Architecture Design of Pattern Detection System for Smart Cities Datasets

Valentina-Camelia Bojan, Ionut-Gabriel Raducu, Florin Pop, Mariana Mocanu, Valentin Cristea
Computer Science Department, Faculty of Automatic Control and Computers, University *Politehnica* of Bucharest
Bucharest, Romania
Emails: bojan.valentinacamelia@gmail.com, gabi.raducu25@gmail.com,
florin.pop@cs.pub.ro, mariana.mocanu@cs.pub.ro, valentin.cristea@cs.pub.ro

***Abstract*—Nowadays, there is more and more interest in the research and development of systems, applications, tools or frameworks for 'smart' environments. We do not want to bother with useless actions or decisions anymore because we want to spent our time doing more valuable activities. This would only be one reason to put the bases and after that to build a platform able to extract patterns and useful information from data measured by devices that monitor the 'smart' environment. A platform of this kind would become the main reason for the environment to be a 'smart' one and for more people to understand the value of such an environment. In this paper we investigate the need of a global and generic platform able to work with many type of datasets, with various systems and to serve different 'Smart cities' applications. Through this platform we aim to unify the need of all 'Smart cities' systems for having and using mined data, patterns extracted from the generated (measured) raw data.**

***Index Terms*—data analysis; distributed processing; machine learning; pattern recognition; time series; smart cities.**

I. Introduction

We live in a society where the concept 'information means power' becomes more and more popular, regarding any business that relies on its customers. Similarly, for us, people, another concept starts to apply very well: 'information means comfort', because the more we know in advance, the more we can save our time from finding certain pieces of information.

The question that rises is 'How can we have any valuable information just a click away?' This desire marks the beginnings of 'Smart cities' technologies, systems, ideas and the tool to achieve all these things is big data. To obtain valuable data we have to store every piece of information that we work with, that we give as output, as opinion, as decision. We have to accept to be a little part from the big system that is a city or even the entire world, so that we can complete the smart ensemble.

Big data is a term that describes datasets with sizes that cannot be processed, analyzed, queried and managed using the traditional tools, frameworks, techniques. Doug Laney used a '3V' model to describe big data. The three V's are volume (amount of data), velocity (speed of data in and out), and variety (range of data types and sources) [1].

Using big data often means to be 'smart' and that is because applications that use big data relies on machine learning systems that provide the necessary tools to dig into the data, put apart the noises and extract information of interest. As Mayer-Schönberger says in his book, big data does not ask why and simply detects patterns [2]. This research has as its purpose the analysis of data for pattern recognition.

Because this is such a big domain, we will target only the systems for 'Smart cities' that often work with time series data. The Internet of Things (IoT) produces an enormous amount of data every day through the huge number of sensors that generate thousands of measurement each second [3], [4]. Almost every object in our everyday life has the ability to emit data [5]. Smart meters in plants, smart shirts for athletes, smart watches that monitors us, smartphones, medical sensors are only a few examples.

No matter the system or the problem that we want to solve, we need to analyze huge amounts of structured data. As far as that goes storing, big data receives support from all the companies that develop solutions of cloud storage. Meanwhile, the domain of analysis is open to research and improvements and we will focus on it. There is no unique data. Everything has a repetitive part and we intend to find it in order to supply valuable data for valuable moments. But none of these can be possible if it is not supported by a strong, scalable, efficient and effective platform of which architecture relies on suitable components for the 'Smart cities' purposes. In this paper we describe a proposal of such an architecture and prove how this proposal can be used for solving some real problems.

'Smart cities' do not mean only storing data generated by our devices, visualize them or make the devices react to certain signals. To build a smart city we need to build a system able to think for us, able to know how to analyze our data and to provide the necessary information that bring us value. For example, what value can give a system that only stores the temperature value and shows their evolution? The answer is: no value. Everything changes for a system that is able to predict the weather. However, to do this it requires algorithms for pattern discovery and for integration with other parameters of interest.

Therefore, the main motivation of this research is to show the strength of a system for 'Smart cities' that relies on machine learning techniques and integrate them into a cloud based infrastructure. Through this platform we want to explore the Big-data in order to detect the unusual data values reported from the sensors, which can be useful in finding the reasons of a failure and preventing it in the future [3]. The detection of patterns in the measured values can also be used

to obtain trends prediction for the observed parameter [6], predictions having a worldwide applicability, some examples being weather and financial forecasting, medical analysis, or traffic congestion.

The objectives of this paper are the following:

- to propose an architecture for a platform that works with Big-Data and applies data-mining algorithms to discover the patterns that exist in the stored data;
- to develop the architecture around a popular processing;
- to show the workflow that the data of a certain dataset follow from the moment when it is generated and stored to the final moment when the platform generates the result;
- to identify some datasets that 'Smart cities' work with and to explain how pattern detection can be useful for them and what problems the presented platform can solve.

In this paper we will start by describing several datasets that are popular in 'Smart cities' applications and for each of them we will try to find the best algorithm for pattern detection and to see what problems the pattern discovery can solve. In order to propose a generic architecture, in the second chapter we will study the similarities between the presented datasets. In chapter 3 we will present the architecture design that we propose for a platform that can detect patterns in Big-Data, in order to describe the workflow corresponding to the platform in chapter 4. We will end this paper with a section highlighting the future work and some conclusions.

II. 'SMART CITIES' DATASETS, PATTERNS AND PROBLEMS

As mentioned earlier, the bases of 'Smart cities' rely on the implantation of sensors in each device or object that can bring us value through its monitoring. Inherently, sensor networks are the major source of Big-Data. In this chapter we will present how exploring the Big-Data from 'Smart cities' systems can solve some of the biggest and obvious problems that we meet daily.

A. Datasets description

Further, we will describe three datasets collected from partners of the CityPulse EU FP7 project [7].

1) Road traffic data: This collection contains several datasets corresponding to the traffic observations made periodically between two locations over a period of 6 months. The raw data comes along with some metadata which shows information about the generated datastreams, such as:

- details about the two observation points (city, street, postal code, coordinates);
- distance between the two points in meters;
- duration of measurements in seconds;
- type of road.

2) Pollution data: The second chosen collection contains datasets of the pollution measurements for the city of Brasov in Romania. This collection is designed to complement the vehicle traffic dataset above, for each traffic observation point a sensor being collocated to measure the air quality. The metadatas that accompanies the raw data are the following:

- timestamp that shows when the measurement took place;
- the coordinates (longitude and latitude) of the sensor;
- measured air parameters (ozone, particulate matter, carbon monoxide, sulfur dioxide, nitrogen dioxide).

3) Parking data: Another popular dataset in 'Smart cities' is the one that contains parking data. The CityPulse EU FP7 project provides a parking data collection from the city of Aarhus in Denmark. There are a total of 8 parking lots providing information over a period of 6 months. The metadata contains the following:

- details about the parking garage: garagecode, city, postalcode, street, housenumber, latitude, longitude;
- the total spaces of the parking garage;
- timestamp that shows when the measurement took place.

B. Finding patterns

Having all these datasets we need to decide what we want to obtain from their processing and analysis and what information can contribute to make the monitored city smarter. The process of discovering patterns in datasets, is known as data mining, which is also referred to as Knowledge Discovery in Databases (KDD) [8].

If we want to put the bases of a generic platform that is able to detect patterns in 'Smart cities' datasets, we need to take into consideration the whole process of KDD, in which the pattern detection is only one step. Our platform needs to have an architecture that supports the data mining process steps prior to the pattern recognition step, such as data collection that supposes the gathering operation from existing databases and data preprocessing that implies several sub-steps such as data cleaning, data integration, data transformation, data reduction or data discretization [8], [9].

Our platform cannot stop after a certain data mining algorithm was applied in order to identify and extract data patterns. We need to think of an architecture that allows the platform to continue the data mining process and offer it the ability to make an evaluation of the obtained pattern features, because not all the results bring valuable information for the studied domain. Because of the fact that several algorithms have associated metrics that allow them to be evaluated regarding their performances, we will design a platform that can measure the value of the patterns right before to output the result.

C. 'Smart cities' problems and solutions

There are uncountable problems that find solutions through our platform, problems related to each 'Smart city' from different domain such as energy, water, air, public services, mobility or buildings. Regarding the 3 chosen datasets, we have to chose the best data-mining algorithm that can provide us the needed information for the problem that we want to solve.

For example, for the traffic dataset, we can apply clustering algorithms if we want to discover the spots that are predisposed to congestions. Along with location information we can also obtain time related information, in this way having a picture about where and when traffic issues rise and sent

this results to the interested people or authorities that can take measures about them. The existing datasets can serve as training sets for supervised learning algorithms and using them the platform can be used to predict the traffic and to route the drivers on different ways so they all would not follow the same route to the destination.

The same algorithms can be applied on the pollution datasets too with the same purposes: to see what the most polluted areas are, when the pollution rises and what the values for the air parameters are in those moments. However, these information can be insignificant if we do not also show the cause of the problem. For this reason, we may use algorithms for mining frequent sequences that can appear between traffic data and pollution data. In this way, the platform will contribute to the reduction of emissions, because the obtained results will be sent to the traffic police or to applications for finding directions to a certain destination in order for them to divert the traffic along other less congested areas to reduce carbon emissions in a particular area.

Parking problems can also be better managed. Monitoring the parking places, cars can be guided to the nearest available parking lots. Finding patterns in the traffic and parking datasets through clustering, classification and frequent sequences algorithms, the platform can find out where the build of new parking places is needed so that cars can find a free spot quicker and not to congest the traffic while searching for a parking place.

III. A DATA-MINING PLATFORM FOR 'SMART CITIES' APPLICATIONS

A. Architecture design

In the previous chapters we put on the table the need for the existence of a generic data-mining platform that serves 'Smart cities' systems and applications and we explained all the problems that this kind of platform could solve. What we have to do next is to present how the architecture of this platform would look like, and how all the member components would interact and work together in order to meet the platform purposes. The first concern, about how the architecture looks like is solved through Figure 1.

In Figure 1, we can observe a layered architecture, the lowest layer working with data while the top layer is working with the applications. We will have a bottom-up approach in out explanation. As we can see, the first layer corresponds to *Data support component*. This component is responsible for the storage of the data that came from various sources, like sensor networks, drones, different systems that log events. All these data producers that accept to serve our platform must send their data to a certain data support from this component. The platform component supports many data supports so that the data source system does not change its working process. For this reason, data can be stored in databases, in files on disks (both supports can be on server machines or on a Cloud storage provider) or in memory.

The next layer of the architecture is represented by *Data gathering component*. This module is designed so as to be able to take (gather) data from the necessary tables/files and transform them into collections of objects (POJOs) based on the metadata that is stored for each used dataset.

The kernel of the platform is the *Processing Engine* layer. This layer contains the *Data processing connectors* that are some pluginable components. If we think about the substeps of the data-mining processing step, we can observe that there is a connector for each substep. Not all the connectors are mandatory for a certain dataset, but if the data have to be processed before applying the pattern detection algorithm, then they have to be defined apriori their usage. We have to mention that is recommended that the connectors definition came along with the metadata of the dataset served by a certain company or system in the moment when they agreed to be part of our platform. The definition of such a connector is in fact the definition of a function that can be applied to an object corresponding to the dataset elements and when it is applied it filters, normalizes or transforms the data.

The most important part of the *Processing Engine* layer is Apache Flink, a hybrid that combines database technology with map/reduce technology [10]. It is a component that explores the data and extracts the existing patterns from it. In Figure 1 we also show the overview of Flink's stack [11] and as we can see in the picture it can be deployed either on the local machine, or on the cluster, or on a cloud machine. The place where the engine is running is not so important, what truly matters being the fact that the platform takes advantage from various deployment environments. In this way, it can start running on a single local machine and when the number of participants and consumers is growing it can easily be exported on a distributed environment, because the engine allows and embraces other deployment environments, too. As the Apache team says, Flink is a 'streaming dataflow engine that provides data distribution, communication, and fault tolerance for distributed computations over data streams' [11]. To do this, it needs a JobManager - the coordinator of the engine - and several TaskManagers - workers that execute parts of the parallel programs. When the *Runtime* component is started, Flink will also start the JobManager and one or more TaskManagers. The *DataSet API* and *FlinkML library* are very important. Before running an algorithm from FlinkML library or a defined one, the data must be pre-processed through DataSet programs that in Flink are regular programs that implement transformations on data sets. In out case, these programs will apply the functions provided by our connectors for filtering, mapping, joining, grouping etc.

Having the result (clusters, a classification tree, a classified data etc.), the engine will store the results that were obtained after a training stage (if it was a supervised learning algorithm) and will send the information as output to the *Application layer*, from where it will be consumed through a REST API or something similar by 'Smart cities' applications or systems.

B. Workflow description

In Figure 2, we present the workflow for out proposed platform that is capable of data-mining in data produced by

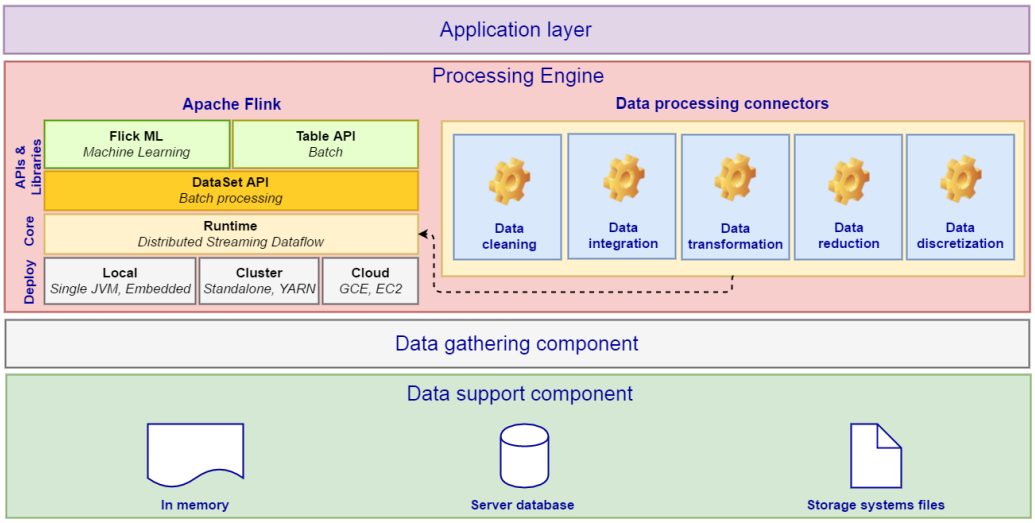

Fig. 1. The architecture design for a data-mining platform that serves 'Smart cities'.

'Smart cities' monitoring tools (sensors, drones, engines able to log certain events etc.).

As we can observe in the figure, there are three flows of data, drawn using different colored arrows. The mauve arrows put in evidence the data generated by different data sources and the fact that all this data is sent to be stored in the *Data Storage component*. The green arrow was used to identify the input of our platform, input that corresponds to requests made by 'Smart cities' systems and applications for valuable information, for mined data. The platform response (the output) is represented by the red arrows.

When a 'Smart city' application generates a request for the platform, this request is first received by the *Gateway* which is in fact the *Application layer* from the above architecture. This component contains implemented services (REST, SOAP etc.) able to serve each application or system that uses the platform. Based on the request, on the entity that created the request and on the data that this entity works with, the Gateway sends the request forward to the *Processing Engine*.

Once arrived in the heart of the platform, the request is intercepted by a module of which purpose is to obtain the necessary data to be mined by the engine. This module sends a query to the *Data gathering component* which queries the *Data storage component* from which obtains raw data. Because of the fact that our engine does not know how to work with raw data, the component that gathers the data also transforms it into a collection of POJOs. In order to be able to do such a transformation, the component uses the metadata that comes along with the raw data from the *Data storage component*. The platform is capable to combine different sources of data and to use several DataSets at once, so it is reasonable to say that after the gathering not only one collection of POJOs can be obtained, but many collections of POJOs.

Having the DataSet(s) we can proceed to the pre-processing stage, where the data is filtered, many DataSets are integrated into only one DataSet and after that it is normalized. All these steps are defined in the *Data processing connectors* and depending on the DataSets involved in the mining process, the steps can differ. In the figure it is also figured that each of these steps are distributed because they are transformed by Apche Flink in tasks that are executed in parallel. In this moment everything is ready for the pattern detection stage, where the platform applies a certain data-mining algorithm. When the algorithm ends its execution, the mining process also ends and the result is available. This result is sent as output to the Gateway that was waiting asynchronously for it so to be able to sent it back to the caller application.

IV. Conclusions and Future Work

In this paper we studied three types of datasets and identified the problems that can be solved for each presented dataset. The purpose of their introduction is to show that each dataset offers different problems to be solved and new problems can arise from the connections that exist between many datasets (like traffic and pollution). We wanted to express the need of a global and generic platform able to work with all these datasets, with various systems and to serve different 'Smart cities' applications. Through this platform we wanted to unify the need of all 'Smart cities' systems for having and using mined data, patterns extracted from the generated (measured) raw data.

After that, we continued by showing and describing an architecture design proposal for the data-mining platform. We presented all the layers of the architecture and spoke about the behaviour of each of them. To exemplify better how the platform would use the architecture described previously, we created a workflow and presented how the data would be transformed by the platform from raw data to valuable information, to data that is able to say something usefull about the system that it represents. We have to mention that this workflow can be applied having in mind any dataset(s), including the datasets given as example in Chapter 2.

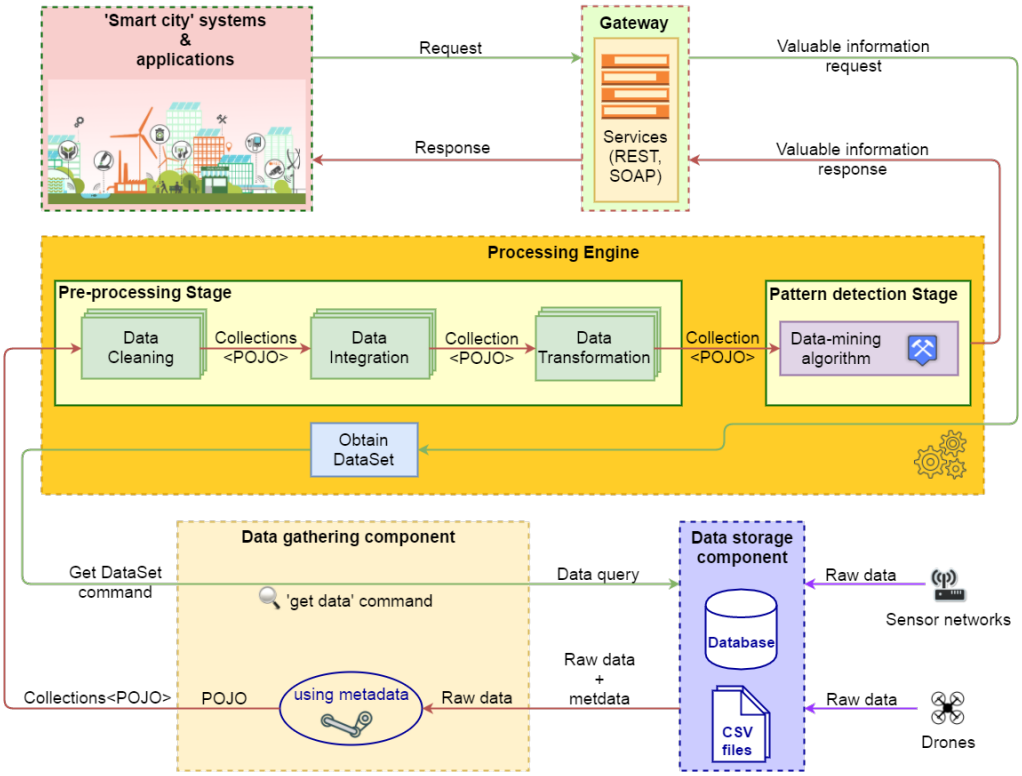

Fig. 2. Workflow for the data-mining platform that serves 'Smart cities' applications.

In our future research, we intend to implement the described platform. We will use the three datasets and for each of them we will provide the suitable metadata, connectors and data-mining algorithm. Once this stage will be completed, we will analyze the performance of the processing engine and we will evaluate the produced results. Having this evaluation, we will be able to improve the algorithm by replacing it with another one more suitable for a certain dataset or by extending it with some optimizations.

We will have to keep in mind that many 'Smart cities' systems work with time-series data, and for this reason, many of the classical methods for pattern recognition will have to be updated to make them compatible with this type of structured data and to make them able to have the same performance as they have for classical data.

We will make each component of the platform real and all the results that we will analyze will be obtained through this platform created for pattern recognition or information extraction. By creating such a useful system, we want to contribute to making the process of offering users various and valuable information that can lead to their comfort, decisions and welfare more easily and accurately.

ACKNOWLEDGMENT

The research presented in this paper is supported by projects: DataWay, PN-II-RU-TE-2014-4-2731; and Data4Water, H2020-TWINN-2015 ID. 690900.

We would like to thank the reviewers for their time and expertise, constructive comments and valuable insight.

REFERENCES

[1] D. Laney, "3d data management: Controlling data volume, velocity and variety," *META Group Research Note*, vol. 6, p. 70, 2001.

[2] V. Mayer-Schönberger and K. Cukier, *Big data: A revolution that will transform how we live, work, and think*. Houghton Mifflin Harcourt, 2013.

[3] E. J. Keogh and P. Smyth, "A probabilistic approach to fast pattern matching in time series databases." in *KDD*, vol. 1997, 1997, pp. 24–30.

[4] M.-S. Chen, J. Han, and P. Yu, "Data mining: an overview from a database perspective," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 8, no. 6, pp. 866–883, Dec 1996.

[5] Time series data is the new big data. http://blog.synyx.de/2014/11/time-series-data-is-the-the-new-big-data/, (visited January 2016).

[6] C. M. Dinu, F. Pop, and V. Cristea, "Pattern detection model for monitoring distributed systems," in *Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), 2011 13th International Symposium on*, Sept 2011, pp. 268–275.

[7] Citypulse dataset collection. http://iot.ee.surrey.ac.uk:8080/index.html, (visited May 2016).

[8] Y.-P. Huang, C.-C. Hsu, and S.-H. Wang, "Pattern recognition in time series database: A case study on financial database," *Expert Systems with Applications*, vol. 33, no. 1, pp. 199–205, 2007.

[9] O. R. Zaiane, "Principles of knowledge discovery in databases - introduction to data mining." Department of Computing Science, University of Alberta, 1999.

[10] V. Markl, "Breaking the chains: On declarative data analysis and data independence in the big data era," *Proceedings of the VLDB Endowment*, vol. 7, no. 13, pp. 1730–1733, 2014.

[11] Apache flink stack. https://flink.apache.org/, (visited May 2016).