

Received September 7, 2021, accepted September 22, 2021, date of publication September 27, 2021, date of current version October 8, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3115848

# Multi-Objective Evolutionary Simultaneous Feature Selection and Outlier Detection for Regression

FERNANDO JIMÉNEZ<sup>1</sup>, ESTRELLA LUCENA-SÁNCHEZ<sup>2,3</sup>, GRACIA SÁNCHEZ<sup>1</sup>, AND GUIDO SCIAVICCO<sup>1b2</sup>

<sup>1</sup>Department of Information Engineering and Communications, University of Murcia, 30100 Murcia, Spain

<sup>2</sup>Department of Mathematics and Computer Science, University of Ferrara, 44121 Ferrara, Italy

<sup>3</sup>Department of Physics, Informatics and Mathematics, University of Modena and Reggio Emilia, 41121 Modena, Italy

Corresponding author: Guido Sciacvico (scvgdu@unife.it)

This work was supported in part by the Spanish Ministry of Science, Innovation and Universities through the SITUS Project under Grant RTI2018-094832-B-I00, in part by the European Fund for Regional Development (EFRD, FEDER), in part by the Emilia-Romagna (Italy) Regional Project 'New Mathematical and Computer Science Methods for the Water and Food Resources Exploitation Optimization,' and in part by the University of Ferrara under the FIR Program through the Project 'Artificial Intelligence for Improving the Exploitation of Water and Food Resources.'

**ABSTRACT** When investigating the causes of contamination in specific contexts, such as in underground water wells, multivariate regression is commonly used to establish possible links between the chemical-physical values of the samples and the levels of contaminant. Two issues often arise from such a statistical analysis: selecting the best predicting variables and detecting the instances that can be suspected to be outliers. In this paper, we propose a comprehensive, integrated, and general optimization model that solves these two problems simultaneously in such a way that outliers can be detected in reference to the specific variables that are selected for the regression, and we implement such an optimization model with a well-known evolutionary algorithm. We test our proposal on data extracted from a project whose aim is to establish the causes of the contamination of underwater water wells in a very specific area of northeastern Italy. The results show that our variable selection and outlier detection algorithm allows the synthesis of very reliable, interpretable, and clean regression models.

**INDEX TERMS** Outlier detection, feature selection, evolutionary computation, multi-objective optimization, underground water contamination.

## I. INTRODUCTION

Statistical and machine learning methods are widely employed in geology and, in general, the study of natural resources (see [1]–[3], among many others), and among the possible models, regression (specifically, linear regression) is one of the most common approaches (see, e.g., [4]–[6]). In particular, in studying problems that involve the contamination of natural resources, such as underground water, linear regression is a key tool that allows us to establish possible explainable relationships between physical-chemical values and contaminant(s) level(s). Linear, and in some cases polynomial, regression is often preferred over other models (nonlinear regression, neural networks, and so on)

The associate editor coordinating the review of this manuscript and approving it for publication was Zhan Bu<sup>1b</sup>.

because it allows an easier interpretation of the results, which in turn leads to more plausible geological models of the events. Regression can be *univariate*, when there is only one independent variable, or *multivariate*; additionally, it can be *linear*, when a linear function is searched for, or *nonlinear*. In certain types of applications, not only is the regressed function interesting but also the precise subset of independent variables that actually play a role is of interest. In the typical interpretation of the output of a linear regression algorithm, for example, those independent variables that are associated with very small constants are considered to be less important; however, one cannot simply identify the importance of a variable only in terms of its coefficient, and furthermore, when the regressed function is not explicit, such as, for example, in random forest, there are no coefficients at all. As a matter of fact, the most general technique for solving

such a problem is *feature selection* [7], while in the classical literature, feature selection is considered to be a technique to reduce data dimensionality and speed up further analysis steps; in real cases, it can also be used to determine exactly which independent variables have a role in a certain learning task.

Associated with feature selection, there are two other common techniques in data analysis, namely, *instance selection* [8] and *outlier detection* [9]. The former can be defined as the selection of a smaller subset of the original set of instances that behaves as or better than the original set in a learning task, while the latter is the task of identifying possible instances whose values could be the product of some instrumental or human mistake and that should not be considered. Instance selection differs from outlier detection in the sense that while the former aims to find a small group, possibly not unique, of representative instances, which can be used instead of the original problem, the latter aims to highlight the nonrepresentative instances caused by measurement error or inherent data variability, which might or might not exist. Outliers are commonly, and informally, defined as *observations that deviate so much from other observations as to increase suspicion that they were generated by other mechanisms* (see, e.g., [10]) and, as such, are not always easy to detect and interpret. For example, while detecting unusual symptoms or test results in a medical context, outliers are typical examples that could indicate potential health problems of a patient, and the entire data mining process is concentrated on their detection; however, if those data are in fact anomalous, they could contaminate the results and reduce the accuracy of the response [11]. Typically in real-life data, outliers are more frequently obtained with human intervention (such as physical-chemical analysis), while automatically generated data are less prone to containing mistakes. Moreover, the sample size and the statistical technique used also determine the importance of their detection: the smaller the sample size is, and the less robust the statistical techniques are, and the higher the influence of the outliers (for example, the sample mean gives a moderately robust estimation of the population central tendency, and thus, one outlier among a large sample will have a limited impact). In statistics, outliers are also classified as *vertical* (when they outlay in the values of independent variables) and *horizontal* (when they outlay in the value of the dependent variable). Feature selection methods that do not incorporate dependencies between attributes are called *univariate* methods, while those methods that evaluate the collective prediction power of subsets of attributes are called *multivariate*. Moreover, depending on the underlying evaluation strategy, feature selection methods are classified as *single* or *subset attribute evaluation*; single evaluation methods can be univariate or multivariate, whereas subset evaluation methods are always multivariate. Feature selection methods are also categorized into filter, wrapper and embedded models: *filters* are algorithms that perform the selection of features using a statistical measure that classifies their significance without making use of

any machine learning algorithm; *wrapper* methods evaluate attributes driven by the performances of an associated learning algorithm, and *embedded* models perform the two operations (selecting variables and building a classifier) at the same time - while these tend to produce very good subsets efficiently, they are usually complex and difficult to implement. Outlier detection methods can follow a similar taxonomy, although the classical literature separates *supervised* from *unsupervised* methods (the latter being far more common) and classifies unsupervised outlier detection methods into *density*- and *distance*-based methods. In algorithmic terms, existing methods are almost universal filters (although there are some embedded methods), because they are focused on finding outliers independently from the learning task.

In this work, we are interested in wrapper multivariate methods, which can be seen as *multi-objective optimization* models. Following the classical multi-objective optimization model for feature selection, our contribution can be summarized as follows:

- First, we define first a multi-objective optimization model for outlier detection.
- Second, we propose a multi-objective optimization model for simultaneous feature selection and outlier detection.
- Third, we implement an evolutionary algorithm to solve the proposed models.
- Fourth, we test our solution on both artificial and real data, comparing the results against several state-of-the-art techniques for outlier detection.

In particular, we test our algorithm in the case of linear regression on data that contain the result of the chemical-physical analysis of groundwater samples collected from a very specific area of northeastern Italy, where traces of pollution due to the abnormal presence of *mercury* (*Hg*) were found. The problem consists of finding which chemical-physical elements are linked to the pollutant and in which way. Our data set is part of a larger project whose aim is to find the origins and causes of pollution in the affected area; linking chemical-physical characteristics to the presence of pollutants can be useful in precisely identifying contaminated underground water bodies and can offer some insights into possible causes. This paper is organized as follows. First, in Section II, we give the necessary background on feature selection, outlier detection, and multi-objective optimization. In Section III, we define the optimization model for outlier detection and for simultaneous feature selection and outlier detection. In Section IV, we describe a solution to the optimization models via evolutionary algorithms, we propose an implementation, and we test our methodology on artificial data sets to evaluate its performances in a controlled environment. Then, in Section V, we test our implementation on real data, and we design a set of experiments devoted to establishing which strategy for feature selection and outlier detection works best. The remainder of the paper presents our conclusions.

## II. BACKGROUND

### A. FEATURE SELECTION

*Feature selection* (FS) is defined as the process of eliminating features from the database that are irrelevant to the task to be performed [7]. Feature selection facilitates data understanding, reduces the storage requirements, and lowers the processing time, which allows the model learning to become an easier process. Feature selection methods that do not incorporate dependencies between attributes are called *univariate* methods; in *multivariate* FS, on the other hand, the assessment is performed for subsets of features rather than single features. In many cases, the dataset will be better characterized by the sole features that have the highest discriminative power than by the whole set of features. However, when an individual feature in a high-dimensional feature space presents a small correlation with the target class, it could still be useful in combination with other features; in this regard, multivariate FS tends to present better results than univariate FS [12]. Feature selection algorithms are also categorized into filter, wrapper and embedded models. *Filters* are algorithms that perform the selection of features using an evaluation measure that classifies their ability to differentiate classes without making use of any machine learning algorithm. *Wrapper* methods select variables driven by the performances of an associated learning algorithm. Finally, *embedded* models perform the two operations (selecting variables and building a classifier) at the same time; in the particular case of regression, a relatively well-known embedded algorithm is the so-called *lasso regression* method [13]. Wrapper methods for feature selection are more common in the literature; often, they are implemented by defining the selection as a search problem and solved using metaheuristics such as evolutionary computation (see, e.g., [14]–[16]). Wrapper schemata are more common in supervised classification rather than regression; however, some effort in this direction has been done (see, e.g., [17]). Other strategies for feature selection include *adaptive boosting* [18]. Feature selection methods based on subset evaluation consist of four steps, usually called subset generation, subset evaluation, stopping criterion, and result validation. *Subset generation* is commonly implemented as a heuristic search algorithm in which candidate subsets are prepared for evaluation. Obviously, the search space for candidate subsets has cardinality  $2^N$ , where  $N$  is the number of features. Examples of subset generation mechanisms include [15], [19]–[27]. During the phase of *subset evaluation*, the goodness of a subset produced by a given subset generation procedure is measured, and examples include [28]–[33]. The *stopping criterion* is established when the feature selection process must finish; it can be defined as a control procedure that ensures that no further addition or deletion of features produces a better subset, or it can be as simple as a counter of iterations. Finally, during *result validation*, the validity of the selected subset is tested.

### B. OUTLIER DETECTION

*Outlier detection* (OD) is defined in [34] as the process of eliminating samples from the database that do not comply with the general behavior of the data model. Such samples, called *outliers*, differ significantly from the remaining set of data, and their existence can be either caused by measurement error or the result of inherent data variability. While OD methods could be classified following a taxonomy similar to the one used for FS methods, in the literature, OD is first separated into supervised and unsupervised methods. *Supervised* OD defines outlier detection as a classification process; it requires a previous labeling of those instances that are known to be outliers, and then, it focuses on training a classifier to recognize further outliers [35]–[39]. When referring to generic outlier detection, *unsupervised* OD, which is much more common than supervised OD, is further separated into *density*- (see, e.g., [40] and many other variants of the LOF algorithm) and *distance*-based methods (see, e.g., [41], [42]). In our terminology, all of these methods can be classified as filters (both unsupervised and supervised, both univariate and multivariate), because the process of detecting outliers is not linked to any successive learning task or algorithm. Univariate, unsupervised filters for OD are also the most commonly available methods; for example, the open-source suite Weka offers the filter *InterquartileRange*, which allows one to detect outliers and extreme values by looking at the values of any attribute under a normal distribution hypothesis. Embedded models for OD in regression have also been attempted [43]–[45]. Finally, *robust regression* (see, e.g., [46]) is a portfolio of regression techniques that are designed to respond better than classic regression techniques when the usual hypothesis on which they are based fails, for example, in the presence of outliers. Therefore, they should be listed as outlier detection techniques. Robust methods are based on either substituting the least square principle with other principles, such as *least absolute deviation*, *m-estimation*, or *least trimmed squares*, or replacing the normal distribution with a *t-distribution*, or on using *unit weights*. Robust regression is not as popular as classical regression; the algorithms are not very widely present in learning and statistical suits, and they have often been criticized for being computationally demanding. Robust regression has also been approached via evolutionary algorithms [47].

### C. COMBINED MODELS

A combination of FS and OD has been attempted in the literature using several different schemata. One of the early approaches is [48]. A more recent diagnostic method based on dummy variables and evaluation-by-plotting, which is specific for linear regression, was presented in [49], and a two-step method specific for linear regression was presented in [50]. On the other hand, OD is similar to the (more common) *instance selection* (IS), which consists of discovering a subset of instances such that a model built with only that

subset has similar, or better, performances than a model built from the entire set of instances [8], and IS has been combined with FS in several experiments whose structure is similar to the model used in this paper. Outlier detection is similar to IS in the sense that in both cases, instances are selected from the data set; however, while IS is focused on finding a set of *representative* instances, the aim of OD is discovering, should they exist, *unrepresentative* instances that should be eliminated. In the literature, IS is often combined with FS [8], [51]–[53]. Combined models for FS and IS range from sequential combinations (FS+IS versus IS+FS) to simultaneous models, although the latter are not usually defined as optimization models.

#### D. MULTI-OBJECTIVE OPTIMIZATION

A *multi-objective optimization problem* (see, e.g., [54]) can be formally defined as the optimization problem of simultaneously minimizing (or maximizing) a set of  $k$  arbitrary functions:

$$\begin{cases} \min / \max & f_1(\bar{x}) \\ \min / \max & f_2(\bar{x}) \\ \dots & \\ \min / \max & f_k(\bar{x}), \end{cases} \quad (1)$$

where  $\bar{x}$  is a vector of decision variables. A multi-objective optimization problem can be *continuous*, in which case we look for real values, or *combinatorial*, where we look for objects from a countably (in)finite set, typically integers, permutations, or graphs. Maximization and minimization problems can be reduced to each other, which means that it is sufficient to consider only one type. A set  $\mathcal{F}$  of solutions for a multi-objective problem is *nondominated* (or *Pareto optimal*) if and only if for each  $\bar{x} \in \mathcal{F}$ , there exists no  $\bar{y} \in \mathcal{F}$  such that (i) there exists  $i$  ( $1 \leq i \leq k$ ) such that  $f_i(\bar{y})$  improves  $f_i(\bar{x})$ , and (ii) for every  $j$ , ( $1 \leq j \leq k$ ,  $j \neq i$ ),  $f_j(\bar{x})$  does not improve  $f_j(\bar{y})$ . In other words, a solution  $\bar{x}$  *dominates* a solution  $\bar{y}$  if and only if  $\bar{x}$  is better than  $\bar{y}$  in at least one objective, and it is not worse than  $\bar{y}$  in the remaining objectives. We say that  $\bar{x}$  is *nondominated* if and only if there is no other solution that dominates it. The set of nondominated solutions from  $\mathcal{F}$  is called the *Pareto front*. Optimization problems can be approached in several ways; among them, *multi-objective evolutionary computation* is a popular choice (see, e.g., [55]–[57]). Within the evolutionary computation paradigm, there are many choices, that range from *ant colony optimization*, to *bee colony optimization*, to *differential evolution*, to *evolutionary algorithms*, among many others. Some very recent contributions, especially applied to the FS problem, include [58]–[60]. Among the several evolutionary algorithms offered in the recent literature, however, the *non-dominated sorted genetic algorithm* (NSGA-II) [61], the second version of the NSGA algorithm, is a very well-known one, easily available as open source code. NSGA-II is an elitist Pareto-based multi-objective evolutionary algorithm that uses a strategy with binary tournament

selection and a rank-crowding better function, where the rank of an individual is defined as its non-domination level in the whole population. While experience indicates that there is no unique evolutionary algorithm that outperforms all others in solving every optimization problem, NSGA-II has been tested in a very wide range of problems showing remarkable performances, and it is available in open source libraries.

#### E. MULTI-OBJECTIVE EVOLUTIONARY FEATURE SELECTION AND OUTLIER DETECTION

Genetic algorithms have always been considered a powerful tool for feature selection [15] and have been proposed by numerous authors as a search strategy in filter, wrapper, and embedded models [62]–[64] as well as feature weighting algorithms and subset selection algorithms [16]. The first evolutionary approach that involves multi-objective optimization for feature selection was proposed in [65] with three criteria: accuracy, number of features, and number of instances. In this approach, the three criteria are aggregated into a single criterion, and then, a single-objective algorithm is used. A formulation of feature selection as a multi-objective optimization problem was presented in [56]. The wrapper approach proposed in [66] accounts for the misclassification rate of the classifier, the difference in the error rate among classes, and the size of the subset using a multi-objective evolutionary algorithm where a niche-based fitness punishing technique is proposed to preserve the diversity of the population. A wrapper approach was proposed in [67], which minimizes both the error rate and the size of a decision tree. Another wrapper method was proposed in [68] to maximize the cross-validation accuracy on the training set, maximize the classification accuracy on the testing set, and minimize the cardinality of the feature subsets using support vector machines applied to protein fold recognition. In [69], two wrapper methods with three and two objectives, respectively, applied to cancer diagnosis were compared. The three-objective version optimizes the sensitivity, specificity and number of genes, while the two-objective version optimizes the accuracy and number of genes. NSGA-II is used as the search strategy, and a support vector machine is used for the classification task. A filter local search embedded multi-objective memetic algorithm was presented in [70], which is a synergy of NSGA-II and a filter method for the identification of relevant features in a multiclass problem. The filter approach proposed in [71] includes measures of consistency, dependency, distance and information, and it is based, again, on NSGA-II. An NSGA-II wrapper approach was also proposed in [72] for named entity recognition. Finally, a modification of the dominance relation is introduced in [73] to treat an arbitrarily large number of objectives and is used in a combination of NSGA-II, logistic regression, and naïve Bayes with Laplace correction as a classification algorithm. Very recent examples of multi-objective feature selection systems can be found in [55], [74], [75]. Finally, there have been a few (simple) attempts to use evolutionary computation to detect outliers; they include,

for example, [76]. A generic formalization of a wrapper methodology for feature selection applied to a data set with  $n + 1$  features and posed as a two-objective optimization problem can be devised from the literature. It simply involves adapting (1) to simultaneously optimize two functions:

$$\begin{cases} \min / \max f(\bar{x}) \\ \min \mathcal{C}(\bar{x}), \end{cases} \quad (2)$$

where  $\bar{x}$ , which takes the values in  $\{0, 1\}$ , represents the set of chosen features (1 means that the feature is selected, and 0 means that it is discarded), and  $\mathcal{C}(\bar{x})$  represents its cardinality:

$$\mathcal{C}(\bar{x}) = \sum_{t=1}^n \bar{x}(t) \quad (3)$$

and  $f(\bar{x})$  are any measures of the goodness of the learning algorithm used when applied to the data set obtained by selecting only those features indicated by  $\bar{x}$ . If we choose to minimize such a function, then possible choices include defining it as the opposite of the accuracy, the correlation coefficient, or the mean squared error, among many others, depending on the specific classification/regression problem to be solved. Observe that applying  $\bar{x}$  entails transforming the original data set by selecting specific attributes while discarding the others; such a concept, as we shall see, can be easily generalized.

### F. EVOLUTIONARY OUTLIER DETECTION

Outlier detection has been approached with evolutionary algorithms in the literature. In [77], for example, OD is solved by harmony search and differential evolution, where a single objective is defined using the sparsity coefficient. Additionally, in [78], the anomaly detection problem is dealt with as a classification problem. Voronoi diagrams are used for the task of classification, and they are generated by many-objective evolutionary computation where objectives are progressively added. They correspond to the classical performance metrics for classification and for Voronoi diagrams (accuracy, recall, number of cells, total empty volume, and so on). Finally, in [79], a multiobjective evolutionary algorithm is proposed to determine the best penalty factors of a sparse group lasso constraint imposed on the learning objective of Adabost, which is combined with an autoencoder for image outlier detection.

### G. AN OVERVIEW OF OUR CONTRIBUTION

In this paper, we first define the OD problem as a multi-objective optimization problem. This definition is rendered with a wrapper methodology that at the expense of being linked to a specific learning model (in our case, regression) can detect outliers in a very effective and easy-to-implement way; as a matter of fact, we use well-known mechanisms whose implementations are widely available. Moreover, because our methodology is multivariate, we evaluate outliers in groups instead of one-by-one. Furthermore,

a wrapper OD method has the advantage of being able to spot outliers in reference to a particular learning model, and those outliers could be overlooked by a blind method. Finally, being multi-objective, our methodology has the advantage of minimizing the number of outliers, which is consistent with considering the original data to be reliable up to a certain point, as opposed to simply eliminating all instances that do not allow us to fit a model. Second, we define the simultaneous FS and OD problem as a multi-objective optimization problem, building on the previous definition and adding a third objective. This proposal has the same advantages as the previous proposal, except that it allows the detection of outliers in specific reference to the selected attribute. Third, we propose and run a testing methodology to evaluate the effectiveness of not only the proposed method against popular outlier detection models but also the several possible variations of our schema that emerge from defining our selection models against each other.

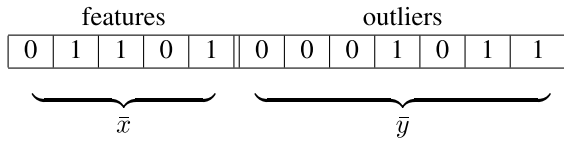
## III. SIMULTANEOUS FEATURE SELECTION AND OUTLIER DETECTION FOR REGRESSION

### A. OUTLIER DETECTION AS A MULTI-OBJECTIVE OPTIMIZATION PROBLEM

Inspired by the optimization model for multivariate wrapper feature selection, which, as we recalled in the previous section, can be defined as the problem of minimizing the cardinality of the selected features while optimizing, at the same time, the performances of the associated learning algorithm, we see that outlier detection can be solved in a very similar way. For a data set with  $m$  instances, let  $\bar{y} = (y_1, \dots, y_m)$  be a vector of decision variables in  $\{0, 1\}$ . We interpret  $\bar{y}$  as the set of selected outliers, in other words, the subset of instances that are not used by the learning algorithm on which the wrapper is built (thus, unlike FS, for  $1 \leq t \leq m$ , we have that  $\bar{y}(t) = 1$  represents *not* selecting the instance  $t$ , while  $\bar{y}(t) = 0$  represents selecting it). Then, we can define the outlier detection problem by adapting (1) as follows:

$$\begin{cases} \min / \max f(\bar{y}) \\ \min \mathcal{C}(\bar{y}), \end{cases} \quad (4)$$

where the choices for  $f$  range, as before, among the possible measures of performance of the learning algorithm. For example, if we had a data set with  $m = 7$ , applying  $\bar{y} = (0, 0, 0, 1, 0, 1, 1)$  encompasses a transformation that consists of eliminating instances 4, 6, and 7. This approach corresponds to evaluating the hypothesis that such instances are, in fact, outliers and that being the case, the regression algorithm will behave better. Observe that minimizing the number of selected outliers is consistent with assuming that if there are outliers in the data set at all, they cannot be too many. While (2) and (4) appear identical, the (fundamental) difference lies in how the decision variables are interpreted, in other words, how the data set is transformed at each iteration of the wrapper. Observe that, for example, in the case of linear regression, the objectives are opposite to each other.



**FIGURE 1.** Example of candidate solution in the simultaneous model. In this example, we select 3 out of 5 features (the second, third, and fifth features) and eliminate 3 out of 7 instances as outliers (the fourth, sixth, and seventh instances).

In particular, observe that there are trivial solutions to a perfect regression, for example, with two points. This circumstance means that if we were not to minimize the number of outliers while maximizing the correlation index, the system would return irrelevant solutions with subsets of points that lie on the same straight line. Moreover, this system is designed for data sets that are supposed to be *relatively clean*, because they are the results of systematic lab analyses. They could contain outliers, but not *too many*; minimizing the number of outliers and setting a predetermined maximum number, therefore, makes sense. Finally, this approach is insensitive to the type of outliers, and it can detect both vertical and horizontal outliers alike.

**B. SIMULTANEOUS FEATURE SELECTION AND OUTLIER DETECTION**

In some problems, the best way to search for a function that fits a data set is performing feature selection and outlier detection simultaneously; as a matter of fact, an instance could be an outlier only under the point of view of certain features, and thus, it should be considered as such only if those features are selected. We can express the process of simultaneous feature selection and outlier detection as a multi-objective optimization problem, as follows. For a data set with  $n + 1$  features and  $m$  instances, let  $\bar{x}, \bar{y}$  be two vectors of decision variables, where  $\bar{x}$  (with range  $n$ ) is defined as in (2), and  $\bar{y}$  (with range  $m$ ) is defined as in (4). The pair  $\bar{x}, \bar{y}$  entails a transformation of the data set that corresponds to selecting those features set to 1 in  $\bar{x}$  and discarding those instances set to 1 in  $\bar{y}$ . Then, a three-objective optimization problem can be defined by adapting (1) as follows:

$$\begin{cases} \min / \max f(\bar{x}, \bar{y}) \\ \min \mathcal{C}(\bar{x}) \\ \min \mathcal{C}(\bar{y}), \end{cases} \quad (5)$$

where  $f(\bar{x}, \bar{y})$  measures, as expected, the performances of the learning algorithm on (i) the instances survived after the transformation entailed by  $\bar{y}$ , and (ii) the features chosen by  $\bar{x}$  only. Thus, continuing with the previous example and further assuming that the original dataset has 5 features (excluding the class), having  $\bar{x} = (0, 1, 1, 0, 1), \bar{y} = (0, 0, 0, 1, 0, 1, 1)$  implies selecting features 2, 3, and 5 while disregarding the instances 4, 6, and 7 (see Fig. 1).

**IV. MULTI-OBJECTIVE EVOLUTIONARY OPTIMIZATION A. IMPLEMENTATION**

As seen in the previous sections, we have defined FS, OD, and simultaneous feature selection and outlier detection as

optimization problems. We choose to approach such optimization problems via an evolutionary algorithm and, in particular, using the well-known algorithm NSGA-II [61], which is available in open source from the suite *jMetal* [80]. As a black box regression algorithm, we used the class *LinearRegression* from the open source learning suite *Weka* [81], with the following default parameters: no embedded feature selection method, elimination of collinear features, and ridge penalty  $10^{-8}$ , run in 10-fold *cross-validation* mode; therefore, in our experiments, we shall be looking for linear regression models. We use a fixed-length representation, where each individual solution consists of a bit set, which is different for each strategy. In simple FS, each individual is of the following type:

$$\bar{x} = (x_1, x_2, \dots, x_n), \quad (6)$$

where, for each  $1 \leq t \leq n$ ,  $x_t = 1$  (resp.,  $x_t = 0$ ) is interpreted as the  $t$ -th attribute being selected (resp., discarded), while in simple OD it is of the type:

$$\bar{y} = (y_1, y_2, \dots, y_m), \quad (7)$$

where, for each  $1 \leq t \leq m$ ,  $y_t = 1$  (resp.,  $y_t = 0$ ) is interpreted as the  $t$ -th instance being discarded (resp., selected). Clearly, the same representation is used in the sequential strategies. In the simultaneous strategy, the representation is the natural generalization of the strategy used in FS and OD:

$$\bar{z} = (z_1, z_2, \dots, z_n, z_{n+1}, z_{n+1}, \dots, z_m). \quad (8)$$

While the outlier detection model is very general, as a rule of thumb, we consider any solution that has too many outliers unacceptable. This choice has effects on both the initial population and the generations; indeed, the initial population is generated randomly, but while individuals (or parts of individuals) used for feature selection are generated in the range of all possible solutions, the individuals (or parts of individuals) used for outlier detection are generated in such a way that the cardinality of each individual is less than or equal to a parameter that we control. When considering that the NSGA-II does not allow the management of external constraints, unacceptable solutions that emerge during the generations are artificially given the worst possible values in each objective and are therefore discarded. Let  $\bar{t}$  be any of  $\bar{x}, \bar{y}$ , and  $\bar{z}$ . In terms of objectives, minimizing the cardinality of the individuals is straightforward, and we do so by using the function  $\mathcal{C}(\bar{t})$  defined in the previous section. To optimize the performance of the learning algorithm used in the wrapper, we define

$$f(\bar{z}) = 1 - |\rho(\bar{t})|, \quad (9)$$

where  $\rho()$  measures the correlation between the stochastic variable obtained by the observations and the linear variable obtained by *LinearRegression* on the data set after the transformation entailed by  $\bar{z}$ , as explained in the previous section. The correlation varies between  $-1$  (perfect negative correlation) and  $1$  (perfect positive correlation), with  $0$  being

**TABLE 1.** Evaluations on the *wood* public artificial data set in 10-fold cross-validation mode, varying the seed.

<i>ex</i>	<i>cc</i>	<i>outliers</i>
1	0.969	4,6,8,19
2	0.969	4,6,8,19
3	0.969	4,6,8,19
4	0.945	1,11,12,14
5	0.950	7,11,14,16,20
6	0.969	4,6,8,19
7	0.943	1,3,11
8	0.947	1,9,11,14
9	0.969	4,6,8,19
10	0.957	1,11,14,15

the value that represents no correlation at all. Defined in this way,  $f$  ought to be minimized. Summarizing, the fitness of individual is computed using the correlation of the linear regression with the selected data (i.e., selected features and/or selected instances after outlier elimination), in 10-fold cross-validation mode, and the number of selected features/outliers. In terms of parameterization, we proceed as follows. In every experiment, the population size is 100. After an initial preliminary test, we found that 10000 total evaluations (100 generations) were sufficient to show convergent behavior. In each experiment, the sequential strategy consists of a first phase of feature selection (or outlier detection) followed by a phase of outlier detection (or feature selection), each of which should be treated as an experiment on its own. Therefore, each of these phases has been granted 10000 evaluations, which means that the last population in a sequential experiment is the result, in fact, of 20000 evaluations. For this reason, in simple strategy experiments, as well as in simultaneous strategy experiments, the number of evaluations has been set to 20000. The maximum percentage of detected outliers is set to 10%. We used the standard crossover and (uniform) mutation, with probabilities (set after an initial explorative analysis) of 0.7 and 0.3, respectively. Finally, in terms of an *a posteriori* decision, we choose to select, in each experiment, the individual with the best value of  $f(\bar{z})$ . The seeds for folder extraction are always fixed from 1 to 10, to ensure the repeatability of the experiments.

## B. TESTING WITH SYNTHETIC DATA

Artificial data sets are generally not a reliable playground for algorithm evaluation. Outlier detection, however, is an exception to this rule: as a matter of fact, knowing the nature and identity of true outliers helps us to establish whether an outlier detection algorithm is at least partially correct. Here, we consider two artificial datasets with two different purposes; for each dataset, we run an experiment as explained above, and we show the results only in terms of the correlation coefficient, outliers, and selected features. In the first experiment, inspired by [82], we use the artificial *wood* dataset distributed within the *R package* [83], which encompasses 20 instances with five features and one dependent variable. Such a data set is declared to have 4 outliers (instances 4, 6, 8, and 19); in other words, 25% of the data are outliers.

**TABLE 2.** Evaluations on our artificial data set in 10-fold cross-validation mode, varying the seed.

<i>ex</i>	<i>cc</i>	<i>attributes</i>	<i>outliers</i>
1	0.973	$X_1, X_2, X_3$	14,27,92,101,102
2	0.995	$X_1, X_2, X_3$	51,52,101
3	0.995	$X_1, X_2, X_3$	11,35,61,65,101
4	0.998	$X_1, X_2, X_3$	101,102,103
5	0.999	$X_1, X_2, X_3$	3,35,36,101,102,105
6	1.000	$X_1, X_2, X_3$	101,102,103,104,105
7	0.993	$X_1, X_2, X_3$	56,97,101,102
8	0.960	$X_1, X_2, X_3$	7,101,102
9	1.000	$X_1, X_2, X_3$	101,102,103,104,105
10	0.976	$X_1, X_2, X_3$	10,51,98,101,102,105

We executed 10 independent runs of our algorithm in *outlier detection mode* (later called OD mode) only, to evaluate whether outliers were detected and how often. The results, shown in Tab. 1, indicate that our method finds all outliers in 5 out of 10 executions and, as expected, the correlation coefficient is the highest when the outliers are found. In the second experiment, we created an artificial data set to better control its content. Our data set contains 105 instances described by 4 attributes  $X_1, X_2, X_3, X_4$  and one dependent variable, which is computed by the function:

$$Y = 2.75 \cdot X_1 + 3.02 \cdot X_2 - 4.123 \cdot X_3. \quad (10)$$

The first 100 instances are computed exactly; the remaining 5 have random values, and they are outliers. Attribute  $X_4$  does not play any role. We executed 10 independent runs of our algorithm in *outlier detection and feature selection mode* (later called FSOD mode) only, to evaluate whether outliers were detected and how often and which features are selected. The results, shown in Tab. 2, once again indicate that our method works as expected; some outliers are found in every execution, all outliers are found in 2 out of 10 executions, and the correct features are selected in all executions. As in the previous case, by applying our *a posteriori* decision policy (in other words, selecting the execution with the highest correlation coefficient), the resulting data set ends up without outliers.

## V. A COMPREHENSIVE EXPERIMENT ON REAL DATA

### A. STRATEGIES

Feature selection and outlier detection, as we have explained, can be performed individually, sequentially, or simultaneously, and it is interesting to establish which strategy performs better. To accomplish this goal, we designed a set of experiments to be performed on the same data set, which was organized as follows:

- *Simple* strategy. This strategy consists of performing feature selection or outlier detection alone. By abusing the notation, we refer to the former as the strategy FS and the latter as the strategy OD.
- *Sequential* strategy. This strategy consists of performing feature selection first to obtain a reduced data set, which in turn is used as the starting data set for the outlier detection phase. We use the symbol FS+OD to denote

such a strategy. The symmetric solution, which consists of outlier detection first followed by feature selection, is denoted by OD+FS.

- *Simultaneous* strategy. This strategy consists of simply applying the simultaneous optimization model, and it is denoted by FSOD.

## B. TEST SETTINGS

Multiple executions (10 in our case) are required to ensure that good results do not occur by chance. As we have already mentioned, for each strategy and each execution, every evaluation is performed in 10-fold cross-validation mode with a fixed seed, and the random choices that occur within the computations (initial population, mutations, and so on) are governed by a seed that changes with the execution, from 1 to 10. In this way, the final results are always comparable to one another. For the original data set, the Weka built-in linear regression algorithm *LinearRegression* with default parameters (as specified in the previous section), in 10-fold cross-validation mode with seeds from 1 to 10, obtained ten correlation coefficients, which we call  $OR_i$  ( $1 \leq i \leq 10$ ). Moreover, we applied the Weka deterministic filter *InterquartileRange* to the original data set and tested the resulting data set with the linear regression algorithm in 10-fold cross-validation mode, obtaining ten correlation coefficients, which we call  $IR_i$  ( $1 \leq i \leq 10$ ); we used the following default parametrization: all columns were analyzed except the target, factor for outlier detection set to 3, and factor for extreme value detection set to 6. Similarly, we applied the standard *robust regression algorithm* ( $RR_i$ ,  $1 \leq i \leq 10$ ) and the standard *lasso regression algorithm* ( $LR_i$ ,  $1 \leq i \leq 10$ ), which were both offered in the R package [83]. We used the following, default, parametrization: for robust regression, we used Huber weights and default  $\Psi$  parameters, and for Lasso regression we used  $\alpha = 1$ , *standardize* set to True, *normalize* set to False, maximum 1000 iterations, negative coefficients allowed. Then, in the simple and simultaneous strategies experiments, we performed 10 executions per strategy (again, with seeds from 1 to 10); out of each final population, we extracted the best individual in terms of the second objective (for us, the correlation coefficient) and named  $FS_i$  (resp.,  $OD_i$ ,  $FSOD_i$ ): to each one of these individuals, we applied the linear regression algorithm, evaluated in cross-validation mode with the corresponding seed. Finally, for the sequential FS+OD strategy, we proceeded as follows: in each execution, after the initial 100 generations of the FS phase, we obtained a final population from which the best (correlation-wise) individual has been extracted; to the latter, we applied 100 further generations to obtain a new final population from which, again, we extracted the best individual, denoted, here, by  $FS+OD_i$ , and whose correlation coefficient has been computed by running the linear regression algorithm and evaluating it in cross-validation mode with its corresponding seed. The correlation coefficients for the individuals  $OD+FS_i$  have been obtained in a symmetric way.

**TABLE 3. Evaluation of linear regression on the original data set in 10-fold cross-validation mode, varying the seed (OR).**

<i>ex</i>	<i>cc</i>	<i>rmse</i>	<i>rae(%)</i>	<i>rrse(%)</i>
1	0.586	0.002	66.418	81.097
2	0.583	0.002	65.797	81.132
3	0.567	0.002	67.456	82.563
4	0.593	0.002	65.494	80.337
5	0.570	0.002	67.211	82.342
6	0.551	0.002	67.047	83.565
7	<b>0.597</b>	<b>0.002</b>	<b>65.331</b>	<b>80.067</b>
8	0.535	0.002	68.295	85.031
9	0.586	0.002	66.034	81.107
10	0.573	0.002	66.527	81.951
<i>avg</i>	0.574	0.002	66.561	81.919

## C. RESULTS EVALUATION

To evaluate the results, we designed the following tests:

- *Interstrategy* test, designed to evaluate which strategy performs better. To accomplish this goal, we considered the set of 10 results of each strategy (in terms of the correlation coefficient) as an independent sample, and we applied an independent group *t*-test to verify the statistical reliability of the results;
- *Post-strategy* test, designed to evaluate the results from a mathematical point of view. To accomplish this goal, we compare the quality of the regression model obtained with the original data with that of the model obtained with the best individual from the simultaneous strategy via a *residual versus fitted* test and a *residual versus leverage* test.

## D. DATA AND RESULTS

To test our system, we considered a data set of physical-chemical samples of underground water in a very specific area in northeastern Italy. Such samples were collected as part of an ongoing investigation commissioned by the local Regional Agency for Environment and Prevention to the University of Ferrara, with the purpose of exploring the causes of a sudden, unexpected spike of *Hg* (*Mercury*) in the underground water. Such data are being used to perform several physical-chemical studies. Among these, we are interested in evaluating whether the amount of *Hg* can be linked to the other parameters in a regression model, which would help in understanding the possible causes. In the relevant area, 92 sampling points (underground water wells) were chosen for this analysis. Samples were collected from 2012 to 2018 in a periodic way at most of these points. Each sample was analyzed from a physical-chemical point of view, and several indicators (in addition to *Hg*) were registered. After normalizing the values, eliminating zero- or low-variance columns, substituting null values with the mean of the corresponding column, the following parameters survived: *Br* (*Bromine*), *Ca* (*Calcium*), *Cl* (*Chlorine*), *Fe* (*Iron*), *HCO3* (*Bicarbonate*), *I* (*Iodine*), *K* (*Potassium*), *Mg* (*Magnesium*), *NH4+* (*Ammonium cation*), *NO3* (*Nitrate*), *Na* (*Sodium*), *SO4* (*Sulfate*), *Hg* (*Mercury*), *T* (*Temperature*), *Eh* (*Reduction potential*), *DO* (*Chemical oxygen demand*), and *C.E.* (*Electric conductivity*). The entire data set of polluted samples has



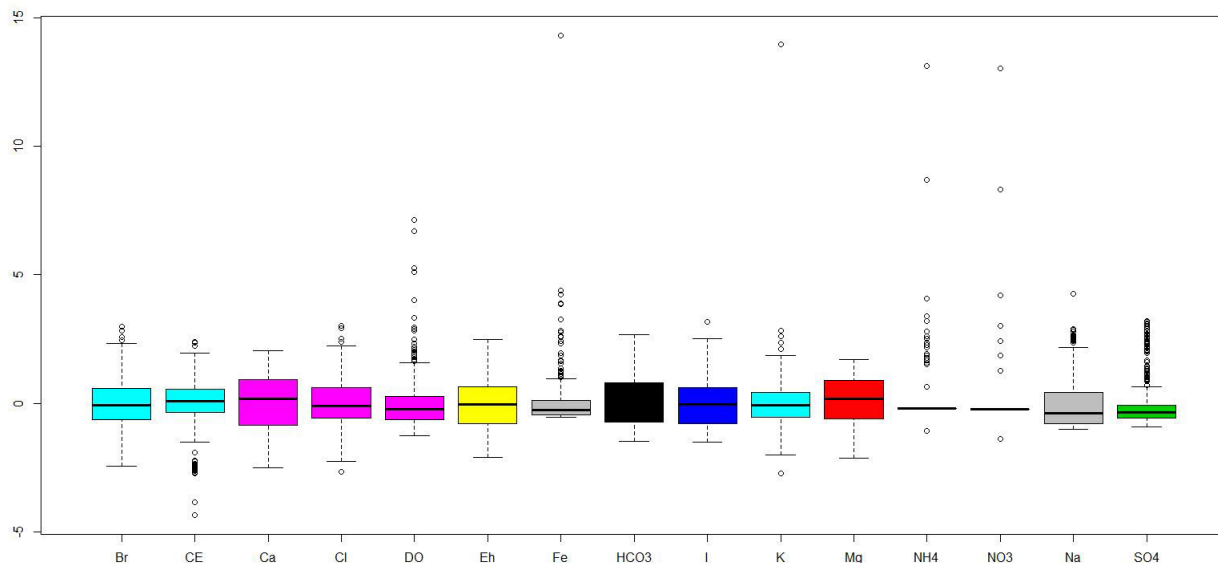


FIGURE 2. Single-attribute outliers.

TABLE 4. Evaluation of linear regression on the data set obtained after eliminating outliers with *InterquartileRange* by Weka, default parameters, in 10-fold cross-validation mode, varying the seed (IR).

ex	cc	rmse	rae(%)	rrse(%)
1	0.567	0.002	68.714	82.513
2	0.563	0.002	68.473	82.665
3	0.550	0.002	67.168	83.375
4	0.564	0.002	68.945	82.478
5	<b>0.591</b>	<b>0.002</b>	<b>68.307</b>	<b>80.567</b>
6	0.564	0.002	68.231	82.679
7	0.508	0.003	69.884	86.141
8	0.558	0.002	67.609	83.011
9	0.574	0.002	67.748	81.986
10	0.574	0.002	68.074	81.875
avg	0.561	0.002	68.315	82.729

TABLE 5. Evaluation of linear robust regression on the original data set with default parameters in 10-fold cross-validation mode, varying the seed (RR).

ex	cc	rmse	rae(%)	rrse(%)
1	0.634	0.002	56.490	78.890
2	0.634	0.002	56.600	78.910
3	<b>0.635</b>	<b>0.002</b>	<b>56.250</b>	<b>78.820</b>
4	0.633	0.002	56.960	78.960
5	0.633	0.002	56.670	78.910
6	0.634	0.002	56.670	78.880
7	0.634	0.002	56.600	78.860
8	0.635	0.002	56.340	78.830
9	0.634	0.002	56.730	78.910
10	0.634	0.002	56.600	78.900
avg	0.634	0.002	56.590	78.890

427 instances. The box plot shown in Fig. 2 refers to the single-attribute outliers that the original problem presents; as seen, it would be quite difficult to pick individual instances as outliers based on this information. First, let us consider Tab. 3, which contains the ten executions in cross-validation mode of the linear regression algorithm, with seeds from

TABLE 6. Evaluation of linear lasso regression on the original data set with default parameters in 10-fold cross-validation mode, varying the seed (LR).

ex	cc	rmse	rae(%)	rrse(%)
1	<b>0.835</b>	<b>0.001</b>	<b>79.070</b>	<b>74.960</b>
2	0.788	0.001	88.170	81.160
3	0.589	0.002	91.690	87.620
4	0.761	0.001	77.060	77.420
5	0.831	0.001	79.000	75.450
6	0.651	0.002	78.960	90.130
7	0.532	0.006	79.610	96.950
8	0.761	0.001	70.940	73.870
9	0.524	0.006	77.190	96.820
10	0.797	0.001	74.530	74.690
avg	0.707	0.002	79.590	82.910

TABLE 7. Evaluation of linear regression on each of the 10 data sets obtained by selecting the best individual from 10 executions; the seed was varied, and the feature selection model (FS) was used.

ex	cc	rmse	rae(%)	rrse(%)
1	0.614	0.002	65.813	78.836
2	0.619	0.002	65.182	78.211
3	0.607	0.002	66.108	79.291
4	0.621	0.002	64.904	78.154
5	0.612	0.002	65.723	78.966
6	0.602	0.002	65.423	79.472
7	<b>0.626</b>	<b>0.002</b>	<b>64.472</b>	<b>77.700</b>
8	0.588	0.002	66.445	80.774
9	0.611	0.002	66.157	79.011
10	0.614	0.002	65.419	78.602
avg	0.611	0.002	65.564	78.901

1 to 10. Use the following indicators: *cc* (the correlation coefficient), *rmse* (the root mean squared error), *rae* (the relative absolute error), and *rrse* (the root relative squared error). It can be immediately observed that the correlation coefficients that emerge from this initial experiment are not very high; this finding could be due to several factors,

**TABLE 8.** Evaluation of linear regression on each of the 10 data sets obtained by selecting the best individual from 10 executions; the seed was varied, and the outlier detection model (OD) was used.

<i>ex</i>	<i>cc</i>	<i>rmse</i>	<i>rae(%)</i>	<i>rrse(%)</i>	<i>out(%)</i>
1	0.735	0.001	60.214	67.556	9.824
2	0.735	0.001	59.077	67.773	9.824
3	0.735	0.001	58.307	67.943	8.421
4	0.682	0.001	64.557	73.253	9.122
5	0.740	0.001	61.115	67.026	9.122
6	0.686	0.001	63.517	72.952	8.070
7	0.722	0.001	60.790	69.268	8.421
8	0.727	0.001	61.738	68.539	9.473
9	0.676	0.001	63.862	73.772	7.368
10	<b>0.757</b>	<b>0.001</b>	<b>59.049</b>	<b>65.010</b>	<b>3.859</b>
<i>avg</i>	0.720	0.001	61.222	69.309	8.350

**TABLE 9.** Evaluation of linear regression on each of the 10 data sets obtained by selecting the best individual from 10 executions; the seed was varied, and the simultaneous feature selection and outlier detection model (FSOD) was used.

<i>ex</i>	<i>cc</i>	<i>rmse</i>	<i>rae(%)</i>	<i>rrse(%)</i>	<i>out(%)</i>	#
1	0.728	0.002	62.085	68.340	7.962	14
2	0.739	0.001	61.271	67.141	5.620	15
3	0.730	0.001	59.182	68.149	9.133	14
4	0.726	0.001	58.168	68.709	8.430	13
5	0.714	0.002	62.386	69.670	9.836	11
6	0.721	0.001	58.938	69.126	8.430	13
7	0.738	0.001	58.813	67.459	8.899	15
8	0.734	0.001	58.795	67.908	6.557	14
9	0.709	0.001	60.878	70.569	6.791	15
10	<b>0.758</b>	<b>0.001</b>	<b>59.194</b>	<b>64.964</b>	<b>9.836</b>	<b>15</b>
<i>avg</i>	0.730	0.001	59.971	68.204	8.149	13.9

including the presence of noise attributes and/or outliers in the instances. Eliminating the outliers that are found by running the Weka built-in tool (as explained in the previous section) does not appear to improve the results, as can be observed in Tab. 4; this observation is not surprising: this method is quite primitive and does not consider the features that are actually used in the subsequent regression step (in fact, it is unrelated to the learning algorithm). The presence of features that are, in fact, noise with respect to the problem is witnessed by Tab. 7, which contains the results of the experiments with the pure FS strategy. When we run the OD experiments, it can be seen that the correlation coefficients increase significantly. The last column of Tab. 8 indicates the rate of outliers that have been selected and eliminated in each best individual of each experiment; as can be seen, especially in some cases, the number is very low, which indicates that a correct choice of (suspected) outliers can truly improve the learning phase, even in the presence of noise features (recall that, in this experiment, there is no feature selection). When we run the simultaneous FSOD experiment, we obtain the results in Tab. 9, which are, as expected, the best results, proving that outliers should be detected relative to the features that are being used in the learning phase. While a simultaneous, integrated, and automated system is certainly preferable over a semiautomated system, such as a sequential strategy, for practical reasons, one can still wonder if selecting features and then detecting outliers, or the other way around,

**TABLE 10.** Evaluation of linear regression on each of the 10 data sets obtained by selecting the best individual from 10 executions; the seed was varied, and the sequential feature selection and outlier detection model (FS+OD) was used.

<i>ex</i>	<i>cc</i>	<i>rmse</i>	<i>rae(%)</i>	<i>rrse(%)</i>	<i>out(%)</i>	#
1	<b>0.746</b>	<b>0.001</b>	<b>61.871</b>	<b>66.373</b>	<b>6.557</b>	7
2	0.708	0.002	63.706	70.473	8.665	6
3	0.698	0.002	64.799	71.455	6.557	6
4	0.699	0.002	63.607	71.345	7.962	7
5	0.677	0.002	63.252	73.393	9.367	6
6	0.690	0.002	63.326	72.113	7.728	5
7	0.724	0.001	62.954	68.685	8.196	7
8	0.729	0.002	61.901	68.210	8.665	6
9	0.738	0.001	60.790	67.352	9.836	8
10	0.680	0.002	63.490	73.410	7.494	9
<i>avg</i>	0.709	0.002	62.969	70.280	8.102	6.7

**TABLE 11.** Evaluation of linear regression on each of the 10 data sets obtained by selecting the best individual from 10 executions; the seed was varied, and the sequential outlier detection and feature selection model (OD+FS) was used.

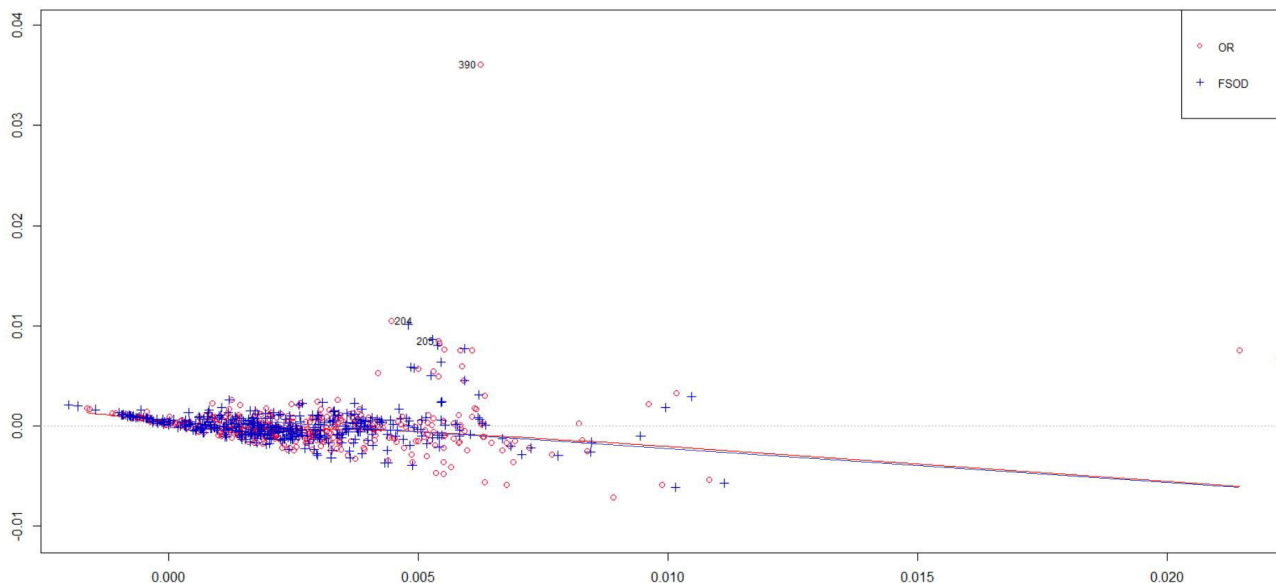
<i>ex</i>	<i>cc</i>	<i>rmse</i>	<i>rae(%)</i>	<i>rrse(%)</i>	<i>out(%)</i>	#
1	<b>0.768</b>	<b>0.001</b>	<b>59.384</b>	<b>63.841</b>	<b>7.728</b>	<b>13</b>
2	0.726	0.001	60.968	68.284	8.665	10
3	0.737	0.001	58.081	67.416	8.665	9
4	0.706	0.002	61.904	70.691	7.962	9
5	0.733	0.001	58.249	67.960	4.918	13
6	0.740	0.001	57.403	67.136	9.601	12
7	0.713	0.002	63.501	70.029	9.836	11
8	0.709	0.002	61.522	70.356	7.728	12
9	0.694	0.002	61.667	71.928	6.791	9
10	0.724	0.002	61.119	68.801	7.728	11
<i>avg</i>	0.724	0.001	60.379	68.644	7.962	10.9

**TABLE 12.** A summary of the results.

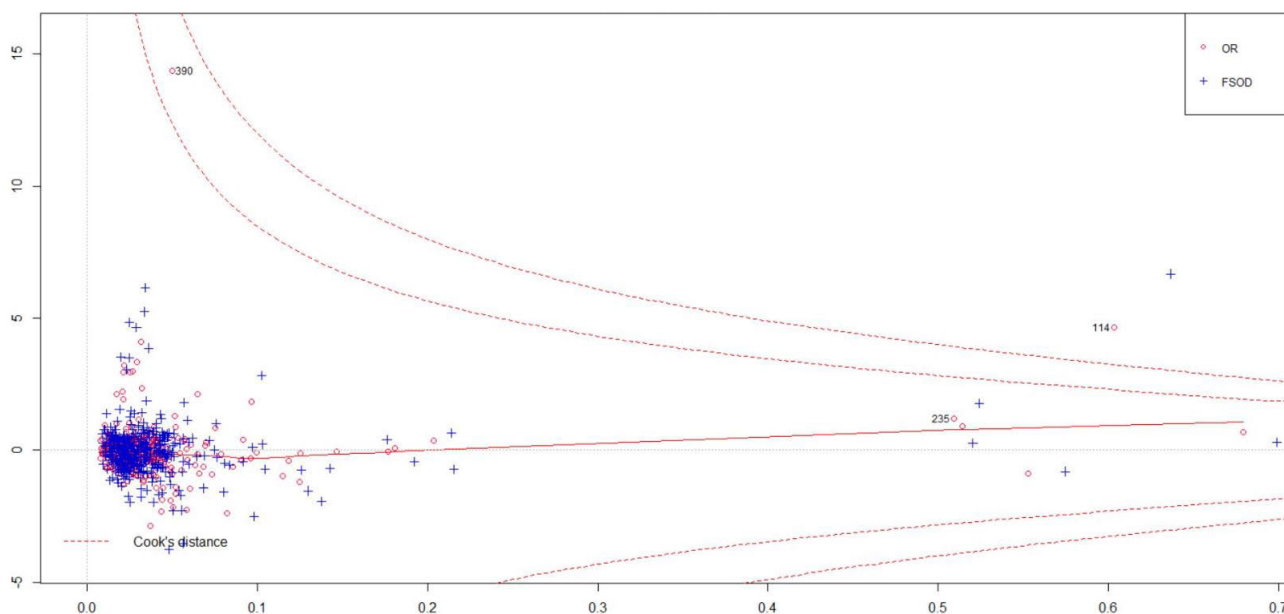
<i>method</i>	<i>cc</i>	<i>rmse</i>	<i>rae(%)</i>	<i>rrse(%)</i>
OR	0.574	0.002	66.561	81.919
IR	0.561	0.002	68.315	82.729
FS	0.611	0.002	65.564	78.901
OD	0.720	0.001	61.222	69.309
FSOD	<b>0.730</b>	<b>0.001</b>	<b>59.971</b>	<b>68.204</b>
FS+OD	0.709	0.002	62.969	70.280
OD+FS	0.725	0.001	60.379	68.644

gives results that are comparable to those obtained with the simultaneous strategy. The results of two experiments that correspond to the two possible sequential strategies, shown in Tab. 10 and Tab. 11, respectively, prove that while both of them are comparable to the simultaneous strategy in terms of improvement over the original problem, the simultaneous strategy still presents a slight advantage over both sequential schemata. Tab. 12 summarizes the average results.

The inter-strategy test shows interesting values. After setting a confidence level of 0.01, as shown in Tab. 14, the differences in the average scores between the simultaneous strategy and pure feature selection, standard outlier detection, and the original data appear to be statistically significant. From Tab. 15, on the other hand, it can be seen that also the improvement of each of our strategies that includes outlier detection over pure feature selection is statistically significant; the improvement of the simultaneous strategy over the



**FIGURE 3.** Residuals (vertical axis) versus fitted (horizontal axis) test; in red, the values for the original problem, and in blue, the values for the problem obtained with the best individual after the simultaneous strategy.



**FIGURE 4.** Residuals (vertical axis) versus leverage (horizontal axis) test; in red, the values for the original problem, and in blue, the values for the problem obtained with the best individual after the simultaneous strategy.

FS+OD strategy is also significant, but at a 0.05 confidence level. Thus, the simultaneous strategy produces better results than any of the other strategies that we have tested, and such improvements have a very high level of reliability in most cases. In Fig. 3 and Fig. 4, we display the results of the tests on the residuals of two models: the original model and the model obtained from the best individual of the simultaneous strategy. In both cases, it is possible to see how the residuals appear clearly closer to the fitting line and more normally distributed in the model obtained by the simultaneous strategy

compared with the original strategy, which indicates the superior quality of the former over the latter.

After having established that FSOD produces the best absolute results and the most statistically reliable findings, it is important to compare its results with those that can be obtained with standard techniques. Applying the filter (as in Tab 4) produces clearly worse results in the same conditions and only slightly better than those that can be obtained from the original data set. Applying robust regression (as in Tab. 5), which, let us recall, should be resilient to the

TABLE 13. Chosen signatures.

ex	attributes	out(%)	cc
1	Br, Cl, Fe, HCO3, Mg, Na	8.430	0.717
2	Ca, Eh, Fe, I	0.936	0.702
3	Ca, Eh, Fe, HCO3, I, SO4	7.728	0.725
4	Ca, Fe, HCO3, NO3	6.088	0.677
5	Ca, Eh, Fe, HCO3, I, SO4	5.854	0.703
6	Br, Ca, Fe, HCO3	5.854	0.703
7	Ca, Eh, Fe, I, K, SO4	8.196	0.709
8	Ca, Eh, Fe, I, NO3	8.430	0.700
9	Cl, Fe, HCO3, Na	5.152	0.701
10	Ca, Fe, HCO3, Mg, Na	8.665	0.704

TABLE 14. p-values obtained from the t-test, inter-strategy test (first part).

	FS	IR	OR
FSOD	< 0.0001*	< 0.0001*	< 0.0001*

TABLE 15. p-values obtained from the t-test, inter-strategy test (second part).

	FSOD	FS+OD	OD+FS	FS	OD
FSOD	-	0.0303 <sup>†</sup>	0.5703	< 0.0001*	0.3229
FS+OD	-	-	0.1321	< 0.0001*	0.3723
OD+FS	-	-	-	< 0.0001*	0.6336

presence of outliers and does not perform feature selection, produces slightly better results than those that can be obtained on both the original data set, the filtered data set, and the data sets obtained after feature selection only but decidedly worse than those obtained by our outlier detection method and our simultaneous method. Finally, lasso regression (shown in Tab. 6) does produce, in some execution, good results; however, the difference between the worst and the best case is so high that the method cannot be considered to be reliable, and on average, the correlation coefficient is still inferior to the average result obtained by our simultaneous technique. Moreover, lasso regression has a feature selection embedded algorithm, and in general, it does not show the selected features or the detected outliers.

E. INTERPRETATION

From the results obtained by FSOD, we can now extract the most interpretable individuals for the problem at hand. Recall that the ultimate aim of analyzing the water samples was to identify, if it exists, the physical-chemical signature of the contamination, in other words, the subset of variables that allow the extraction of a linear regression model to predict the level of contamination whose reliability, after having excluded potential outliers, is sufficiently high to allow a geochemical interpretation. Therefore, from each of the ten executions of the FSOD method, we extract the best individual (in terms of the correlation coefficient) with fewer than six variables, and we study such individuals to identify patterns. Tab. 13 shows such individuals. We can see that Ca, HCO3, Fe, and Na are the variables that apparently present the highest correlation with the contaminant.

VI. CONCLUSION, LIMITATIONS, AND FUTURE WORK

In this paper, we have approached two common problems associated with multivariate regression: selecting the best variables that predict the independent variable(s) and detecting the instances that can be suspected to be outliers. While these problems have been widely debated in the literature, they are often considered separately and with different heterogeneous methods. We have proposed a comprehensive, integrated optimization model that solves them simultaneously in such a way that outliers can be detected in reference to the specific variables that are selected for the regression. We implemented such an optimization model with a well-known evolutionary algorithm, and we tested it on real data taken from a problem of natural resource exploitation. The results show that our algorithm, when used to detect outliers without feature selection, already produces better models than those produced by standard outlier detection methods and those produced by standard feature selection methods. Additionally, they show that simultaneous feature selection and outlier detection, as we expected, produces the best results, comparable with those produced by attacking the two problems sequentially (in particular, first outlier detection and then feature selection), yet it is still superior (in terms of the correlation coefficient of the extracted linear model) to the latter. Since it is based on an evolutionary algorithm, one of the main limitations of our approach is the computation time, which is much higher than, for example, embedded outlier detection methods. Moreover, our method can be improved by including instance selection for a better result: the idea is that while outliers are discarded, the most representative instances are selected, and both choices depend on the selected features. Finally, our approach does not take into account different possibilities to solve the underlying multi-objective problem: in the future, we plan to explore not only modern evolutionary algorithms, such as binary differential evolution with self-learning or self-adaptive particle swarm optimization, but also non-evolutionary techniques.

REFERENCES

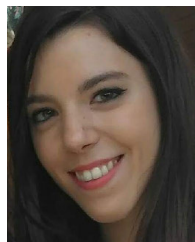
- [1] F.-J. Chang, C.-W. Huang, S.-T. Cheng, and L.-C. Chang, "Conservation of groundwater from over-exploitation—Scientific analyses for groundwater resources management," *Sci. Total Environ.*, vol. 598, pp. 828–838, Nov. 2017.
- [2] M. Zamanirad, A. Sarraf, H. Sedghi, A. Saremi, and P. Rezaee, "Modeling the influence of groundwater exploitation on land subsidence susceptibility using machine learning algorithms," *Natural Resour. Res.*, vol. 29, pp. 1–15, May 2019.
- [3] A. Khalil, M. N. Almasri, M. McKee, and J. J. Kaluarachchi, "Applicability of statistical learning algorithms in groundwater quality modeling," *Water Resour. Res.*, vol. 41, no. 5, pp. 1–16, May 2005.
- [4] B. Payandeh, "Some applications of nonlinear regression models in forestry research," *Forestry Chronicle*, vol. 59, no. 5, pp. 244–248, Oct. 1983.
- [5] A. Salimi, J. Rostami, C. Moormann, and A. Delisio, "Application of non-linear regression analysis and artificial intelligence algorithms for performance prediction of hard rock TBMs," *Tunnelling Underground Space Technol.*, vol. 58, pp. 236–246, Sep. 2016.
- [6] S. V. Archontoulis and F. E. Miguez, "Nonlinear regression models and applications in agricultural research," *Agronomy J.*, vol. 107, no. 2, pp. 786–798, Mar. 2015.

- [7] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, Mar. 2003.
- [8] J. T. de Souza, R. A. F. do Carmo, and G. A. C. de Lima, "On the combination of feature and instance selection," in *Machine Learning*, Y. Zhang, Ed. IntechOpen, Feb. 2010, pp. 157–172. [Online]. Available: <https://www.intechopen.com/chapters/10444>, doi: 10.5772/9153.
- [9] B. Jiang and J. Pei, "Outlier detection on uncertain data: Objects, instances, and inferences," in *Proc. IEEE 27th Int. Conf. Data Eng.*, Apr. 2011, pp. 422–433.
- [10] D. Cousineau and S. Chartier, "Outliers detection and treatment: A review," *Int. J. Psychol. Res.*, vol. 3, no. 1, pp. 58–67, 2010.
- [11] K. Singh and S. Upadhyaya, "Outlier detection: Applications and techniques," *Int. J. Comput. Sci. Issues*, vol. 9, no. 1, pp. 307–323, 2012.
- [12] W. Ni, "A review and comparative study on univariate feature selection techniques," Ph.D. dissertation, Dept. Elect. Eng. Comput. Sci., Univ. Cincinnati, Cincinnati, OH, USA, 2012.
- [13] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Roy. Statist. Soc. B, Methodol.*, vol. 58, no. 1, pp. 267–288, 1996.
- [14] M. Dash and H. Liu, "Feature selection for classification," *Intell. Data Anal.*, vol. 1, nos. 1–4, pp. 131–156, 1997.
- [15] H. Vafaie and K. D. Jong, "Genetic algorithms as a tool for feature selection in machine learning," in *Proc. 4th Int. Conf. Tools Artif. Intell. (TAI)*, 1992, pp. 200–203.
- [16] J. Yang and V. Honavar, "Feature subset selection using a genetic algorithm," *IEEE Intell. Syst. Appl.*, vol. 13, no. 2, pp. 44–49, Mar. 1998.
- [17] A. Khalili, "An overview of the new feature selection methods in finite mixture of regression models," *J. Iranian Stat. Soc.*, vol. 10, no. 2, pp. 201–235, 2011.
- [18] J. P. Barddal, F. Enembreck, H. M. Gomes, A. Bifet, and B. Pfahringer, "Boosting decision stumps for dynamic feature selection on data streams," *Inf. Syst.*, vol. 83, pp. 13–29, Jul. 2019.
- [19] R. Caruana and D. Freitag, "Greedy attribute selection," in *Proc. 11th Int. Conf. Mach. Learn.*, 1994, pp. 28–36.
- [20] A. Marcano-Cedeño, J. Quintanilla-Domínguez, M. G. Cortina-Januchs, and D. Andina, "Feature selection using sequential forward selection and classification applying artificial metaplasticity neural network," in *Proc. 36th Annu. Conf. IEEE Ind. Electron. Soc. (IECON)*, Nov. 2010, pp. 2845–2850.
- [21] S. F. Cotter, K. Kreutz-Delgado, and B. D. Rao, "Backward sequential elimination for sparse vector subset selection," *Signal Process.*, vol. 81, no. 9, pp. 1849–1864, Sep. 2001.
- [22] Narendra and Fukunaga, "A branch and bound algorithm for feature subset selection," *IEEE Trans. Comput.*, vols. C–26, no. 9, pp. 917–922, Sep. 1977.
- [23] P. Gupta, D. Doermann, and D. DeMenthon, "Beam search for feature selection in automatic SVM defect classification," in *Proc. Object Recognit. Interact. Service Robots*, 2002, pp. 212–215.
- [24] G. Nandi, "An enhanced approach to las Vegas filter (LVF) feature selection algorithm," in *Proc. 2nd Nat. Conf. Emerg. Trends Appl. Comput. Sci.*, Mar. 2011, pp. 1–3.
- [25] A. G. M. Al-Dawoodi, "An improved bees algorithm local search mechanism for numerical dataset," Ph.D. dissertation, School Comput., Univ. Utara Malaysia, Changlun, Malaysia, 2015.
- [26] L. Cervante, B. Xue, M. Zhang, and L. Shangi, "Binary particle swarm optimisation for feature selection: A filter based approach," in *Proc. IEEE Congr. Evol. Comput.*, Jun. 2012, pp. 1–8.
- [27] Z. Yong, G. Dun-Wei, and Z. Wan-Qiu, "Feature selection of unreliable data using an improved multi-objective PSO algorithm," *Neurocomputing*, vol. 171, pp. 1281–1290, Jan. 2016.
- [28] C. H. Chen, "On information and distance measures, error bounds, and feature selection," *Inf. Sci.*, vol. 10, no. 2, pp. 159–173, Jan. 1976.
- [29] L. Sun, J. Xu, and Y. Tian, "Feature selection using rough entropy-based uncertainty measures in incomplete decision systems," *Knowl.-Based Syst.*, vol. 36, pp. 206–216, Dec. 2012.
- [30] A. Al-Ani and M. Deriche, "Feature selection using a mutual information based measure," in *Proc. Object Recognit. Interact. Service Robots*, vol. 4, 2002, pp. 82–85.
- [31] S. K. Das, "Feature selection with a linear dependence measure," *IEEE Trans. Comput.*, vol. C-20, no. 9, pp. 1106–1109, Sep. 1971.
- [32] A. Arauzo-Azofra, J. M. Benitez, and J. L. Castroi, "Consistency measures for feature selection," *J. Intell. Inf. Syst.*, vol. 30, no. 3, pp. 273–292, 2008.
- [33] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artif. Intell.*, vol. 97, nos. 1–2, pp. 273–324, 1997.
- [34] S. S. Sane and A. A. Ghatol, "Use of instance typicality for efficient detection of outliers with neural network classifiers," in *Proc. 9th Int. Conf. Inf. Technol. (ICIT)*, Dec. 2006, pp. 225–228.
- [35] G. H. John, "Robust decision trees: Removing outliers from databases," in *Proc. 1st Int. Conf. Knowl. Discovery Data Mining*, 1995, pp. 174–179.
- [36] D. B. Skalak and E. L. Rissland, "Inductive learning in a mixed paradigm setting," in *Proc. 8th Nat. Conf. Artif. Intell.*, 1990, pp. 840–847.
- [37] S. Hawkins, H. He, G. J. Williams, and A. R. Baxter, "Outlier detection using replicator neural networks," in *Proc. 4th Int. Conf. Data Warehousing Knowl. Discovery*, 2002, pp. 170–180.
- [38] N. Goernitz, M. Kloft, K. Rieck, and U. Brefeld, "Toward supervised anomaly detection," *J. Artif. Intell. Res.*, vol. 46, pp. 235–262, Feb. 2013.
- [39] H. D. Kuna, R. García-Martínez, and F. R. Villatoro, "Outlier detection in audit logs for application systems," *Inf. Syst.*, vol. 44, pp. 22–33, Aug. 2014.
- [40] M. M. Breunig, H. P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying density-based local outliers," *SIGMOD Records*, vol. 29, no. 2, pp. 93–104, 2000.
- [41] E. M. Knorr, R. T. Ng, and V. Tucakov, "Distance-based outliers: Algorithms and applications," *Very Large Data Bases J.*, vol. 8, nos. 3–4, pp. 237–253, 2000.
- [42] S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient algorithms for mining outliers from large data sets," *ACM SIGMOD Rec.*, vol. 29, no. 2, pp. 427–438, 2000.
- [43] S. Hadi and J. Simonoff, "A stepwise procedure for identifying multiple outliers in linear regression," in *Proc. Meeting Amer. Stat. Assoc. Sect. Stat. Comput.*, 1990, pp. 137–142.
- [44] U. López, L. Trujillo, Y. Martínez, P. Legrand, E. Naredo, and S. Silva, "RANSAC-GP: Dealing with outliers in symbolic regression with genetic programming," in *Proc. 20th Eur. Conf. Genetic Program.*, in Lecture Notes in Computer Science, vol. 10196, 2017, pp. 114–130.
- [45] U. Lopez, L. Trujillo, and P. Legrand, "Filtering outliers in one step with genetic programming," in *Proc. 15th Int. Conf. Parallel Problem Solving Nature*, in Lecture Notes in Computer Science, vol. 11101, 2018, pp. 209–222.
- [46] R. Maronna, D. Martin, and V. Yohai, *Robust Statistics: Theory and Methods*. Hoboken, NJ, USA: Wiley, 2006.
- [47] R. Nunkesser and O. Morell, "An evolutionary algorithm for robust regression," *Comput. Statist. Data Anal.*, vol. 54, no. 12, pp. 3242–3248, Dec. 2010.
- [48] J. Hoeting, A. E. Raftery, and D. Madigan, "A method for simultaneous variable selection and outlier identification in linear regression," *Comput. Statist. Data Anal.*, vol. 22, no. 3, pp. 251–270, Jul. 1996.
- [49] R. S. Menjoge and R. E. Welsch, "A diagnostic method for simultaneous feature selection and outlier identification in linear regression," *Comput. Statist. Data Anal.*, vol. 54, no. 12, pp. 3181–3193, Dec. 2010.
- [50] S.-S. Kim, S. H. Park, and W. J. Krzanowski, "Simultaneous variable selection and outlier identification in linear regression using the mean-shift outlier model," *J. Appl. Statist.*, vol. 35, no. 3, pp. 283–291, Mar. 2008.
- [51] D. Fragoudis, D. Meretakis, and S. Likothanassis, "Integrating feature and instance selection for text classification," in *Proc. 8th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2002, pp. 501–506.
- [52] C.-F. Tsai, W. Eberle, and C.-Y. Chu, "Genetic algorithms in feature and instance selection," *Knowl.-Based Syst.*, vol. 39, pp. 240–247, Feb. 2013.
- [53] K. Yu, X. Xu, M. Ester, and H.-P. Kriegel, "Feature weighting and instance selection for collaborative filtering: An information-theoretic approach," *Knowl. Inf. Syst.*, vol. 5, no. 2, pp. 201–224, Apr. 2003.
- [54] Y. Collette and P. Siarry, *Multiobjective Optimization: Principles and Case Studies*. Berlin, Germany: Springer, 2004.
- [55] F. Jiménez, G. Sánchez, J. M. García, G. Sciacicco, and L. Miralles, "Multi-objective evolutionary feature selection for online sales forecasting," *Neurocomputing*, vol. 234, pp. 75–92, Apr. 2017.
- [56] C. Emmanouilidis, A. Hunter, J. Macintyre, and C. Cox, "A multi-objective genetic algorithm approach to feature selection in neural and fuzzy modeling," *Evol. Optim.*, vol. 3, no. 1, pp. 1–26, 2001.
- [57] A. Mukhopadhyay, U. Maulik, S. Bandyopadhyay, and C. A. C. Coello, "A survey of multiobjective evolutionary algorithms for data mining: Part I," *IEEE Trans. Evol. Comput.*, vol. 18, no. 1, pp. 4–19, Feb. 2014.
- [58] Y. Liu, D. Gong, J. Sun, and Y. Jin, "A many-objective evolutionary algorithm using a one-by-one selection strategy," *IEEE Trans. Cybern.*, vol. 47, no. 9, pp. 2689–2702, Sep. 2017.
- [59] Y. Xue, B. Xue, and M. Zhang, "Self-adaptive particle swarm optimization for large-scale feature selection in classification," *ACM Trans. Knowl. Discovery Data*, vol. 13, no. 5, pp. 1–27, Oct. 2019.

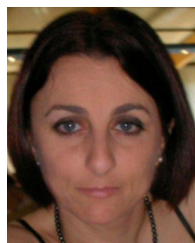
- [60] Y. Zhang, D.-W. Gong, X.-Z. Gao, T. Tian, and X.-Y. Sun, "Binary differential evolution with self-learning for multi-objective feature selection," *Inf. Sci.*, vol. 507, pp. 67–85, Jan. 2020.
- [61] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. London, U.K.: Wiley, 2001.
- [62] M. E. ElAlami, "A filter model for feature subset selection based on genetic algorithm," *Knowl.-Based Syst.*, vol. 22, no. 5, pp. 356–362, Jul. 2009.
- [63] R. C. Anirudha, R. Kannan, and N. Patil, "Genetic algorithm based wrapper feature selection on hybrid prediction model for analysis of high dimensional data," in *Proc. 9th Int. Conf. Ind. Inf. Syst. (ICIIS)*, Dec. 2014, pp. 1–6.
- [64] J. Huang, Y. Cai, and X. Xu, "A hybrid genetic algorithm for feature selection wrapper based on mutual information," *Pattern Recognit. Lett.*, vol. 28, no. 13, pp. 1825–1844, May 2007.
- [65] H. Ishibuchi and T. Nakashima, "Multi-objective pattern and feature selection by a genetic algorithm," in *Proc. Genetic Evol. Comput. Conf.*, 2000, pp. 1069–1076.
- [66] J. Liu and H. Iba, "Selecting informative genes using a multiobjective evolutionary algorithm," in *Proc. Congr. Evol. Comput. (CEC)*, 2002, pp. 297–302.
- [67] A. G. F. Pappa and C. Kaestner, "Attribute selection with a multi-objective genetic algorithm," in *Proc. 16th Brazilian Symp. Artif. Intell.*, 2002, pp. 280–290.
- [68] S. Y. M. Shi, P. N. Suganthan, and K. Deb, "Multiclass protein fold recognition using multiobjective evolutionary algorithms," in *Proc. IEEE Int. Geosci. Remote Sens. (IGARSS)*, Oct. 2004, pp. 61–66.
- [69] J. García-Nieto, E. Alba, L. Jourdan, and E. Talbi, "Sensitivity and specificity based multiobjective approach for feature selection: Application to cancer diagnosis," *Inf. Process. Lett.*, vol. 109, no. 16, pp. 887–896, Jul. 2009.
- [70] Z. Zhu, Y. Ong, and J. Kuo, "Feature selection using single/multi-objective memetic frameworks," in *Multi-Objective Memetic Algorithms*, in *Studies in Computational Intelligence*, vol. 171. Berlin, Germany: Springer, 2009, pp. 111–131.
- [71] M. Venkataradi and K. Rao, "A multiobjective genetic algorithm for feature selection in data mining," *Int. J. Comput. Sci. Inf. Technol.*, vol. 1, no. 5, pp. 443–448, 2010.
- [72] A. Ekbal, S. Saha, and C. S. Garbe, "Feature selection using multiobjective optimization for named entity recognition," in *Proc. 20th Int. Conf. Pattern Recognit.*, Aug. 2010, pp. 1937–1940.
- [73] D. A. C. Reynolds and M. Chantler, "Feature selection for multi-purpose predictive models: A many-objective task," in *Proc. 11th Int. Conf. Parallel Problem Solving Nature*, 2010, pp. 384–393.
- [74] F. Jiménez, R. Jódar, M. Martín, G. Sánchez, and G. Sciavicco, "Unsupervised feature selection for interpretable classification in behavioral assessment of children," *Expert Syst.*, vol. 34, no. 4, pp. 1–15, 2017.
- [75] F. Jimenez, C. Martínez, E. Marzano, J. T. Palma, G. Sánchez, and G. Sciavicco, "Multiobjective evolutionary feature selection for fuzzy classification," *IEEE Trans. Fuzzy Syst.*, vol. 27, no. 5, pp. 1085–1099, May 2019.
- [76] L. Chomatek and A. Duraj, "Multiobjective genetic algorithm for outliers detection," in *Proc. IEEE Int. Conf. Innov. Intell. Syst. Appl. (INISTA)*, Jul. 2017, pp. 379–384.
- [77] G. J. Krishnai and V. Ravi, "Outlier detection using evolutionary computing," in *Proc. Int. Conf. Informat. Anal.*, Aug. 2016, pp. 1–6.
- [78] L. Martí, A. Fanni-Tchango, L. Navarro, and M. Schoenauer, "Progressively adding objectives: A case study in anomaly detection," in *Proc. Genetic Evol. Comput. Conf.*, Jul. 2017, pp. 593–600.
- [79] Z. Chen, C. K. Yeo, B. S. Lee, C. T. Lau, and Y. Jin, "Evolutionary multi-objective optimization based ensemble autoencoders for image outlier detection," *Neurocomputing*, vol. 309, pp. 192–200, Oct. 2018.
- [80] J. J. Durillo and A. J. Nebro, "jMetal: A Java framework for multi-objective optimization," *Adv. Eng. Softw.*, vol. 42, no. 10, pp. 760–771, 2011.
- [81] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*, 3rd ed. San Mateo, CA, USA: Morgan Kaufmann, 2011.
- [82] G. Williams, R. Baxter, H. He, S. Hawkins, and L. Gu, "A comparative study of RNN for outlier detection in data mining," in *Proc. IEEE Int. Conf. Data Mining*, Dec. 2002, pp. 709–712.
- [83] R Core Team, "R: A language and environment for statistical computing," R Foundation for Statistical Computing, Inst. Statist. Math., Univ. Wien, Tech. Rep., 2013.



**FERNANDO JIMÉNEZ** received the M.D. degree in computer science from the University of Granada, Spain, and the Ph.D. degree in computer science from the University of Murcia, Spain. He is currently an Associate Professor with the University of Murcia. His research interests include evolutionary computation, multi-objective constrained optimization, soft computing, evolutionary fuzzy systems, and data mining.



**ESTRELLA LUCENA-SÁNCHEZ** is currently pursuing the Ph.D. degree with the University of Modena and Reggio Emilia, Italy, and with the University of Ferrara, Italy. She specializes in intelligent data mining and knowledge extraction from biological, chemical, and physical data, with classical and new methods and techniques.



**GRACIA SÁNCHEZ** received the M.D. degree in computer science from the Polytechnic University of Valencia, Spain, and the Ph.D. degree in computer science from the University of Murcia, Spain. She is currently an Associate Professor with the University of Murcia. Her research interests include multi-objective constrained optimization, soft computing, evolutionary fuzzy systems, and data mining.



**GUIDO SCIAVICCO** received the M.D. and Ph.D. degrees in computer science from the University of Udine, Italy. He is currently an Associate Professor with the University of Ferrara, Italy. He has been researching in temporal, spatial, and modal logic for over ten years, especially in satisfiability and model-checking problems. Recently, he has focused on studying temporal data mining problems with applications to real-world cases.

...