World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:8, No:1, 2014

MATLAB-Based Graphical User Interface (GUI) for Data Mining as a Tool for Environment Management

M. Awawdeh, A. Fedi

***Abstract*—**The application of data mining to environmental monitoring has become crucial for a number of tasks related to emergency management. Over recent years, many tools have been developed for decision support system (DSS) for emergency management. In this article a graphical user interface (GUI) for environmental monitoring system is presented. This interface allows accomplishing (i) data collection and observation and (ii) extraction for data mining. This tool may be the basis for future development along the line of the open source software paradigm.

***Keywords*—**Data Mining, Environmental data, Mathematical Models, Matlab Graphical User Interface.

I. Introduction

THE management of environmental emergency is one of the most interested field that scientists are working to develop, where rapid environmental changes call for continuous surveillance and on-line decision making. The complexity of environment problems make necessary the development and applications of new tools capable of processing not only numerical aspects, but also the experience from experts and wide public participation, which are all needed in decision system.

As a part of decision support system for environmental emergency management, the data mining plays a main role in extracting data, analysis, and prediction. For Environment Monitoring System (ENMS), data come from measuring stations (i.e. meteorological ones), and the measurements flow from several sensors to support decision makers.

In this paper, we show a Matlab Graphical User Interface (GUI) as a tool for environmental applications. The aspect of environment emergency management we consider here regards data collecting and prediction, as its role in supporting decision making. We mention the data here as sensors measurements over real time observing. In the last years, a huge number of potentially useful methods and software tools have been proposed including methods for environment surveillance. Our tool's additive is to connect the monitored data after processing and extracting with powerful tools of Matlab for data mining, using Prediction, Classification, and Neural Network tools.

M. Awawdeh is with the Department of Mechanical, Energetic, Management, and Transport Engineering DIME- University of Genova, P.le Kennedy Pad. D, 16129 Genova, Italy (e-mail: awawdeh@dime.unige.it).

A. Fedi is with Acrotec Srl, Via A. Magliotto 2, 17100 Savona, Italy (e-mail: adriano@acrotec.it).

We present the data mining algorithms and methods application that meeting our project phases in a sequence regarding to [2].

The Tool is a contribution work in a project, named Integrated Network for Emergency (NIE). The interface is connected to the other parts of the project to complete a comprehensive system for environment management. Our role in the project is to support the decision making by scientific prediction tools.

II. Basics

A. Project Overview

The interface is a part of a project for environment surveillance, named Integrated Network for Emergency "N.I.E". This project has been performed by cooperated work of the University of Genova, "FadeOut" Company; it works on software programming, and "ACROTEC" Company, which is the main executor of this project. The main goals of the project are to monitor environmental changes, take measurements using multi sensors in different measuring stations as real time observing, display data on website, in addition to interactive modeling, broadcasting and early warning system, and environment measurements analysis depending on mathematical models using the definition of scientific prediction.

We will not discuss in this paper the whole project, whereas mentioned in the introduction, our role is to design the mathematical models for supporting the decision-making in environment emergency state; as a part of our work, the proposed interface has been designed.

B. Environmental Monitoring and Data Mining

Many environmental systems involve processes which are not yet well known, and for which no formal models are established at present. Because the consequences of an environmental system changing behavior or operating under abnormal conditions may be severe, there is a great need for Knowledge Discovery (KD) in the area.

Great quantities of data are available, but as the effort required to analyze the large masses of data generated by environmental systems is large, much of them are not examined deeply and the associated information remains unexploited. The special features of environmental processes demand a new paradigm to improve analysis and consequently management. Approaches beyond straightforward application of conventional classical techniques are needed to meet the

challenge of environmental system investigation. Data mining techniques provide efficient tools to extract useful information from large databases, and are equipped to identify and capture the key parameters controlling these complex systems [1].

The interactive Graphical User Interface proposed in this work helps in extracting, analyzing, and exporting data using the power of Matlab programming regards data mining techniques.

C. Working Prerequisites and Required Connections

From section (II.A), we showed that our interface is a part of comprehensive system, so we have initial tasks to run our interface, where this tool is not an open source yet and the data come to our interface by importing it from the server of the executive company (ACROTEC). The software and connections we use for this interface are: (1) Licensed Matlab Program version (R2011b) from MATHWORK, and a package of JAVA library. (2) CISCO Systems VPN Version 5.0.07.0240, connection is required to access the data in the company's core server.

III. Our Interface and It's Applications

A. Database and Data form

The database includes measurements from sensors over real time observing. These data flow to the core server in the surveillance management room of the executive company. The form of data we need to import in our interface is not compatible with the form of data in the server for two main reasons: (1) The form of data is not a numerical form so we couldn't manage by Matlab functions. (2) The programming language, which has been used to construct the data in the core server, is X-DROPS language; this language is a customized programming language, developed by ACROTEC Company for this project. This code is able to connect with MATLAB but the code itself is not readable by MATLAB. So the interface has to make the connection between MATLAB and X-DROPS to create the readable data for our interface before progressing its phases. The data flow to our interface as shown in Fig. 1.

The sensors are distributed in many locations and they are connected with the stations over transmission lines. The blocks "ACROTEC Databases" and "Data reprocessing" are our data source; because our system is not connected directly to the sensors. The connection with ACROTEC server provides our interface with all required data over on-line connections and it supports the feedback process.

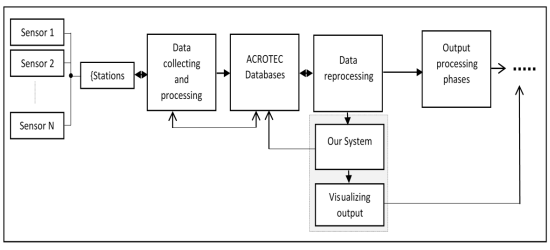

Fig. 1 Data flow block diagram from the source (Sensors) to our system (Interface)

Data preparing and integration is occurring in both stages, firstly in the core server and then in our interface processes corresponding to the target of data extracting. The first filtration of data occurs in the core server, and it is not shown in our system, but another phases of data filtering and errors formulation have been compiled with data selecting and extracting, to provide our interface by full access to process and extract data from the database tank. All data processing phases in our interface is dependent processes, changeable, and editable upon user targets.

B. Data Initializing and Collecting

The initialization of data occurs over the connection between Matlab and X-DROPS; the VPN connections build the bridge access to the database in the core server. This phase creates the connection with X-DROPS and after this initialization the interface is ready to start collecting data and processing them.

After this step, we implement the filtration as an in-filtering process, which filters the extracted data corresponding to the output coming from the phase of "Data reprocessing" (see Fig. 1). Depending on the data collecting target as we will show in the following section, the data flow inside our interface as shown in Fig. 2.

C. Interface Structure and Its Feature

The interface includes three phases of data processing, (1) Extracting data from the core database and build the bridge of data exchange between X-DROPS and Matlab. (2) Collecting data from sensors depending on specific search criteria determined by user. (3) Building databases inside Matlab and exporting these data to other connected interfaces as shown in (Fig. 1). The Data collecting features depend on user commands where there are two modes for collecting data: (1) Collect measurements depending on the type of sensors determined by user. (2) Collect measurement under the determinant of the geographical points. The modes of these phases are shown in the interface as multi-input choices for user.

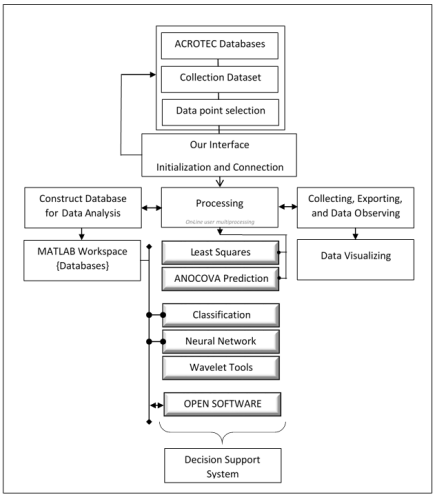

Fig. 2 Data flow and its destination inside our interface structure

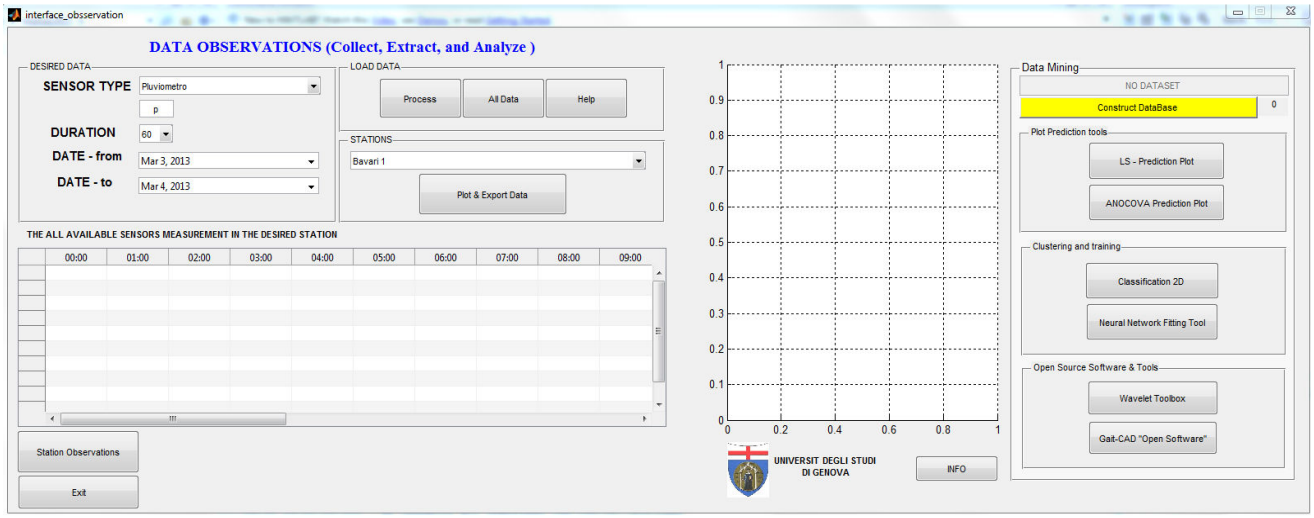

Fig. 3 Get Observations interface screenshot at the starting mode

The external database from which our interface extract data, is a collection of sensor measurements over time. It is composed by a matrix of observations, constructed as [stations×time] dimension. We collect the measurement as a function of (x, y), where x is the time in minutes, and y is the measurements. Fig. 3 shows the interface in the starting mode. From now on, we will call the interface as "Get Observations" interface.

At the starting mode, after the auto-initializing (where the initialization is occurring automatically when user open the interface) the system is ready to process the input and extract data up to user choices. Here, all input data are required to start processing regardless to the desired output. Required input and available output are shown in Table I.

TABLE I
INTERFACE INPUT AND OUTPUT DATA

Required input	All available output
Sensor type	Collecting all available data
Time-Duration	Plotting 2D (x, y)
Desired date (period)	Exporting to Data Mining
Station's name	Creating Datasets
	Constructing Databases

In Table II, we show the interface article rules corresponding to the three modes of data collection.

TABLE II
INTERFACE ARTICLES AND PROCESSES

Interface Article	Processing rule
Process	Observations of particular type of sensors in all stations in Genova
All Data	Observations of particular type of sensors in all stations in Italy
Station Observation	Observations of all sensor in one station in Genova

From Fig. 3, one can see that user has lists of input choices. The choices are corresponding to these questions from the user: What is the type of sensor? What is the observation period? For which date users want to get the measurements? and in which station?

We will not study the interface from the point view of system analysis, because we do not show each article of the interface and its rules. The goal of this paper is to present an easy-developed tool for environment emergency management based on Matlab programming and data mining methods. The Interface is divided into three phases: (1) The Input phase where user shall choose from the lists. (2) The visualized output, which is in two faces: a graph visualizing, and a table of collected measurements. (3) Data mining phase; contains the plot prediction tools, data clustering, data training tools, and open source software.

The data processing generates two types of extracted data: an auto-exported data shown in the interface-visualized output. This type feeds two levels of data mining: the "Least Squares" and "ANOCOVA" prediction tools. The other type of the generated data is the Databases; in this type, user constructs and exports data for the data mining phase from the tool "*Construct Database*" as shown in Fig. 3. The user can generate six different databases with six different input choices. Fig. 4 shows an example of a generated database.

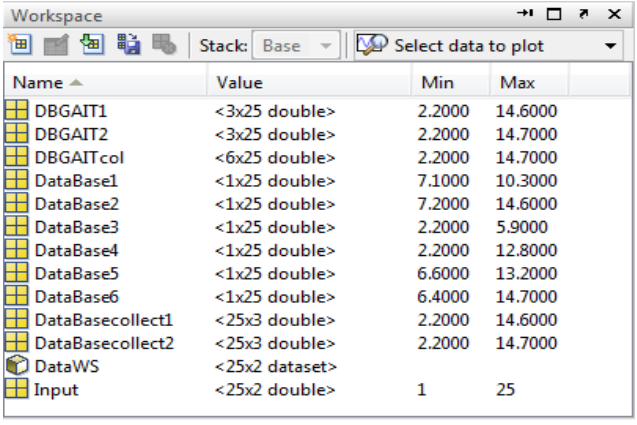

Fig. 4 An example of Databases have saved in the Workspace of Matlab

Here, we show an example of data observation with the specific inputs.

Example 1: (1) Sensor type: Thermometer. (2) Observing each 60 minutes. (3) In 11-December-2012. (3) Desired station is "Pegli2" station (see Fig. 5).

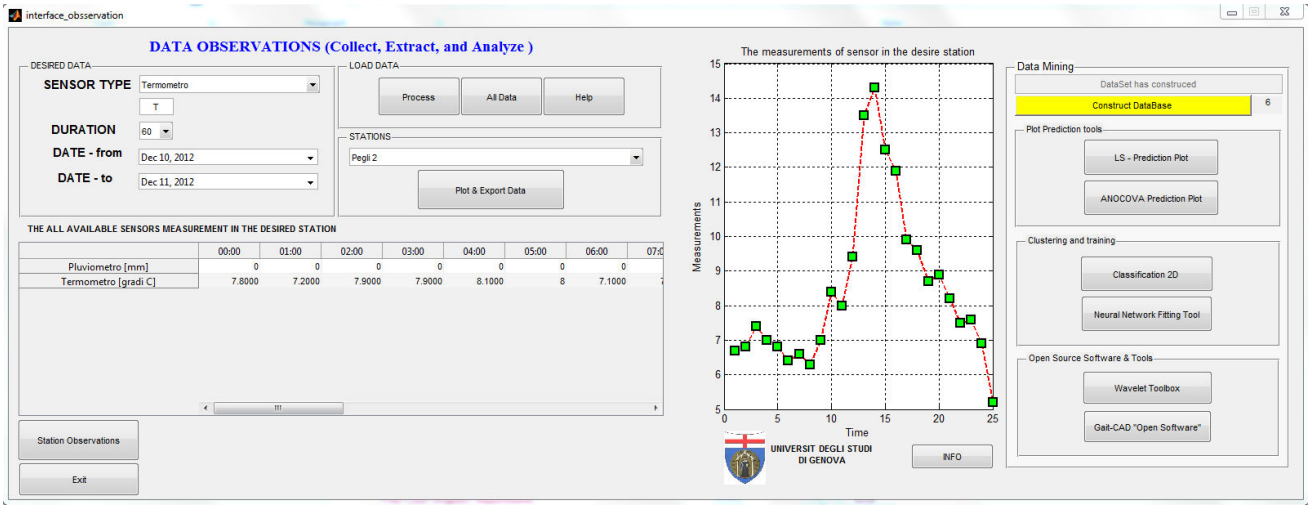

Fig. 5 Get Observations interface screenshot, example 1 at the Running mode

D. Data Mining Methods and Applications Inside

The two high-level primary goals of data mining in practice tend to be prediction and description. Prediction involves using some variables or fields in the database to predict unknown or future variables of interest and description focuses on finding human-interpretable patterns describing the data. The goal of prediction and description can be achieved using variety of particular data-mining methods [2]. Here, we show some of these methods corresponding to its rules in our system.

Before proceeding with our interface phases, we should mention one of the important steps in data mining applications. That is *"Outlier Detection"*. This step is a primary one in many data mining application. Johnson defines an outlier as "observation in a data set which appears to be inconsistence with the reminder of that set of data" [3]. In our interface, we use "Tukey's method – Boxplot" and a customized-weighted least squares method with robust regression to detect outliers. Tukey's method of constructing a boxplot, is a graphical tool to display information about continuous univariate data, such as the median, lower quartile (Q1), upper quartile (Q3), lower extreme, and upper extreme of a data set [4].In this mode to minimize the influence of outliers, we fit our data using robust-least squares regression using "Bisquare Weight"; method -Matlab support- this method minimizes the weight sum of squares. We have programmed all these mathematical procedures to simplify that for users, since not all users in the data analysis field are able to deal with the pure mathematical models. One of the future work we hope to progress-in is to develop this filter (i.e. outlier detection filter) to be an industrial filter.

We don't apply this process for all data, but upon user request in the data analysis phases (just to display all data including its extreme value to be notified from user then to export it for detecting outlier before the analysis). Matlab code has been programmed to process the data using Tukey's method. The code consists of data processing for outlier detection using the definitions of quartile, boxplot, and outlier removal, with the ability to exchange the value of outlier with some proper values that reduce the effect of extreme values over regression analysis.

Fig. 6 shows a snapshot for data (measurements for six days) including outliers, which have been detected by Tukey's method.

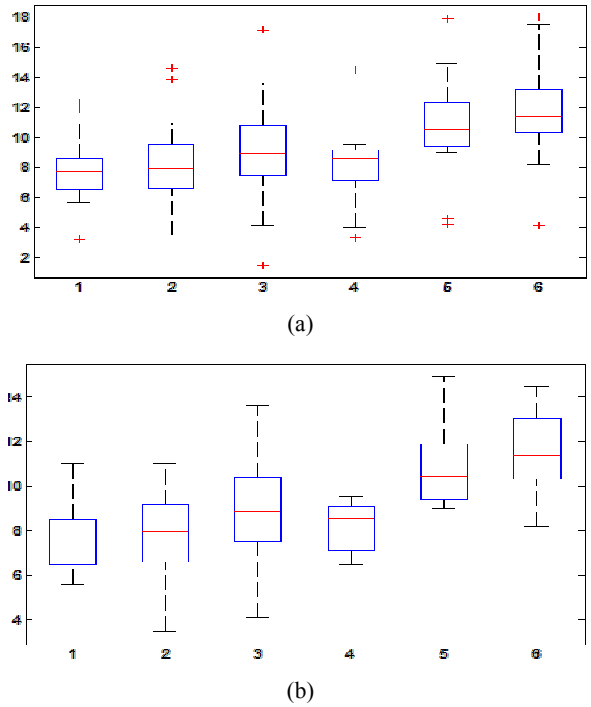

Fig. 6 (a) Outlier detection removal display snapshot. Outliers have been removed. (b) Outlier detection graphical display snapshot. The red crosses are the outliers

Classification and Clustering, Classification is learning a function that maps (classifies) a data item into one of several predefined classes [5]. Clustering is a common descriptive task where one seeks to identify a finite set of categories or cluster to describe the data [6]. In our system, we use Fuzzy C-Means Clustering (FCM); FCM is a data clustering technique in which a dataset is grouped into n clusters with every data point in the dataset belonging to every cluster to a certain degree [7]. The following example (example 2), shows a 2D classification using Fuzzy C-Means techniques for measurements taking from ("Thermometer" sensor) in particular station ("Pegli2") where these variables are observed each 30 minutes for 24 hours in some day.

Example 2: A plot of 2D classification for measurements from "Thermometer" sensor. We use (3) clusters for this dataset, where the number of desired clusters is upon user choice, taking into account the dataset size. (See Fig. 7)

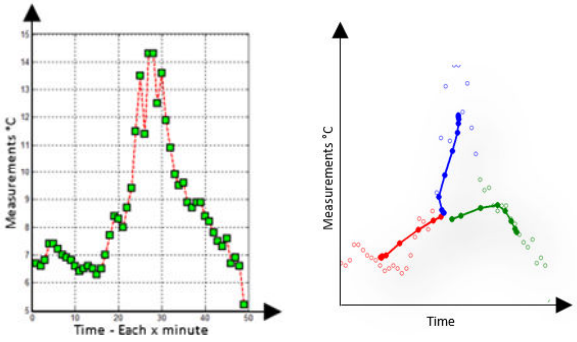

Fig. 7 An example of 2D classification – 3 Cluster, for measurement of "Thermometer" sensor; observations through 24hours each 30 minutes

Regression is learning a function that maps a data item to real-valued prediction variables [2]. In the Data mining panel as shown in Fig. 5, we implemented two panels; one as "*Plot Prediction tools*" and the other for "*Classification and Neural Network"*. In the "Plot Prediction tools", firstly we have Least Squares Prediction Plot, which is nonlinear regression method for prediction. We use the Matlab function "*Polytool*" (Interactive polynomial fitting) in a least squares sense. The data for this prediction plot have already implemented from the processing phase under the process of "*Plot & Export Data*" where the values of (x, y) are exported as we mentioned before. User can use the interface to explore the effects of changing the fit parameters and to export fit results to the workspace. From example 2, one can see the plotting of the measurements; these variables are exported to the least square plotting and its prediction curve is shown in Fig. 7 as quartic mode.

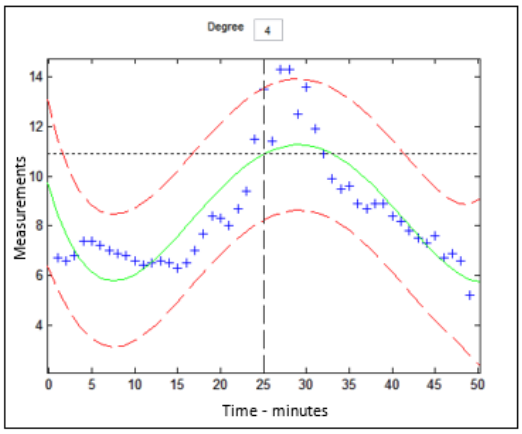

Fig. 8 A prediction plot of quartic model, for the measurements of example 2, in the interface of Least Squares, one can change the degree to see different models (Linear, quadratic, cubic, etc.). In addition, to export variables to the workspace

The second tool of plot prediction tools is "*one-way analysis of covariance*" (ANOCOVA) models.We implement the measurements (x, y) as auto-exported variables from graph. ANOCOVA models process our measurements, and plot the prediction curves corresponding to five different models (same mean, separate means, same line, parallel lines, and separate lines), and respect to our modes of "*grouping*", we grouped the measurements as periods of measurements observations where we designed the modes as 24 hours of observations over measurements each 30 minutes, as these period of time: (00:00-06.00, 06.00-12.00, 12.00-18.00, 18.00-00.00). This powerful tool provides us by three types of outputs: (1) An interactive graph of the data and prediction curves. (2) An ANOVA table. (3) A table of parameter estimates. Example 3 (Fig. 9) illustrate ANOCOVA prediction plot.

Example 3: ANOCOVA prediction plot for the variables from example 2, with these features: Model: *Separate Lines*, Group: *All Group*.

For these two types of prediction plots (Least squares and ANOCOVA), we use a design process of data mining algorithms (see Figs. 9-11). The idea of this design was developed by Mikut et al. [8].

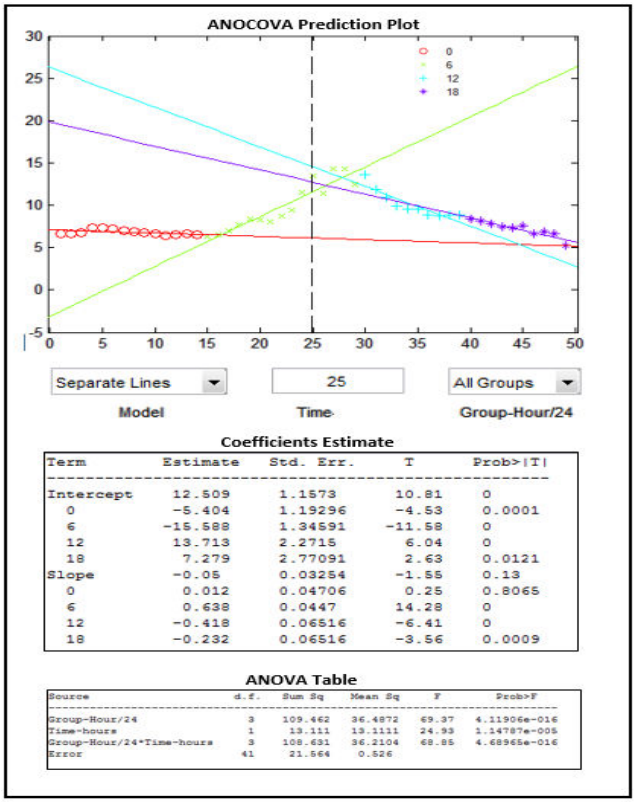

Fig. 9 An example of ANOCOVA prediction, for the same measurement in the example 2, from this figure one can see the three output we mentioned above in that example. We apply ANOCOVA using the model "*separate lines*" and for all groups (0,6,12,18). The sensor measurements are 49; they are the reading each 30 minute through 1 day (24 hours). The user can choose any model to see different fits, in addition, to choose any of the four groups where one each demonstrates the measurements from 00.00 o'clock with different numbers of variables for each group

R. Mikut et al. [9]implemented the standardized data mining process proposed from [8] in the toolbox Gait-CAD. This tool box is an open source software bases on Matlab and it operates by a graphical user interface. We edited some of their design process blocks to meet our need of processing.

The "*Error formulation*" has been integrated with the source code of our interface where the *detection process* of measurements collections is running over predefined variables and structures of the sensors measurements and corresponding to pre GIS definition, where we constructed the source code to face the on-line data detection.

Nonlinear Regression and Classification methods, these methods consist of a family of techniques for prediction that fit linear and nonlinear combinations of basis functions (Sigmoids, splines, polynomials) to combinations of the input variables [2]. Neural networks method is one of the most common methods of data mining, it used for classification, clustering, feature mining, prediction, and pattern recognition. One of neural network types is *"Feed-forward networks"*, that regards the perception back propagation model and the function network as representatives, and it is mainly used in the areas such as prediction and patter recognition.

We have chosen this type of neural networks to train our data (measurements in the database). As we mentioned in section (III), through the data collection process we construct databases respect to our destination of monitoring. These databases flow in two way (i) as available variables in the workspace and (ii) as a saved file in the directory *(.dat extension files*). These data are constructed up to the input variables, which have been processed.

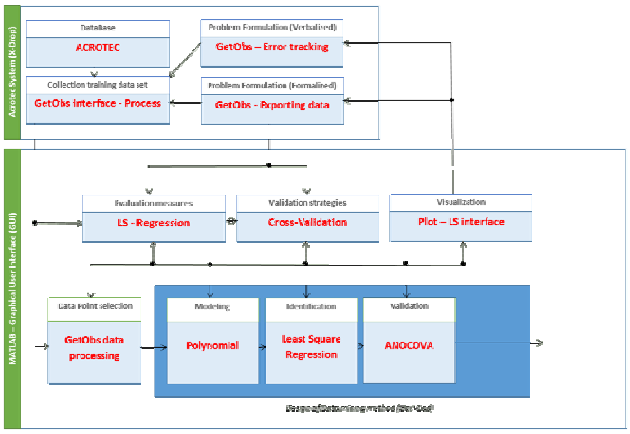

Fig. 10 Design process of data mining algorithms for Least Squares. The design of this algorithm fit with our need of output and support the on-line models. Moreover, the processing operations in the "Get Observation" interface have been constructed to fit with the future development

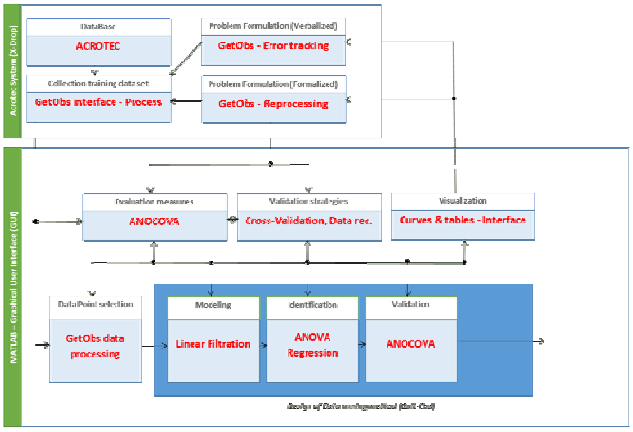

Fig. 11 Design process of data mining algorithms for ANOCOVA predictions

These data will be trained with *"Levenberg-Marquardt backpropagation algorithm"*, The Levenberg-Marquardt algorithm, which was independently developed by Kenneth

Levenberg and Donald Marquardt, provides a numerical solution to the problem of minimizing a nonlinear function. It is fast and has stable convergence [10]. It gives a good exchange between the speed of the Newton algorithm and the stability of the steepest descent method.

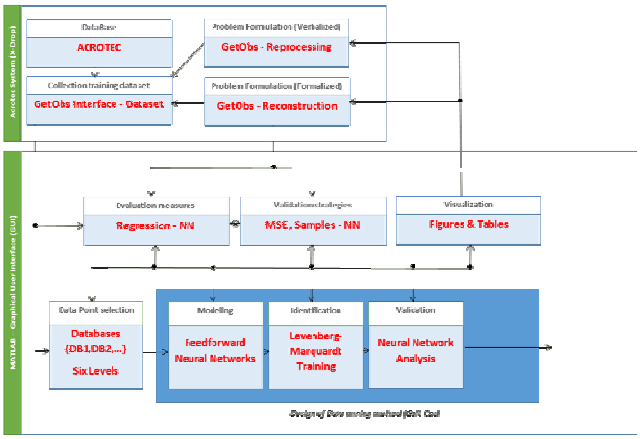

Fig. 12 Design process of data mining algorithms for Feedforward neural network training

The update rule of Leven-Marquardt algorithms can be written as:

$$w_{k+1} = w_k - (J_k^T J_k + \mu I)^{-1} J_k e_k \text{, where}$$

w : *The weight,* J *: Jacobian,* μ *: Combination coefficient which is always positive,* I *: The identity matrix.*

As the combination of steepest descent algorithm and the Gauss-Newton algorithm, The Levenberg-Marquardt algorithm switches between the two algorithms during the training process. (See [10]).

The design process for data mining algorithms for our neural network has been published to fit with the future opportunity of developing the interface as open source software.

Therefore, we use again the design of [8] to show the design of data mining method (see Fig. 12).

Example 4, this example shows the regression for a small dataset (see Figs. 13 and 14), which has been constructed as measurements from thermometer sensor during three days over observing each 60 minutes. The input file has been imported as 'DBASE1', which is a [3x25] matrix, representing static data: 25 samples of 3 elements, with these percentage selections: 70% of samples for training, 15% for validation, and 15% for testing.

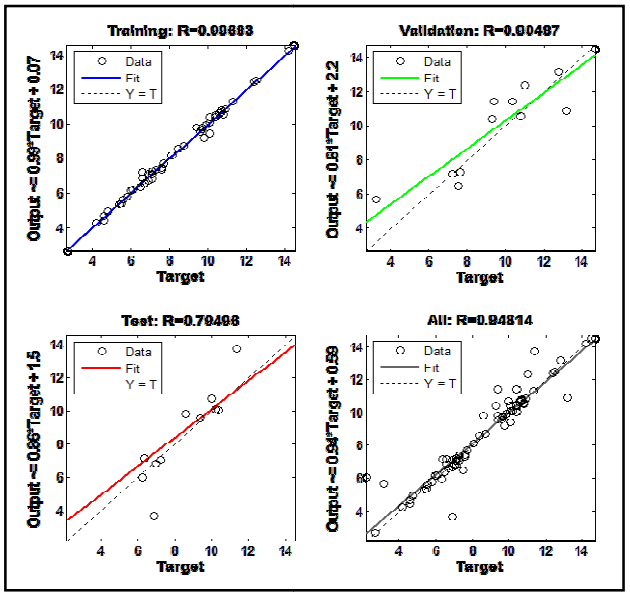

Fig. 13 Regression, dataset of example 4

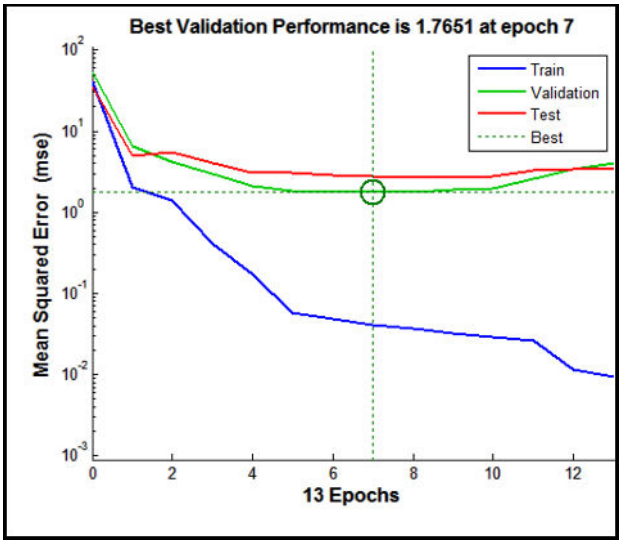

Fig. 14 Neural network training performance, dataset of example 4

Interface & operations	Modeling & Identification	Programming based	Output patterns
Get Observation	Main interface (Data extracting and fitting) based on Matlab & XDrops, a combination programming.		
Least Squares	Polynomial – Regression	Matlab- GUI	Prediction curve
ANOVOCA	Linear filtration	Matlab- GUI	Prediction curves & tables
Fuzzy C-Mean	Classification	Matlab- GUI	Data clustering
Neural Network	Feedforward neural network , Levenberg-Marquardt	Matlab toolbox	Regression, fitting, training data, and results

Fig. 15 Interfaces output and programming based, "*Get Observation*" mentions our interface and which it includes the other interfaces in its panels

IV. Data Mining Algorithms and Review

There are three primary components in any data-mining algorithm: (i) model representation (ii) model evaluation, and (iii) search. [2],[12]-[16].

Model representation our interface is operated by a graphical user interface based on Matlab. This representation can be divided into two faces: (i) Hidden programming for data collecting which based on X-DROPS code, at the end of this stage the system has been initialized and the discoverable patterns are described into graphical user interface based on Matlab. (ii) Matlab programming, which the interface has been built and executed over Matlab code using (GUI) tools. Interactive graphical user interface describe the discoverable patterns as graphs, tables, analysis, numerical results, etc. As mentioned before we can summarize the representation for our interface and interfaces inside as shown in Fig. 15.

Another component of data mining algorithms is *Model-evaluation criteria*, as mention in [2], this component are quantitative statements of how well a particular pattern (a model and its parameters) meets the goal of knowledge discovery in database (KDD) process. KDD is the "*nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data*" [2]. Our interface meets this definition by its application. The understandable-constrains plot of measurements, which have been extracted corresponding to specific user criteria and have been exported to the data models (predictive models, data training pattern, clustering, and data fitting) give a clear idea for the observer about the desired data with their analysis. All of models have been tested under a randomly user input choices; test set has been used to examine the predictive model accuracy with other multi-input levels.

Search method, the third components of data mining algorithm, which it consists of two components: (1) parameter search and (2) model search. In this stage data mining task is reduced to purely an optimization task: find the parameters and models from selected family that optimized the evaluation criteria [2], [11]. (See Fig. 16).

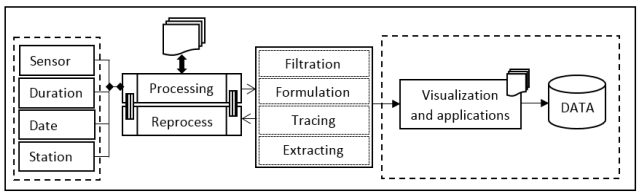

Fig. 16 Interface data flow and process from the search model up to extracted data. More details about the search model have been discussed in the previous sections of this paper

V. Conclusion

The aim of these tools is to provide an interface for applying data mining methods in the environmental applications. This interface plays an intermediate-cooperated role between two fields: environmental monitoring and data mining. We use this interface to collect data from sensor (extract form database) and applying the data mining methods on these sets of data, finally for extracting theses data in understandable and flexible model to be used in the decision support system. The system has been designed to fit with real on-line observing and the algorithms have been constructed flexibly to meet with future needs. This interface will be developed in an upgrading-phase to be as an *open source software*.

Acknowledgment

Authors would like to extend grateful thanks to professor Angelo Alessandri and professor Patrizia Bagnerini from the department of mathematical engineering and simulation in the university of Genova, Cosimo Versaci from ACROTEC Company, for there guidance and their appreciated efforts. In special way, the author wishes to acknowledge all members of the project (N.I.E), in providing the data on which this toolwas based.

References

[1] Jessica Spate, Karina Gibert, Miquel S`anchez-Marr`e, Eibe Frank, Joaquim Comas, Ioannis Athanasiadis, Rebecca Letcher, "Data Mining as a Tool for Environmental Scientists". *Al Magazine* Vol.17.1996.
[2] Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth, From Data Mining to Knowledge Discovery in Databases, Article.
[3] Richard Johnson and Dean Wichern, "Applied Multivariate Statistical Analysis", 1992.
[4] John W. Tukey, "Exploratory data analysis", 1977.
[5] Weiss, Sholom M. and Kulikowski, Casimir A. "Classification and prediction methods from statistics, neural nets, machine learning, and expert systems", 1991.
[6] Anil K. Jain and Richard C. Dubes. "Algorithms for data clustering", 1988.
[7] Math Works, http://www.mathworks.it/index.html
[8] Mikut, R.; Reischl, M.; Burmeister, O.; Loose, T.: Data Mining in Medical Time Series. Biomedizinische Technik, vol. 51, pp. 288–293, 2006.J.
[9] R. Mikut, O. Burmeister, S. Braun, M. Reischl, "the open source matlab toolbox gait-cad and its application to bioelectric signal processing", Paper, Institute for Applied Computer Science, Forschungszentrum Karlsruhe GmbH, Germany.
[10] Hao Yu and B. M. Wilamowski, "Levenberg–Marquardt Training" Industrial Electronics Handbook, vol. 5 – Intelligent Systems, 2nd Edition, chapter 12, pp. 12-1 to 12-15, CRC Press 2011.
[11] NIE project documents, ACROTEC Company. http://www.acrotec.it
[12] Muhammad Aqil, Ichiro Kita, Akira Yano, and NishiyamaSoichi. "Decision Support System for Flood Crisis Management using Artificial Neural Network", International Journal of Electrical and Computer Engineering 1:5 2006
[13] David Hand, Principle of Data Mining. Massachusetts Institute of Technology, 2001.
[14] Feng Jiansheng. KDD and its applications, BaoGangtechniqyes. 1993.
[15] Xianjun NI, Research of Data Mining Based on Neural Networks. World Academy of Science, Engineering and Technology, 39 2008.
[16] Ben-Gal, Data Mining and Knowledge Discovery Handbook, chapter one."Kluwer Academic Publisher, 2005".