Abstract — The paper presents a tool supporting the smart pre-elaboration and analysis of large datasets that implements complex algorithms for variable selection and outliers detection. These pre-elaboration stages are fundamental for further data exploitation in modelling and classification tasks. Moreover, the identification of anomalous data and of the most relevant variables affecting a process or a phenomenon is crucial in data mining, as it supports the extraction of knowledge from the data.

The proposed algorithms come equipped with a user-friendly graphical interface that helps the interpretation of the analysis and can be applied to any context, especially in industrial ones.

Keywords - component; pre-processing; data mining; interface; variable selection; outlier detection

I. INTRODUCTION

Variable selection and outliers detection are important pre-processing stages in machine learning and data mining applications. In modelling applications, applying artificial intelligence techniques (e.g. neural networks), the data quality and an accurate selection of the input variables is relevant for the development of a consistent model. Indeed, many real-world datasets are affected by errors and variables redundancy, therefore the preliminary improvement of the quality of data is fundamental for their further exploitation in any kind of application. Variable selection is a procedure addressed to identify the variables that mostly affect the target to be predicted in order to create a subset, which includes only the features that are uncorrelated and informative [1]. This procedure is principally significant dealing with industrial datasets as the number of input variables can be considerable with respect to the number of available samples [2]. Nowadays in the industrial context a huge number of features are usually measured by sensors which are located, for instance, along the whole production chain and the reduction of the feature space is essential in order to perform a consistent prediction model. Moreover, a careful selection of the variables which mostly affect a phenomenon is of utmost importance in order to improve process monitoring and control [3, 4]. The issue of variable selection is largely investigated in literature for traditional purposes such as prediction [5][6], classification [7-12] or clustering [13]. On the other hand, outliers are anomalous patterns that deviate from the rest of data [14]. They can occur due to different reasons, such as measurements errors or variability of the investigated phenomenon. An outlier can be an anomalous pattern to be removed or a rare event to be detected, as an indicator of a fault, an intrusion attempt or an anomaly. Therefore, outlier detection is essential in many different applications, such as fraud detection, financial analysis, network intrusion detection and many others [15-19].

In this paper two software modules are described, implementing two innovative methods of variable selection and anomalous data detection developed in C#; both modules are equipped with an intuitive graphical interface in order to simplify data loading, algorithms startup and results analysis. The innovative aspect regards the implementation of a user-friendly graphical interface that helps the interpretation of the analysis and can be applied to any context due to its modular nature.

The paper is organized as follows: Section II reviews traditional variable selection methods and outlier detection techniques, moreover, in addition to the state of the art, two innovative methods are described concerning the two main topic that have been implemented in the interface. Section III explains the particular industrial application, while Section IV describes the functionality of the implemented interface. Finally section V proposes some concluding remarks.

II. DATA PRE-PROCESSING

A. Review on Variable Selection

The difficulty of selecting the most significant input variables can be principally due to the large size of the initial inputs set, the correlations among variables which cause redundancy and the occurrence of variables which do not affect the considered phenomenon or process. The ideal set of inputs should comprise only the variables needed to describe the behavior of the considered process with no or minimal redundancy in order to create an accurate, effective and more interpretable model [20]. In literature, variable selection algorithms are classified into 3 main approaches: filter, wrapper and embedded.
The filter methods can be considered as a pre-processing phase, they are independent from the learning algorithm carried out and their computational burden is restrained. These approaches consist on trying statistical tests to perform the selection by computing a pertinence score for each variable: the variables with higher score are selected. A popular example of filters is the correlation-based approach [21] but recently, several more sophisticated approaches have been proposed [22-23]. The wrapper approaches are introduced in 1997 [24] and they consider the learning algorithm as a black box exploiting its performance to select a subset of variables through their predictive power. A recognizable wrapper method is the exhaustive search or brute force method which analyses all combinations of variables but if the number of available input variables are significant this approach becomes impracticable. Also, other sub-optimal methods, based on genetic algorithms have been proposed in order to overcome this problem [25-26].

Finally the embedded approaches perform the variable selection as part of the learning stage and are usually particular of a specific learning machine. The main advantage of embedded approaches lies in the association with the learning algorithm. The embedded methods have a lower computational cost than wrapper approaches but they are too specific of a given algorithm.

Recently, hybrid variable selection approaches have been introduced in order to get the benefits of some approaches and overcome the shortcomings of others [27].

The software component presented in this work belongs to wrapper approaches. It is based on genetic algorithms [6], it is implemented in C# and comes with a Graphical User Interface GUI. The proposed approach selects the best set of variables to be fed in input to a neural network within a regression problem. The chromosomes are binary coded and their length is equal to the number of variables, so that each gene corresponds to an input variable. When the gene is unitary it means that the associated variable has been selected. The fitness function is calculated as the performance of a feed forward neural network but it can be modified according to the characteristics of specific applications. The fitness function is calculated for each chromosome of the population and crossover and mutation operators are applied.

The generic scheme of the proposed approach is shown in Figure 1. The algorithm stops when a fixed number of iterations are reached or a plateau for the fitness function is achieved. In order to obtain a stability of the variable selection algorithm, several iterations are run and the variables with a frequency in the winner chromosome of at least 80% are selected and considered significant to explain the particular application to be studied.

B. Review on Outlier Detection

The most popular outlier detection methods can be categorized into four main classes: distance-based, density-based, clustering-based and distribution-based.

The distance-based approaches were introduced in 1998 [28] as a consequence of the following definition of outlier: “An object x in a dataset T is a DB(p,D)-outlier if at least fraction p of the objects in T lie at a distance greater than D from x”. These approaches are based on the definition of distance between a point and the complete dataset and depend on the distance metric that is used. A classical distance-based method implies the evaluation of the Mahalanobis distance between the selected object and the center of the mass of the whole dataset [29].

Density based approaches, introduced in 2000 [30] attribute to each datum a degree of outlierness on the basis on how much it is “isolated” with respect to his neighborhood.

Recently, hybrid variable selection approaches have been introduced in order to get the benefits of some approaches and overcome the shortcomings of others [27].

The software component presented in this work belongs to wrapper approaches. It is based on genetic algorithms [6], it is implemented in C# and comes with a Graphical User Interface GUI. The proposed approach selects the best set of variables to be fed in input to a neural network within a regression problem. The chromosomes are binary coded and their length is equal to the number of variables, so that each gene corresponds to an input variable. When the gene is unitary it means that the associated variable has been selected. The fitness function is calculated as the performance of a feed forward neural network but it can be modified according to the characteristics of specific applications. The fitness function is calculated for each chromosome of the population and crossover and mutation operators are applied.

The generic scheme of the proposed approach is shown in Figure 1. The algorithm stops when a fixed number of iterations are reached or a plateau for the fitness function is achieved. In order to obtain a stability of the variable selection algorithm, several iterations are run and the variables with a frequency in the winner chromosome of at least 80% are selected and considered significant to explain the particular application to be studied.

B. Review on Outlier Detection

The most popular outlier detection methods can be categorized into four main classes: distance-based, density-based, clustering-based and distribution-based.

Clustering based approach classify as outlier a sample which does not belong to any cluster, after an appropriate clustering operation. The clustering method which is used in the implemented approach is the fuzzy C-means (FCM) algorithm, described in [31].

Finally the distribution-based approaches identify an outlier as a sample that does not fit well with a standard distribution. The most popular distribution-based approach is
the Grubbs test [32], which classifies as outliers the samples that deviate from the Gaussian distribution function.

The approach which is implemented in the tool which is described in the present paper combines different traditional approaches through a Fuzzy Inference System (FIS) [33-34].

The proposed approach is automatic and self-adaptive, as several parameters that are required for the combination of the approaches are automatically calculated by using the available dataset, without the need for a-priori assumptions.

The first step of the proposed fuzzy approach consists in the extraction of four features for each pattern lying in the range [0; 1] and calculated by applying the four different outlier detection methods. The four features represent the outlierness degree that is obtained by exploiting the following classical outlier techniques, namely:

- distance-based method using the Mahalanobis distance;
- density-based method computing the local outlier factor [30];
- clustering-based method applying the FCM technique [35]. The number of clusters, which is required by the algorithm, is automatically evaluated through a validity method [36];
- distribution-based method determined by applying the Grubbs algorithm separately to each component of the data vector. In this case, the output of the Grubbs test is null in the case the datum is certainly not considered an outlier, unitary otherwise.

The four features are fed in input to a FIS, which outputs an outlierness index in the range [0; 1]; if the index is close to one, the associated pattern is classified as an outlier [37]. Figure 2 depicts the scheme of the proposed method.

III. THE INDUSTRIAL CONTEXT

The design of Complex Mechanical Dynamic Systems requires huge volume of data, to be collected from different analyses such as fluid-dynamic, thermodynamic, mechanic, vibrations and noise, materials, resistance. These data need to be exploited through an integrated approach for both technical goals, such as machine design optimization and components residual life prediction, and management tasks by exploiting big data analytics, which face issues of complexity, loss of information and bottlenecks.

The PROMAS project, within which the described tool has been developed, aimed at the creation of hardware and software infrastructure, for supporting the machine and complex system design, through the integration of different disciplines, following the Model Driven Engineering (MDE). The infrastructure has been located in the facilities of Scuola Superiore Sant’Anna in Pisa (Tuscany, Italy) and has been arranged in four fundamental elements:

- Computational “core” with its local storage and communication systems. Communications are exclusively “trusted”. Those system are predisposed to interact with innovative components, to manage complex computations when, during simulations and analysis, big data are involved.
- Data-centered Approach, where Dassault iSight, IChrome, ModeFrontier, and Phoenix applications are integrated and managed by different tools and entities, through a dynamic and safe/secure database with adequate dimensions.
- Advanced tools for analysis and big data managing.
- Simulation tools of complex, non-linear systems.

This infrastructure allows easy access to data, simulation and analysis also from remote sites to all authorized entities, on the base of confidentiality and access rights.

The Key objectives of the PROMAS project are:

- Integrations of different disciplines through the creation of hardware and software infrastructure for supporting the machine and complex system design, following the Model Driven Engineering (MDE).
- Models Creation supported by software implementation working on complex data;
- Simulation supported by integrated systems, on the base of collected data;
- Fast Data Analysis due to fast communications and time reduction, realized with the integrated architecture;
- High Performance Computation due to time reduction and best performance in terms of speed of system integrations;
- Photonics deployed for digital and fast communications, internal and external the system.

The data analytics tool, which is described in this paper, targets the data analytics goal though an easy-to-use and standalone tool, which can be used by the platform users or can be downloaded. The strategic context of the PROMAS project is referred to the European planning “POR FESR 2014-2020”. Strategic pillars are oriented to push and
motivate the production system of Tuscany to innovate, bringing modernization, being always competitive on the market of interest.

IV. INTERFACE

The above-described algorithms for selecting variables and detecting outliers were implemented in C# and equipped with a graphical interface that allows their easy execution. Also by users without great experience in data analytics. This choice derives from the need to realize a fast and usable tool independent on any software license. The two main algorithms are implemented in order to exploit at best the computational power of the platform via multi-threading, by parallelizing some parts of the algorithm in order to improve their efficiency even when treating very large databases, as usual in the context of industrial systems.

For both analyses, the dataset should be available as a .csv table, which is loaded in a simple way. The module carries out the required analysis and displays the result. The main characteristics of the analysis and filtering modules are described in the following.

A. Variable Selection Interface

Figure 3 shows the main screenshot of the graphical interface for the GA-based variable selection described in section II. Figure 3 refers to a regression application implemented using the popular MLP neural network. This network topology is the one most commonly used, but it is automatically sized considering the number of samples available and the number of selected variables in order to avoid problems of overfitting or mal-conditioning. The module automatically splits the dataset into two subsets, one for the training and one for the validation phase.

As the initial population is randomly selected, a variable can be included in the winner chromosome even if it is not correlated with the target to predict. In order to solve this problem and to enhance the algorithm stability [38-39], the GA is run more times (typically 10) and all winner chromosomes are stored. Finally, the presentation frequency of each variable in the winning chromosomes is computed and only the variables that are present in more than the 80% of the winning chromosomes are considered as really affecting the target to be predicted.

The interface, visualizes in the top left window (see the red rectangle in Figure 3) the variables selected by each GA run and the performance (the prediction error in this particular case), of the winning chromosome in the training set and in the validation set. The lower left window, (highlighted by a green rectangle in Figure 3) shows the final outcome and displays the selected variables set and the corresponding value of the fitness function.

B. Outlier Detection Interface

Figure 4 shows the main screen of the graphical interface for FIS-based outlier detection algorithm, which is described in detail in the section II. In particular, Figure 2 refers to an application that exploits a synthetic database of two-dimensional data. The interface allows selecting the representation of data in the plot placed in the right side of the window, i.e. it is possible select the couple of variables reported on the horizontal and vertical axes of the diagram. The data, which are identified as outliers, are highlighted in blue color, while the others are represented in red color. The left window reports the analysis of each sample, i.e. the value of the 4 indexes used to evaluate the probability that a datum is an outlier, and the final “outlierness degree” in the range [0, 1]. A further interesting functionality of the interface is the visualization of the samples detected as outliers according to the traditional approaches, in order to compare the results obtained with the FIS-based approach.

In the interface, there are also two other panels on the left: the blue box shows the results of the analysis performed using the different techniques on the complete dataset, while the green box shows the results of the outlierness degrees obtained only considering samples that have been classified as outliers (representing a subset of the previous window). Figure 5 shows the outcome of the same analysis performed on an industrial database having a dimensionality much greater than the previous one.

V. CONCLUSION

The paper describes two software modules implementing advanced variable selection and outlier detection procedures on a novel hardware and software infrastructure, for supporting the design of complex machineries and systems, through the integration of different disciplines. The modules have been specifically designed for exploiting the computational capability of the platform and for processing large volumes of data. The modules are available as service on the platform but can also be downloaded and locally used. The platform will be running also after the completion of the project in whose context it has been developed and will allow also external user to exploit the provided systems and services, including, among others, data analytics.

ACKNOWLEDGMENT

The work presented in this paper was developed within the project entitled ”Piattaforma Integrata Avanzata per la Progettazione di Macchine e Sistemi Complessi” (PROMAS), which was co-funded under Tuscany POR FESR 2014-2020.
REFERENCES


Figure 3. Executive module for variable selection.

Figure 4. Executive module for variable selection.
Fig 5. Executive module for outlier detection: example on real industrial data