Weather Data Analysis and Sensor Fault Detection Using An Extended IoT Framework with Semantics, Big Data, and Machine Learning

Aras Can Onal*, Omer Berat Sezer*, Murat Ozbayoglu*, Erdogan Dogdu†
* Department of Computer Engineering
TOBB University of Economics and Technology
Ankara, Turkey 06560
Email: {arascanonal, oberatsezer, mozbayoglu}@etu.edu.tr
† Department of Computer Engineering
Cankaya University, Ankara, Turkey 06790
Georgia State University (adj.), Atlanta, GA, USA 30302
Email: {edogdu}@cankaya.edu.tr

***Abstract*—In recent years, big data and Internet of Things (IoT) implementations started getting more attention. Researchers focused on developing big data analytics solutions using machine learning models. Machine learning is a rising trend in this field due to its ability to extract hidden features and patterns even in highly complex datasets. In this study, we used our Big Data IoT Framework in a weather data analysis use case. We implemented weather clustering and sensor anomaly detection using a publicly available dataset. We provided the implementation details of each framework layer (acquisition, ETL, data processing, learning and decision) for this particular use case. Our chosen learning model within the library is Scikit-Learn based k-means clustering. The data analysis results indicate that it is possible to extract meaningful information from a relatively complex dataset using our framework.**

***Keywords*—Internet of things; machine learning; framework; big data analytics; weather data analysis; anomaly detection; fault detection; clustering**

I. Introduction

Due to the advancements in sensor technologies, embedded microcomputers and high-speed communication networks, big data analytics (both batch and streaming) applications are becoming more and more common. For effective utilization of such applications, frameworks integrating big data, IoT and machine loftware earning are proposed [20]. Even though these fields are still in their emerging stages, integration is implemented through some tools or libraries [13], [14], [11]. These efforts eventually will provide easier solutions for data analysis implementations on various problems such as intelligent transportation systems, financial applications, smart grids, environment monitoring, health, home automation, etc. [8].

In this study, we present a weather clustering model implemented using our Big Data IoT Framework. The implementation details are provided for each layer, hence other data analytics problems can be handled using the framework in a similar fashion.

The rest of the paper is structured as follows. The related work is presented in Section II. The overall framework infrastructure is explained in Section III. The weather data used in this study and the suggested analysis are presented in Section IV along with how the weather data analysis problem is solved using the framework. The results of clustering analysis are presented and discussed in Section V. We conclude and give future research directions in Section VI.

II. Related Work

With the current developments in new technologies, IoT sensors and devices are widely used in daily life. Aviation, communication, environment monitoring, health, home automation, traffic, vehicle communication [8] are the examples of usage areas of IoT sensors and devices. It is stated that the number of connected electronic devices to the internet will reach 50 to 100 billions [20], [15].

The foresight of the increasing use of IoT devices suggest that new frameworks and solutions for IoT devices are needed and also developed by organizations, companies and the academic world. There are lots of IoT frameworks and platforms that are being used in our daily lives for collecting and analyzing data. These are reviewed and surveyed in several works [13], [14], [11]. Some of popular frameworks include: AllJoyn, AirVantage, Brillo Carriots, Devicehub.net, EvryThng, Ericsson IoT-Framework, IoTivity, Intel IoT Platforms, LinkSmart, OpenIoT, OpenMTC, OpenRemote, Pentaho, Platform.io, realTime.io, SensorCloud, SkySpark, Statistica (Dell), Tellient, The Thing System, ThingSpeak, ThingSquare, ThingWorx, IBM Watson IoT Platform, Zetta.

Although many IoT frameworks and solutions are proposed and developed, there are still problems and missing features in this area. Interoperability, storing large datasets and implementing learning algorithms are the examples of such issues that can be improved. To cope with the interoperability problem, semantic Web solutions are used, such as modeling data using Resource Description Framework (RDF), Web Ontology Language (OWL)[1] protocols, using standard data formats such as Turtle, N3, and JSON-LD. In addition, in literature,

[1] http://www.w3.org/TR/owl2-overview/

978-1-5386-2715-0/17/$31.00 ©2017 IEEE

common vocabularies to represent knowledge are developed to support semantificiation. IoT-related ontologies are proposed and developed to provide a common language to express things, relationships towards higher levels of interoperability. In IoT world, Semantic Sensor Network (SSN)[2] Ontology is developed to express and model sensors, their properties and observations. There are also extended SSN Ontologies that are used for different IoT domains (e.g. smart home automation [19]). In addition, these proposed extended SSN ontologies support different features namely IoT Resources, IoT Services, Observation and Measurement, Physical Locations [22], IoT sensors' semantic definition, sensor discovery, and service infrastructure [12][6]. Besides, there are newly developed semantic IoT frameworks, middlewares and platforms in the literature [1][5][21][20].

Machine learning algorithms are also frequently used in IoT frameworks. Most of the frameworks have low level (data, device, energy, security, etc.) management functionalities. However, IoT devices can be smarter with the implementation of machine learning algorithms. These algorithms are mainly divided into three categories: supervised, unsupervised, and reinforcement learning. Supervised learning is a learning method that can be implemented if the data is labelled. Naive Bayes, support vector machines, decision trees, artificial neural networks, linear regression are the examples of the supervised learning methods. Unsupervised learning is another learning method that works without labelled data. Data is clustered using unsupervised learning methods. Unsupervised learning algorithms provide the detection of anomalies with the clustering advantage. K-means, fuzzy clustering, DBSCAN (density-based spatial clustering of applications with noise) are the examples of the unsupervised learning algorithms. Unsupervised learning methods can easily be used for sensor fault and anomaly detection in IoT world. Reinforcement learning is the third type of learning method that is used in control, simulation-based solutions, games, etc. It has a feedback signal to optimize the output of the model.

IoT sensors are used everywhere in all sorts of applications. Weather monitoring and forecasting a widely used IoT application area that can be further utilized in application areas such traffic monitoring and management [16][2], agriculture and food production [9], travel planning, smart cities [23][17] and smart grids, social life and many areas of our daily lives [4]. Earlier approaches utilized simple statistical methods to predict, for example agricultural production using weather data [9], but new research use more big data solutions and machine learning methods for prediction using weather and other IoT sensor data. Chin et al. [3] have shown the correlation between weather-based conditions and short-cycling behaviour using big data collected from IoT weather sensors.

IBM Bluemix framework provides a Weather API (Application Programming Interface)[3], which utilizes IoT and big data analytics to serve weather information via API calls. IBM Bluemix is also capable of producing weather alerts, daily forecasts, historical data.

Due to the wide application areas of IoT weather and other sensors, fault and anomaly detection is an important problem in especially wide-scale deployments of sensors [7][18][10]. Here we attempt to detect anomalies in weather sensors using a big data and deep learning method.

III. Proposed IoT Framework

Our previously proposed "An Extended IoT Framework with Semantics, Big Data, and Analytics" [20] framework consists of five layers, and these are data acquisition, extract-transform-load (ETL), semantics, learning and action layers that can be seen at Figure 1. The data acquisition layer is responsible for acquiring data from sensors, or datasets can be bulk-loaded to the framework. This layer is not responsible for parsing or evaluating data. After acquiring data, the data acquisition layer conveys data to the ETL layer. The ETL layer parses data, produces semantically annotated data in RDF (Resource Description Framework) format. The RDF formatted file is then used for semantic rule reasoning and the data objects are later used in learning layer. For big data storage a NoSql database is used to store sensor data. Semantic rule reasoning layer processes RDF data according the rules and produces inferred data. A CSV (Comma Separated Values) file containing resulting IoT data is transferred to the learning layer. The learning layer preprocesses the incoming data, extracts relevant features and applies machine learning methods. Machine learning algorithms are then run in parallel in distributed computer servers to process big data in a faster way. The last layer is action layer, which is responsible for evaluating the results produced in learning layer.

In this work, we employ the following steps to process weather observation data. In the data acquisition layer, LinkedSensorData and LinkedObservationData are acquired via bulk loading method. Then, the acquired data is directly transferred to ETL layer for further processing. In ETL layer, sensor data and observation data values are parsed with using regular expression, features and their values are grouped by their time frame and CSV formatted file is produced for further usage. The CSV file is produced by using Node.js. After producing file, Python script which is written for clustering is called by Node server. This script reads previously created file and the results are produced. Since the dataset does not have label, k-means unsupervised clustering method is employed in the learning layer. Since the semantic rule reasoning and action layer are currently under development, rule reasoning isn't used in this scenario. Python-shell is used for running Python script[4]. As the development continues, creating CSV file phase will be changed. Rather than creating CSV file, data object that stores parsing values will be directly conveyed to script.

IV. Use-Case Scenario

In this use-case scenario, we used LinkedSensorData and LinkedObservationData[5] which contain different weather sensors such as air temperature, wind speed, relative humidity, pressure, visibility, etc. LinkedSensorData is a RDF dataset that defines approximately 8000 weather sensors information (ex: latitude, longitude, type, etc.) in United States. LinkedObservationData is also a RDF dataset that contains values

[2]SSN Ontology, https://www.w3.org/2005/Incubator/ssn/ssnx/ssn

[3]https://developer.ibm.com/recipes/tutorials/forecast-weather-data-from-ibm-bluemix-weather-apis/

[4]https://github.com/extrabacon/python-shell

[5]http://wiki.knoesis.org/index.php/LinkedSensorData

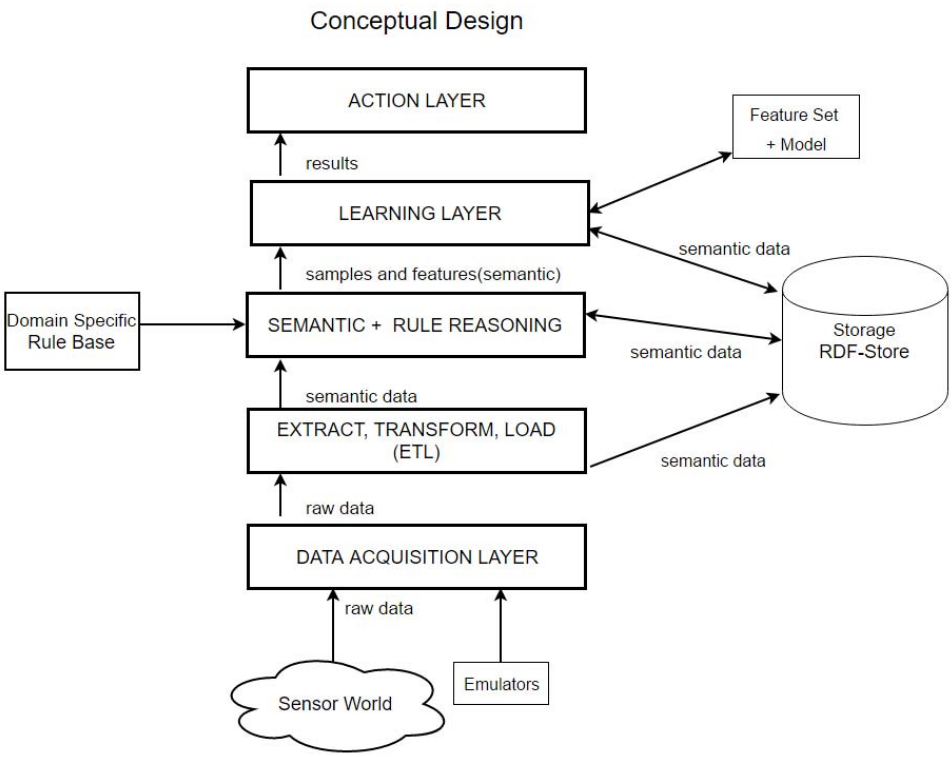

Fig. 1. Conceptual design of our extended IoT framework [20]

of different types of sensors which are defined in LinkedSensorData. In our scenario, we used the sensor data of August 29, 2005.

As mentioned in Section III, proposed conceptual extended IoT framework is used to implement the use-case scenario. In this scenario, there are different phases that show the task of the layers in the framework. Figure 2 illustrates these phases namely: Data acquisition phase, ETL (Extraction, Transform, Load) and semantic phase, and learning phase.

Each dataset contains two different n3 files which contain LinkedSensorData and LinkedObservationData. Besides, each sensor file contains different chunk of feature set. LinkedSensorData is used in order to obtain sensor location, altitude, latitude and longitude. LinkedObservationData is used for obtaining observation values which are pressure, relative humidity, air temperature, dew point, wind direction, wind gust, wind speed and visibility.

Since LinkedObservationData contains lots of irrelevant structure, regular expressions are employed to fetch the desired chunk of data from the file. As a first step, LinkedSensorData is processed with the help of regular expression and stored the location information in the location map created for this task. Then, a class structure is created to hold all sensor features and stored location information for each different sensor types. As a second step, LinkedObservationData is processed, and again with regular expression, the observation values are fetched and stored these values in a class instance which is mentioned above. CSV file creation strategy is centered around the acquisition time of the sensor value and the sensor name. The file is scanned for the different time that sensor values are taken, then the values are grouped by their time information and row is created. Since each sensor file can contain different chunk of feature set(some have windspeed, some have not for example), N/A is set for not existent feature information. In short, conversion code is developed that can give sensor values with desired time frame such as 18.00 PM-19.00 PM and CSV file is created.

An example LinkedSensorData which belongs to 3CLO3 sensor can be seen in following snippet:

```
sens-obs:point_3CLO3 a wgs84:Point ;
      wgs84:alt "20"^^xsd:float ;
      wgs84:lat "46.22"^^xsd:float ;
      wgs84:long "-124.00"^^xsd:float .
```

Sensor name, altitude, latitude and longitude values can be obtained easily from this RDF notation. Similary, LinkedObservationData uses the same notation:

```
sens-obs:
      MeasureData_WindSpeed_3CLO32005823_172000
      a om-owl:MeasureData;
      om-owl:floatValue "300.0"^^xsd:float ;
      om-owl:uom weather:degrees .
```

In this example, the feature name is MeasureData, it is taken at 23 August 2005, 17:20 PM. Its value is 300. All other features uses same notation, therefore it can be obtained easily with regex, which we used for parsing process.

In the learning phase, hidden data can be extracted and machine learning methods are used to get further information. In this scenario, weather sensors are needed to process to get further information. Sensor data is not labelled by sensors.

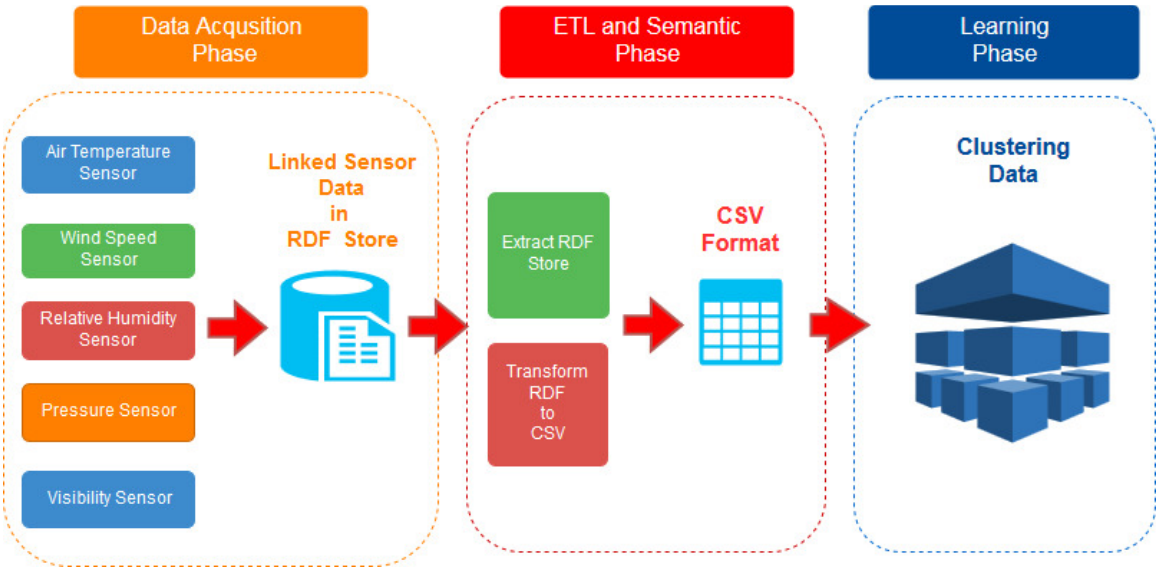

Fig. 2. Use-case Scenario of our proposed IoT Framework

So, we aim to cluster data to detect hidden information. In our approach, we used K-Means clustering algorithm to find general pattern of the data, sensor faults and sensor anomalies. Learning algorithm is implemented in Python[6] and Scikit-learn[7], Pandas[8], Numpy[9] and Matplotlib[10] library and frameworks are used. Generalized proposed learning algorithm is illustrated in Algorithm 1. The machine learning implementation of the proposed model is available in GitHub[11] can be downloaded from [12].

V. Evaluation

We performed data clustering using the aforementioned publicly available open weather data. We limited the number of clusters to 2,3 and 4 mainly due to easier interpretation of the results. When more than 4 clusters are used, the regions started disappearing, so we chose the maximum k value as 4. In the next subsections, more explanation is provided for individual feature selections for different clustering scenarios.

A. Air Temperature - Humidity and Wind Speed Characteristics:

Data clustering results represent a good distinction between the geographical regions of continental USA, i.e. The Rocky Mountains clearly provide a good borderline between the two distinct weather data clusters. The interesting point is the model creates almost 2 identical sized cluster blocks (one cluster has 3687 data points, the other has 3319). The data is geographically divided almost equally as can be seen from Figure 3. The region that is east of Rocky Mountains is a few degrees cooler, but considerably more humid compared

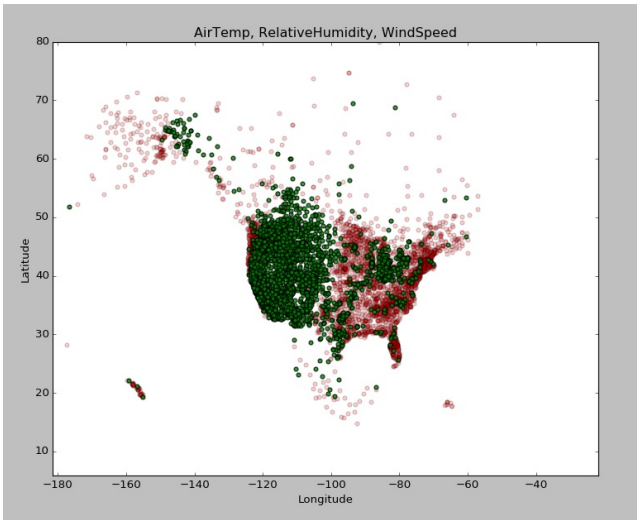

Fig. 3. AirTemp, Relative Humidity, WindSpeed 2 Clusters

to the region that is west of the Rocky Mountains. The most distinctive feature seems to be the relative humidity, as such the temperature is on average 8 degrees warmer (or hotter) in the West region, but the relative humidity is considerably lower (23% to 70%) that might compensate for the higher temperature. We need to take into consideration that the weather snapshot is taken in August 29, one of the warmest times of the year.

Using 2 clusters, the geographical distinction is clear (see Figure 3). Meanwhile when we perform 3 clustering (Figure 4), we also face interesting results, such that similarities start occurring between east and west coast lines, whereas the mountain and plateau region between the Rockies and West Coast Line remains as a unique region. However, the Midwest and the Plains becomes as a transition region between the Rockies and East Coast. We can also observe similar characteristics on the West Coast, the California and Pacific Northwest

[6]https://www.python.org/
[7]http://scikit-learn.org
[8]http://pandas.pydata.org/
[9]http://www.numpy.org/
[10]https://matplotlib.org/
[11]https://github.com/
[12]https://github.com/omerbsezer/IoTWeatherSensorsAnalysis

Algorithm 1 Generalized Pseudocode of IoT Framework Algorithm

```
procedure ALLPHASES()
    Extract, Transform, Load Phase:
    readLinkedSensorDataFileDirectory();
    locationMap = createLocationMapFromLinkedSensorDataFile() // getting altitude, longitude, latitude
    locationObject = fillLocationInformationFromLinkedSensorDataFile()
    locationMap.put(sensorName, locationObject)
    readLinkedObservationDataFileDirectory()
    sensorValueList = createSensorValueList()
    foreach(file)
        uniqueTimeFrames = findUniqueTimeFrames()
        createSensorValueList
        foreach(uniqueTimeFrame)
            sensorValue = fillSensorValueInformation()
            sensorValue.setLocation(locationMap.get(sensorName))
            sensorValueList.add(sensorValue)
    createCsvFileFromSensorValueList()
    Learning Phase:
    dataFrame = pandas.readcsv('file.csv')
    table = dataFrame.pivottable(index = ["Name"])
    table = table.dropna()
    if(numberOfSensorTypes == 1) :
        no need to normalize sensor data
    elif(numberOfSensorTypes == 2) :
        tableNormalized = normalize2values(table)
    elif(numberOfSensorTypes == 3) :
        tableNormalized = normalize3values(table)
    k = numberOfClusters
    cluster = sklearn.cluster.KMeans(nclusters = k)
    if(numberOfSensorTypes == 1) :
        table["Cluster"] = cluster.fitpredict(table[table.columns[:]])
    else :
        table["Cluster"] = cluster.fitpredict(tableNormalized[tableNormalized.columns[:]])
    plotResults() in terms of Clusters, Sensor Positions(Latitude, Longitude) using MatplotLib
    Evaluation:
    evaluateResults()
```

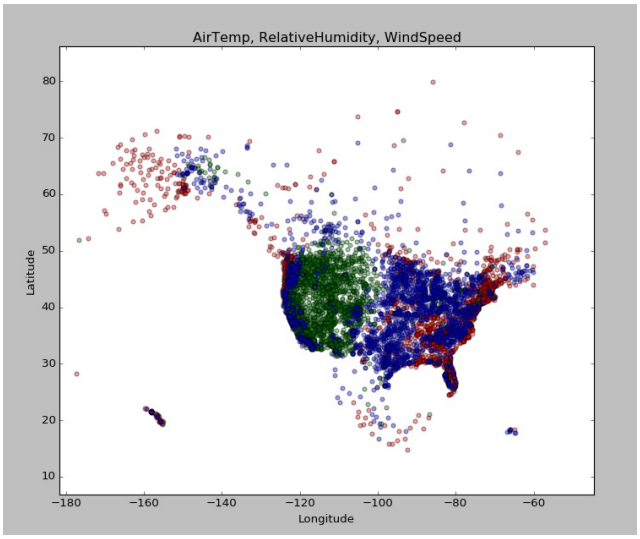

Fig. 4. AirTemp, Relative Humidity, WindSpeed 3 Clusters

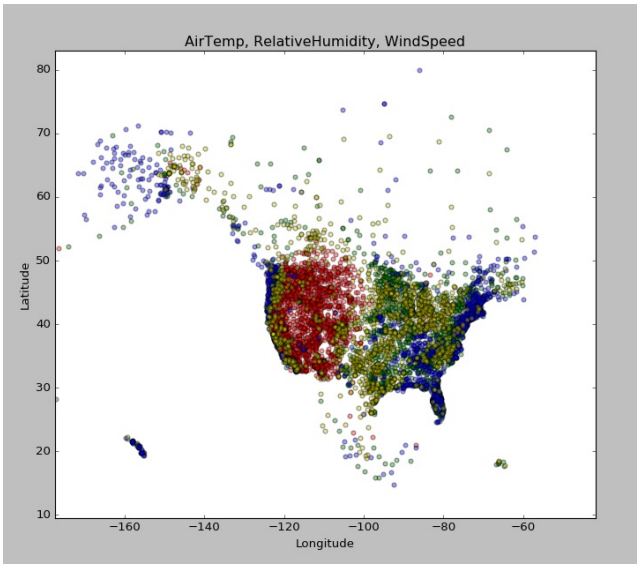

Fig. 5. AirTemp, Relative Humidity, WindSpeed 4 Clusters

Mountain ranges provide a transitive region just like East Coast. The distinctions between the temperatures and relative humidity follow the same properties as the 2-cluster case, this time the transition region has a temperature and relative humidity readings just in between the two aforementioned regions (Rocky Mountain range and the coastlines). Once again, wind speed seems to be not a representative factor when used along with other weather characteristics. It is also noteworthy to mention that the data distribution among the clusters is also relatively uniform just as in the 2-cluster case. Table I, Table II, and Table III show the sensor clustering results such as centroid points, cluster color, cluster name, maximum and minimum points in each clusters.

Using a 4-cluster analysis (Figure 5) does not provide any more interesting outcome. The results are very similar to 3 cluster case, the region between Rocky Mountains and the west coast states remains to be one cluster chunk, whereas the transition regions between the inland and the coastlines are also visible from the Figure. The cluster data uniformity is also still intact.

B. Wind Speed - Relative Humidity

The analysis for 2-3-4 clustering for these 2 features is almost identical to Air Temperature - Humidity and Wind Speed analysis. This is due to relative humidity being the most important distinctive feature of the weather data. The geographical regions formed as a result of clustering show high similarities with the 3-feature combination case.

C. Air Temperature - Relative Humidity

We might assume the corresponding geographical regions for this analysis would be similar to Air Temperature - Humidity and Wind Speed analysis, however, interestingly there are some noticeable differences between the two sets of results. First of all, the regions are not equally divided as in the former case. The two weather clusters are distributed within each other geographically. Even though, it can be claimed that the North East Region, West Coast, and the Great Lakes region form one cluster and the rest of the continental USA belongs to the other cluster, some exceptions exist (Parts of Florida, parts of Midwest, parts of Southwest). Hence, it is not easy to characterize this particular clustering result geographically. However, since this is a single snapshot of data, different clustering results might be obtained some other time.

When we perform 3-cluster analysis (Figure 7), we observe some interesting phenomena. Even though the results look similar to 2-cluster case, the cluster that is formed in the region that is between Rocky Mountains and West Coast extends itself to the Plains up to Ohio valley (Figure 6). This is generally consistent with the summer temperature maps, Northern Plains represent a lower temperature center, surrounded by higher temperature regions formed in Northwest, Southwest and Southern Plains. This higher temperature region generally follows a diagonal path between Northern Texas, Oklahoma, all the way up to the Great Lakes. A similar diagonal formation is visible in the Mississippi Valley all the way to the Atlantic Ocean through the Carolinas. However, strange enough, when we divide the data into 4-clusters, this phenomena is not visible (Figure 8). This result indicates the importance of choosing the

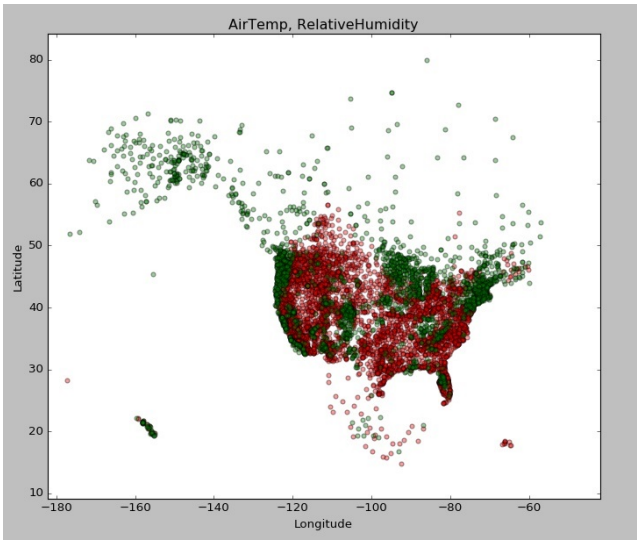

Fig. 6. Air Temperature Relative Humidity 2 Clusters

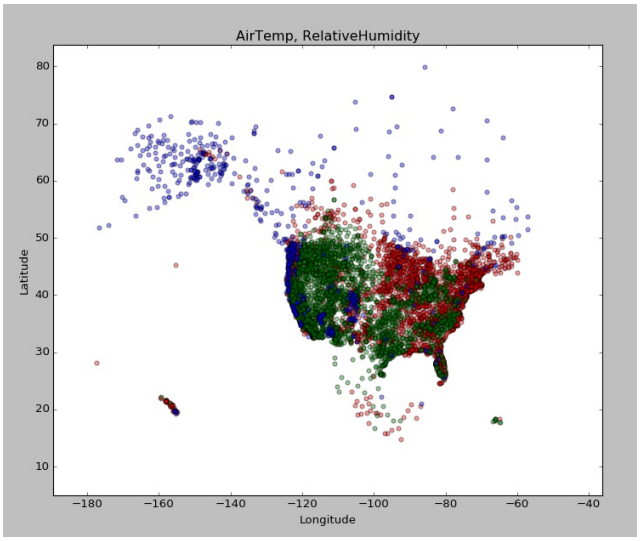

Fig. 7. Air Temperature Relative Humidity 3 Clusters

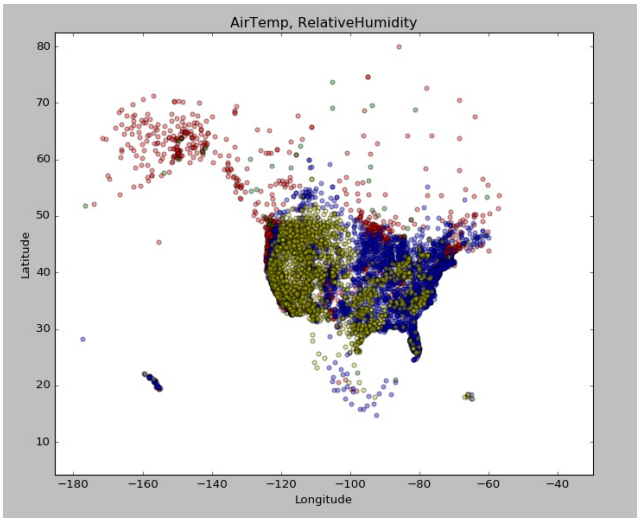

Fig. 8. Air Temperature Relative Humidity 4 Clusters

TABLE I. 1-TYPE SENSORS CLUSTERING RESULTS

Figure / Sensor Name	Cluster	Cluster Color	Cluster Size	Centroids	Maximum	Minimum
AirTempC2	C0	Red	2373	62.56	73.6	-60.3
	C1	Green	6047	84.57	151.0	73.6
AirTempC3	C0	Red	4364	87.56	151.0	79.7
	C1	Green	500	43.96	57.9	-60.3
	C2	Blue	3556	71.86	79.7	58.0
AirTempC4	C0	Red	2716	90.84	151.0	84.6
	C1	Green	3931	78.33	84.6	70.3
	C2	Blue	107	15.44	38.7	-60.3
	C3	Yellow	1666	62.19	70.3	39.2
PressureC2	C0	Red	165	24.97	26.9	17.9
	C1	Green	197	28.87	29.9	26.9
PressureC3	C0	Red	167	29.16	29.9	27.6
	C1	Green	151	26.02	27.5	24.6
	C2	Blue	44	22.91	24.4	17.9
PressureC4	C0	Red	103	24.94	25.9	23.2
	C1	Green	155	29.27	29.9	28.1
	C2	Blue	87	26.89	28.0	25.9
	C3	Yellow	17	21.29	22.9	17.9
WindSpeedC2	C0	Red	5551	3.055	7.1	0.0
	C1	Green	1869	11.26	100.0	7.2
WindSpeedC3	C0	Red	767	14.84	100.0	10.9
	C1	Green	3665	1.64	4.3	0.0
	C2	Blue	2988	6.91	10.8	4.3
WindSpeedC4	C0	Red	3271	1.34	3.9	0.0
	C1	Green	784	14.52	44.3	10.6
	C2	Blue	3363	6.54	10.5	4.0
	C3	Yellow	2	100.0	100.0	100.0
RelativeHumidityC2	C0	Red	3431	23.15	46.8	0.0
	C1	Green	3813	70.52	655.0	47.0
RelativeHumidityC3	C0	Red	3814	70.03	148.2	46.7
	C1	Green	3427	23.12	46.5	0.0
	C2	Blue	3	655.0	655.0	655.0
RelativeHumidityC4	C0	Red	2482	52.17	66.8	35.3
	C1	Green	2720	18.42	35.3	0.0
	C2	Blue	2039	81.77	148.2	67.0
	C3	Yellow	3	655.0	655.0	655.0

correct number of clusters in the analysis. The difference in the clustering results of 3 and 4 clusters is quite remarkable.

D. Visibility

When visibility is used as the sole distinctive feature, the 2-3-4 clustering from the data did not provide meaningful results. This might be as a result of either the lack of importance of visibility as a feature for weather analysis, or the particular snapshot taken on August 29 2005 was not enough. It is possible to get better results, or at least some distinctive characteristics if the snapshot was taken in different seasons, i.e. visibility might be a more important factor during winter time or rainy seasons.

E. Pressure

When we observe the pressure data, it is seen that most of the sensors had missing values, hence it was more difficult to analyze the results. However, even with the lack of data, the existing results are still consistent with the geographical regions formed through other feature selections. (Figure 9 - only 2-cluster is enough)

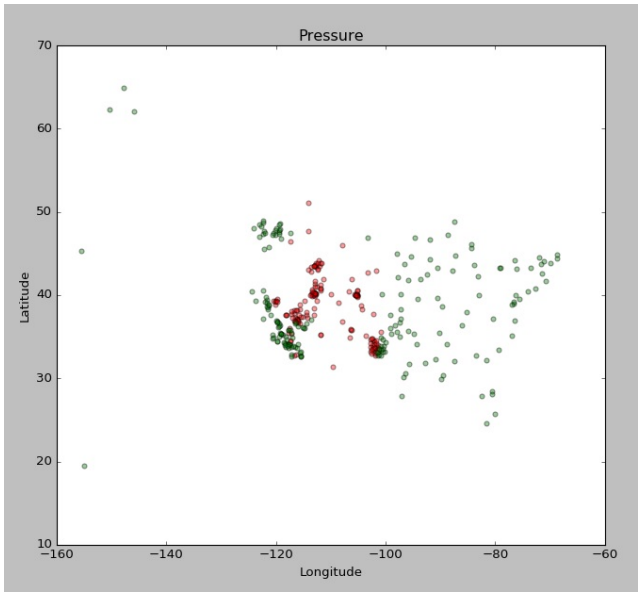

Fig. 9. Pressure 2 Clusters

F. Air Temperature

Using only air temperature as the feature for clustering results in an interesting outcome if we observe the 2-cluster case (Figure 10). It divides the North America (where the sensors are located) geographically that is consistent with the country borders of USA and Canada. Even though there are some pockets of other cluster data within the continental USA cluster, it still represents a complete block. The clustering drastically changes for 3-cluster representation (Figure 11). The analysis is similar to the 3-cluster formation in Air Temperature - Relative Humidity case. (Figure 7)

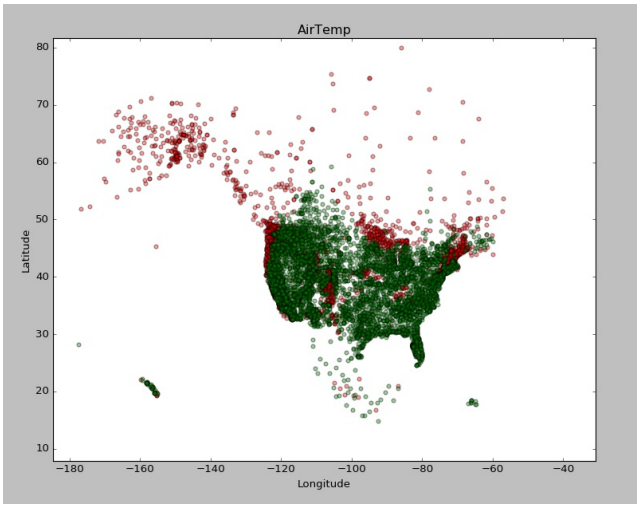

Fig. 10. Air Temperature 2 Clusters

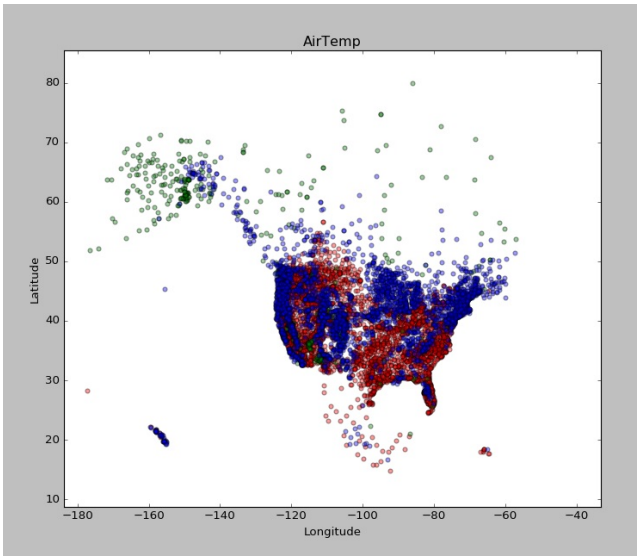

Fig. 11. Air Temperature 3 Clusters

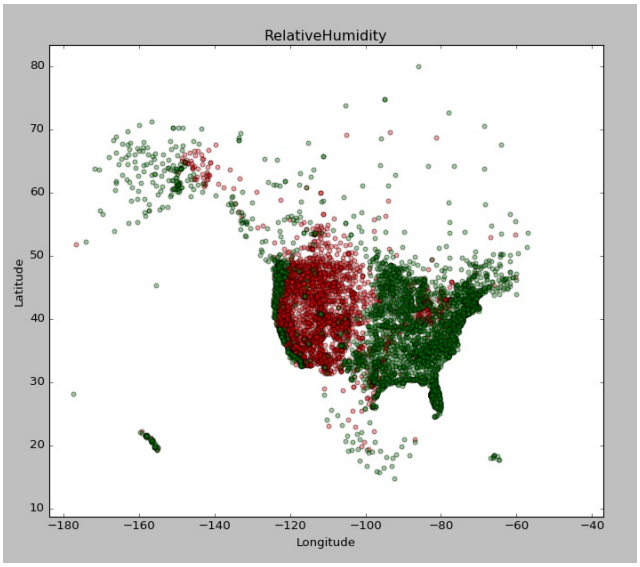

Fig. 12. Relative Humidity 2 Clusters

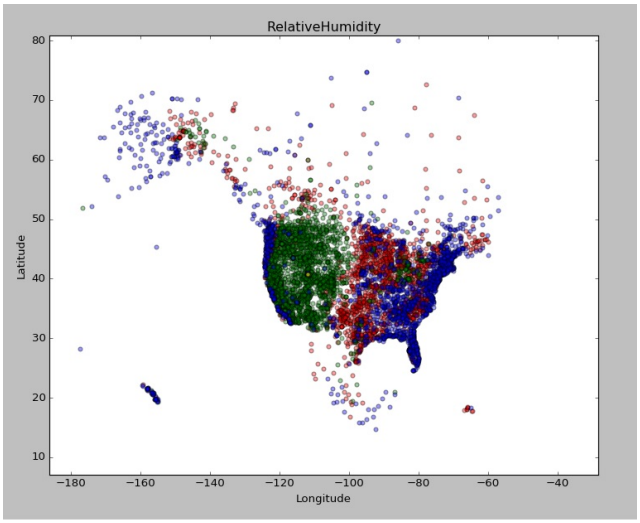

Fig. 13. Relative Humidity 4 Clusters

G. Wind Speed

It looks like, especially for the particular time snapshot taken, wind speed is not an important factor by itself for the weather data analysis. The 2 clusters formed do not represent a clear geographical distinction between them, in addition the centroid values are 3 mph and 11 mph, respectively for the lower and upper wind speed clusters which indicates an insignificant distinction between the two clusters.

H. Relative Humidity

Relative humidity, being probably the most distinctive feature of all, provides clustering results that are consistent with the case where all features are used together (Air Temperature - Relative Humidity and Wind Speed). This indicates that for most weather analyses (at least for the particular time of the season), humidity is a very distinctive factor. However some minor differences are also visible. Ohio valley and the narrow diagonal line between Texas and the Great Lakes belong to the same cluster that is to the west of Rockies, however that was not the case in the Air Temperature - Relative Humidity and Wind Speed analysis. Otherwise, the two analyses are very similar. This is applicable to 2-3-4 cluster cases (Figure 12, Figure 13).

I. Sensor Fault Analysis

We used weather data clustering also for faulty sensor detection. Without loss of generalization, when we performed clustering analysis on the wind speed and relative humidity alone, some sensor data results were substantially different from their neighboring regions. We provided an example use case for detecting the faulty sensors (Figure 14, Figure 15). In that particular case the faulty sensors form separate clusters, far away from the other data points. One important observation is this: for these faulty sensors the number of clusters mostly

TABLE II. 2-TYPE SENSORS CLUSTERING RESULTS

Figure Name	Cluster	Cluster Color	Cluster Size	Relative Humidity Centroids	Air Temp Centroids
AirTempRelativeHumidityC2	C0	Red	4822	85.72	38.65
	C1	Green	2390	66.27	66.36
AirTempRelativeHumidityC3	C0	Red	3412	76.14	63.51
	C1	Green	3156	88.39	27.87
	C2	Blue	644	50.99	63.20
AirTempRelativeHumidityC4	C0	Red	1222	62.32	69.03
	C1	Green	78	13.84	21.70
	C2	Blue	3300	79.02	59.09
	C3	Yellow	2612	89.48	24.47

TABLE III. 3-TYPE SENSORS CLUSTERING RESULTS

Figure Name	Cluster	Cluster Color	Cluster Size	Air Temp Centroids	Relative Humidity Centroids	Wind Speed Centroids
AirTempRelativeHumidityWindSpeedC2	C0	Red	3687	75.87	70.02	4.65
	C1	Green	3319	83.26	23.34	5.59
AirTempRelativeHumidityWindSpeedC3	C0	Red	1932	72.85	81.98	4.47
	C1	Green	2612	84.06	18.51	5.67
	C2	Blue	2462	79.51	52.35	4.98
AirTempRelativeHumidityWindSpeedC4	C0	Red	2358	84.70	16.98	5.79
	C1	Green	1757	77.86	65.69	4.62
	C2	Blue	1196	70.60	88.15	4.44
	C3	Yellow	1695	79.71	44.14	5.09

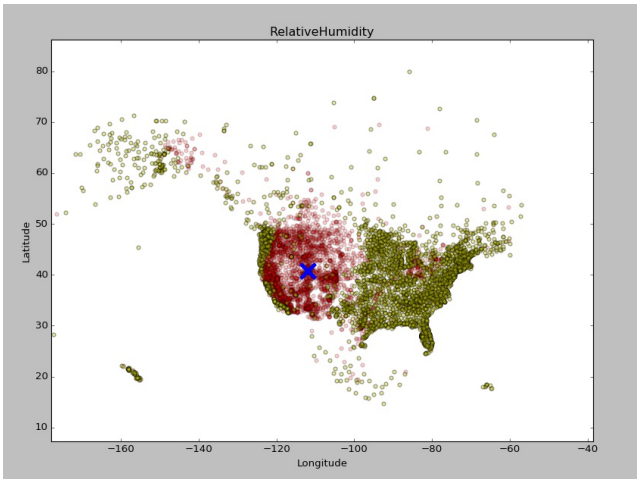

Fig. 14. Relative Humidity 3 Clusters, Sensor Fault

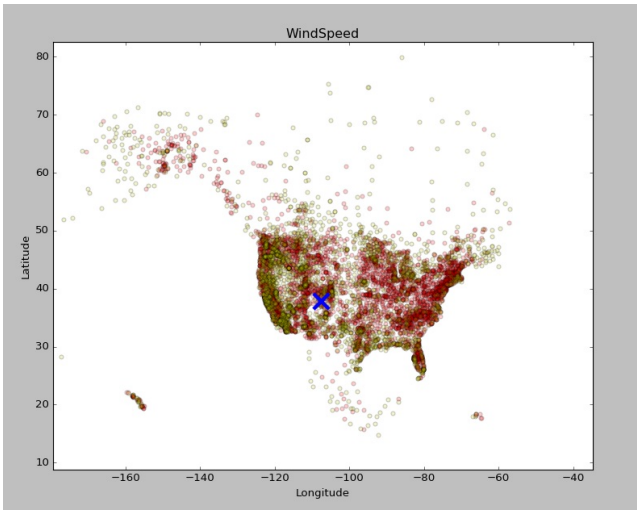

Fig. 15. Wind Speed 3 Clusters, Sensor Fault

do not prevent the detection of the faulty sensor as such, they always form a separate cluster, since their data readings are very different from the others.

Even though we used wind speed and relative humidity as the sole feature for anomaly detection, it is possible to use other features, or the combination of features to detect faulty sensors and abnormal data. More analyses might be needed for a more comprehensive model (e.g. time series sensor data might be analyzed in a sequence).

VI. CONCLUSION

In this study an extended IoT Framework that integrates the data retrieval, processing, and learning layers is presented with a use case on weather data clustering analysis. The learning model we developed uses clustering unsupervised learning method in the learning phase of the framework in order to best utilize the associated big data for this problem. The US Weather data captured from 8000 different weather stations around North America is acquired through log files. This data is acquired and processed through Node.js calls, and submitted to learning phase for the learning process. In this particular study, air temperature, wind-speed, relative humidity, visibility, and pressure data are used in the data analysis. Traditional k-means clustering algorithm is applied and the results are presented. As an interesting phenomena, we observed that the data clustering matches the geographical alignment of the stations. In other words, some of the important geographical regions within the North American continent (and the continental USA) form distinct weather clusters and easily differentiated from each other. In addition, possible sensor faults and anomalies are emerged with using clustering method. This use case allowed us to present an example of how such a IoT Big Data framework can be used for such implementations.

REFERENCES

[1] 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.

[2] L. Chapman, D. Young, C. Muller, P. Rose, C. Lucas, and J. Walden, "Winter road maintenance and the internet of things," in *Proceedings of the 17th International Road Weather Conference*, vol. 18, 2014.

[3] J. Chin, V. Callaghan, and I. Lam, "Understanding and personalising smart city services using machine learning, the internet-of-things and big data," in *Industrial Electronics (ISIE), 2017 IEEE 26th International Symposium on*, IEEE, 2017, pp. 2050–2055.

[4] F. DaCosta, *Rethinking the Internet of Things: a scalable approach to connecting everything*. Apress, 2013.

[5] A. Gyrard, P. Patel, A. Sheth, and M. Serrano, "Building the Web of Knowledge with Smart IoT Applications (Extended Version)," 2016.

[6] 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.

[7] M. A. Hayes and M. A. Capretz, "Contextual anomaly detection in big sensor data," in *Big Data (BigData Congress), 2014 IEEE International Congress on*, IEEE, 2014, pp. 64–71.

[8] S. S. Institute, *IoT Mind Map-Application Domains*. [Online]. Available: http://www.sis.se/PageFiles/15118/IoT%20Mind%20Map-Application%20Domains.pdf (visited on 03/23/2016).

[9] M. Lee, J. Hwang, and H. Yoe, "Agricultural production system based on iot," in *Computational Science and Engineering (CSE), 2013 IEEE 16th International Conference on*, IEEE, 2013, pp. 833–837.

[10] L. Liu, D. Liu, Y. Zhang, and Y. Peng, "Effective sensor selection and data anomaly detection for condition monitoring of aircraft engines," *Sensors*, vol. 16, no. 5, p. 623, 2016.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] P. Pyykönen, J. Laitinen, J. Viitanen, P. Eloranta, and T. Korhonen, "Iot for intelligent traffic system," in *Intelligent Computer Communication and Processing (ICCP), 2013 IEEE International Conference on*, IEEE, 2013, pp. 175–179.

[17] M. M. Rathore, A. Ahmad, A. Paul, and S. Rho, "Urban planning and building smart cities based on the internet of things using big data analytics," *Computer Networks*, vol. 101, pp. 63–80, 2016.

[18] B. Saneja and R. Rani, "An efficient approach for outlier detection in big sensor data of health care," *International Journal of Communication Systems*, 2014.

[19] O. B. 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 *Collaboration Technologies and Systems (CTS), 2015 International Conference on*, IEEE, 2015, pp. 12–18.

[20] O. B. Sezer, E. Dogdu, M. Ozbayoglu, and A. Onal, "An extended iot framework with semantics, big data, and analytics," in *Big Data (Big Data), 2016 IEEE International Conference on*, IEEE, 2016, pp. 1849–1856.

[21] 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.

[22] 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.

[23] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of things for smart cities," *IEEE Internet of Things journal*, vol. 1, no. 1, pp. 22–32, 2014.