Real Time Analysis of Sensor Data for the Internet of Things by means of Clustering and Event Processing

Hugo Hromic[1], Danh Le Phuoc[2], Martin Serrano*[2], Aleksandar Antonić[3], Ivana P. Žarko[3], Conor Hayes[1] and Stefan Decker

NUI Galway – Information Mining and Retrieval Unit[1]
Insight Centre for Data Analytics
Galway, Ireland
{hugo.hromic, conor.hayes}
@insight-centre.org

NUI Galway – IoT Unit [2]
Insight Centre for Data Analytics
Galway, Ireland
{danh.lephuoc, martin.serrano,
stefan.decker}@insight-centre.org
(*corresponding author)

University of Zagreb [3]
Faculty of Electrical Engineering
and Computing
Zagreb, Croatia
{aleksandar.antonic,
ivana.podnar}@fer.hr

Abstract— **Sensor technology and sensor networks have evolved so rapidly that they are now considered a core driver of the Internet of Things (IoT), however data analytics on IoT streams is still in its infancy. This paper introduces an approach to sensor data analytics by using the OpenIoT[1] middleware; real time event processing and clustering algorithms have been used for this purpose. The OpenIoT platform has been extended to support stream processing and thus we demonstrate its flexibility in enabling real time on-demand application domain analytics. We use mobile crowd-sensed data, provided in real time from wearable sensors, to analyse and infer air quality conditions. This experimental evaluation has been implemented using the design principles and methods for IoT data interoperability specified by the OpenIoT project. We describe an event and clustering analytics server that acts as an interface for novel analytical IoT services. The approach presented in this paper also demonstrates how sensor data acquired from mobile devices can be integrated within IoT platforms to enable analytics on data streams. It can be regarded as a valuable tool to understand complex phenomena, e.g., air pollution dynamics and its impact on human health.**

Keywords: Cloud Computing, Interoperability, Linked Data, Intelligence Server, Sensor Data, Services, Applications.

I. Introduction

Sensor technology and sensor networks have evolved so rapidly that they are now considered a core driver of the Internet of Things (IoT). As a result, the IoT is in the economic focus of the electrical manufacturing industry. A recently published 2014 Gartner curve[2] set the IoT in the top of the hype curve of emergent technologies. As happens to all rapidly growing technologies, IoT may be heading for the 'Trough of Disillusionment' in the hype cycle. Its recovery will depend on the ability to deploy various value-adding ICT solutions and services using IoT collected data rather than on the deployment of more interconnected devices. Mostly the term IoT is associated with devices (Things) that are linked to create a network of communicating objects [1]. However the IoT term has tended to exclude the co-existing software services that are used to build an ecosystem of services, devices and applications driven mainly by IoT collected data [2]. It is widely recognized that IoT scenarios need "*intelligence*" supported by information servers, and as result of monitoring, statistical and analytical processes. Providing the means of analysing and extracting information from IoT data seems to be the necessary next step in the evolution of this technology. Analytics services will need to be realised as an integral feature of any IoT platform consisting of -physical and/or virtual- things. In a general sense, analytics refers to the process of transforming data into interpretable or actionable information. In the IoT scenario, a real time data enable on-demand analytics. In this paper we provide a proof-of-concept solution that uses an analytics interface to enable real time interpretation of IoT data. The use case for evaluating the proposed solution is a mobile crowd-sensing application for air quality monitoring in a smart city environment, where users provide data streams with wearable sensors. The real data acquired during a system trial is analysed and visualised.

The structure of the paper is as follows: Section II presents the design principles for the intelligent servers that were implemented in the framework of the OpenIoT project and extended to enable analytical services on top of the collected IoT data set. Section III describes the intelligent mobile sensing capabilities in more detail and its extensions to support IoT analytics services. Section IV describes the experiments and trials to showcase the data analytics performed on live sensor streams acquired through the OpenIoT platform in the air-quality crowd-sensing use case. Section V concludes the paper.

II. OpenIoT Real Time Data Processing by using Intelligent Servers

In this section we describe the methods and core functionality of intelligent servers implemented and demonstrated in the framework of the EU co-funded FP7-287305 OpenIoT project. Generally speaking, an intelligent server component acts as a gateway between two different applications or sources of information. In the framework of the OpenIoT project, the Intelligent OpenIoT server is the gateway for the Extended Global Sensor Networks (X-GSN) middleware (based on the GSN[3]) and the Linked Sensor Middleware (LSM-Light). X-GSN is responsible for providing an interface for the raw and heterogeneous sensor data. LSM-Light is responsible for organizing the X-GSN

[1] EU FP7 OpenIoT project - www.openiot.eu

[2] http://www.gartner.com/newsroom/id/2819918

[3] http://github.com/LSIR/gsn

978-1-4673-6432-4/15/$31.00 ©2015 IEEE

data to make semantic annotations to the collected data becoming Linked Data and also to annotate the data to enable sophisticated sensor discovery and service orchestration. Both X-GSN and LSM-Light provide various means for filtering, aggregating and managing the data before being handed over to the overlying applications. In the process of performing analytics for IoT data, the intelligent information server enables service providers to deploy cloud/utility-based infrastructures supporting the delivery of IoT services by responding to appropriate end-user requests by means of specific interfaces in the form of stream processing queries.

In order to enable analytics a crucial activity is the integration between IoT systems, which ideally must be performed in cloud environments, i.e., environments comprising virtual IoT data "entities" describing sensors, actuators and smart devices alike their readings Up-to-date several approaches have described the benefits of a sensor-based distributed computing infrastructure [3][4][5] without however providing a systematic and structured solution to the formulation and management of resources for enabling analytics on IoT collected data. Similarly, state-of-the-art participatory sensing infrastructures [6] and services [7][8], provides instantiations of cloud-based and utility-based sensing services (such as the "Location-as-a-Service" [9]), without however providing any middleware framework and disciplined approach to deploying and providing such analytical insights about the collected data.

Analytics should play a crucial role in the design of IoT middleware platforms, , so that an intelligent server can fulfil its two major requirements. First, the designed and implemented platform must allow for easy integration of data coming from all kinds of Internet connected devices, particularly physical sensors. Second, the raw input data coming from those connected devises should be semantically annotated as well as transformed to comply with the Linked Data principles (See Figure 1). Further requirements are the need to offer dynamic data source selection and orchestration functionalities in order to facilitate analytical operations.

The OpenIoT edge server has been extended towards endowing more intelligence to enable analytical functionality. Mainly the Global Sensor Networks (X-GSN) and the Linked Sensor Middleware (LSM-Light) have been re-engineered for this purpose. Sensors and Internet-connected objects need to be registered with X-GSN. The registration process essentially requires the definition of a wrapper to transform the incoming sensor data into a common format. X-GSN introduces the notion of a virtual sensor to abstract from the device-specific data representation. A virtual sensor can be a physical sensors or any other kind of continuous data source. Moreover, a virtual sensor can represent a set of data sources. LSM-Light, in turn, considers virtual sensors as input data sources. Data coming from virtual sensors are transformed into a Linked Data representation, i.e., RDF (Resource Description Framework), and annotated according to the supported ontology. This transformation is, again, done using wrappers that add annotations to the data.

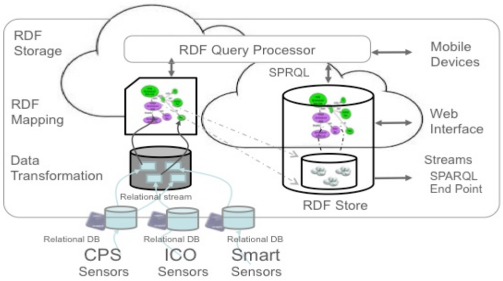


Figure 1. Intelligent Server Functional Diagram.

Working on annotated RDF data, LSM-Light provides even more sophisticated means to filter and aggregate data. It allows SPARQL (SPARQL Query Language) queries over stored sensor data, but also includes more relevant operations enabling analytics via the execution of continuous queries – formulated in CQELS (Continuous Query Evaluation over Linked Streams) [10] as an extension of the SPARQL language. LSM-Light features a SPARQL endpoint and streaming channels (e.g., WebSockets, XMPP, PubSubHubbub) to forward request data to applications enabling CQELS.

The architecture of the Intelligent OpenIoT Server comprises several software components. All components used for the implementation and trials are open source:

- *ZooKeeper*[4]: Apache ZooKeeper is an open source distributed configuration service, synchronization service and naming registry for large distributed systems.
- *Storm*[5]: Storm is a free and open source distributed real time computation system. Storm makes it easy to reliably process unbounded streams of data, doing for real time processing what Hadoop did for batch processing.
- *HBase*[6]: Apache HBase is an open source, non-relational, distributed database modelled. It provides a fault-tolerant storage for large quantities of sparse data. HBase features compression, in-memory operation, and Bloom filters on a per-column basis.
- *ZeroMQ*[7]: ZeroMQ is a high-performance asynchronous messaging library for scalable distributed or concurrent applications. It supports message queues, but unlike other message-oriented middleware, ZeroMQ runs without a dedicated message broker.

The architecture of the intelligent server corresponding to the cloud environment is shown in Figure 2. The Execution Coordinator coordinates the cluster of operator containers using coordination services provided by Storm and HBase, which share the same Zookeeper cluster. The

[4] http://zookeeper.apache.org
[5] http://storm-project.net/
[6] http://hbase.apache.org/
[7] http://www.zeromq.org/

Global Scheduler uses Nimbus, an open source EC2/S3-compatible Infrastructure-as-a-Service implementation, to deploy the operators' code to Operator Containers and monitor for failures. Each operator container node runs a Storm supervisor that listens for continuous processing tasks assigned to its machine via Nimbus. The processing tasks that need to process the persistent data use the HBase Client component to access data stored in HBase. The machines running an operator container also hosts the HDFS DataNodes of the HBase cluster. The DataNodes are accessed via the operator container's HRegionServer component of HBase.

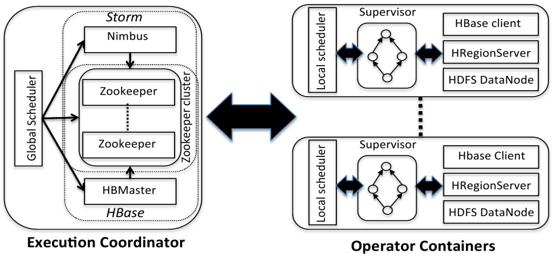


Figure 2: Intelligent Server Cloud infrastructure.

III. Intelligent Capacities for Internet Connected Objects Enabling Analytics in Cloud

Mobile sensors are used to opportunistically transmit the sensed data, mainly into the cloud via intelligent systems, and thus have the potential to generate huge amounts of streaming data falling in the Big Data domain. In this paper we don't provide details about sensor data harvesting and data transmission into the cloud in order to enable analytics on top of the collected data; details can be found at [11]

3.1 Data Import and Raw Data into X-GSN

X-GSN is a middleware designed to facilitate the deployment and programming of sensor networks. X-GSN runs on one or more computers composing the backbone of the acquisition network (see Figure 3). A set of wrappers allows feeding raw data into X-GSN system. Wrappers are used to encapsulate the data received from the data source into the standard X-GSN data model.

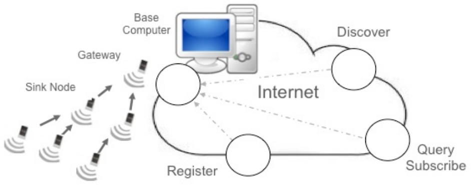


Figure 3: Data acquisition network in X-GSN.

3.2 Data Transformation into LSM-Light

There are two ways to import/stream data into LSM-Light – particularly the data as output from X-GSN in the context of OpenIoT: pull-based and push-based. In the pull-based approach, LSM-Light periodically polls a data source. In contrast, the push-based approach enables data sources (e.g., X-GSN) to actively send data to LSM-Light. Both mechanisms are used within the OpenIoT infrastructure, featuring different advantages and disadvantages depending on the actual use case.

3.3 Filtering Capabilities and Edge Intelligence

To limit the number of sensors and/or sensor data points made available for defining new data services is a fundamental feature of an IoT platform. For example, sensors might be filtered according to their location or manufacturer, and sensor data might be filtered according to the minimum level of accuracy. The universal need for such filtering capabilities indicates that it is optimal to implement them within the edge intelligent OpenIoT Server,

Edge Intelligent Server handles data streams that are integrated in the cloud via the sensor middleware (i.e. X-GSN). X-GSN allows data collection from different types of sensors (such as physical devices, signal processing algorithms, information fusion algorithms), which are integrated on the basis of X-GSN's "virtual" sensor concept.

Edge Intelligence OpenIoT Servers support filters that are compliant with the EPC Global Architecture [8] and standards, and more specifically filters that are compliant to the EPC ALE [9] (Application Level Events) and EPC LLRP specifications.

3.3.1 LSM-Light – Semantic Level

LSM-Light transforms the data from virtual sensors into Linked Data stored in RDF. The de-facto language to query RDF is SPARQL[10]. A SPARQL query is a so-called one-shot query, i.e., the query is issued to and executed by the system, immediately returning a result (that might be empty). In the context of LSM-Light, or IoT applications in general, such queries typically refer to queries about sensor metadata and historical sensor readings.

SPARQL queries are executed once over the entire collection and discarded after the results are produced, queries over Linked Stream Data are continuous. Continuous queries are first registered in the system, and continuously executed as new data arrives, with new results being output as soon as they are produced. For processing continuous queries over Linked Stream Data, the LSM provides the CQELS engine [10] a developed tool that runs in cloud systems. The query processing in CQELS is done in a push-based fashion, i.e., data entering the query engine trigger the processing. The continuous queries are expressed in the CQELS language, which is an extension of the SPARQL 1.1 standard.

[8] http://www.gs1.org/gsmp/kc/epcglobal/architecture
[9] http://www.gs1.org/gsmp/kc/epcglobal/ale
[10] http://www.w3.org/TR/rdf-sparql-query/

3.4 Aggregation Capabilities

Often individual sensor readings are not relevant but instead an aggregated value is required. For example, to increase the accuracy, rooms are equipped with a set of low-cost temperature sensors. From an application perspective, however, only the average room temperature is of interest. It is therefore meaningful to provide application-independent support for the aggregation of sensor data in the edge server. The three levels (physical, virtual (X-GSN) and semantic (LSM) feature different aggregation capabilities.

IV. EXPERIMENTS AND TRIALS

This section describes an OpenIoT use case in the area of Smart Cities to showcase the data analytics performed on live sensor streams acquired through the OpenIoT platform. Urban crowd sensing for air quality monitoring is selected as a representative use case since air quality should be monitored in city areas densely, both in time and space, to understand air pollution dynamics and its impact on human health [11]. The use case involves citizens carrying wearable sensors for air quality monitoring who contribute big data streams by use of smartphones to the OpenIoT platform. The acquired data set provides the means for new discoveries that are otherwise not feasible by existing static meteorological stations. The data stream analytics can help ecologists, public health officers and city officials to understand urban dynamics and put new perspectives in population-wide empirical public health research.

4.1 USE CASE DESCRIPTION

The air quality monitoring application integrates the data produced by low-cost wearable mobile sensors measuring temperature, humidity, atmospheric pressure, and levels of air pollutants. A wearable sensor shown in Figure 4 is custom designed and built from off-the-shelf components, and communicates over Bluetooth with a smartphone. It incorporates electrochemical gas sensors for measuring atmospheric sub-ppm level concentrations of carbon monoxide (CO), and either nitrogen dioxide (NO2) or sulphur dioxide (SO2). Two gas sensors are integrated onto a single sensing node to reduce its size, so that it can be easily mounted on a bicycle or worn on clothes or backpacks. A rechargeable Li-Ion battery with estimated autonomy of 3 days powers the sensor. The communication protocol between a sensor and smartphone enables an application user to start or stop the sensing process on sensor nodes, as shown on the right-hand side smartphone in Figure 4. After the sensing process is started, measurements are produced periodically on the sensing node and are delivered to the mobile application running on the smartphone that decides whether to transmit the data into the OpenIoT cloud. The smartphone on the left-hand side in Figure 4 shows the perspective of a user's application that receives air quality alerts in the geographical area where the user is currently residing.

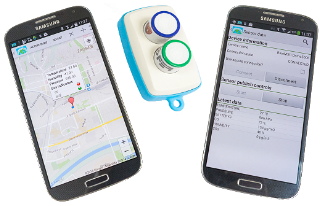


Figure 4. Wearable sensor and mobile application.

In our evaluation we use a real sensor dataset from an air quality measurement campaign SenseZGAir conducted in the City of Zagreb in July 2014. Twenty (20) volunteers were collecting measurements using wearable sensors to acquire the data for further processing within the OpenIoT platform. They have produced a dataset comprising 16,835 data points, where each data point includes 5 sensor readings (temperature, humidity, atmospheric pressure, CO level and NO2 or SO2 level) and is enriched by coordinates (latitude and longitude) provided by a smartphone. Each data point has 7 dimensions. The volunteers have covered the area of 144 km^2 during the campaign and have crossed in total a distance of 758.6 km, either on bicycles or on foot. The available sensors and related variables are in Table 1.

Table 1: Variables measured during the SenseZGAir campaign.

Variable	Name	Sensor Description
Long	Longitude	Geographical Longitude in decimal degrees.
Lat	Latitude	Geographical Latitude in decimal degrees.
T	Temperature	Air temperature in Celsius degrees.
P	Pressure	Atmospheric pressure in hPA.
H	Humidity	Air humidity in %.
CO	CO	Carbon monoxide level in micrograms per m^3.
NO2	NO$_2$	Nitrous dioxide level in micrograms per m^3.
SO2	SO$_2$	Sulfur dioxide level in micrograms per m^3.

4.2 CORRELATION ANALYSIS FOR IOT SENSOR DATA

The goal of the SenseZGAir analytics is to understand how various variables relate to each other in time and geographically. For this task we construct *correlation graphs* that evolve over time. A correlation graph $G = (V, E)$ is an undirected complete graph (K_n) where the vertices (V) represent n variables being measured during the experiment, while the edges (E) represent correlations between them. The visual representation of a correlation graph is shown in Figure 5, where edge colours depict the type of correlation (positive, negative or none at all) between two variables while edge widths are proportional to correlation strength.

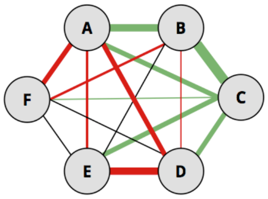


Figure 5: Correlation Graph for variables A to F.

When building this graph we first need to choose an adequate statistical measure to assess the actual correlation magnitude (or strength) between two variables which is assigned to edge weights *W*(*e*), *e* ∈ *E*. The literature suggests a linear relationship between Temperature and Humidity, and also between air pollutant gases such as CO, NO2 and SO2 [12],[13]. From this evidence we assume that the sensor variables we are using *may* have a linear correlation. We can then employ the well-known Pearson's Correlation Coefficient for a sample (denoted as *r*), as shown in Equation 1, where *X* and *Y* are a pair of variables under analysis, while *X*' and *Y*' are their respective means.

$$r_{X,Y} = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}}$$

...(1)

The Pearson's Coefficient is convenient for our purposes because it represents the correlation between two variables in the closed range [−1, 1]. When *r* ≈ 1, it implies a perfectly positive correlation (both variables have a linear relationship). When *r* ≈ −1, it means a perfectly negative correlation (both variables have an inverse linear relationship). Finally when *r* = 0, this indicates that no linear correlation exist between the variables (however, a non-linear relationship may still exist). Thus we use the absolute magnitude of the coefficient (|*r*|) as an edge weight and its sign (±) as the edge colour *C*(*e*), *e* ∈ *E*. For the special case when *r* = 0 we use a distinctive colour. This is summarized in Equations 2 and 3 respectively.

$$W(e = (X,Y)) = |r_{X,Y}|$$

...(2)

$$C(e = (X,Y)) = \begin{cases} red & \text{if } r_{X,Y} < 0 \\ green & \text{if } r_{X,Y} > 0 \\ gray & \text{otherwise} \end{cases}$$

...(3)

To further improve graph visualisation by incorporating some notion of the values associated with the involved variables, we decided to add weights to graph vertices that are proportional to the observed variables and to size them accordingly. Visualising independent variables together is a challenging task mostly because variables are often given in different scales. How can we know which values are considered low or high? In our particular Air Monitoring experiment we are interested in summarizing the observed values to enable comparison between their values, i.e., to indicate whether measurements are substantially higher or lower than normal values.

We decided to model vertex weights *W*(*v*),*v* ∈ *V* as a normalised mean of the variable observations while minimizing the effects of different scales that each variable may possess. Under the assumption that no a priori knowledge about the variables involved exists, a reasonable approach is to normalise variable observations by statistical outlier detection. Our reasoning here is that vertex weights should be balanced if the observed values are within a historic average for the same variable, but are marked as low or high if spurious averages go below lower or upper relative thresholds.

In statistics there are many techniques for outlier detection to identify data values that might not be real observations of a variable. We decided to employ a simple outlier detection based on inter-quartile range analysis. The first quartile (*Q*1) is defined as the median value where at least 25% of all the data points sit. Analogously, the third quartile (*Q*3) is the median value where at least 75% of all the data points are. The inter-quartile range (*IQ*) is then *Q*3 − *Q*1 and we can define an outlier detection range as [*Q*1 − *kIQ*, Q3 + *kIQ*]. The non- negative constant *k* is used to set different sensitivities for this range. In descriptive statistics two values of *k* are commonly used: *k* = 1.5 to generate a mild outliers detection range, and *k* = 3 to generate a major outliers range. Since we do not expect values to fall very often into the major outliers range, we decided to use the mild range instead. Furthermore, all of the variables in our experiment have non-negative values. This allows us to simplify the normalisation to use only the positive side of the mild range, as defined in Equation 4.

$$W(v = X) = \frac{\bar{X}}{Q3 + kIQ}, k = 1.5$$

...(4)

4.1.2 Time and Location Partitioning

Correlation graphs can provide a good overview on how sets of variables correlate. However, we are interested to investigate how these variables evolve over time and in different geographical regions. Simply using time windows can do partitioning data over time. For our experiment we decided to split the data into 2 hours non-overlapping time windows, producing in total 28 windows for the data available in the experiment.

The second partitioning task is then to further divide the data into interesting geographical areas or zones. As mentioned before, all observations are times tamped and have an associated Coordinates variable. We use the coordinates to spatially group the data by means of the k-means clustering algorithm with predefined cluster centroids. In the case of our experiment we want the centroids to be the central points for different areas of interest of the city, e.g., green, industrial or residential areas. The k-means algorithm is particularly useful here; it generates Voronoi-like partitions, which are very convenient in the geographic context.

4.3 SenseZGAir Experiment

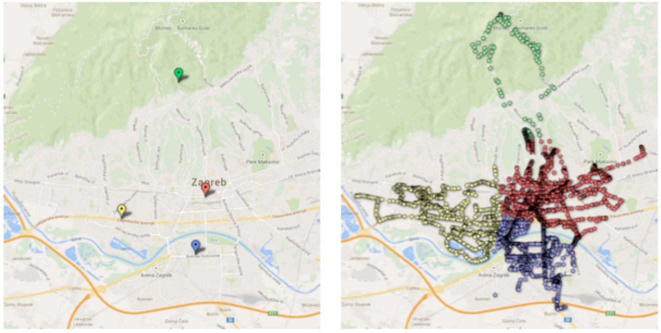
The SenseZGAir experiment is performed on the dataset summarized in Table 2.

Table 2: Summary of SenseZGAir dataset

Time Range	2014-07-04 08:38:47 to 2014-07-10 12:49:30
Number of Datapoints	16,835
Unique Coordinates	13,905
$T/P/H$ **Readings**	16,835 (100%)/16,835 (100%)/16,811 (99.8%)
$CO/NO2/SO2$ **Readings**	16,815 (99.9%)/7,505 (44,6%)/9,295 (55.2%)

It can be observed that the number of readings acquired by NO2 and SO2 sensors is approximately one half of the total number of readings since each sensing nodes includes either an NO2 or SO2 sensor.

We selected four centroids for the City of Zagreb to initiate the k-means algorithm, as shown in Figure 6a. The four centroids correspond to the northern green area, the western and eastern zones above the river Sava and the southern zone below it. After running k-means over all the data points the resulting zones can be seen in Figure 6b.

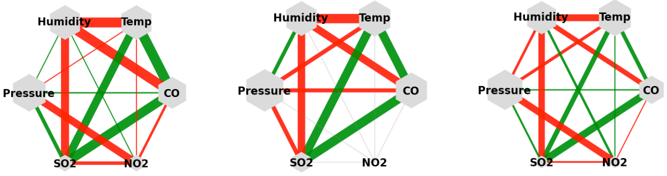


(a) Selected input centroids (b) Resulting zones

Figure 6: Geographical partitioning of the dataset using the modified k-means algorithm.

The number of assigned data points per zone is as follows: 307 points for the northern green zone (Zone 2), 2148 points for the western yellow zone (Zone 3), 7019 points for the eastern red zone (Zone 4) and 4431 points for the southern blue zone (Zone 1). In addition to geographical partitioning we also performed time splitting using 2-hour time windows. The zone and time partitioning of the data set has resulted in the total of 97 computed correlation graphs.

An example set of correlation graphs built for Zone 1 is shown in Figure 7. Three consecutive time windows during a hot day are displayed for this zone (the southern area of the City of Zagreb, below the river). The correlation graphs reveal interesting relationships between the measured variables and can serve as a useful visual tool to environmental scientists or as input for further analytical systems.

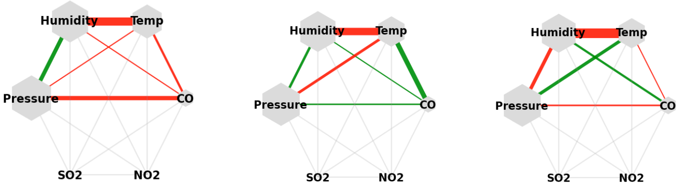


(a) 2014-07-08 10:00 to 11:59 (b) 2014-07-08 12:00 to 13:59 (c) 2014-07-08 14:00 to 15:59

Figure 7: Correlation graphs built for Zone 1 during three consecutive time windows.

Firstly the graphs show a strong inverse correlation between Humidity and Temperature, this is an identified trend [12]. Furthermore, CO and SO2 also exhibit a strong correlation that may indicate that both gases are produced by the same sources, in this case traffic [13]. This strong correlation weakens in the last time window when both CO and SO2 levels drop. At the same time the inverse correlation between CO and Humidity also weakens, suggesting a possible effect of Humidity on the contaminant [14]. Conversely, some variables exhibit a change in the sign of their correlation, as can be seen for Pressure and CO levels. A number of interesting observations can be further drawn based on air quality analyst assessments by following the graphs.

A second set of correlation graphs is shown in Figure 8. In this case the graphs show the evolution during the night for Zone 4, which is the biggest zone constructed by k-means and located in the eastern part of the city. Again, interesting behavioural patterns for the studied variables can be observed. For example, there are varying effects between Temperature, CO levels and Pressure. However, the interesting observation is that these correlations switch signs during this pulsation. This can be also observed for Humidity and Pressure during the same period. This dynamicity may suggest that during low population activity, low quantities of pollutant gases are easily affected by environmental conditions. Those patterns are clearly different from the observations for Zone 1 during the day.



(a) 2014-07-10 00:00 to 01:59 (b) 2014-07-10 02:00 to 03:59 (c) 2014-07-10 04:00 to 05:59

Figure 8: Correlation graphs built for Zone 4 during three consecutive time windows.

For experimental purposes a different zone configuration was used. We split the biggest Zone 4 (east side of the city at figure 4) into two equally distributed sub-zones generating a new 5-zone configuration. We observe that most of the patterns are now distributed between the two new sub-zones, and some new patterns have emerged. This suggests that geographical division of the data is an influential step for construction of correlation graphs, while mobile crowdsensed data can be a valuable source of information for zone-based environmental analysis.

Finally, we experimented with different window size strategies, from 15 minutes up to 12 hours. For deciding on what windowing setting to adopt, we must trade-off between two aspects: latency of the analysis and quality of the computed correlations. If the window size is too small, the analysis is more updated but the correlations might be not trustful. On the other hand if the window size is too big, the correlations are of higher quality but computing them requires more waiting time. As an example see Figure 9 where the number of data points per window across all windows for three different window sizes (30 minutes, 2 hours and 6 hours) can be seen. To the right of the figure, the correlation graphs for each window size at the same time (the dotted line) are displayed. It can be observed that for the small size setting of 30 minutes the correlation graph is not able to compute correlations for the SO2 variable nor provide a good quality Pearson's Coefficient for the other variables. On the other hand the 6 hours settings gives a very fine-grained correlations graph at the cost of having to wait for a quarter of the day for data. The middle point setting of 2 hours provides a reasonable approximation in quality for the computation of the correlations with the benefit of a much lesser latency for obtaining this result.

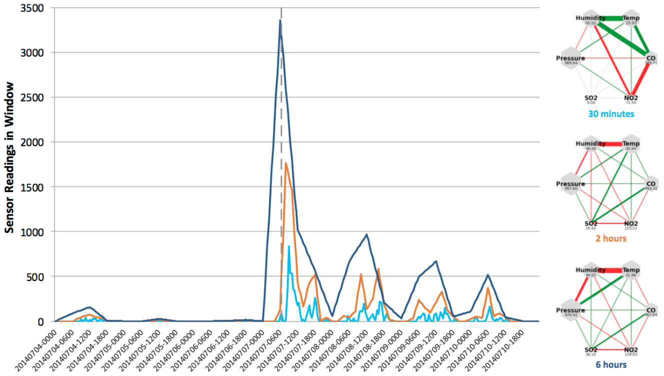


Figure 9: data points for different window size analysis.

We have developed an interactive web-based tool to explore the correlation graphs built and the zones found by k-means that behaves according to distributed sensor model [15]. This tool presents a navigable map based on the Google Maps technologies with all the points inside all the zones found by k-means using different colours. The centroids are also shown with a distinctive pin. Furthermore, the tool also presents the correlation graphs for each zone using navigable time windows. We believe that this tool can be an important analytical framework for a specialized end-user, e.g. a City Manager, who needs to monitor city-wide dynamics and variables such as those explored in this paper.

V. Conclusions

The paper introduces the OpenIoT approach to data stream analytics by use of intelligent servers running in cloud environments and edge servers for real time data acquisition, annotation and processing of sensor data. The design and specification of the OpenIoT edge servers is optimal for real time analysis of (raw) sensor data. The design of the intelligent servers is a step towards enabling IoT data analytics on top of huge IoT streams. The main characteristic of the Intelligent OpenIoT server system presented is that IoT adaptations/extensions have been added towards satisfy initial analytical work on the collected IoT data streams.

The described SenseZGAir trials have collected a real mobile crowd-sensed data collection that is used for investigating air quality conditions by means of analytics. In particular correlations between a number of sensor variables and correlation evolution over time have been identified. The resulting visual representation of correlations provides valuable information for domain experts and can be regarded as a useful tool for investigating and understanding complex phenomena, e.g., air pollution dynamics, human health, Etc.

Acknowledgments

Part of this work has been carried out in the scope of the ICT OpenIoT Project, which is co-funded by the European Commission under seventh framework program, contract number FP7-ICT-2011-7-287305-OpenIoT and also Science Foundation Ireland (SFI) CLIQUE project under Grant No. 08/SRC/I1407 (Clique: Graph & Network Analysis Cluster). Partially the Insight Centre for Data analytics with Grant No. 12/RC/2289 (Insight) and the Croatian Science Foundation project Human-centric Communications in Smart Networks (HUTS).

References

[1] A J Jara, RM Silva, JS Silva, MA Zamora, AFG Skarmeta, Mobile ip-based protocol for wireless personal area networks in critical environments. Wireless Personal Communications 61 (4), 711-737.

[2] Harald Sundmaeker, Patrick Guillemin, Peter Friess, Sylvie Woelfflé (eds), "Vision and Challenges for Realising the Internet of Things", ISBN 978-92-79-15088-3, © European Union, March 2010.

[3] Hock Beng Lim, Yong Meng Teo, Protik Mukherjee et Al. "Sensor Grid: Integration of Wireless Sensor Networks and the Grid", Proc. of the The IEEE LCN. 91-99, Nov 15-17, 2005.

[4] Lina Yu, Xiang Sun et Al. "Research on Resource Directory Service for Sharing Remote Sensing Data under Grid Environment", gcc, pp.344-347, 2009 8th Intl. Conf. on Grid and Cooperative Computing, 2009.

[5] Kapadia, A.; Myers, S.; XiaoFeng Wang; Fox, G. "Secure cloud computing with brokered trusted sensor networks", Intl. Sym. on Collaborative Technologies and Systems (CTS), 7-21 May 2010.

[6] Catherine Havasi, James Pustejovsky, Robert Speer, Henry Lieberman, "Digital Intuition: Applying Common Sense Using Dimensionality Reduction", IEEE Intelligent Systems, pp. 24-35, July 2009.

[7] Nicolas Maisonneuve, Matthias Stevens et Al. "NoiseTube: Measuring and mapping noise pollution with mobile phones", Proc. of the 4th Intl. ICSC Symposium, Thessaloniki, Greece, May 28-29, 2009.

[8] Page, X. and A. Kobsa "Navigating the Social Terrain with Google Latitude". iConference 2010, Urbana-Champaign, IL, p.174-178.

[9] Marshall Kirkpatrick "The Era of Location-as-Platform Has Arrived", ReadWriteWeb, January 25, 2010

[10] D. Le-Phuoc, M. Dao-Tran, J. Xavier Parreira, and M. Hauswirth. "A native and adaptive approach for unified processing of linked streams and linked data". ISWC2011, page 370-388. Springer, (2011)

[11] Martin Serrano, Hoan Nguyen, Danh Le Phuoc, Manfred Hauswirth, John Soldatos, Nikos Kefalakis, Prem Jayaraman, Arkady Zaslavsky "Defining the stack for service delivery models and interoperability in the Internet of Things: A practical case with OpenIoT-VDK" IEEE Journal on Selected Areas in Telecommunications 2014.

[12] D. Le-Phuoc, M. Dao-Tran, J. Xavier Parreira, and M. Hauswirth. "A native and adaptive approach for unified processing of linked streams and linked data". ISWC2011, p.p. 370-388. Springer, 2011.

[13] H.-Y. Liu, E. Skjetne, M. Kobernus, Mobile phone tracking: in support of modelling traffic-related air pollution contribution to individual exposure and its implications for public health impact assessment, Environmental Health 2013, 12:93.

[14] T Wang, TF Cheung, YS Li, XM Yu, DR Blake, et al. Emission characteristics of CO, NOx, SO2 and indications of biomass burning observed at a rural site in eastern china. Journal of Geophysical Research, 107(D12), 2002.

[15] Mark G Lawrence. The relationship between relative humidity and the dewpoint temperature in moist air: A simple conversion and applications. Bulletin of the American Meteorological Society, 86(2):225-233, 2005.