minute
- Second
- Zone
- Latitude
- Longitude
- Altitude

Random Forest and Linear Regression have been used for checking the accuracy of the model. We have chosen these two algorithms because of their ability to work with numeric class attribute. The algorithms model the data captured by the sensors for predicting the Temperature sensor values. The accuracy/error results with # of Instances has been shown in Table 1. The prediction has been made on the monthly captured different files. It shows the precision of prediction for different subsets of dataset. The correlation column shows the closeness of actual and predicted Temperature sensor's value. Fourth and fifth columns represent Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) for predicting Temperature Value. Thus, we can say that Random Forest algorithm is independent of number of Instances and worked better then Linear Regression algorithm. The results illustrate that it is feasible to predict a sensor, given the other feature variables i.e. different sensors, specific location (latitude, longitude, altitude, zone), date (Date, Month, Year) and time (Hour, Minute, Second).

We also perform feature selection method to know the important features in the observed data with the "Temperature Value" as a class label. The Correlation-based Feature Selection (CFS) algorithm [3] of feature selection with Greedy Ranker has been used with various subset of datasets. We found for 70% of subsets the algorithm ranked the attributes in following order: Humidity, Latitude, Longitude, Hour, Zone, Altitude, Date, Pressure, Second, Month, Minute, Second, Day of Week. For example, Fig. 7 shows the relationship of Temperature with Humidity & Pressure sensors for a subset of dataset. It shows Humidity is negatively correlated to Temperature while Pressure is constant with varying Temperature. This analysis shows the correlation between different sensors can be exploited for predicting an unavailable(undeployed, failed, broken) sensor.

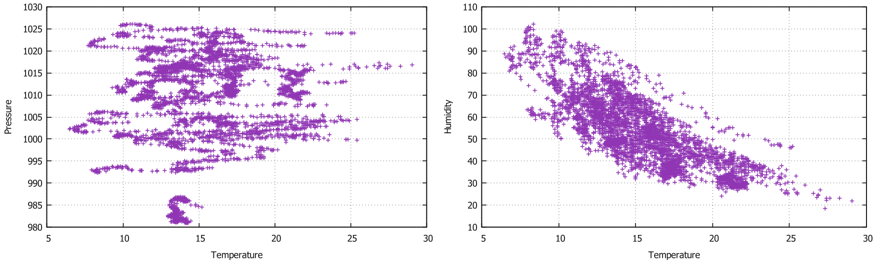


Fig. 7. SmartMe temperature and pressure, humidity correlation

Table 1. Analysis on different instances where, # of attributes = 13, Class label = Temperature value

Algorithm	Instances	Correlation	MAE	RMSE
Linear regression	4717	0.8963	1.2638	1.6148
Random forest		0.9943	0.2613	0.3956
Linear regression	4987	0.8835	1.3347	1.7247
Random forest		0.9986	0.185	0.307
Linear regression	7801	0.8354	2.5056	3.1417
Random forest		0.9948	0.4156	0.606
Linear regression	10000	0.8013	1.2437	1.5647
Random forest		0.9786	0.176	0.214
Linear regression	15233	0.9269	0.9901	1.2892
Random forest		0.9969	0.1849	0.2727

This scenario can help in the situation where buying large number of specific sensors would be too costly. Thus, the substitution of a certain subset by other low costly but related sensors would be beneficial in terms of cost effectiveness. It would also help to diagnose the problematic sensor that behaves incorrectly in presence of other sensors.

5 Conclusion and Future Work

We proposed a novel approach for converting real-time as well as stored sensor data into RDF format. The #SmartME project infrastructure has been exploited to enable real-time sensor data conversion and storage. The semantic conversion annotates the raw sensor data, provides a unified model for sensors and observations description, and also assists application developer in terms of deep data acquisition as well as quick application building. Also, we observed that correlation between sensors can be leveraged for prediction of data from unavailable sensor. For sensor data prediction we used a data mining tool and analysed the correlation between two sensors and also some environmental parameters. We found that we can rely on the sensor as well as environmental parameters to predict values of specific sensors. In this paper we report some preliminary results of the outgoing work.

References

1. Bruneo, D., Distefano, S., Longo, F., Merlino, G.: An IOT testbed for the software defined city vision: the # smartme project. In: 2016 IEEE International Conference on Smart Computing (SMARTCOMP), pp. 1–6. IEEE (2016)

2. Consoli, S., Mongiovic, M., Nuzzolese, A.G., Peroni, S., Presutti, V., Reforgiato Recupero, D., Spampinato, D.: A smart city data model based on semantics best practice and principles. In: Proceedings of the 24th International Conference on World Wide Web, pp. 1395–1400. ACM (2015)
3. Han, J.: Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers Inc., San Francisco (2005)
4. Kojima, K.: Prediction of individual thermal sensation using unspecified sensors in sensor networks. In: International Conference on Control, Automation and Systems, ICCAS 2008, pp. 123–126. IEEE (2008)
5. Longo, F., Bruneo, D., Distefano, S., Merlino, G., Puliafito, A.: Stack4things: a sensing-and-actuation-as-a-service framework for IOT and cloud integration. Ann. Telecommun. **72**, 1–18 (2016)
6. Maddox, T.: Smart cities: 6 essential technologies (2016). http://www.techrepublic.com/article/smart-cities-6-essential-technologies/. Accessed 26 Mar 2017
7. Neuhaus, H., Compton, M.: The semantic sensor network ontology. In: AGILE Workshop on Challenges in Geospatial Data Harmonisation, Hannover, Germany, pp. 1–33 (2009)
8. Pfisterer, D., Romer, K., Bimschas, D., Kleine, O., Mietz, R., Truong, C., Hasemann, H., Kröller, A., Pagel, M., Hauswirth, M., et al.: Spitfire: toward a semantic web of things. IEEE Commun. Mag. **49**(11), 40–48 (2011)
9. Ploennigs, J., Schumann, A., Lécué, F.: Adapting semantic sensor networks for smart building diagnosis. In: Mika, P., Tudorache, T., Bernstein, A., Welty, C., Knoblock, C., Vrandečić, D., Groth, P., Noy, N., Janowicz, K., Goble, C. (eds.) ISWC 2014. LNCS, vol. 8797, pp. 308–323. Springer, Cham (2014). doi:10.1007/978-3-319-11915-1_20
10. Ploennigs, J., Schumann, A., Lecue, F.: Extending semantic sensor networks for automatically tackling smart building problems. In: Proceedings of the Twenty-first European Conference on Artificial Intelligence, pp. 1211–1212. IOS Press (2014)
11. Sheth, A., Henson, C., Sahoo, S.S.: Semantic sensor web. IEEE Internet Comput. **12**(4), 78–83 (2008)
12. Wei, W., Barnaghi, P.: Semantic annotation and reasoning for sensor data. In: European Conference on Smart Sensing and Context, pp. 66–76. Springer (2009)