Intelligent Data Classification and Aggregation in Wireless Sensors for Flood Forecasting System

El Mabrouk Marouane
LaSIT Laboratory
Faculty of Sciences
UAE
Tetuan, Morocco

Ezziyyani Mostafa
LaSIT Laboratory
Faculty of Sciences and Technologies
UAE
Tangier, Morocco

Essaaidi Mohamed
LaSIT Laboratory
Faculty of Sciences
UAE
Tetuan, Morocco

***Abstract*—Flood is a major natural hazard in the world. For the period 1996-2005, about 80% of global natural disasters were meteorological or hydro. The floods have affected an average of 66 million people per year between 1973 and 1997. By these statistics, floods are considered the disasters that produce the most damage. Thats why we must handling floods data with great caution since human life is at stake. In this paper, we present an intelligent model that gathers data received from the wireless sensors and reach them in an intelligent way on one hand, on the other hand, it detects erroneous or redundant data to classify them to just have reliable and adequate data to be stored in the database in order to be processed in the decision support system for real time flood forecasting by using of the multi-agent system (MAS) to process data, eliminate redundancy, non-useful and erroneous data and establish collaboration between mobile agents to send the results to the base station.**

Keywords*—flood, forecasting, wireles sensor network, multi-agent system, jade, mobile agent

I. Introduction

Technological advances in the field of wireless communications and microelectronics during the last decade, allowed the emergence of wireless sensor networks (WSN) [1]. They are composed of a large number of nodes, often deployed randomly and densely in the interested field. [2]

WSN consists of a sensor assembly, provided with a capture unit, a microcontroller, a battery, a radio module and a storage unit. The capture unit collect physical data of the environment, temperature, pressure, etc [3]. The transmission unit transmits the collected data across a radio link to the base station, through byte packets based on a communication protocol. The nature of the data collected depends on the application and the sensor type used. The application areas of WSNs are many and varied (health, environment, agriculture, military, etc.) [4]

However, [5] WSNs are characterized by limited capacity in terms of computing potency, storage, and energy, which represents a challenge for designing algorithms and programs for this type of networks, thus the importance for researchers to have tools for modeling and simulation to study and evaluate their performance. [6]

For multi-agent approach, it showed its strength for distributed problems for which it is difficult to anticipate all the situations encountered. The division of an application to decentralized cooperative entities transforms the analytical problems from a global level and a local level and reduces the design complexity. [7]

Distributed systems involve many entities, often autonomous from an energy point of view. These entities are hardware (sensors) and software (mobile agents).They are responsible for complex tasks with very different natures: measurement acquisition, action, behavior, computing and communication tasks. [8]

II. Using Mobile Agents and Multi-Agent Systems in WSN

In this part, we introduce the wireless sensor networks, the multi-agent system concept as well as use of the multi-agent systems in wireless sensor networks.

A. Wireless Sensor Networks (WSN)

The wireless sensor network is a particular family of ad-hoc networks [9]. It consists of autonomous sensors distributed randomly in an area of interest in order to monitor physical or environmental parameters (eg. Temperature, sound, pressure, etc.) and transmit data via the network to the base station. [10] [11]

Figure 1 shows an exemple of wireless sensor networks :

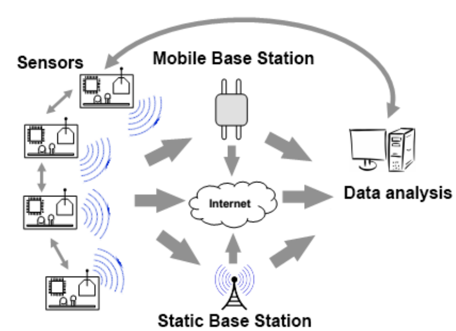


Fig. 1: Example of wireless sensor network

978-1-4799-7391-0/14/$31.00 ©2014 IEEE

Sensor is an electronic component that can collect data and transmit them to a coordinator node or a base station in full autonomy. [11]

It is composed of four basic components which are:

- Microcontroller (a programmable integrated circuit).
- Radio module for wireless communication.
- Battery.
- Storage unit

Figure 2 shows an exemple of sensor :

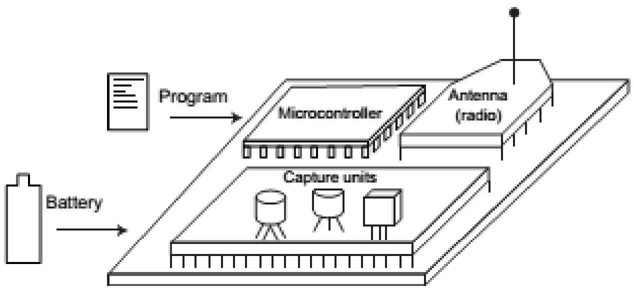


Fig. 2: Example of Sensor

WSN are characterized by limited capacity in terms of computing potency, storage, and energy, which represents a challenge for designing algorithms and programs for this type of networks, thus the importance for researchers to have tools for modeling and simulation to study and evaluate their performance [6]. here it comes the importance of combining between multi-agent systems and wireless networks of sensors to have best results in terms of collaborative decisions

B. Multi-Agent Systems (MAS)

Multi-agent systems represent a technology of choice for design and implementation of distributed and collaborative applications. A multi-agent system (MAS) is a set of agents which evolve in a common environment. [12]

This set of agents, not necessarily intelligent, constitutes a complex system which unlock a collective intelligence. This intelligence comes from the emergence of a global behavior. [12]

An agent is a reactive computing entity, pro-active and with social skills, able to act autonomously in an environment [13]. Responsiveness refers to the maintenance of a constant link with its environment to meet the changes that occur. Pro-activity means that the system generates and satisfies goals: his behavior is not directed only by events. Social skills indicate that the system is able to interact or cooperate with other systems. Agents can be described at different levels of abstraction: model, architecture and implementation. The agent model describes the agent, its properties and how it can be represented. The architecture is an intermediate level between the model and implementation. It organizes the construction of the agent, its capabilities and its control, accordance with the model. The implementation provides the practical realization of the agents architecture using languages or programming tools. [12]

C. Using Mobile Agents and Multi-Agent Systems in WSN

Smart components are found in several fields of network management [11] : configuration, security, breakdowns and accounting. This intelligence comes usually from distributed artificial intelligence.

Recently, there are more and more attempts to confrontation between the two research areas that are multi-agent networks and wireless sensor systems. The distributed and open nature of sensor networks makes the multi-agent approach particularly suitable. The fact that this approach provides an external representation of the interactions and organization offers several features such as control by an external observer. We have identified several areas of multi-agent systems applications in the context of wireless systems: routing information, data fusion, research and description of services. [11]

Data fusion

[14] The agents are used to achieve an open and flexible assembly cell structure in which data collected by the agents are merged to create a virtual environment. The Unmanned Ground Vehicle Program ARPA project [15] uses multi-agent approach to merge data from several military vehicles in order to have a global view of the operating field from different local views.

Research and description of services

Research and description of services is a very active research area where the multi-agent systems are used in the context of ad hoc networks [10] [16] [17] [18] [19]. FIPA [20], who works on the standardization of interactions in multi-agent systems, is also interested in this family of communication networks [21]. Some works on this standard have been made and some interesting results were measured for certain types of ad hoc networks [22].

The routing information

Few works, dealing with the routinog informatin problem in ad hoc networks of autonomous physical entities, using multi-agent systems. However, we can cite related works using mobile and focused on optimization of message delivery time agents [23]. Its technique relies on management of affinities between agents (similar to routing techniques called link state). The ActCom military project [24] in which the routing information is essential: the objective is to manage the communications exchanged between soldiers and between soldiers and headquarters via satellite. Other approaches of distributed artificial intelligence related to multi-agent systems are used.

III. Proposed model of data classification, aggregation and wireless sensor filtering for real-time flood forecasting

This section presents the main idea of the classification of data received from wireless sensors using multi-agents in order to filter data for the real-time floods forecasting.

The problem we must deal with is that data received from wireless sensors may be erroneous, redundant or non-useful, but our model is a model of real-time flood forecasting, so we treat a disaster that can put society in danger, so we must be sure of data we are working with in order to provide the best decisions in a timely manner.

In order to do it, we propose an intelligent model of classification and aggregation for data received from wireless sensors, the model divide data into two classes valid and invalid using a proposed algorithm for data comparison. The data obtained are reliable. We used the multi-agent systems for the processing of the classification and aggregation according to the proposed algorithm. the system has five agents :

- Verification Agent : responsible for verifying the functional and non-functional sensors.
- Rainfall Agent : responsible for classifying and aggregating data received from the Rainfall sensor.
- Flow Velocity Agent : responsible for classifying and aggregating data received from the Flow Velocity sensor.
- Water Level Agent : responsible for classifying and aggregating data received from the Water Level sensor.
- Interaction Database Agent : responsible for saving the valid and invalid data in their database and to monitor the number of times a sensor has sent erroneous data to change the state of the sensors by changing its Flag.

Figure 3 summarizes the main idea of the classification :

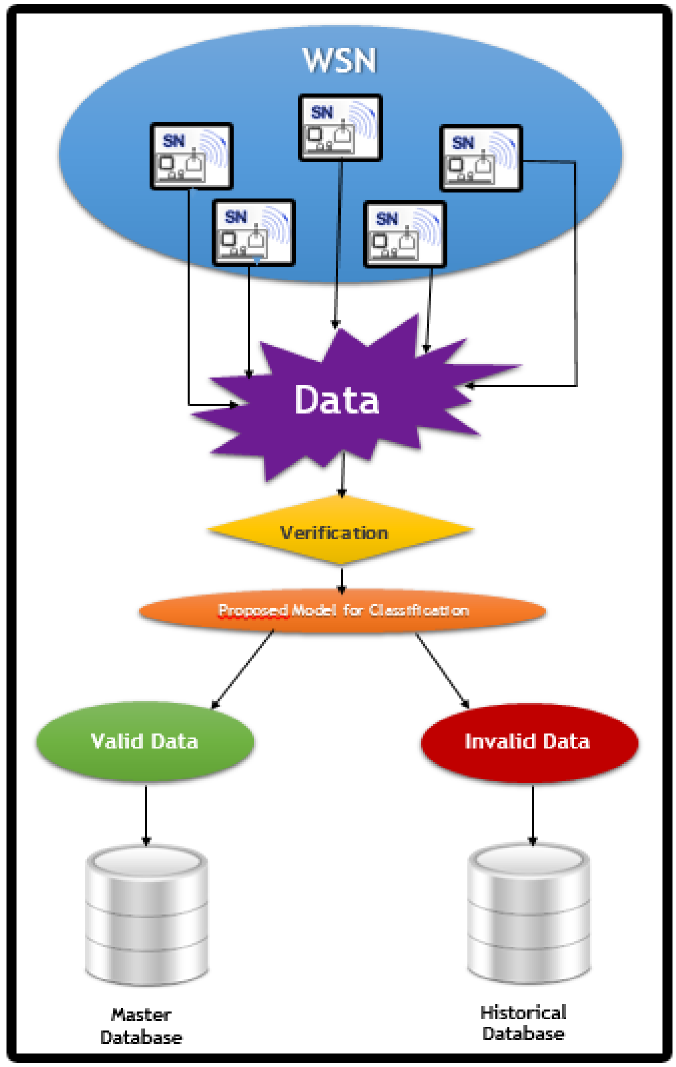


Fig. 3: Main idea of the proposed model

Figure 4 shows the architecture of the proposed model :

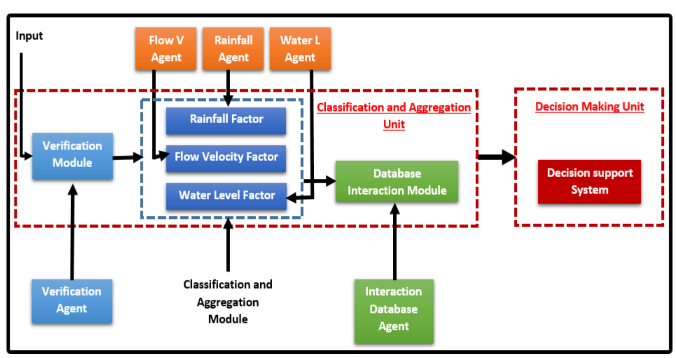


Fig. 4: Proposed model architecture

In the next section, we will present and we will explain all the phases that constitute our proposed model.

A. Sensors verification phase

Before sending data received from the wireless sensors to our model, it access a very important step. This step serves to identify the sensors that are functional and non-functional and also serves to identify the sensors that will be considered by the model when processing data.

Identification is made by using a table called “Verification” which have two fields: SENSOR_ID and FLAG, SENSOR_ID is the number ID of the sensor in the wireless sensors network, FLAG is a boolean field, it takes “No” if the sensor is non-functional so it needs maintenance or change, therefore, the system sends warning messages on the platform, emails and sends also SMS to managers in order to take the necessary, and it takes “Yes” if the sensor is functional, so data sent from sensors with FLAG = “Yes” will be taken into consideration during the processing of the classification.

B. Data Aggregation and Classification phase

$E_x = \{D_1, D_2, D_3, ..., D_n\}$ is the vectors set of all data received from sensors installed in the region with $D_n = \{d_1, d_2, d_3, ..., d_m\}$ a vector of “m” data sent by sensor “n” at instant t.

The first step for the model is to calculate a square matrix for the comparison and classification as follows:

$$M_C = \begin{pmatrix} 0 & & & & \\ T_{(d2,d1)} & 0 & & & \\ T_{(d3,d1)} & T_{(d3,d2)} & 0 & & \\ T_{(d4,d1)} & T_{(d4,d2)} & T_{(d4,d3)} & 0 & \\ \vdots & \vdots & \vdots & \ddots & 0 \\ T_{(dn,d1)} & T_{(dn,d2)} & T_{(dn,d3)} & \cdots & T_{(dn,dm)} \end{pmatrix}$$

With :

$$T_{(d_i,d_j)} = \sum_{i=2}^{n} \sum_{j=1}^{i} \left(1 - \frac{\min(d_i, d_j)}{\max(d_i, d_j)}\right) \quad (1)$$

When the matrix is ready to be used, during the classification, the model distinguish the data received from wireless sensor into two list valid and invalid. To provide the possibility to the model to distinguish between valid and invalid data, we proposed a tolerance percentage to set a a tolerance margin for data, it’s through this percentage that we can identify data that are on or outside the tolerance interval and therefore have two lists of valid and invalid data.

The tolerance percentage is calculated by the following formula for each sensor :

$$P_{TC} = \frac{(2 \times \Delta_C) + T_F}{100} \quad (2)$$

With :

- Δ_C : Uncertainty designated by the fabricator company.
- T_F : Uncertainty that may be attached to a factor in the study area (Precipitation, Flow, ...)

So, after using the tolerance percentage in the comparison and classification process, we obtain two valid and invalid class, now it remains only the database interaction phase. The next section presents the data backup processes in the master database and historical database.

C. Database Interaction phase

For valid class, the model takes the closest data to the average of valid data. Firstly, it calculate the average and the distance between the average each valid value, then it compare the distances, finally it takes the data whose distance is the smallest of all the calculated distances, and at this point, we finally obtained the most appropriate value of all the valid values.

For the invalid class, since we have a historical database, firstly it save the invalid data and the sensor ID for each sensor, then it calculates how many times a sensor sent erroneous data, since we deal with sensitive catastrophe, then the error rate is fixed at five times, finally if a sensor has sent invalid data five times or more, we deactivate it by changing its FLAG in the “Verification” table in order to indicate to the managers to maintain it or change it.

D. Algorithm of Proposed Model

In this section, we will present the algorithm of proposed model for data classification and aggregation of flood forecasting system. This is the proposed algorithm:

Fonction VerificationFunctionalSensors(Flag)

```
Data : List SensorsFunctionalList;
begin
    SensorsList ← ResearchFunctionalSensorsDB(Flag);
    return SensorsFunctionalList;
end
```

Fonction VerificationNonFunctionalSensors(Flag)

```
Data : List SensorsNFunctionalList;
begin
    SensorsList ←
    ResearchNonFunctionalSensorsDB(Flag);
    return SensorsNFunctionalList;
end
```

Procédure SavingFinalValidData(ValidDataList)

```
Data : Double Average; Double FinalValidData;
begin
    Average ←
    CalculateAverageDataValidList(ValidDataList);
    FinalValidData ← CalculateSmallestDistanceAver-
    ageData(ValidDataList,Average);
    SavingDB(SensorID, FinalValidData);
end
```

Procédure SavingNonValidData(NonValidDataList)

```
begin
    SavingDB(SensorIDList, NonValidDataList);
    ErroneousNumberTime(SensorIDList);
end
```

Procédure ErroneousNumberTime(SensorIDList)

```
begin
    foreach Sensor do
        if Erroneous Number Time of This Sensor <= 5
        then
            ErroneousNumberTime++;
        else
            ChangeFlag(SensorID);
        end
    end
end
```

Procédure ChangeFlag(SensorID)

```
begin
    EditFlag(SensorId, False);
end
```

IV. Results and analysis

the aim of our application is to simulate the proposed model using multi-agent systems through the JADE framework for classification and aggregation of data received from the wireless sensors. As already mentioned in the previous sections, our model consists of three phases: verification, classification & aggregation and finally database interaction, each phase is managed by one or more agents to complete the process of the phase.

since we work on a disaster which is very dangerous and can expose human life in danger, so we set the broken down rate to 80%, however it has fixed the number of sensors for each factor to 5, therefore if more than one sensor is broken down, we must absolutely doing the necessary maintenance and repair.

Now, we will introduce the application in which we have implemented our proposed model while simulating with the dataset presented in the following tables:

Rainfall Sensor Data							
Experiences	D_1	D_2	D_3	D_4	D_5	Δ_C	T_C
1	20	**30**	22	21	24	1.2	5
2	**20**	36	37	39	38	1.2	5
3	41	43	**70**	42	40	1.2	5
4	50	54	51	52	**42**	1.2	5
5	61	62	60	**73**	64	1.2	5
6	**50**	70	71	74	72	1.2	5
7	81	82	84	80	**10**	1.2	5
8	91	**03**	90	92	94	1.2	5
9	4	2	**27**	1	5	1.2	5
10	107	105	109	**84**	108	1.2	5
11	20	**30**	22	21	24	2	10
12	**20**	36	37	39	38	2	10
13	41	43	**70**	42	40	2	10
14	50	54	51	52	**42**	2	10
15	81	82	80	**73**	84	2	10
16	**50**	70	71	74	72	2	10
17	81	82	84	80	**10**	2	10
18	91	**03**	90	92	94	2	10
19	4	2	**27**	1	5	2	10
20	107	105	109	**84**	108	2	10

Algorithme 1 : DataClassificationAggregationAgentSensors

```
Data : List SensorsFunctionalList;
List SensorsNFunctionalList;
List FunctionalDataList;
List ValidDataList;
List NonValidDataList;
List DataSensorFileList;
Matrix M;
Double P_TC;
Double Δ_C, T_F;
begin
    SensorsFunctionalList ←
    VerificationFunctionalSensors(True);
    SensorsNFunctionalList ←
    VerificationNonFunctionalSensors(False);
    if SensorsNFunctionalList <> empty then
        MessageWarningSMSEmail(SensorsNFunctionalList);
    end
    if SensorsFunctionalList <> empty then
        DataSensorFileList ← ReadFile(File);
        foreach Sensor Functional from
        DataSensorFileList do
            FunctionalDataList ←
            TempList[DataSensorFunctional];
        end
        for i ← 2 To FunctionalDataList.size() - 1 do
            for j ← 1 To i do
                M[i, j] ←
                1 − Min(ValidDataList[i],ValidDataList[j]) / Max(ValidDataList[i],ValidDataList[j]);
            end
        end
        P_TC ← ((2×Δ_C)+T_F) / 100;
        for i ← 2 To FunctionalDataList.size() - 1 do
            for j ← 1 To i do
                if P_TC − M[i, j] ≥ 0 then
                    if This sensor does't exist in the list
                    then
                        ValidDataList[i] ←
                        FunctionalDataList[i];
                    end
                end
                if This sensor does't exist in the list then
                    NonValidDataList[i] ←
                    FunctionalDataList[i];
                end
            end
        end
        SavingFinalValidData(ValidDataList);
        SavingNonValidData(NonValidDataList)
    end
    Print("No sensor to handle !!!");
end
```

Flow Velocity Sensor Data							
Experiences	D_1	D_2	D_3	D_4	D_5	Δ_C	T_C
1	66	64	67	68	**55**	2	9
2	129	**100**	130	128	131	2	9
3	7	5	6	**35**	8	2	9
4	62	61	**88**	60	64	2	9
5	81	82	80	**73**	84	2	9
6	136	138	**80**	137	139	2	9
7	55	57	54	56	**91**	2	9
8	**123**	91	90	92	94	2	9
9	41	**19**	45	43	42	2	9
10	177	179	176	**116**	178	2	9
11	66	64	67	68	**55**	5	18
12	129	**100**	130	128	131	5	18
13	7	5	6	**35**	8	5	18
14	62	61	**88**	60	64	5	18
15	81	82	80	**73**	84	5	18
16	136	138	**80**	137	139	5	18
17	55	57	54	56	**91**	5	18
18	**123**	91	90	92	94	5	18
19	41	**19**	45	43	42	5	18
20	177	179	176	**116**	178	5	18

Water Level Sensor Data							
Experis	D_1	D_2	D_3	D_4	D_5	Δ_C	T_C
1	**4.58**	1.20	1.21	1.19	1.18	2	8.5
2	1.66	1.68	1.67	**2.68**	1.69	2	8.5
3	0.16	**1.95**	0.18	0.17	0.19	2	8.5
4	0.82	0.80	0.81	0.84	**2.37**	2	8.5
5	2.51	2.52	**0.54**	2.50	2.54	2	8.5
6	**1.45**	0.33	0.31	0.34	0.32	2	8.5
7	0.61	0.62	0.64	0.60	**2.84**	2	8.5
8	3.31	**3.01**	3.30	3.34	3.32	2	8.5
9	1.33	1.31	**2.57**	1.34	1.30	2	8.5
10	4.50	4.52	4.51	**0.84**	4.54	2	8.5
11	**4.58**	1.20	1.21	1.19	1.18	4	16
12	1.66	1.68	1.67	**2.68**	1.69	4	16
13	0.16	**1.95**	0.18	0.17	0.19	4	16
14	0.82	0.80	0.81	0.84	**2.37**	4	16
15	2.51	2.52	**0.54**	2.50	2.54	4	16
16	**1.45**	0.33	0.31	0.34	0.32	4	16
17	0.61	0.62	0.64	0.60	**2.84**	4	16
18	3.31	**3.01**	3.30	3.34	3.32	4	16
19	1.33	1.31	**2.57**	1.34	1.30	4	16
20	4.50	4.52	4.51	**0.84**	4.54	4	16

A. Sensors verification phase

The platform alerts users that these sensors shonw in the message are broken down, so it sends text messages SMS and Emails to managers to repair and maintain or to change sensors reported and it displays functional and non functional sensors, after that the Verification Agent send an ACL Message to inform the Rainfall Agent, the Flow Velocity Agent and Water Level Agent that the verification phase is completed and these agents must begin the data Aggregation and Classification phase.

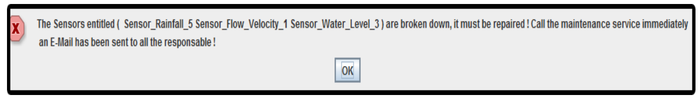


Fig. 5: Sensors verification phase

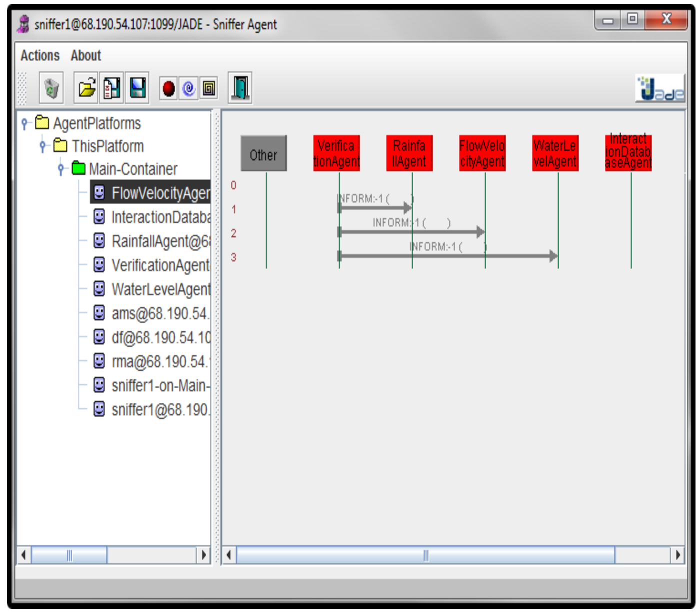
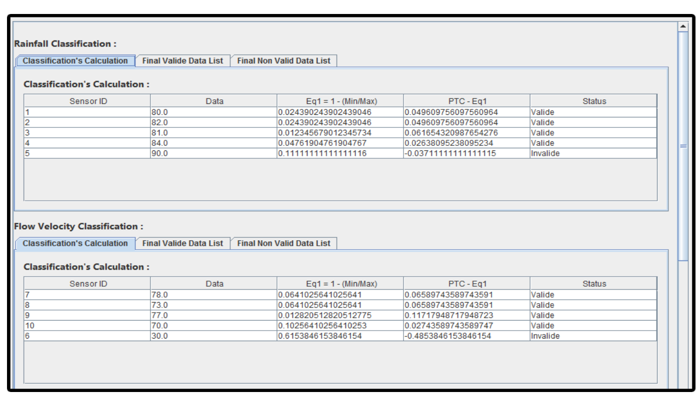


Fig. 6: Agents communication durant the verification phase

B. Data Aggregation and Classification phase

As already mentioned in the previous sections, this phase serves to do the data Aggregation and Classification process by using the three agents. The plateform shows all the process with all calculation made, it shows the classification calculation, the final valid data and the final invalid data. The three agents sends an ACL Message to inform the Interaction Database Agent that the data Aggregation and Classification phase is completed and this agent must begin the database interaction phase.



Rainfall Classification :

Classification's Calculation | Final Valide Data List | Final Non Valid Data List

Classification's Calculation :

Sensor ID	Data	Eq1 = 1 - (Min/Max)	PTC - Eq1	Status
1	80.0	0.024390243902439046	0.049609756097560964	Valide
2	82.0	0.024390243902439046	0.049609756097560964	Valide
3	81.0	0.012345679012345734	0.061654320987654276	Valide
4	84.0	0.04761904761904767	0.02638095238095234	Valide
5	90.0	0.11111111111111116	-0.03711111111111115	Invalide

Flow Velocity Classification :

Classification's Calculation | Final Valide Data List | Final Non Valid Data List

Classification's Calculation :

Sensor ID	Data	Eq1 = 1 - (Min/Max)	PTC - Eq1	Status
7	78.0	0.0641025641025641	0.06589743589743591	Valide
8	73.0	0.0641025641025641	0.06589743589743591	Valide
9	77.0	0.012820512820512775	0.11717948717948723	Valide
10	70.0	0.10256410256410253	0.02743589743589747	Valide
6	30.0	0.6153846153846154	-0.4853846153846154	Invalide

Fig. 7: Computing steps for classification

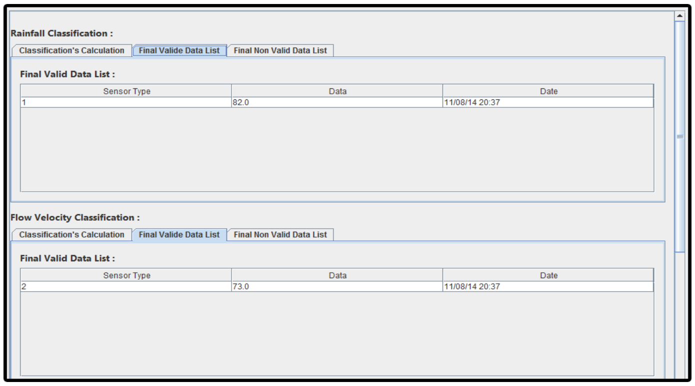


Fig. 8: Final Valid Data

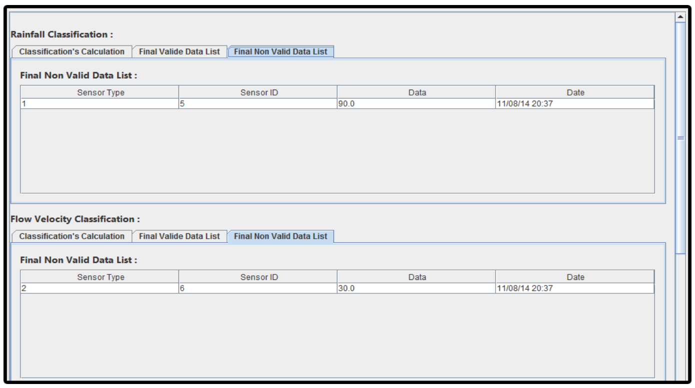


Fig. 9: Final Non Valid Data

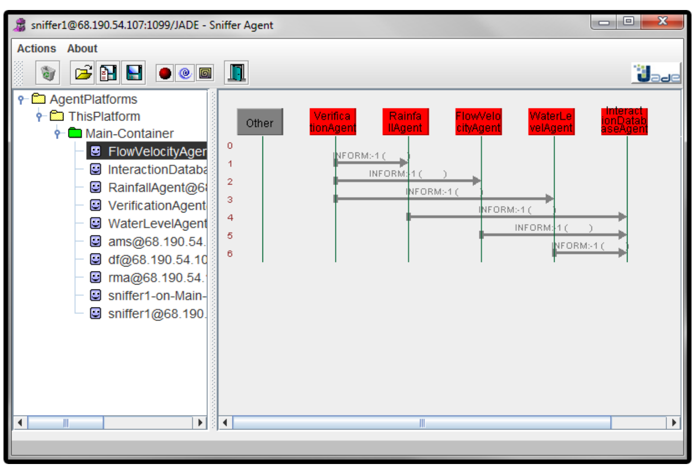
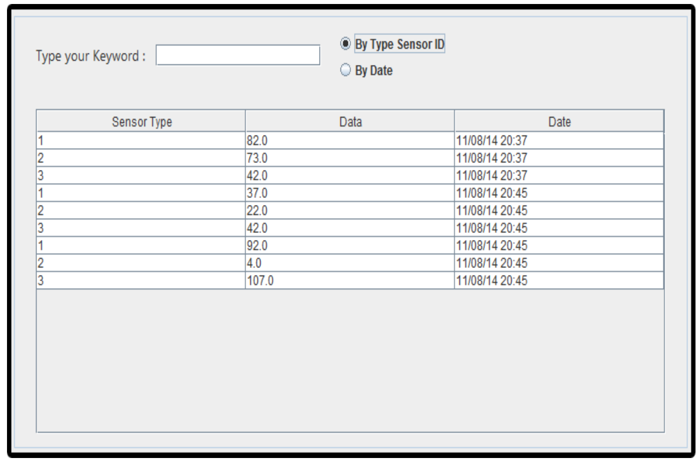


Fig. 10: Agents communication durant the data Aggregation and Classification phase

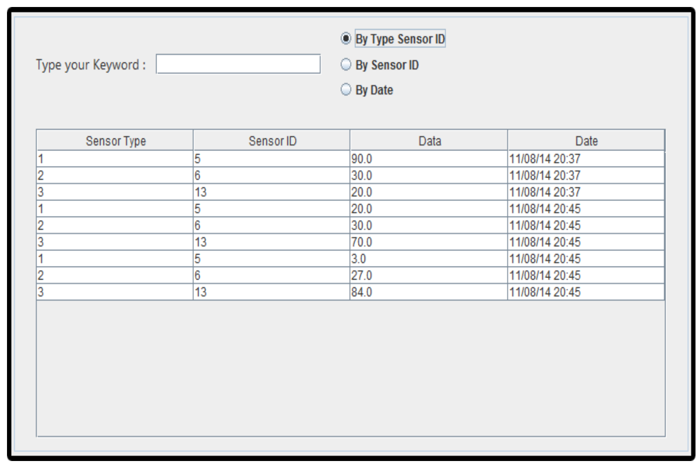
C. Database Interaction phase

As already mentioned in the previous sections, this phase serves to save the final valid and invalid data in their data bases. The plateform shows the final valid list and the final invalid list.The Interaction Database agent sends an ACL Message to itself to inform that the database interaction phase is completed and the whole process is complited and the classification process must be stopped.



Sensor Type	Data	Date
1	82.0	11/08/14 20:37
2	73.0	11/08/14 20:37
3	42.0	11/08/14 20:37
1	37.0	11/08/14 20:45
2	22.0	11/08/14 20:45
3	42.0	11/08/14 20:45
1	92.0	11/08/14 20:45
2	4.0	11/08/14 20:45
3	107.0	11/08/14 20:45

Fig. 11: Valid Data List



Sensor Type	Sensor ID	Data	Date
1	5	90.0	11/08/14 20:37
2	6	30.0	11/08/14 20:37
3	13	20.0	11/08/14 20:37
1	5	20.0	11/08/14 20:45
2	6	30.0	11/08/14 20:45
3	13	70.0	11/08/14 20:45
1	5	3.0	11/08/14 20:45
2	6	27.0	11/08/14 20:45
3	13	84.0	11/08/14 20:45

Fig. 12: Non Valid Data List

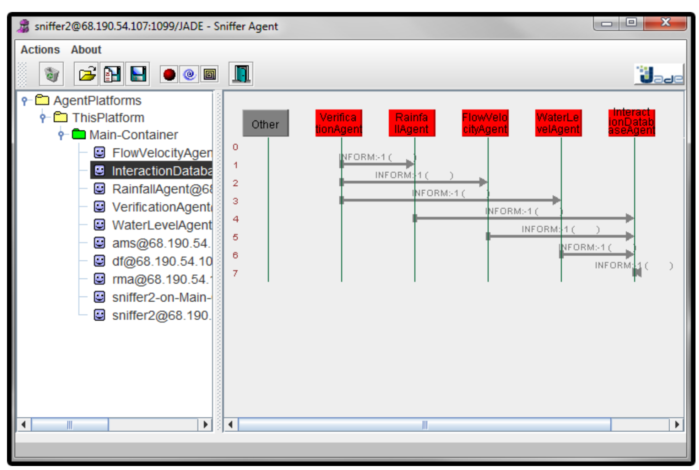


Fig. 13: Agents communication durant the database interaction phase

From the experiments already performed in the simulation phase, we noticed that when we increase the Δ_C and T_C the error rate also increases and vice versa, however, we treat a disaster that could put human lives in danger so we need to handle these data with great caution, thus the tolerance rate parameters must be reduced in order to select only data that are reliable.

V. Conclusion

In this paper, we have presented a new optimal model based on multi-agent systems for treatment of Dataset received from wireless sensors in order to classify data into two classes Valid and Invalid data to overcome the flood disaster. To make it, we proposed an optimal model for classification and aggregation which is divided into 3 phases :

- Sensors verification phase : to identify sensors that are not functional to report this breakdown to managers and to eliminate data received from non-functional sensors from process of classification to get only reliable data.
- Data Aggregation and Classification phase : for the classification process by using the proposed model already presented in this paper.
- Database Interaction phase : finally saving the valid and invalid data in their database by using the process already expressed in this paper.

to provide this data to the decision support system to decide about the impact of flooding since this disaster is the most expensive type of natural disaster regarding the damages.

The further work will focus on the implementation of a simulator of the Historical and Real Time Dataset model to simulate the results and to do necessary comparative studies to finalize the decision support system for real time flood forecasting.

References

[1] E. A. L. X. L. Philip Baldwin, Sanjeev Kohli and Y. Zhao, "Visualsense : Visual modeling for wireless and sensor network systems," in *UCB ERL Memorandum*, UCB/ERL, 2005, p. M05/25.

[2] A. F. G. Chelius and E. Fleury, "Worldsens: Development and prototyping tools for application specific wireless sensors networks," in *Proceeding of the International Conference on Information Processing inSensor Networks*, Boston, USA, 2007.

[3] M. P. L. Z. Gilbert Chen, Joel Branch and B. Szymanski, "Sense : A wireless sensor network simulator," *Advances in Pervasive Computing and Networking, Springer US*, pp. 249–267, 2005.

[4] M. S. A. Kpke and K. Wessel, "Simulating wireless and mobile network in omnet++ the mixim vision," in *Proceedings of the 1st international conference on Simulation tools and techniques for communication, network and systems and workshops*, 2008.

[5] Y. K. Wen-Hwa Liao and C.-M. Fan, "Data aggregation in wireless sensor networks using ant colony algorithm," *Journal of Network and Computer Applications*, vol. 31, no. 4, pp. 387–401, November, 2008.

[6] H. M. Huifang Chena and T. Mizuno, "Adaptive data aggregation scheme in clustered wireless sensor networks," *Computer Communications*, vol. 31, no. 15, pp. 3579–3585, Septempber 25, 2008.

[7] L. M.-B. A. Sardouk and D. Gaiti, "Agent-cooperation based communication architecture for wireless sensor networks," in *Proceeding of Wireless Days, WD '08, 1st IFIP*, Dubai, UAE, November 24-27, 2008, pp. 1–5.

[8] D. O. Richard Tynan, David Marsh and G. OHare, "Intelligent agents for wireless sensor networks," in *Proceeding of AAMAS05*, Utrecht, Netherlands, July 25-29, 2005, pp. 1179–1180.

[9] D. A. M. V. Milanosic N., Malek M., "Routing and security in mobile ad hoc networks," *D. L. Carver Edition, IEEE Computer*, vol. 37, pp. 61–65, February, 2004.

[10] I. H. K. Ian F. Akyildiz, "Wireless sensor and actor networks: research challenges," *Ad Hoc Networks*, vol. 2, no. 4, pp. 351–367, October, 2004.

[11] S. K. M. Z. T. Raghavendra, C.S., *Wireless Sensor Networks*. Springer, 2004.

[12] K. L.-B. Yoav Shoham, *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press, December 15, 2008.

[13] M. Wooldridge, *Intelligent agents*. G. Weiss (ed.), Multiagent systems : A modern approach to Distributed Artificial Intelligence, MIT Press, 1999.

[14] W. T. E. A.-D. A. G. Z. Petriu E. M., Patry G. G., "Intelligent robotic sensor agents for enviroment monitoring," in *Proceedings of the IEEE International Symposium on Virtual and Intelligent Measurement Systems*, May, 2002, pp. 14–19.

[15] H. L. B. Cook D. J., Gmytrasiewcz P., "Decision-theoric cooperative sensor planning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 10, pp. 1013–1023, October, 1996.

[16] G.-R. C. B. P. T. Campo C., Marin A., "Service discovery in pervasive multiagent systems," in *Proceedings of the Workshop on Ubiquitous Agents on embedded, wearable, and mobile devices, ACM*, July, 2002.

[17] X. L. J. A. F. T. Chen H., Chakraborty D., "Service discovery in the future electronic market," in *Proceedings of the 17th National Conference on Artificial Intelligence, AAAI Press/MIT Press*, Austin, Texas, USA, July, 2000, pp. 1–6.

[18] L. B., "Discovery of infrastructure in multi-agent systems," in *Proceedings of Second International Joint Conference on Autonomous Agent and Multi-Agent Systems*, Melbourne, Australia, 2003.

[19] T. S. K. D. K. A. G. G. J. A. F. T. Ratsimor O. V., Chakraborty D., "Allia : Alliancebased service discovery for adhoc environments," in *Proceedings of ACM Mobile Commerce Workshop, ACM*, Atlanta, USA, September, 2002.

[20] FIPA, *FIPA ACL Message Representation in Bit-Efficient Specification, Specification number SC00069*, December, 2002. [Online]. Available: http://www.fipa.org

[21] H. H. Berger M., Watzke M., "Towards a fipa approach for mobile ad hoc environments," in *Proceeding of the 8th International Conference on Intelligence in next generation Networks*, Bordeaux, France, May, 2003.

[22] L. M. Helin H., "Performance analysis of software agent communication in slow wireless networks," in *Proceedings of the 11th International Conference on Computer Communications and Networks (ICCCN 2002)*, Austin, April, 2002, pp. 354–361.

[23] B. S. Choudhury R. R., Paul K., "Topology discovery in ad-hoc wireless networks using mobile agents," in *Proceedings of the 2nd ACM International Workshop on Mobile Agents for Telecommunication Applications, Springer-Verlag*, September, 2000, pp. 1–16.

[24] G. R. S., "Soldiers, agents and wireless networks : A report on a military application," in *Proceedings of the Fifth International Conference and Exhibition on the Practical Application of Intelligent Agents and Multi-Agents*, Manchester, England, April, 2000.