

Towards a Real Time Distributed Flood Early Warning System

EL MABROUK Marouane
CA-GIP, CA, Versailles
France

Abstract—Since the beginning of humanity, floods have caused a lot of damages and have killed many people. So far, floods are causing a lot of losses and damage in many countries every year. When facing such a disaster, effective decisions regarding flood management must be made using real-time data, which must be analyzed and, more importantly, controlled. In this paper, we present a distributed decision support system that can be deployed to support flood management decision makers. Our system is based on Multi-Agent Systems and Anytime Algorithm, and it has two modes of processing: a Pre-Processing mode to test and control the information sent by sensors in real-time and the Main Processing mode, which has three different parts. The first part is the Trigger Mode for monitoring rainfall and triggering the second part, the offline mode, which predicts the flood based on historical data without going through the real-time decision support system. Finally, the online mode predicts the flood based on real-time data and on a combination of communications among different modules, hydrodynamic data, Geographic Information System (GIS), decision support and a remote sensing module to determine information about the flood.

Keywords—Flood; forecasting; distributed decision support system; multi-agent system; anytime algorithm

I. INTRODUCTION

Water, a source of life and, energy, reveals to man his image. When he contemplates, prostrated on the shore of a lake or favorite ocean, a man draws happiness and peace; however, floods are aquatic phenomena that man respects and of which he is afraid. Occurring as a result of heavy rainfall, which is occasionally regular, lasting, intense and violent, floods leave behind desolation, misery and destruction [1].

Based on a report published by the United Nations, floods are considered one of the most damaging natural disasters in the world. Generally, the number of people affected and the economic damages resulting from floods are increasing [2]. In addition, floods are responsible for one-third of all damages caused by natural disasters [3].

The annual damage due to floods worldwide was estimated at 50 billion US dollars per year in the decade from 1990 to 1999, and an increase in costs of 4% per year seems to be emerging [2]. However, despite the constant average number of natural disasters, the number of people affected and the amount of damage recorded is increasing, reflecting not only better identification of disasters but also an increase in the concentrations of people in areas at risk, especially in flood-prone areas [3].

In recent years, Morocco has experienced several floods that caused several deaths and injuries, and serious material and human damages [4]. It is for this reason that flood forecasting has become an important area of research; it is important for national economic planning and saving people's lives. In recent years, flood forecasting technology has begun to use computer science, and the flood forecasting abilities have been greatly improved through database technology, artificial intelligence technology and several decision support systems developed using Web and other technologies [4].

Flood risks cannot be completely eliminated, but with warning systems and decision support systems that can be used to prevent floods, we can reduce their impact. Decision Support Systems (DSSs) are defined as tools that help end users (usually called decision makers) choose from a set of possible decisions or alternatives. In the domain of flood management, DSSs are used for many tasks and are an important part of the decision-making process [5].

DSSs consider three phases of flood management: pre-flood management, operational flood management and post-flood management. DSS tools for floods must provide accurate flood forecast information for the first phase and useful strategies for the second one. This information can be used for warning messages aimed at the general population or towards selected groups and services, such as forest rangers, police forces, fire brigades, military forces and other intervention groups [5].

The rest of the paper is organized as follows: Section 2 introduces proposals for real-time flood monitoring published by other authors. In addition, DSSs and real-time distributed DSSs are presented in Section 2. In Section 3, we introduce our distributed system for flood management that is based on multi-agent systems; it is divided into several parts and modes. In Section 4, we will present the conception, design and implementation of the proposed system. The results and discussion are presented in Section 5. Finally, in Section 6, the conclusion and recommendations are given for further work.

II. STATE OF THE ART

A. Related Work

Since the 1960s, flood forecasting has been the subject of extensive research and has resulted in countless operational developments in many countries because of the human and economic effects associated with these events. Information systems are increasingly used in flood forecasting. Since, several researchers have focused their work on intelligent

systems for decision support for the prediction of floods. In [6], the authors developed a DSS for flood management in urban areas. Their system has several modules, including a module for the analysis of sea level and wind data based on the 48-hour weather forecast. According to the models included (BSM-2010, FMI), the system can calculate the water level in the watershed to trigger the warning.

In [7], the authors proposed a decision support system known as RAMFLOOD DSS that includes several modules and methods for processing the data used in the system, ie, data about land use, topographic data, etc. They also proposed a digital model for hydrodynamic modeling that merged hydraulic methods with soil data using neural networks. The authors in [8] designed and developed DESMOF, a decision support system for the Red River watershed. This system has four components and is able to combine intelligent knowledge with database and model data to support the decisions.

In [9], a real-time flood management system was developed by BCEOM, CS SI and Meteo France. The system includes four parts. The first part is the acquisition and validation of hydrometeorological data (meteorological data and soil data), the second part is for hydrological forecasting calculations, the third is for identifying natural hazard scenarios and the last is to trigger the evacuation plan. The author in [10] developed and proposed an adaptive system based on a multi-agent system using the AMAS approach [11] for real-time flood forecasting. This system is designed to forecast floods in an entire watershed. Its architecture is based on two levels: the first level is for the agents to calculate the water levels. Each agent is designed to provide the variation in the water level for an hour. The second level is for wireless sensors, and each agent in the first level manages these sensors to observe the actual influence of each entry.

B. Real-Time Distributed Decision Support Systems

The growth of the use of computer systems in daily life has pushed the scientific community to focus on improving their reliability. Thus, ensuring proper functioning has become a necessity. To ensure proper operation and the quality of the processing and the results of these systems, we must think in terms of performance, i.e., the processing time, execution time and response time [12]. Hence, using a distributed decision support system is required to ensure resource sharing, fault tolerance, inter-process communication and to facilitate communication between users.

1) *Decision support systems*: Decision support systems increasingly rely on computer systems that are responsible for providing to the decision maker many relevant elements as quickly as possible to support in their decision-making [13].

The concept of the DSS is extremely vast, and its definition often depends on the author's perspective. Simonovic [14] defined a DSS as a system that allows decision-makers to make judgments using computer output based on a user interface that provides the information needed for the decision-making process.

These systems assist with decisions that are related to various problems (structured, semi-structured and

unstructured) using all available information. They use several approaches for processing and responding to the problem requested, such as quantitative models and database elements. They are an integral part of the decision maker's approach when processing and solving a problem.

Decision support systems are often multi-user distributed information systems. Therefore, it is necessary that they rely on distributed computer systems: users in multiple locations who cooperate using computers. That is why, as part of our work, we focused on the conception of a decision support system that is physically and logically distributed using multi-agent systems. First, though, what is a distributed decision support system?

2) *Distributed decision support systems*: The current trend of globalization has transformed the way decisions are made; organizations are now often present at different locations spread around the world. Consequently, data, interfaces and users are often in geographically separate locations. However, before presenting distributed decision support systems, we must first define distributed systems.

Distributed systems includes all of the systems used by several autonomous entities, with each communicating with its neighbors through either communication channels or shared variables. Each entity evolves according to its own state and those of its neighbors. The challenge of distributed algorithms is to succeed in making all of these entities collaborate to achieve a global task [12].

This make distributed systems more powerful. First, they can be more reliable because every function is duplicated several times. When one of the entities fails, another can take over the work and the results of the processing can be stored in several locations. Second, by having several entities, the response and processing times can be kept short by using parallel computation. This property and the fault tolerances of the distributed systems give the systems the potential to be much more powerful than any traditional operating system [15].

It should be noted here that the concept of a DSS is not incompatible with the distribution of knowledge and users but simply that in the classical architecture of a DSS, we have a central synchronization source (often a central database).

The evolution of the concept from the classic DSS to the distributed DSS occurs mainly at the level of the topology used to connect the various components. Gachet [16] defined the Distributed DSS as a set of components and services arranged in a dynamic and unreliable network, whether it is composed of hardware or software services, working together to assist in decision-making.

Many topologies are used in distributed systems, but three main topology types are used for communication: centralized topology, hybrid topology and decentralized topology. Hybrid topology represents an intermediate stage between the other two topologies [16].

3) *Real-Time Multi-Agent system*: The implementation of Anytime Algorithms turns any system into a real-time system.

The necessity of using a distributed system forced us to explore multi-agent systems. Flood disasters require a distributed system that responds quickly because flooding is a disaster that can endanger human lives; thus, using these two approaches is essential. Real-time multi-agent systems are multi-agent systems that account for real-time constraints.

a) *Anytime Algorithm*: An anytime algorithm is an algorithm that exchanges execution time for a high-quality result. Allowing more time to execute a task results in an output of higher quality. Because the execution times are so long, this approach risks increased response times in non-traditional intelligent systems. Fortunately, these algorithms find their places in the field of artificial intelligence because, in this area, the construction of reasoning would occur progressively over time and address some of the deadline needs [17,18].

b) *Multi-Agent System*: From the concept of multi-agent system disengages immediately the idea of a multiple agents system, the concept of agents remains the backbone of it. An agent is a real or virtual entity that evolves in an environment that is able to perceive, act and communicate with other agents, and that exhibits autonomous behavior that can be observed through interactions with other agents and the goals it pursues. Thus, multi-agent systems are very suitable for modeling phenomena in which the interactions between the various entities are too complex to be understood by classical modeling tools [19]. They are increasingly used in disaster management scenarios, especially flood events, because they can represent autonomous entities with behaviors that include cooperating, negotiating and communicating with others.

The approach accounts for the real-time aspect of multi-agent systems and is the approach used most often [20,21], because it enables the design of applications that provide answers in time, even if the results provided are partial or approximate.

Claude Duvallet and Bruno Sadeg [22] proposed a new approach that accounts for the real-time aspect in a multi-agent system: the ANYMAS model, which is a multi-agent system model based on the anytime approach. In this model, the real-time aspect has two levels (agent itself and multi-agent systems).

In the next section, we introduce and present our proposed Distributed Decision Support System for Real-Time Flood Management.

III. PROPOSED DISTRIBUTED DECISION SUPPORT SYSTEMS FOR REAL-TIME FLOOD MANAGEMENT

Our need to design and develop a physically and logically distributed system led us to perform several studies on distributed approaches to choose the most appropriate approach for our case. The advantages that can be provided by the multi-agent approach, namely distributed, parallel, collaborative, hybrid, flexible, recursive, adaptive, cooperative, and intelligent distributed processing and a scalable environment, led us to choose this approach, because it provides us all of the desired advantages in our distributed decision support system.

Our system is a system for flood forecasting and warnings that performs several tasks; the multi-agent approach allowed us to assign agents to specific tasks to perform intelligent, distributed, adaptive and scalable processing to reach cooperative and collective decisions by entering all of the agents into the global processing system.

Our decision support system is a distributed system based on multi-agent systems and Anytime Algorithms for flood disaster management. It is from here that we can see the originality and strength of our proposed system since, in the literature, few examples like this exist.

The coupling of multi-agent systems with Anytime Algorithms makes our distributed decision support system reliable, fast, strong and original compared to other decision support systems for flood disaster management.

Our system is divided into two parts. The first is the pre-processing phase, which aims to:

- Ensure the best functioning of the wireless sensors.
- Test the reliability of data received from these sensors.
- Classify the received data as valid or invalid data.
- Choose the most appropriate data to be saved in the database.

The second part, the main processing phase, aims to:

- Simulate floods and run comparison modules for decisions regarding the occurrence of flooding in the offline mode.
- Support decisions in the online mode using an intelligent distributed architecture based on multi-agent systems in real-time flood forecasting.
- Simulate hydrological data using hydrological modelling modules.
- Use remote sensing to determine land use in areas under surveillance to identify areas at risk.
- Use a GIS module, which allows for the review of all knowledge related to flood risks and provides an overview of the risks to provide support in disaster management.
- Use knowledge bases to determine the constraints for the system to obtain the most intelligent decisions possible.

Fig. 1 shows the general architecture of our distributed decision support system.

Our system is a system based on distributed architecture and client-server architecture with a centralized topology. This topology allows for easy control over data and users and simple implementation [16].

This is, of course, an important factor in the choice of topology for facilitating the implementation of distributed decision support systems. Fig. 2 displays the system's distributed architecture.

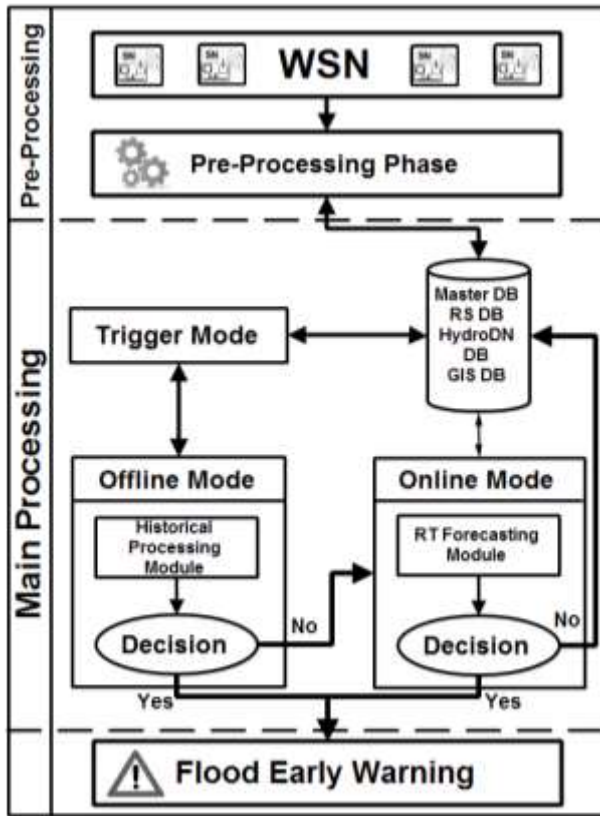


Fig. 1. General Architecture of our Decision Support System.

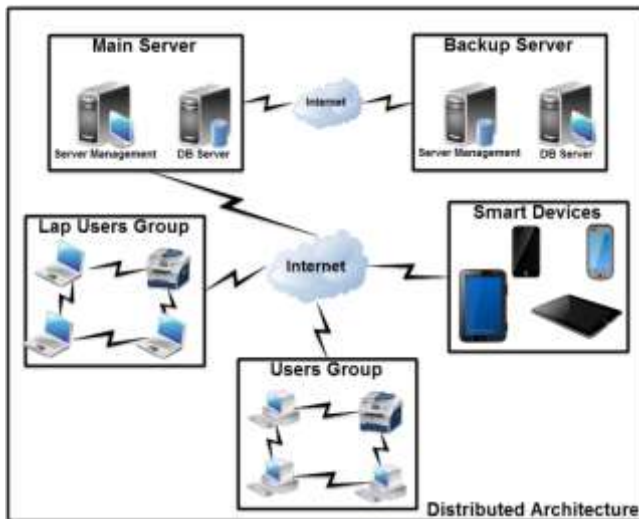


Fig. 2. Proposed Distributed Architecture.

A. Pre-Processing Phase: The Intelligent Proposed Model for Classification and Segregation [23]

The pre-processing phase is used to monitor the function and performance of sensors installed in flood-prone areas. This model is based on multi-agent systems and is used to classify the data received into valid and invalid data. We have proposed three stages with five agents to perform the necessary processing to classify the data to be stored in the database.

Data Aggregation and Classification Stage is the responsible for this data aggregation and classification of data received from the WSN and there are three agents responsible for the classification process, the Rainfall Agent, the Runoff Agent and the Water Level Agent. Each of these agents does a processing according to the following proposition:

$$M(C) = \begin{pmatrix} 0 & & & & & \\ T_{(d_2,d_1)} & 0 & & & & \\ T_{(d_3,d_1)} & T_{(d_3,d_2)} & 0 & & & \\ T_{(d_4,d_1)} & T_{(d_4,d_2)} & T_{(d_4,d_3)} & 0 & & \\ \vdots & \vdots & \vdots & \ddots & 0 & \\ T_{(d_n,d_1)} & T_{(d_n,d_2)} & T_{(d_n,d_3)} & \dots & T_{(d_n,d_{n-1})} & 0 \end{pmatrix} \quad (1)$$

Where

$$T_{(d_i,d_j)} = \left(1 - \frac{\min(d_i,d_j)}{\max(d_i,d_j)} \right) \text{ Where } \begin{cases} i \in [2, n] \text{ Where } n \in \mathbb{N} \\ j \in [1, n-1] \text{ Where } n \in \mathbb{N} \end{cases} \quad (2)$$

The d_i and d_j are a value from the dataset sent by sensors to be processed in the proposed matrix in position i and j . n is the number of lines and columns in the matrix

To range data according to the tolerance specification, we proposed the tolerance percentage and here is its proposed formula:

$$P_{TC} = \frac{(2 \times \Delta_C) + T_F}{100} \quad (3)$$

Where

- Δ_C : Sensor uncertainty designated by the fabrication company.
- T_F : Uncertainty that may be attached to a factor in the study area (Rainfall, Runoff, ...).

Here is the architecture of this proposed Pre-Processing phase presented in the Fig. 3:

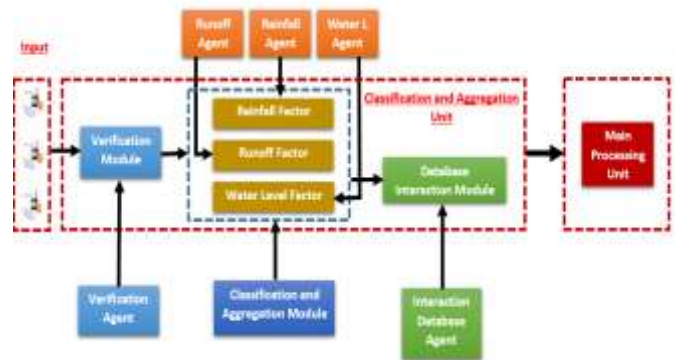


Fig. 3. Pre-Processing Mode Architecture.

B. Main Processing: Trigger Mode

In this sub-mode, we proposed a multi-agent layer responsible for monitoring rainfall and its intensity. When rain is particularly heavy, the agent triggers the double forecasting and warning system (offline and online), sends ACL messages to the system to inform it that rainfall has become stronger, and waits for confirmation from the system. This confirmation allows the server responsible for the online processing to be monitored for proper functioning (Fig. 4).

C. Main Processing (Offline Mode): Proposed Model for Decisions based on Historical Processing

The offline mode is based on similarities between historical data and data received in real time. In this mode, we proposed a model responsible for flood forecasting in the offline mode based on historical data. Our model compares the similarities between two multidimensional objects: historical data and real-time data received from wireless sensors. We proposed an equation that includes all of the factors (rainfall, runoff and water levels). Instead of comparing each factor in the historical data with its corresponding real time data, we compare the result of the proposed equation calculated for the historical and real-time data. In the proposed equation:

$$F(Z) = \frac{aF_1 \times cF_3}{bF_2 \times TSFV} \tag{4}$$

Where

F_1, F_2, F_3 and $TSFV$ represent rainfall (mm), runoff ($\frac{m}{s}$), water levels (m) and the time between the previous forecast value at an instant (t) and the current forecast value at an instant ($t + 1$) in second (s), respectively. For the first record $TSFV = 1$, F_H and F_{RT} are respectively, historical data and real-time data functions for factors that give us values ($V_1, V_2, V_3, V_4, \dots, V_n$) to compare in our Historical Processor Model that has two levels for comparison.

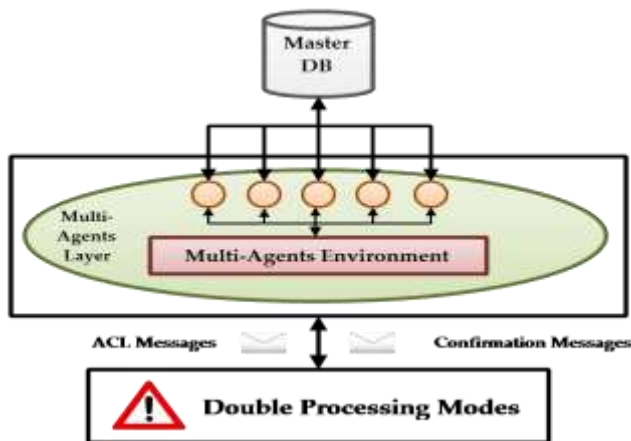


Fig. 4. Trigger Mode Architecture.

(a, b, c) are coefficients and $(a, b, c) \in \mathbb{R}_+^*$. These coefficients are the percentages of influence for each factor that leads to flooding in the affected area.

1) *Proposed model*: Our proposed model is divided into two stages: the splitting stage and the double validation stage.

a) *Splitting Stage*: First, data are loaded from the database, and then the splitting module divides the historical data into categories. The splitting agent calculates the set for all historical data for each record in the database using the equation proposed in the previous section. Next, it divides the historical data into categories depending on the interval (end of flood, beginning of flood). Each distribution includes a set of records beginning with the first record saved directly after the flooding until the next flood event occurred. This distribution then represents one case out of all of the cases of flooding that have occurred in the area. The system compares the real-time data with these distributions to determine if there are any similar cases in the past. The distributions are then saved in the local database. When the agent finishes processing the data, it sends the ACL messages to trigger the Historical Processing Model. Fig. 5 shows how the proposed model divides the data and creates the distributions.

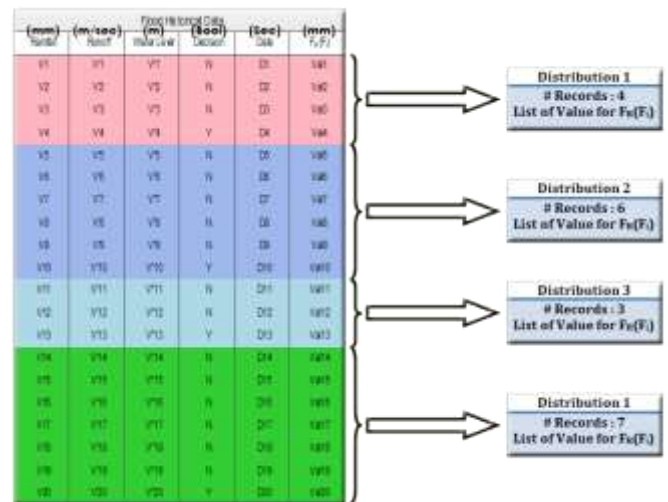


Fig. 5. Splitting Process.

b) *Double Validation Stage*: This stage includes two levels. The first level is for simulation by the variance coefficient, and it has three modules: the first module is the historical data variance coefficient calculation module; it is managed by the variance coefficient calculation agent $AMVCH_i$. This agent calculates the variance coefficient for each distribution that was generated in the previous stage. The second module is the real time data variance coefficient calculation module, managed by the $AMRTC$ agent. This agent calculates the variance coefficient for the real-time data. The third module is the comparison module. The comparator mediator agent allows for comparisons between the historical coefficients and the real-time coefficients. If the coefficients are similar, they will be included in a list of similar

distributions, and it will be sent to the second level. The agent sends ACL messages to inform and trigger the second level. Otherwise, it sends ACL messages to the decision support system to trigger the real-time online forecasting mode for flood disaster management.

The second level is for the Gini index simulation [24]. Our proposed model is based on using inequality measures to perform the comparisons between the generated distributions and the real-time data received from the wireless sensors. The Gini Index is one of the best methods for comparing distributions across different categories because of its easy and simple interpretation using the Lorenz curve diagram. The Gini index also has an important advantage in that it can be used to indicate changes in the distributions. Thus, in this level, there are five modules to validate the similarity of the historical and real-time data. The first module is a module for the interaction with the database. It is managed by the *AMLDDb* agent, which selects the selected distributions already sent by the first level from the database and loads it into the GINI index historical data calculation module. This second module is for calculating the GINI index for historical data. It is managed by the *AMGH_i* agent, which calculates the GINI index for the selected distributions. Additionally, there is a module managed by the *AMGRT* agent responsible for calculating the Gini index for real-time data received from wireless sensors. At the end of the processing, the *AMGRT* agent saves the results in the Master database. The fourth module is used to perform the comparison between the selected distribution and the real-time data based on the results of the two previous modules. During the comparison, the comparator mediator agent of the GINI index is responsible for determining the similarity between the selected distribution and the real-time data using a distance comparison.

If there is a similarity, the system continues to the last module to give the final decision about the likelihood of a flood event occurring, otherwise it proceeds to a real-time decision using the online mode. The final module is the Loopback module managed by the *AMLB* agent. To ensure that there are similarities between the distributions and real-time data, we designed our system to compare 90% of the distribution records with the data received in real time to decide and trigger the early flood warning because if there is 90% similarity, we can be absolutely certain that there will be flooding, and the accuracy of the system's decision will be high. The *AMLB* agent is responsible for testing the number of records in the real-time data. If it reaches 90% of the

number of records of the selected distributions, it triggers the early flood warning. Otherwise, it proceeds to a real-time decision using the online mode.

2) *Architecture of the proposed model:* Fig. 23 (see Appendix) presents the architecture of the offline forecasting mode.

3) *Flow chart of the proposed model:* In this section, we will present the flow chart of the offline forecasting mode.

Fig. 6 presents the flow chart of the first level of validation by the variance coefficient, and Fig. 7 presents the flow chart of the second level of validation by the GINI index.

a) Variance Coefficient Validation Flow Chart

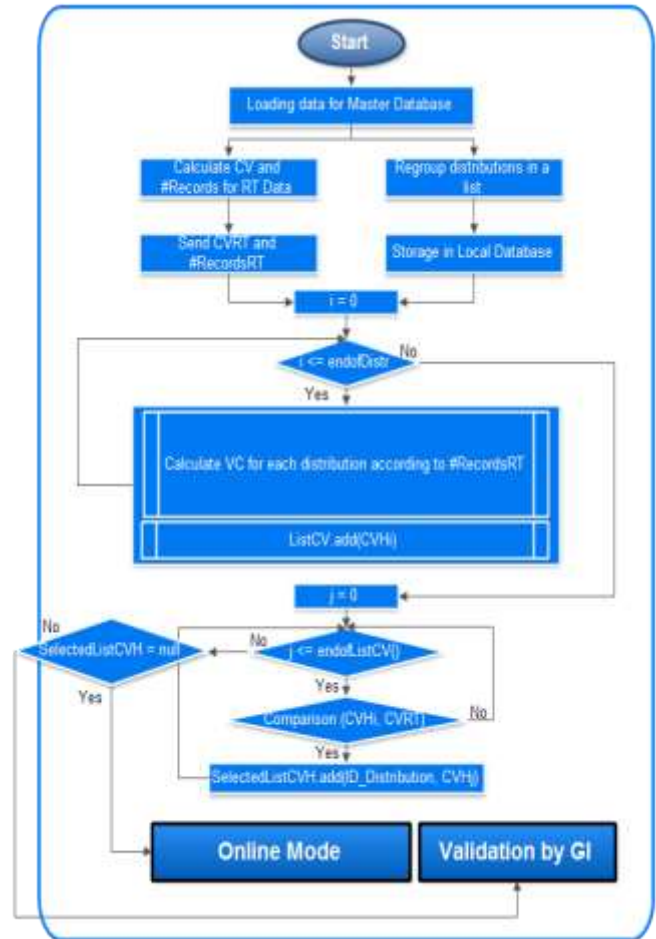


Fig. 6. Variance Coefficient Validation Flow Chart.

b) GINI Index Validation Flow Chart

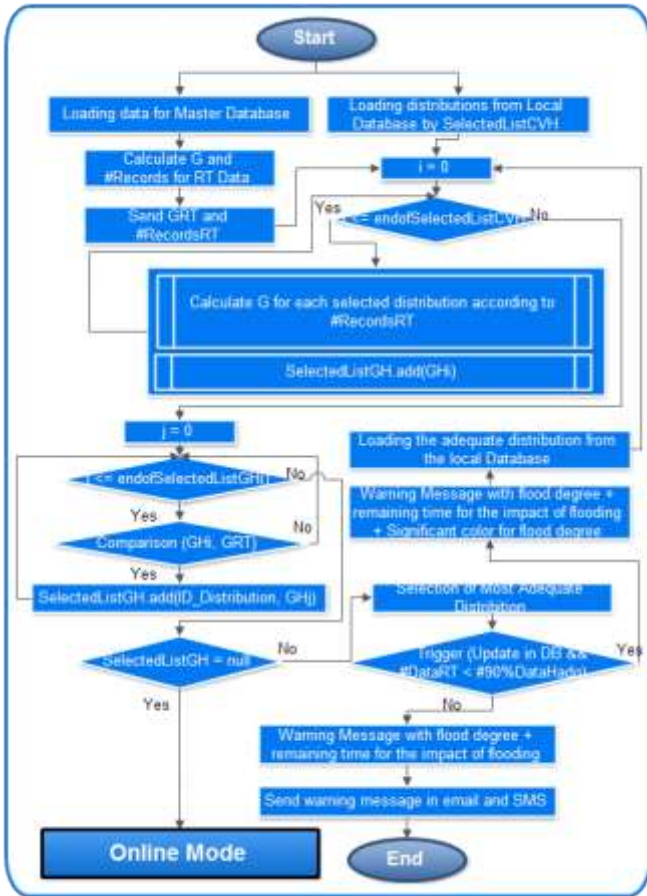


Fig. 7. GINI Index Validation Flow Chart.

4) *Validation by GINI Index:* We use historical data (Table I) and real-time data (Table II) to make the simulation and the validation with the GINI index. As already mentioned, for a first forecasting of a region, we process its historical data before starting the real-time forecasting in order to calculate the distributions and its variance coefficients and the GINI coefficients and Lorenz curves to be used in comparison with the real-time data during the real-time forecasting. Next, the results are stored in the database.

The validation with the GINI index is triggered only if it found similarity between historical distributions and real-time distributions in the first level of validation by the variance coefficient. The comparison is based on the number of records of real-time data received from sensors. The system gives a decision if and only if 90% the record sum of the historical distribution \geq the record sum of the real-time distribution. If $G \leq 0.1$ so the distributions are similar. Therefore, the system triggers the warning procedure. If there is no similarity with the historical distributions, so the system proceeds to the online mode to make the forecasting and warning. The following are two distributions, with which we will illustrate an example of our system functioning using the validation with the GINI index.

TABLE I. HISTORICAL DATA

Rainfall	Runoff	Water Level	FH(Z)
1	80	0,11	0,001375
2	295	0,33	0,00223729
4	157	0,17	0,00433121
9	221	0,14	0,00570136
6	181	0,26	0,00861878
4	77	0,17	0,00883117
18	186	0,11	0,01064516
12	118	0,20	0,02033898
22	105	0,14	0,02933333
11	86	0,35	0,04476744
12	35	0,22	0,07542857
51	434	0,70	0,08225806

TABLE II. REAL-TIME DATA

Rainfall	Runoff	Water Level	FRT (Z)
1	80	0,01	0,000125
2	120	0,02	0,00033333
4	107	0,09	0,00336449
6	181	0,18	0,00596685
11	106	0,15	0,01556604
12	124	0,19	0,0183871
18	86	0,11	0,02302326
9	50	0,24	0,0432

The Table III (see Appendix) shows the calculated data to plot the Lorenz curve (Fig. 8) for comparison of the historical and real-time distributions.

The Table IV (see Appendix) allows calculating the GINI index of the processed distributions

$$\text{Gini Index} = 1 - (0,531125 + 0,315821) = 1 - 0,846946 = 0,153054$$

It is observed that the real-time distribution and the historical distribution are similar in about 85% and even if it has not reached 90% of the distribution of historical data yet. If G is less than or equal to 0.1 so both of distributions will be similar, therefore the system will trigger the warning. The following curve in the Fig. 8 shows the comparison between the historical and the real-time distributions.

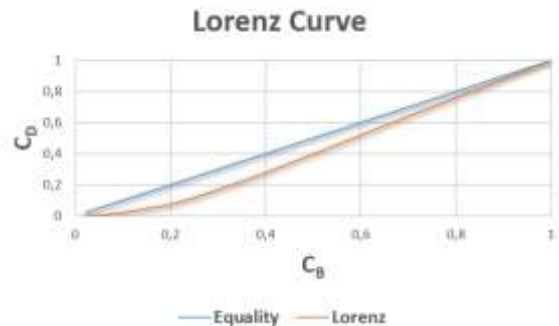


Fig. 8. Lorenz Curve of the Distributions.

Using this technique, the system can decide about the similarity of distributions even if the distributions are different.

D. Main Processing (Online Mode): Decision Support System for Real-Time Flood Management

In this mode, we proposed an expert system based on multi-agent systems and Anytime Algorithms for real-time flood forecasting and warning. Using several modules communicating with each other to provide the best decisions in a short amount of time forced us to design a collaborative system that can quickly determine the likelihood of a flood event. To accomplish this, we used ANYtime Multi-Agent Systems (ANYMAS) [22]. Each anytime agent is responsible for managing a module. The system has four modules in its online mode, which are described below.

1) *Remote sensing module*: This module presented in the Fig. 9 is used to identify the land use in at-risk areas where our system is installed. It has four processing levels. The first level is for receiving satellite images transmitted by the Land 7 satellite, which has as ETM+ sensor. The second level is for mosaicking and super positioning the satellite pictures over each other to obtain the resulting picture. The third level is for the supervised classification of the satellite pictures to analyze and extract the land use of areas under control, and finally, the fourth level is for the generation of the land-use map. This module is surveillance by the *AMRS* Remote Sensing agent. This agent is responsible for all of the data processing inside the module and for communicating with other modules.

2) *Hydrodynamic module* [25]: This module (Fig. 10) is responsible for estimating the risk of flooding, as it allows the hydrological behaviour of the watershed in reference floods to be understood. In Morocco, empirical models are often used to estimate the maximum runoff at the outlet of the watershed. Our module is divided into two levels. The first level is for hydrological modelling using a proposed model for short-term forecasting

Here is the proposed system:

$$(S) = \begin{cases} aX_1 + bY_1 + cZ_1 - Cn = 1 \\ aX_2 + bY_2 + cZ_2 - Cn = 1 \\ aX_3 + bY_3 + cZ_3 - Cn = 1 \end{cases} \quad (4)$$

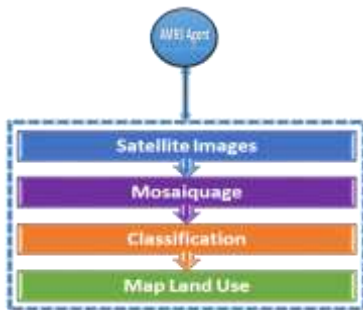


Fig. 9. Remote Sensing Module.

Where

$(X_1, Y_1, Z_1, X_2, Y_2, Z_2, X_3, Y_3, Z_3, Cn) \in \mathbb{R}_+^*$ and (X, Y, Z, Cn) are respectively the rainfall, the runoff, the water level and the evacuation of the area under control.

The solutions of this system are the values with which our expert system detects and makes the flood forecasting and triggers warnings based on the following generic function:

$$F(Z) = aX + bY + cZ - Cn \quad (6)$$

and the empirical model of Hazan-Lazarevic for medium and long-term forecasting.

The runoff is given by the following equation:

$$Q_{(T_1)} = Q_{(T_2)} \times \frac{1 + k \ln T_1}{1 + k \ln T_2} \quad (7)$$

Where:

T_1 and T_2 are respectively the return period and $k \in [0,8, 1,2[$

The second level is for hydraulic modeling using HEC-RAS software. This module is managed by the *AMH* agent, which is responsible for the processing.

3) *GIS Module*: This module introduced in the Fig. 11 aims to provide information about the magnitudes of floods, their causes, and the areas that are at risk. The combination of GIS information layers provides information about the vulnerability of a given area. The layers with which the GIS Module works include occupation records, land-use maps already created in the remote sensing module, urban maps, flood maps, digital ground models and hydrological data. In the ArcGIS software package, we worked with the HecGeoRas module, which is used to model the hydraulic system in a given area to be studied in terms of the rivers and their morphologies so that they can be processed in the hydraulic model using HEC-RAS software.

This module is managed by the *AMS* agent; this agent is responsible for communicating with the ArcGIS software package and other system modules to receive the hydrological data from the watershed morphology to be studied (runoff and the water levels) that are sent by the hydrodynamic module. It also receives the land-use maps sent by the remote sensing module and loads real-time data.

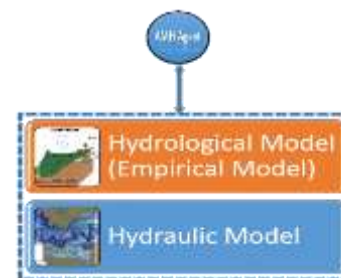


Fig. 10. Hydrodynamic Module.



Fig. 11. GIS Module.

The agent communicates with the remote sensing module and the hydrodynamic module using the Soil and Water Assessment Tool (SWAT) [26], which is a hydrological model linked to the ArcView interface in the ArcGIS software package. It is written in Fortran 90 and can predict the effects of changes in land use on water, sediment and chemical compounds; it also accounts for cultural practices. This is a physically based, semi-distributed model.

4) *Decision support module*: This module is an expert module for decision support capable of accounting for the information received from the other modules and executing the reasoning already developed in the knowledge base. It includes three parts:

- Facts base
- Rule base
- Inference engine

Our inference engine is able to use the facts and the rules from the knowledge base to produce new, partial decisions until the final decision about the impact of the flooding is made. This module also serves to create a custom database to be used in the offline mode and to calculate the forecast parameters that have been proposed and the coefficients of the factors that lead to floods (rainfall, runoff and the water levels). These parameters are the basis of our proposed system for real-time forecasting and are used by the inference engine; we developed the system and rules for testing the impact of flooding based on these parameters. This module is managed by the *AMDSS* agent, which is responsible for receiving the data sent by the other modules and transferring them to this module. After the processing and decision-making, the agent saves all of the processed data in the database to be used in the offline mode in the future. Fig. 12 displays the architecture of the Decision Support module.

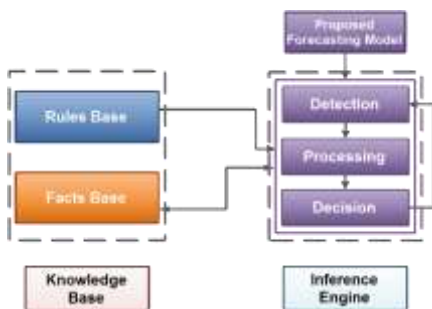


Fig. 12. Decision Support Module.

5) *Architecture of the proposed model*: In this section, we will present the architecture of the real-time flood forecasting model in the online mode. This is displayed in Fig. 13.

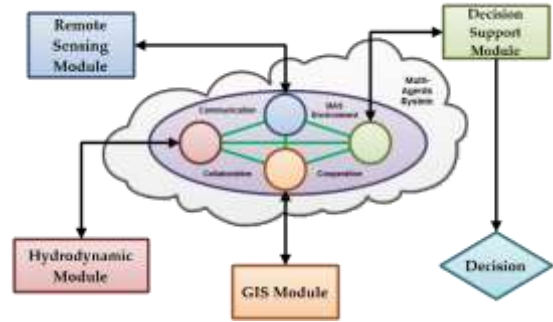


Fig. 13. Architecture of the Proposed Online Model.

The Fig. 14 shows how the different modules communicate in the online mode.

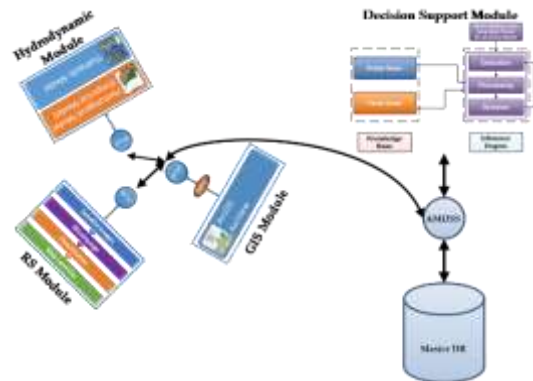


Fig. 14. Modules Communication in the Online Mode.

IV. DESIGN AND IMPLEMENTATION

In the system design phase, we chose visual modelling as a design model for this system because of its many benefits, including simplicity, universality, conciseness and expressiveness. The language used is UML, which is a graphical language for processing and modelling data. The modelling was performed on three levels, structural modelling, behavioural modelling and interaction between objects modelling. [23].

A. Design

1) *Structural and statistical modelling*: The deployment diagram illustrates the distributed architecture dedicated to our decision support system. The decision makers can log in from multiple devices, i.e., PCs, smartphones and tablets, through the Internet or intranet. We use one server for the Database Management System (DBMS) and another server for the decision support system. The deployment diagram is presented in Fig. 15.

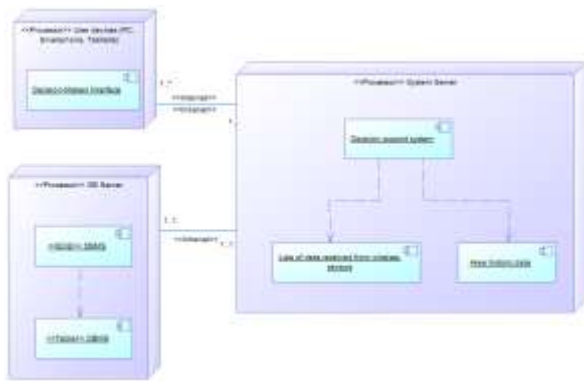


Fig. 15. Deployment Diagram.

2) *Behavioural modelling*: Behavioural diagrams focus on the dynamic behaviour of the system, the behavioural diagrams that we used are the use case diagram and sequence diagram. The main actors in our system are the decision makers who watch over the control and management of floods, responsible and administrators. The use case diagram is shown in the Fig. 16.

The sequence diagram is shown in the Fig. 17.

3) *Interactional and dynamical modelling*: The diagrams of interaction between objects modeling used to model the dynamic behavior of the system, and to indicate how the objects interact at run time. The most interesting diagram of this kind of modeling is the communication diagram. The communication diagram is shown in the Fig. 18.

B. Implementation

In this section, we will present our distributed decision support system for real-time flood forecasting and warning.

1) *Pre-Processing*: In this stage, as already described and shown in [23], the system first verifies the functioning of the wireless sensors installed in the surveillance area to identify the functioning and non-functioning sensors. The system disables the wireless sensors that are broken to remove them from the forecasting process and to have them repaired later.

Fig. 19 presents the classification interfaces of the proposed model for classification and aggregation, where the most appropriate value to be stored in the database is obtained from all of the valid data. In this stage, we also obtain the invalid values and the sensors that sent the erroneous values. The full details and information about this stage is presented in [23].

2) *Main processing (offline mode)*: In this phase, we proposed dividing the historic floods in the area into distributions by calculating the proposed function that includes all of the factors in a single equation.

Then, we proposed comparing these distributions with the data in real time using the double validation process that consists of two levels of comparisons: level 1 for comparisons using the variance coefficient and level 2 for comparisons using the GINI index (Fig. 20).

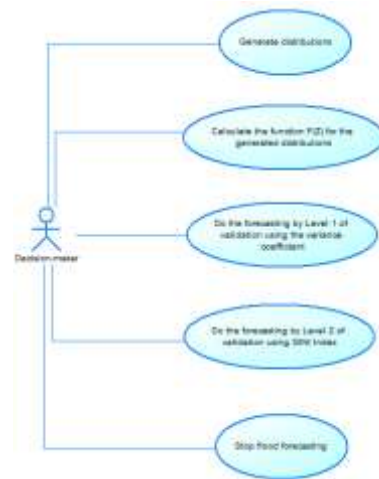


Fig. 16. Use Case Diagram for Main Processing Phase (Offline Mode).

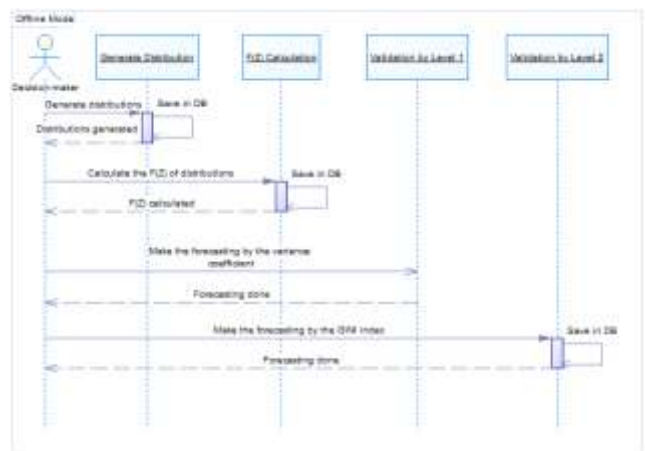


Fig. 17. Sequence Diagram for Main Processing Phase (Offline Mode).

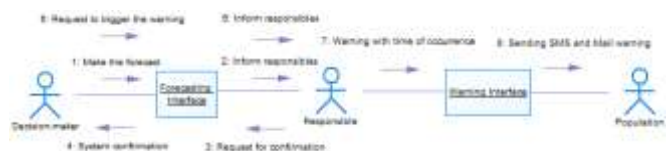


Fig. 18. Communication Diagram for Main Processing phase (Offline Mode).

Search ID	Date	Obs (x - 1) (Observed)	FTC (F(x))	Status
1	10/2	0.024280023902390239	0.0000000000000000	Valid
2	10/3	0.048560047804780478	0.0000000000000000	Valid
3	10/4	0.072840071707171717	0.0000000000000000	Valid
4	10/5	0.097120095609560956	0.0000000000000000	Valid
5	10/6	0.121400119501195119	0.0000000000000000	Valid
6	10/7	0.145680143401434143	0.0000000000000000	Valid
7	10/8	0.170000167301673167	0.0000000000000000	Valid
8	10/9	0.194280191201912191	0.0000000000000000	Valid
9	10/10	0.218560215102151215	0.0000000000000000	Valid
10	10/11	0.242840239002390239	0.0000000000000000	Valid

Fig. 19. Classification and Aggregation Process by the Proposed Model.



Fig. 20. Double Validation Process.

3) *Main processing (online mode)*: In this phase, we can forecast flood in real-time for the short, medium and the long term using a proposed model for short term forecasting and according to a proposed model by Hazan-Lazarevic [25] (Fig. 21). The full details and information about this stage is presented in [25].



Fig. 21. Short-Term Forecasting and Warning.

V. RESULTS AND DISCUSSION

In order to test the performance of our proposed decision support system, we were interested by the response time when dealing with flood events. First, we chose 20 areas in which to experiment where we could perform our experiments using historic data. For each area, we performed four experiments, and in each experiment, we tried to create forecasts and warnings for the areas. For each set of areas, we noted the required response times.

The experiments were first completed using only the online mode. Next, the Pre-Processing mode was used with the online mode, followed by the Pre-Processing mode with the offline and online modes and, finally, the entire distributed decision support system was used.

For all of the experiments, we noticed that the response time increased when the number of areas to be monitored increased, which is normal because that increases the amount of data. Using only the online mode resulted in a long response time for a single area. We noticed that more than five minutes were needed to respond because the amount of data in the database was large and it was not classified, so all of the data had to be processed before the forecasts and warnings could be made, which affected the response time.

In the second experiment, we observed the importance of classifying and aggregating the data before saving it in the

database because the response time decreased compared to the first experiment. In the third experiment, we used the three proposed modes: The Pre-Processing mode for classifications, the offline mode for decisions without the online mode and, of course, the online mode.

It was also noted that the response time decreased significantly compared to the two experiments already conducted due to using the pre-processing mode for classifications and aggregations, and due to the offline mode that helps determine the risk of flooding without using the online mode by comparing the new data with historical data.

Using our entire distributed decision support system, we noticed that the response time decreased by approximately five times compared with the first experiment thanks to the various proposed models and modules, the distributed architecture and the real-time multi-agent environment, which helps the system process data from several locations using several agents where each is responsible for performing a given process. Fig. 22 summarizes what we have just explained.

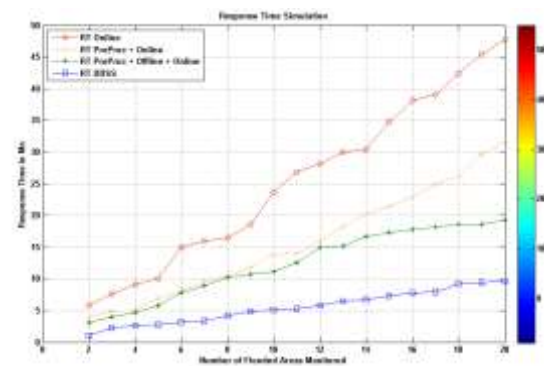


Fig. 22. Simulation of the Response Time of the System.

VI. CONCLUSION AND FUTURE WORK

In this paper, we presented our distributed decision support system for real-time flood forecasting. It has two main parts, the pre-processing part used to monitor the sensors and to classify data received from these sensors so that only the appropriate data are included. Three stages have been proposed for this process: the sensor verification stage, the Data Aggregation and Classification stage and the Database interaction stage.

The second part is the main processing part, and it has three modes:

- **Trigger mode**: This mode is responsible for monitoring rainfall to trigger the proposed systems.
- **Offline mode**: In this mode, we can predict floods without using the online mode by using a proposed model for processing and comparing historical and real-time data. It has two stages, the splitting stage for dividing the historical data into distributions to be used in the double validation stage and the double validation stage for the processing needed to calculate the similarities between the historical and real-time data to determine the likelihood of a flood.

- Online mode: In this mode, we use an expert system based on multi-agent systems and Anytime Algorithms for real-time flood forecasting and warning. This mode has four modules for real-time flood forecasting: the remote sensing module to identify land use, the hydrodynamic module for hydrologic and hydraulic modelling, the GIS module to identify flood-prone areas and, finally, the Decision Support Module for receiving and processing all of the data using a proposed model that manipulates the knowledge base through the inference engine.

Our future work will focus on improving the system by choosing specific criteria to improve. We will also design and develop multi-area forecasting and warning methods to improve our system such that it can monitor several areas at the same time without decreasing the performance and response times.

ACKNOWLEDGMENT

We are so thankful to Mathematics and Applications Laboratory and LIROSA Laboratory, Abdelmalek Essaadi University, Tangier and Tetuan, Morocco for providing support for this research.

REFERENCES

- [1] Jordan F., Modèle de prévision et de gestion des crues. Optimisation des opérations des aménagements hydroélectriques à accumulation pour la réduction des débits de crue, Ph.D. dissertation, Ecole Polytechnique Fédérale de Lausanne, 2007.
- [2] Onu, Guidelines for Reducing Flood Losses. International Strategy for Disaster Reduction. United Nations 2004.
- [3] Doussin N., Mise en œuvre locale d'une stratégie globale de prévention du risque d'inondation : le cas de la Loire moyenne, Ph.D. dissertation, Université De Cergy-Pontoise, 2009.
- [4] Elmabrouk M., Ezziyyani M., and Essaaidi M., Toward an optimal model based on inequality measures for treatment of historical & real time flood's datasets, in Proceeding of the International IEEE Colloquium of International On Information Science and Technology (CIST'14), Tetuan, Morocco, 2014.
- [5] Hernandez J. G., 'Flood management in a complex river basin with a realtime decision support system based on hydrological forecasts', Ph.D. dissertation, Ecole Polytechnique Fédérale de Lausanne, 2011.
- [6] Krzhizhanovskaya V., Melnikova N., Chirkin A., Ivanov S., Boukhanovsky A., and Sloot P., Distributed simulation of city inundation by coupled surface and subsurface porous flow for urban flood decision support system, *Procedia Computer Science*, Vol. 18, pp. 1046 – 1056, 2013.
- [7] Castellet E. B. I., Valentin M. G., Ripolles J. D., Onate E., Piazzese J., And Corestein G., Decision support system for flood risk assessment and management, in Proceeding of the 7th International Conference on Hydroinformatics (HIC'06), Nice, France, 2006.
- [8] Ahmad S. and Simonovic S., An intelligent decision support system for management of floods, *Water Resources Management*, Vol. 20, No. 3, pp. 391 – 410, 2006.
- [9] Raymond M., Peyron N., Bahl M., Martin A., and Alfonsi F., Espada : Un outil innovant pour la gestion en temps réel des crues urbaines, *NOVATECH*, pp. 793 – 800, 2007.
- [10] George J. P., Gleizes M. P., Glize P., and Rgis C., Real-time Simulation 'for Flood Forecast: an Adaptive Multi-Agent System STAFF, in Proceeding of the Symposium on Adaptive Agents and Multi-Agent Systems (AISB). University of Wales, Aberystwyth: Society for the Study of Artificial Intelligence and the Simulation of Behaviour, pp. 109–114, 2003.
- [11] George J. P., The AMAS Theory for Complex Problem Solving based on Self-Organizing Cooperative Agents. First European Workshop on Multi-Agent Systems (EUMAS'03). Oxford, UK, 2003.
- [12] Messika S., Méthodes probabilistes pour la vérification des systèmes distribués, Ph.D. dissertation, Ecole Normale Supérieure de Cachan, 2004.
- [13] Levine P. and Pomerol J. C., Systèmes interactifs d'aide à la décision et systèmes experts, *Traite des nouvelles technologies*, Paris, Hermes, 1989.
- [14] Simonovic S. P., Decision support system for flood management in the red river basin, Report to the Int. Joint Commission Red River Basin Task Force, Ottawa, Washington D.C, Tech. Rep. 1999.
- [15] Mullender S. J., 'Introduction to distributed systems', in Proceeding of the CERN School of Computing (CERN'92), Italy, 1992.
- [16] Gachet A., A new vision for distributed decision support systems, in Proceeding of the DSIage 2002 Conference, Ireland, pp. 341–352, 2002.
- [17] Boddy M. and Dean T., Solving time-dependent planning problems, in Proceeding of the 7th National Conference of Artificial Intelligence, Minneapolis, Minnesota, pp. 979–984, 1989.
- [18] Horvitz E. J., Reasoning about beliefs and actions under computational resource constraints, in Proceeding of the 1987 Workshop on Uncertainty in Artificial Intelligence, 1987.
- [19] Oufella S., Hamdadou D., and Bouamrane K., Conception d'un système interactif d'aide à la décision collective en localisation spatiale, in Proceedings of the 2nd Conference Internationale sur l'Informatique et ses Applications (CIIA'09), Saïda, Algeria, 2009.
- [20] Salvant T. and Brunessaux S., Multi-agent based time-critical decision support for c31 systems, in Proceeding of the Practical Applications of Intelligent Agents and Multi-Agents, PAAM, London, 1997.
- [21] Salvant T., CAAM, un modèle d'agent Anytime coopératif pour l'aide à la décision en temps critique, Ph.D. dissertation, ENST, Paris, 1998.
- [22] Duvallet C. and Sadeg B., Des systèmes multi-agents Anytime pour la conception de systèmes d'aide à la décision, *Technique et Science Informatiques*, Vol. 23, No. 8, pp. 997 – 1025, 2004.
- [23] Elmabrouk M. and Gaou S., Proposed Intelligent Pre-Processing Model of Real-Time Flood Forecasting and Warning for Data Classification and Aggregation, *International Journal of Online Engineering (iJOE)*, Vol 13, No 11, pp 4 – 24, 2017.
- [24] Gini C., Measurement of Inequality of Incomes, *the Economic Journal*, Vol. 31, No. 121, pp. 124 – 126, 1921.
- [25] Elmabrouk M., Ezziyyani M., Sadouq Z. A., and Essaaidi M., New expert system for short, medium and long term flood forecasting and warning, *Journal of Theoretical and Applied Information Technology (JATIT)*, Vol. 78, No. 2, pp. 286 – 302, 2015.
- [26] Baker T. J. and Miller S. N., Using the soil and water assessment tool (SWAT) to assess land use impact on Water Resources in an East African watershed, *Journal of Hydrology*, Vol. 486, pp. 100 – 111, 2013.

APPENDIX

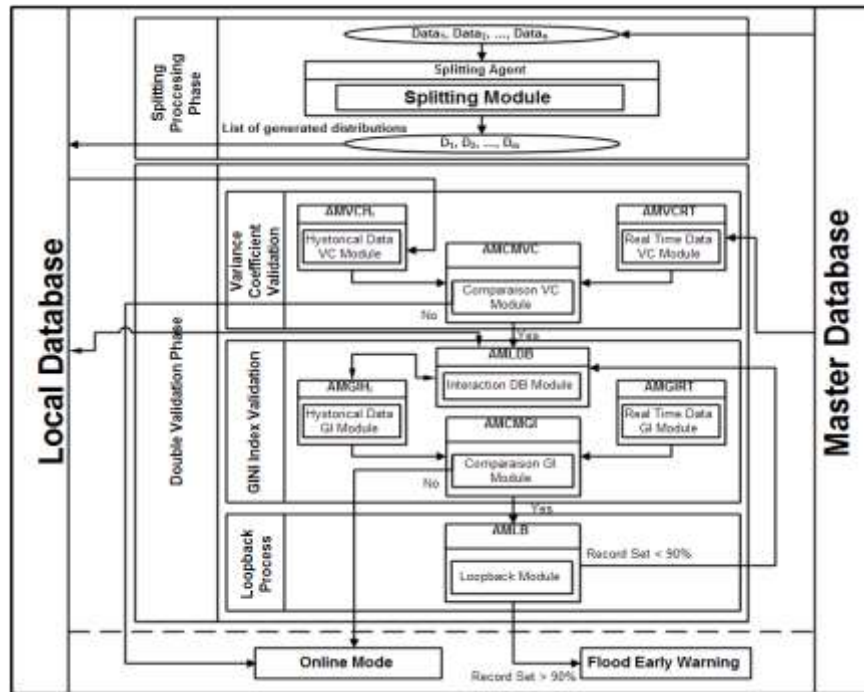


Fig. 23. Simulation of the Response Time of the System.

TABLE III. LORENZ CURVE TABLE

	B	D				
	FH(Z) / TotalFH(Z)	FRT(Z) / TotalFRT(Z)	D / B	CB	CD	 B - D
1	0,02214921	0,00113671	0,05132075	0,02214921	0,00113671	0,0210125
2	0,03603943	0,00303121	0,08410809	0,05818865	0,00416792	0,03300822
3	0,06976938	0,03059571	0,43852631	0,12795803	0,03476363	0,03917367
4	0,09184047	0,05426083	0,59081609	0,2197985	0,08902446	0,03757964
5	0,13883579	0,14155312	1,01957225	0,35863429	0,23057758	0,00271733
6	0,14225708	0,16720703	1,17538637	0,50089136	0,39778461	0,02494995
7	0,17147777	0,20936694	1,22095676	0,67236914	0,60715155	0,03788917
8	0,32763086	0,39284845	1,19905812	1	1	0,06521759
Total	1	1				0,26154808

TABLE IV. GINI INDEX CALCULATION

	B			
	$F_H(Z) / Total F_H(Z)$	C_D	$B * CD_i$	$B * CD_{i-1}$
	0	0	0	0
	0,02214921	0,00113671	0,000026	0,000041
	0,03603943	0,00416792	0,000151	0,000291
	0,06976938	0,03476363	0,002426	0,003193
	0,09184047	0,08902446	0,008177	0,01236
	0,13883579	0,23057758	0,032013	0,032802
	0,14225708	0,39778461	0,056588	0,068212
	0,17147777	0,60715155	0,104113	0,198922
	0,32763086	1	0,327631	0
Total			0,531125	0,315821

This section presents all the abbreviations used in this article.

TABLE V. ABBREVIATIONS

Abbreviation	Full explanation
ACL	Agent Communication Language
AMAS	Adaptive Multi-Agents System
AMDSS	Agent Mobile for Decision Support System
AMGHi	Agent Mobile Gini Historic data
AMGRT	Agent Mobile for Real Time data
AMH	Agent Mobile for Hydrodynamic module
AMLB	Agent Mobile for LoopBack module
AMLDDDB	Agent Mobile for Loading Data from DataBase
AMRS	Agent Mobile for Remote Sensing
AMRTCV	Agent Mobile Real Time Coefficient of Variance
AMS	Agent Mobile for Geographic Information System Module
AMVCHi	Agent Mobile Variance Coefficient Historic data
ANYMAS	ANYtime Multi-Agent Systems
Bool	Boolean
CVH	Coefficient of Variance of Historic data
DBMS	DataBase Management System
DSS	Decision Support System
FH(Fi)	Function of Historic data for Factor i
G	Gini
GI	Gini Index
GI Module	Gini Index Module
GIS	Geographic information system
GIS DB	Geographic information system DataBase
Hi	Historic data
HydroDN DB	HydroDyNamic DataBase
m/s	Meter/Second
m	Meter
mm	Millimeter
MAS	Multi-Agents System
RT	Real Time
s	Second
Sec	Second
SD	Standard Deviation
SWAT	Soil and Water Assessment Tool
VC	Variance Coefficient
VC Module	Variance Coefficient Module
Water L	Water Level
WSN	Wireless Sensors Network