

The Multi-Agent System Solutions for Big Multi-sensor Data Management

Zakarya Elaggoune *, Ramdane Maamri and Imane Boussebough

*Lire Laboratory, University of Constantine 2 – Abdelhamid Mehri
Constantine, Algeria, 25000*

Abstract

This research realizes the multi-agent community to handle the rinsing challenges in smart cities by the combined use of multi-fuzzy-agent systems for extracting relevant data from big noisy data, and the multi-agent systems-of-systems for adaptive big data processing. A multi-agent-based large wireless sensor network is used to show the approach used by the multi-fuzzy-agent systems, it exploits sensors as autonomous fuzzy agents that measure the relevance of the collected data and eliminate irrelevant ones. The aim of the multi-agent systems-of-systems is to handle the super velocity of big data flood in an adaptive way, which break down the rigidity of current big data systems that are forced to restart the data analysis process periodically to integrate the newly generated data into the analytical cycle.

Keywords: *Big data, Multi-agent system, Wireless sensor network, Fuzzy logic, System of systems, Relevant data*

1. Introduction

“smart city” is defined as the intelligent use of advanced technologies to detect, prepare, analyze and integrate big quantities of data sets of main systems in operating cities. Moreover, a smart city can bring clever reactions to different sorts of daily needs, including individuals livelihood, environment preservation, systems security, public health, transport monitoring, and commercial and industrial operations [1].

Smart cities represent an impending need, and are typically the real sort of “smart globe” applied to personalized areas to release the clever and integrated management of cities. In smart cities, different data sets are constantly analyzed to provide smart management, smart building models, and intelligent planning ideas, therefore, big data sets are treated as an engine of any smart system [2].

There are three principal levels to build a smart city - the perception level, the network level and the application level - [3], wherein each level massive data are analyzed and processed.

Big data is the fuel of smart cities, after data gathering process at the network level, usually, the data is transferred to a big data cloud server for applying data mining and machine learning technologies in order to discover behavior patterns or anomalies and then suggest the optimal solutions.

At the beginning of the Big Data era, three main challenges inherent to the characteristics of big data appeared (the "3V" initial Big Data):

1. Volume: data sets with enormous size and complexity (many features),
2. Velocity: fast generation of data arriving in continuous flows,
3. Variety: Different types of data come in different forms.

These challenges, also known as "data flood", have pushed storage systems and processing techniques to their limits at that time. After becoming familiar with the first three challenges, the new techniques began to perform well, but soon the flood of data overwhelmed these techniques. Indeed, as the volume of data increased and sources multiplied, raw data became increasingly poor and useful information became scarcer. Increasingly, the usefulness and reliability of data and their sources have been questioned. Hence the emergence of two new challenges taking the "3V" challenges of big data to "5V". [4] define the new two challenges as follows:

4. Value: the usefulness of the data or more precisely the amount of useful information among the flooded data,
5. Veracity: Reliability and confidence attributed to the data and its sources.

In this paper, we explore the possibility of using multi-agent systems to deal with these new big data. Figure 1 illustrates an overview of the integration of Multi-Agent Systems (MASs)

* Corresponding author. Tel.: +213 792620762

E-mail: zakarya.elaggoune@univ-constantine2.dz

© 2019 International Association for Sharing Knowledge and Sustainability.

DOI: 10.5383/JUSPN.11.02.004

into the big data decision support cycle. The different MASs and their roles are listed in the following paragraphs.

- *Preprocessing multi-agent group.* The flood of data brings with it a heap of incompleteness, uncertainty, redundancy, and irrelevant data. This noise must be filtered, and these data must be refined and transformed into smart data before the processing step. In other words, to ensure the value of information, transforming big data into smart data is a mandatory step. It is in this stage (preprocessing) that the role of multi-fuzzy-agent systems appears, where the fuzzy agents are distributed on the different nodes of big data servers. A fuzzy logic reasoning is applied by those agents to determine the rate of the relevance of data, and if the defuzzification output is above the percentage of relevance predetermined by the user, then the data will be considered as relevant, else it will be considered as irrelevant and it will be neglected. In addition, cooperation between agents is needed to eliminate distributed redundancy, cover incomplete data gaps, and measure data veracity.

The major novelties of using multi-fuzzy-agent systems are: (a) the ability to estimate relevance rate and filter out the noise through the use of MASs technology that executes fuzzy algorithm; (b) fuzzy agents look at the data in imprecise terms, then responds with precise actions (a precise problem-solving methodology), which can remedy the problem of incompleteness and uncertainty data; (c) fuzzy logic enables low-cost micro-controllers to perform functions traditionally performed by more powerful expressive machines enabling lower cost product to execute advanced features; and (d) the relevance rate threshold accepted is not constant, which means that this approach does not depend on a specific application, and users can configure the relevance rate threshold accepted taking into consideration the domain of the application (it is a generic approach).

- *Processing multi-agent group.* After the data refinement step comes the data processing step. Usually, data processing and analysis phase can only begin after grouping a large batch of data, because data mining techniques work better on large data sets. But nowadays, the speed at which the data is generated poses a problem at this level, where a large amount of data that hides behind a lot of information and knowledge is generated during the time taken in the analysis of the current batch, and the decision made by the end-user will not be based on real-time results, because a large amount of data is queued for processing. What is more, current big data systems restart the data mining process at each new batch to integrate incoming

data to the decision support cycle (rigidity in data management).

The aim of the multi-agent systems of systems is to adapt the big data processing process to support the newly arrived data without having to restart the process each time. The neighborhood between agents is defined by time, where two neighboring agents represents two successive periods. There is a coordinator agent in the head of each MAS, this last considers the others agents as independent systems, each with its own agenda and capabilities, and its main role is to wrap all these capabilities in a single global system. In this way, when new data arrives, it can be processed directly without stopping the entire process, and after that, the coordinator agent gathers the newly obtained results with the older ones in the form of one single batch.

To sum up, multi-agent systems of systems can handle the dynamism of big data in an adaptive way, which break down the rigidity of current big data systems.

With the recent increase in the number of smart and portable devices and other measuring instruments in ambient applications, smart cities and Internet of Things (IoT) [5], we are just beginning to address every aspect of this new big data. In the smart cities context, we can extract two main rising challenges from this new big data:

- *Gathering relevant information from large scale wireless sensor networks in an energy-efficient manner.* A Wireless Sensor Network (WSN) contains a big quantity of sensor nodes with small batteries, which are deployed randomly over an area to gather data. Considering that many of the acquired data could be redundant or irrelevant, gathering only useful data is an important factor to deal with this new big data. Furthermore, and based on the fact that processing data locally in the sensor node and eliminating unimportant data consumes much less energy than transmitting the raw data directly (the energy cost needed for sending 1KB over a range of 100m is approximately equal to that required to perform 03 million instructions), filtering out the noise (irrelevant data) contributes to increasing the quality of information in one hand, and reduce the energy consumption of each sensor node, which leads to the extension of the life-time of the WSN in other hand.
- *Managing the super velocity of the big data in an adaptive way.* One of the benefits of big data is the exploitation of a large amount of data in several

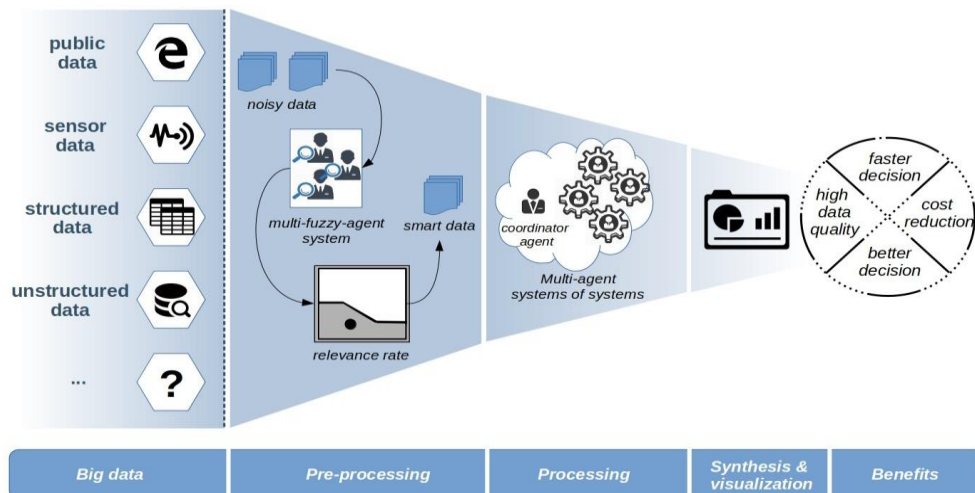


Fig. 1. An overview of the proposed approach

needs, like business strategies, medicine and marketing. For reliable data mining exploitation, the analytical tools stop the data mining algorithm to integrate the new data stream to the knowledge discovery process, and the process resumes after data acquisition. Restarting the processes periodically (each time fresh data arrives) consumes much energy, and it takes a long time, this is particularly true in the context of smart cities, where everything is monitored and data is streamed continuously to the servers for big data analysis. Accordingly, processing data continuously in an adaptive way is a necessary task.

The main issues of managing big data in smart cities are the energy preservation and the assurance of the value of the data gathered in the WSNs level; and the adaptation of data processing processes to support the dynamicity of the data in the servers level. To remedy these problems, a three-tier smart city system is suggested. The whole system is shown in Figure 2, and it consists of three parts:

1. Smart Wireless Sensor Network (SWSN). Which contains a group of fuzzy agents, where each agent is embedded in a sensor node. Neighboring agents communicate with each other to extract smart data. The fuzzy agent group ensures the value of the data gathered and increase the life-time of the network.

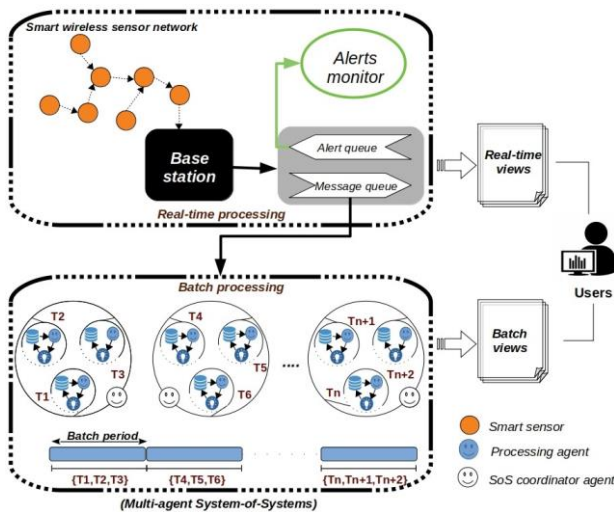


Fig. 2. A Smart City Architecture based MAS

2. Message Oriented Middleware (MOM). A MOM [6] is a distributed component that provides asynchronous messaging between the preprocessing module (SWSN) and processing module (the processing and storage system), providing independence of operation for both modules.
3. Multi-Agent System-of-Systems (MASoS). This group of agents is located in the processing module, the main role of this MAS is the continuous processing of data and the adaptation to the dynamism of the data, which offers real-time views to the end-users.

The rest of this paper is organized as follows: Section 3 is devoted to the smart wireless sensor network. Section 4 presents the multi-agent systems-of-systems model. The conclusion is reported in section 5.

2. First-Tier: Smart Wireless Sensor Network

WSNs have given rise to many research problems to improve network performance, including maximizing their lifetimes. WSNs literature recognizes that data is an important consumer of energy [7]. One technique to minimize energy consumption is the clustering technique; It is used to partition the network into groups with one Cluster-Head (CH) for each. The latter supports the exchange of data with the sink node (the base station).

During the last decade, multi-agent systems (MAS) have greatly developed, and are applied to different fields such as simulation, image processing and robotics. spider colonies and Ant agents are examples of MAS, which are applied in wireless sensor networks for gathering data, detection of shortest paths, routing, etc. Recent works have focused on the use of the paradigm of MAS to easily describe new types of sensor nodes and to allow more generic modeling.

Our goal is to design a smart wireless sensor network that collects a maximum of data and extract only the relevant ones to minimize energy consumption. To achieve this objective, we propose to integrate, into each CH, a fuzzy agent to process data, eliminate non-useful data, and reduce redundancy. Each CH in the network is seen as an autonomous fuzzy agent with its own attitudes and characteristics towards the different events they receive.

This solution is useful for intelligent processing of sensor data by the nodes in terms of assessment of the importance of data, redundancy reduction, and the elimination of irrelevant ones. For more efficiency, we have proposed the use of the clustering technique to send data easily to the Sink and for better organization. We can use the algorithm Low Energy Adaptive Clustering Hierarchy (LEACH) or any other efficient algorithm to decompose the network into clusters, each with a cluster-head.

2.1. Fuzzy Agent Role Behaviors

The aim of the WSN is to collect the maximum data and eliminate the irrelevant or redundant ones [8]. Preprocessing data locally in the sensor node and reducing the size of the information sent consumes less energy than sending raw data directly (without preprocessing) [9].

Each Cluster-Head in the network is associated with a fuzzy agent (FA), the principal role of the FA is to use fuzzy logic to estimate the relevance of the data and to eliminate the unimportant data. Hence, we have defined two main points fuzzy agents to extract the relevant information, which means to reduce the power of each node and to extend the life of the WSN:

1. Degree of the relevance of data: the degree of relevance of the data strongly depends on the desired application. This parameter is calculated locally in the sensor node. The fuzzy agent can estimate the degree of relevance of the data collected. This information is taken into account if it's the primary information containing the required information. For example, for air pollution monitoring, the node records the latest collected data to compare with the new ones collected. The fuzzy agent considers data as relevant if the difference between the two values is greater than a predetermined threshold. However, if the difference increases, the fuzzy agent consider that these data have a higher priority, so the degree of relevance increases.

- Inter-sensor-nodes redundancy elimination: typically, the sensor nodes are randomly deployed. so, many sensor nodes will cover the same geographical points, which means that they will give the same information (redundancy). In this case, the fuzzy agent will compare the values collected by each sensor node with its neighbors for eliminating the inter-sensor-nodes redundancy.

Figure 3 illustrates the fuzzy logic used by the agent to estimate the relevance of the data.

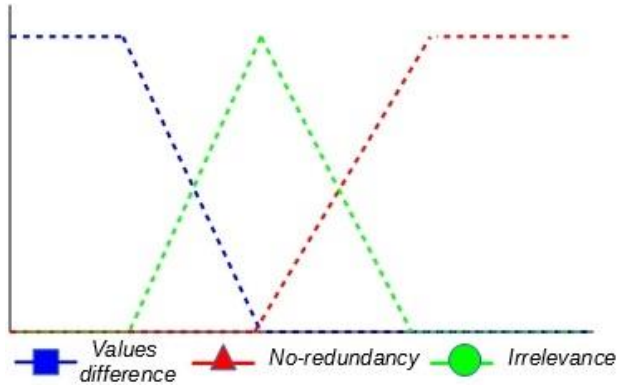
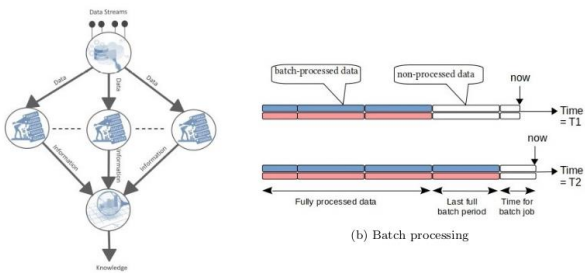


Fig. 2. Degree of the relevance of data

3. Second-Tier: Adaptive Multi-Agent System-of-Systems for Big Data Processing

The most used process for big data analysis is the distributed pipeline (Figure 4-a). This model has been proposed to circumvent the super velocity problem and reduce the processing time by means of parallelism. This pipeline is based on the MapReduce pattern and its famous Hadoop framework. However, this model relies on batch processing and does not really focus on real-time processing, which leaves always a portion of non-processed data (Figure 4-b).



(a) The distributed big data analytics pipeline
Fig. 3. MapReduce Model

Other architectures have extended this model, trying to support the real-time processing, in the following paragraphs, we will discuss the two most used architectures: Lambda Architecture and Kappa Architecture.

• **lambda Architecture (LA):** “The LA aims to satisfy the needs for a robust system that is fault-tolerant, both against hardware failures and human mistakes, being able to serve a wide range of workloads and use cases, and in which low-latency reads and updates are required. The resulting system should be linearly scalable, and it should scale out rather than up.” [10]

This is what it looks like, from a high-level point of view:

- All streamed data is sent to both the batch and the speed layers;
- The Batch layer pre-calculate the batch views;
- The serving layer indexes the batch views so that they can be queried in a low-latency way;
- The speed layer indemnify the high latency of updates to the serving layer and process only recent data;
- Any incoming query can be resolved by merging results from real-time and batch views.

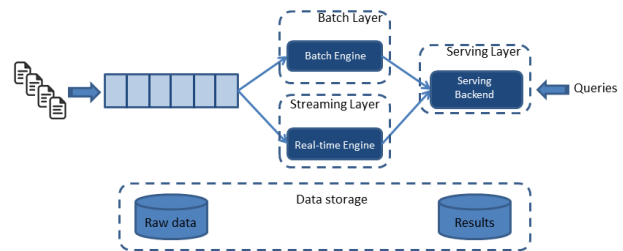


Fig. 4. Lambda Architecture

The idea behind these layers was to balance batch processing with real-time processing, where, the speed layer will provide real-time results into serving layer, and if any data will be missed during the streaming or if any wrong results will be produced by real-time processing, then batch job will compensate this lack of data and updates the serving layer, so providing accurate results. The only drawback of this architecture is that is very difficult to build the pipeline and maintain analysis logic in both batch and speed layer.

• **Kappa Architecture (KA):** “Kappa Architecture is a simplification of Lambda Architecture. A Kappa Architecture system is like a Lambda Architecture system with the batch processing system removed. To replace batch processing, data is simply fed through the streaming system quickly.” [11]

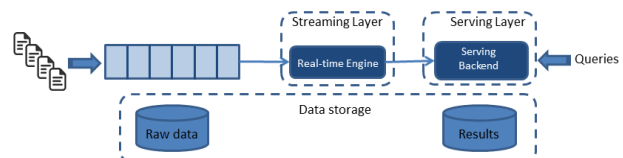


Fig. 5. Kappa Architecture

One of the disadvantages of the lambda architecture, as detailed above, is that the analyzer has to keep coding and executing the same logic twice, and this is avoided in the kappa architecture. The data storage system used in KA is more restricted and needs to be a log and unmodifiable file system because this type of data storage systems can hold the data for a while, so it can be reprocessed with another logic to discover additional knowledge. However, the kappa architecture should be considered an alternative to the lambda architecture only in applications that do not require unbounded retention data storage systems.

In addition to the limitations already cited of these three architectures, a common key disadvantage of these architectures is that they do not adapt with dynamic data, to put it another way, the processing process must periodically restart to integrate new data arriving, which results in high costs and slows down the decision.

3.1. Multi-Agent Systems-of-Systems for Big Data Processing

Systems of Systems (SoS) are systems that describe the large-scale incorporation of several independent self-contained

systems to fulfill global goals and multi-system requests [12]. The principal drivers behind the idea of systems of systems are various yet inter-related, namely,

- The growing number of companies that use multiple services from different systems, which led to increases in interacting systems.
- The need to balance autonomy and cooperation in interconnected systems.
- The complexity of systems is rising.

This kind of drivers is presented in the dimensions that define SoS, typically, the distributed environment of the systems, the autonomy and independence of each sub-system, and their evolutionary development as one global system.

In this section, we present a new model for dynamic big data processing based on multi-agent systems-of-systems.

Our MAS is composed of a group of worker agents distributed on the nodes of the storage system (each agent is deployed on a node). Each worker agent applies data analysis algorithms to its local data. Moreover, neighboring agents can cooperate with each other to perform distributed batch processing of data analysis.

The neighborhood of the agents is defined by time, from which two neighboring agents represent two successive periods, each agent represents a part of the batch, and a certain number of neighboring agents represents the full batch period (Figure 7).

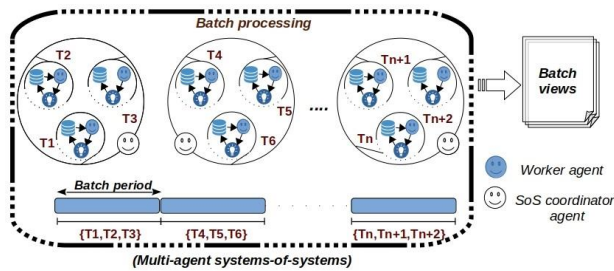


Fig. 6. Multi-agent systems-of-systems

The essence of the system-of-systems is the independence of the individual sub-systems, that is what led us to use a Multi-Agent System of Systems (MASoS) , where each agent is represented as a subsystem of the global system.

There is a coordinator agent in the head of each MASoS, this last considers the others agents as independent systems, each with its own agenda and capabilities, and its main role is to periodically wrap all these capabilities in a single global system. In this way, when new data arrives, it can be processed directly without stopping the entire process, and after that, the coordinator agent gathers the newly obtained results with the older ones in the form of one single batch.

There are several models of SoS, the MAS used in this research is based on the Wave model [13]. The wave model elements are shown in Figure 8 and described in more detail, below.

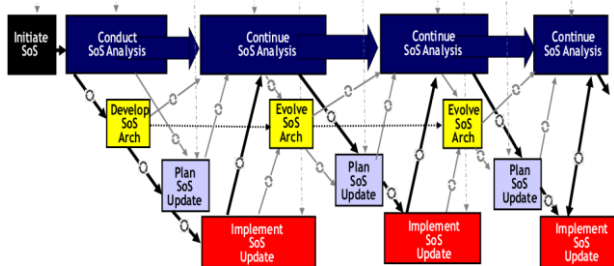
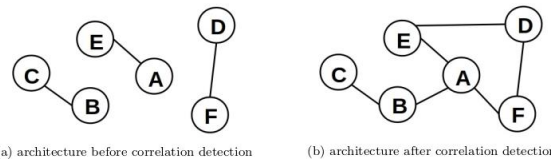


Fig. 7. The wave model of SoS development

The steps used in the MASoS model to integrate new incoming data in the data analysis cycle in an adaptive way, and avoid restarting the process each time are described below:

1. The coordinator agent **initiates the MASoS**, and provides the foundational information to start the data analysis process, including an understanding of objectives, the data analysis algorithms used, the beginning and the end of the full batch period, the worker agents included-in this period and the configuration of the neighborhoods.
2. The coordinator agent **conducts the MASoS analysis**, it communicates periodically with the worker agents to calculate the rate of evolution of the analyzes compared to the final objectives. When the MASoS reaches the final results, the coordinator agent stops the process.
3. Neighboring worker agents can interact with each other to **develop and evolve the MASoS architecture**. Basically, only one-step neighboring worker agents communicate with each other to perform distributed batch processing of data analysis, but, if two neighbor worker agent detects a high correlation (weak or strong) between the data stored in their nodes, then the neighborhood network will be extended, and these one-step neighboring worker agents will interact with neighbors of its neighbors in order to detect others potential correlations (Figure 9). The coordinating agent takes into account this development, and it will update the architecture in



(a) architecture before correlation detection (b) architecture after correlation detection

Fig. 8. An example of a possible MASoS architecture before and after correlation detection

the next iteration.

4. After each iteration, the coordinator agent **updates the MASoS plan**, it evaluates the volume of incoming new data and assigns a new worker agent to process this data if it is needed, it evaluates correlations and level of progress, and lastly, it defines the plan for the next MASoS upgrade cycle.
5. Afterwards, the coordinator agent **implements MASoS Updates**, which means that it communicates the new plan, the new architecture, and the new objectives to the worker agents, resulting in a new MASoS product baseline.
6. Workers agents continue data analysis, and the coordinator agent continues monitoring the MASoS involving ongoing analysis, and ongoing evolution of the system including the new data stream.

Another way to process a larger amount of data, which means discovering more relationships and correlations, is the expansion of the MASoS, where, each MASoS become a sub-system, and each group of neighboring sub-systems will configure the overall MASoS (Figure 10). This property is used to widen the batch period.

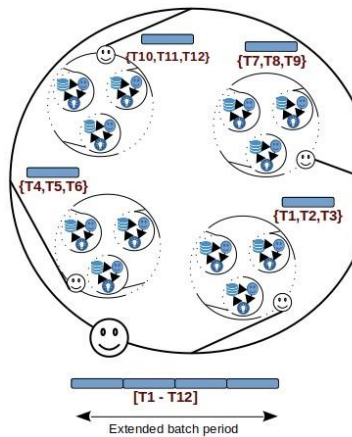


Fig. 9. Extended Multi-agent systems-

This MASoS can handle the dynamism (super velocity) of big data, which break down the rigidity of current big data systems. Moreover, unlike the MapReduce model, this model offers incremental real-time views to the user as shown in Figure 11.

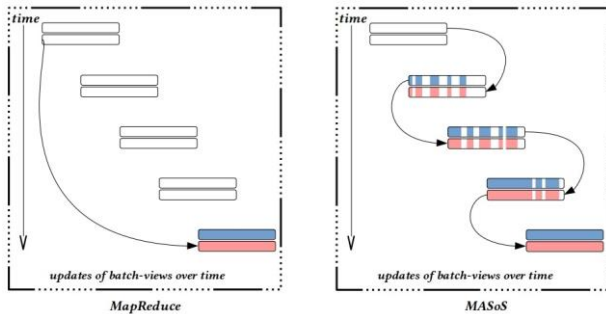


Fig. 10. Updates of batch-views over time

To sum up, it can be said that the proposed MASoS has several driving characteristics that allow it to handle the flooding data in a flexible and adaptive way, which avoids restarting decision support cycle with each new data streamed. These characteristics are listed below:

1. Multiple overlapping iterations of evolution;
2. Ongoing data analysis;
3. Continuous external data integration;
4. Architecture evolution;
5. The continuous update of the results.

4. Conclusion

This paper has proposed the use of multi-agent systems to deal with the growing challenges in the field of Big Data by the combined use of multi-fuzzy-agent systems for extracting smart data from big noisy data, and the multi-agent systems-of-systems for adaptive big data processing. To demonstrate the usefulness of our approach, we illustrated the use of the proposed system in the context of smart cities, where, a two-tier smart city driven by multi-agent systems was proposed. The first tier consists of a large-scale wireless sensor network powered by fuzzy agents. The main goal of the multi-fuzzy-agent system was the elimination of noisy data collected by the sensors based on the user-defined accepted relevance threshold. In the second tier, we described and discussed how multi-agent systems-of-systems can be applied to handle the super velocity of big data, it can break down the rigidity of current big data

systems and can process big data continuously without having to restart the analysis periodically.

The next stage of our research will be feeding fuzzy agents with reinforcement learning, where the perception of what constitutes a smart data will be particular to each agent, and each agent will learn to use the best methods to produce truthful smart data with the minimal energy. In the other hand, we will implement the MASoS and check the efficiency of the architecture compared to other big data architecture.

References

[1] M. I. Pramanik, W. Zhang, R. Y. K. Lau, C. Li. A framework for criminal network analysis using big data, in: 2016 IEEE 13th International Conference on e-Business Engineering (ICEBE), 2016, pp. 17–23. doi:10.1109/ICEBE.2016.015. <https://doi.org/10.1109/ICEBE.2016.015>

[2] M. I. Pramanik, R. Y. Lau, H. Demirkan, M. A. K. Azad, Smart health: Big data enabled health paradigm within smart cities, Expert Systems with Applications 87 (2017) 370 – 383. <https://doi.org/10.1016/j.eswa.2017.06.027>

[3] K. Su, J. Li, H. Fu, Smart city and the applications, in: 2011 International Conference on Electronics, Communications and Control (ICECC), 2011, pp. 1028–1031. doi:10.1109/ICECC.2011.6066743. <https://doi.org/10.1109/ICECC.2011.6066743>

[4] H. V. Jagadish, J. Gehrke, A. Labrinidis, Y. Papakonstantinou, J. M. Patel, R. Ramakrishnan, and C. Shahabi, “Big data and its technical challenges,” Commun. ACM, vol. 57, no. 7, pp. 86–94, Jul. 2014. [Online]. Available: <http://doi.acm.org/10.1145/2611567>

[5] M. Samaniegoa, R. Deters, Virtual Resources & Blockchain for Configuration Management in IoT in: Journal of Ubiquitous Systems & Pervasive Networks , Volume 9, No. 2(2017) pp. 1-13.

[6] C.-C. Hung, C.-C. Hsieh, Chapter 5 - big data management on wireless sensor networks, in: H.-H. Hsu, C.-Y. Chang, C.-H. Hsu (Eds.), Big Data Analytics for Sensor-Network Collected Intelligence, Intelligent Data-Centric Systems, Academic Press, 2017, pp. 99 – 116. <https://doi.org/10.1016/B978-0-12-809393-1.00005-2>

[7] Sardouk, A., Rahim-Amoud, R., Merghem-Boulaiah, L., Gaiti, D., 2009. Information-importance based communication for large-scale wsn data processing, in: Wozniak, J., Konorski, J., Katulski, R., Pach, A.R. (Eds.), Wireless and Mobile Networking, Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 297–308. https://doi.org/10.1007/978-3-642-03841-9_27

[8] V.K. Singha, S. Vermaa, M. Kumara, Evaluation of Privacy Preserving In-Network Aggregation for Different Routing Structures in WSNs in: Journal of Ubiquitous Systems & Pervasive Networks , Volume 9, No. 2(2017) pp. 15-19

[9] Akyildiz, I., Su, W., Sankarasubramaniam, Y., Cayirci, E., 2002. Wireless sensor networks: a survey. Computer Networks 38,393–422. [https://doi.org/10.1016/S1389-1286\(01\)00302-4](https://doi.org/10.1016/S1389-1286(01)00302-4)

[10] M. Hausenblas, N. Bijnens, Lambda architecture. URL <http://lambda-architecture.net/>

- [11] S. Uesugi, Kappa architecture. URL <http://milinda.pathirage.org/kappa-architecture.com/>
- [12] E. Alonso, Multi-agent systems : A new paradigm for systems of systems, in: Eighth International Conference on Systems 2013 (ICONS 2013).
- [13] J. Dahmann, G. Rebovich, J. Lane, R. Lowry, K. Baldwin, An implementers' view of systems engineering for systems of systems, in: Systems Conference (SysCon), 2011 IEEE International, IEEE, 2011, pp. 212–217. <https://doi.org/10.1109/SYSCON.2011.5929039>