Pressure drop modelling in sand filters in micro-irrigation using gradient boosted regression trees

P.J. García Nieto[a,*], E. García–Gonzalo[a], G. Arbat[b], M. Duran–Ros[b], F. Ramírez de Cartagena[b], J. Puig-Bargués[b]

[a]Department of Mathematics, Faculty of Sciences, University of Oviedo, 33007 Oviedo, Spain

[b]Department of Chemical and Agricultural Engineering and Technology, University of Girona, 17071 Girona, Catalonia, Spain

Abstract

Filters are essential for guaranteeing the good performance of microirrigation systems. Pressure losses across filters should be known for the proper design and management of this irrigation equipment. Pressure losses produced by filtering media in sand filters can be computed using Ergun or Kozeny-Karman equations, which require knowledge, among other parameters, of the sphericity of the filter medium. As this parameter is not easy to determine, it is useful to explore the performance of alternative computing methods that can avoid requiring knowledge of sphericity. In this paper, taking as starting point the nonparametric machine learning approach known as the gradient boosted regression tree (GBRT) approach and hybridising it with the differential evolution (DE) technique, the pressure drop in sand filters used in microirrigation has been modelled. For different filtering materials such as modified glass, crushed glass, silica sand and glass microspheres, experimental data of pressure drop for velocities between 0.004 and 0.025 m s^{-1} was collected and the model built. The results demonstrated that DE–GBRT–based model was able to accurately predict pressure drop. The model also allowed ranking of

*Corresponding author. Tel.: +34-985103417; fax: +34-985103354.
E-mail address: lato@orion.ciencias.uniovi.es (P.J. García Nieto).

the importance of the independent variables examined within the model. Taking into account this ranking, and using only the main variables, a simplified method with an improved coefficient of determination was constructed.

Keywords: Regression trees; Gradient boosting; Differential evolution; Drip irrigation; Sand filters

Nomenclature

ABC	Artificial bee colony
b_{jm}	Constant value calculated for the region R_{jm}
C_o	Cover of the GBRT algorithm
$CART$	Classification and Regression Trees
D_{eq}	Equivalent diameter, m
DE	Differential evolution
F_m	Weak model that predicts the mean y of the training set
F_q	Frequency of the GBRT algorithm
$F_0(x)$	Constant function
$\hat{F}(x)$	an estimate of the function $F^*(x)$
G_a	Gain of the GBRT algorithm
GA	Genetic algorithm
$GBRT$	Gradient boosted regression tree
GR	Parameter that controls the recombination rate

H	Set of arbitrary differentiable functions
h	Weak learner function
$h_i(x)$	Weighted sum of functions
$h_m(x)$	Decision tree
J_m	Number of terminal nodes in the tree model
NP	Noisy random vectors
L( )	Loss function
m	Weighted medium mass, kg
m_{og}	Overall mass of the grains, kg
MARS	multivariate adaptive regression splines
MCW	Minimum child weight of GBRT algorithm
MDS	Minimum delta step of GBRT algorithm
n	Number of observed data
N	Number of grains
Nrounds	Maximum number of iterations of the GBRT algorithm
p	Index of the individual in the population
PSO	Particle swarm optimization
r_{im}	Pseudo-residuals
RMSE	Root mean square error
R^2	Coefficient of determination
SR	Subsample ratio of the GBRT algorithm
SS_{tot}	Total sum of squares
SS_{reg}	Regression sum of squares
SS_{err}	Residual sum of squares

$\mathbf{t}_m^g$	Trial vectors
V_m	Medium volume, m^3
V_w	Volume of the additional water, m^3
V_f	Final volume of the water and medium mixture, m^3
V	Mean flow velocity, m s^{-1}
$\mathbf{x}_p^g$	Original vectors
$(\Delta p / \Delta L)$	Pressure drop per unit length
Δ	Step value over each tree's weight estimation
ε	Medium porosity
η	Learning rate of GBRT algorithm
ρ_b	Bulk density of each medium, kg m^{-3}
ρ_r	Real density of each medium, kg m^{-3}
ϕ	Sphericity factor
γ	Minimum loss reduction of the GBRT algorithm
Ω	Penalty function that controls the model complexity

1. Introduction

Proper irrigation water filtration is essential to ensure the successful continuous long-term operation of microirrigation systems (Clark, Haman, Prochaska, & Yitayew, 2007). By following good maintenance practices, which includes filtration, the longevity of some subsurface microirrigation systems have reached 26.5 years (Lamm & Rogers, 2017). Screen, disc, media and hydro-cyclone filters are common filter types that are used in microirrigation systems. The choice of filter type will basically depend on the quality of water source, the flow rate of the irrigation system and the desired filtered water quality for avoiding emitter clogging (Clark et al, 2007).

Irrigation engineers require knowledge of the pressure drop across the filter to properly design and manage this important system component which is related to water and energy consumption as well as pollutant removal efficiency (Duran-Ros, Puig-Bargués, Arbat, Barragán, & Ramírez de Cartagena, 2009). Mathematical models have been developed using dimensional analysis for describing pressure drops across screens (Wu, Chen, Liu, Yin, & Niu, 2014b; Zong, Zheng, Liu & Li, 2015), disc (Yurdem, Demir, & Degirmencioglu, 2008; Wu et al., 2014a), hydrocyclone (Yurdem, Demir, & Degirmencioglu, 2008) and in sand media filters (Elbana, Ramírez de Cartagena, & Puig-Bargués, 2013). These models did not consider the specific effect of the different filter components (filtration zone and auxiliary elements) on pressure loss. In sand media filters, pressure loss clearly vary across the filter media, the underdrain and diffuser platter, and the backflushing valve (Bové et al., 2015b; Burt, 2010; Mesquita, Testezlaf & Ramirez, 2012).

Bové et al. (2015a) experimentally analysed the pressure drop across different sand and recycled glass media in a microirrigation sand filter. Although the Ergun equation showed the best prediction accuracy for predicting the pressure drop, multi linear regression equations had better performance than the Kozeny–Carman equation, which is a simplification of the Ergun equation. However, these equations require parameters defining the filter media such as equivalent diameter and sphericity which are difficult to obtain.

García-Nieto et al. (2017) used a hybrid model artificial bee colony (ABC)-multivariate adaptive regression splines (MARS) which satisfactorily computed pressure loss across filtration beds without the need for sphericity. This work suggests that other alternative methods, specifically a hybrid methodology that combines the gradient boosted regression tree (GBRT) approach with the differential evolution (DE) optimisation algorithm (Storn, & Price, 1997; Price, Storn, & Lampinen, 2005; Feoktistov, 2006; Rocca, Oliveri, & Massa, 2011), could also be used to predict pressure drops in the granular filters used in microirrigation systems.

GBRT models are supervised machine learning procedures that can be used for multivariate classification and regression (Vapnik, 1998; Friedman, 2002; Schapire, 2003; Bühlman & Hothorn, 2007; Hastie et al., 2003). GBRT models build competitive, highly robust procedures that are particularly appropriate for treating not very clean data (Hastie et al., 2003). They are very flexible models that can be easily be customised for any data-driven task. They are straightforward to implement and have been very successful in data-mining and machine-learning challenges (Natekin & Knoll, 2013). One of the reasons for their success could be that tree boosting takes the bias-variance trade-off into consideration while fitting the models (Nielsen, 2016). For example, GBRT models have been effective in predicting biological parameters in environmental problems such as forecasting wind variables (Landry et al., 2016), solar power generation prediction (Persson et al., 2017) and short-term waste estimation (Johnson et al., 2017).

Differential evolution (DE) is a metaheuristic evolutionary global method, derived from genetic algorithm (GA), intrinsically capable of solving multidimensional optimisation problems involving continuous variables. As with other evolutionary computation algorithms such as particle swarm optimisation (PSO) (Eberhart et al., 2001; Clerc, 2006; Olsson, 2011) or ant colony optimisation (Dorigo & Stützle, 2004), DE is a bio-inspired algorithm that generates high-quality solutions to optimisation problems by means of bio-inspired operators such as mutation, recombination and selection (Storn & Price, 1997; Price, Storn & Lampinen, 2005; Simon, 2013; Yang et al., 2013).

The main objective of the present study was to develop a hybrid algorithm using DE optimising GBRT parameters (DE–GBRT) to predict the pressure drop per unit length $\left(\Delta p / \Delta L\right)$ across sand and recycled glass media from the physical input parameters of the filtration media used in media filters.

2. Materials and methods

2.1. Experimental setup

The experimental setup providing the considered data set is described in Bové et al. (2015a). In a laboratory filter, which was a scaled version of a commercial microirrigation media filter (Arbat et al., 2013), pressure losses of four different filtration materials (silica sand, crushed recycled glass, surface modified glass and microspheres) with grain sizes between 0.63 and 1.50 mm were measured at surface velocities ranging from 0.004 to 0.025 m s^{-1} under pressures ranging between 4,631 and 275,630 Pa.

2.2. Variables involved in the model

Data obtained from the experiment gave the pressure drop per unit length $\left(\Delta p / \Delta L\right)$, as the output variable. The input variables were the filter media type (as category), media bulk and real density, porosity, equivalent diameter, sphericity or shape factor, flow surface velocity and average grain size. Procedures for obtaining these variables are described in Bové et al. (2015a).

2.3. Computational procedure

2.3.1. Gradient boosting regression tree (GBRT)

Gradient boosting is a machine learning method used for classification and regression that constructs a model from a set of weak models or learners, that are, usually, decision trees. It builds the model by stages, as is typical for boosting methods, and obtains a single strong ensemble model optimising a differentiable loss function (Breinman et al. 1984; Vapnik, 1998; Friedman et al., 2000; Friedman, 2001; Friedman, 2002; Schapire, 2003; Bühlman & Hothorn 2007; Hastie et al., 2003).

It can be described as a least-squares regression method, where the aim is to teach a model F to predict the values $\hat{y} = F\left(x\right)$, minimizing the mean squared error $\left(\hat{y} - y\right)^2$, being y the true values from the training set. At each stage $1 \le m \le M$ of gradient boosting, we have a weak model F_m that predicts the mean y of the training set. The gradient boosting algorithm improves F_m constructing a new model F_{m+1} that adds an estimator h to improve the previous model $F_{m+1}\left(x\right) = F_m\left(x\right) + h\left(x\right)$. To find h, the gradient boosting method takes into account that, to have a perfect h (Friedman et al., 2000; Schapire, 2003; Bühlman & Hothorn, 2007; Hastie et al., 2003):

$$F_{m+1}(x) = F_m(x) + h(x) = y \tag{1}$$

that is,

$$h(x) = y - F_m(x) \tag{2}$$

Thus, gradient boosting will perform the fitting of h to the residual $y - F_m(x)$. In each stage, F_{m+1} is constructed as a correction of its predecessor F_m. We can generalise this explanation to other loss functions different from squared error, taking into account that residuals $y - F(x)$ are the negative gradients of the loss function $\frac{1}{2}(y - F(x))^2$.

As in other supervised learning problems, we have an output variable y and a set of input variables x. The objective is to find an estimate $\hat{F}(x)$ of the function $F^*(x)$ that minimises the value of some loss function $L(y, F(x))$ using a training set $\{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$ of already known values of x and their corresponding values of y, (Friedman, 2002; Schapire, 2003; Bühlman & Hothorn, 2007; Hastie et al., 2003; Mayr et al., 2014a,b; Taieb & Hyndman, 2014; Döpke et al., 2017):

$$\hat{F} = \arg\min_F \mathrm{E}_{x,y}\left[L(y, F(x))\right] \tag{3}$$

The gradient boosting method approximates y with a weighted sum of functions $h_i(x)$ from some class H, called weak learners:

$$F(x) = \sum_{i=1}^{M} \gamma_i h_i(x) + \text{const} \tag{4}$$

Using the empirical risk minimisation principle, the method looks for an approximation $\hat{F}(x)$ that minimises the average value of the loss function on the training set. It starts with a model, that consists in a constant function $F_0(x)$, and step by step expands its value in a greedy way (Hastie et al., 2003; Taieb & Hyndman, 2014; Döpke et al., 2017):

$$F_0(x) = \arg\min_{\gamma} \sum_{i=1}^{n} L(y_i, \gamma) \tag{5}$$

$$F_m(x) = F_{m-1}(x) + \arg\min_{h \in H} \sum_{i=1}^{n} L\left(y_i, F_{m-1}(x_i) + h(x_i)\right) \tag{6}$$

where $h \in H$ is a weak learner function.

As choosing the best function h at every stage for an arbitrary loss function L is a computationally infeasible optimization problem, a simplification is applied and a steepest descent method is used to solve this minimization problem. Given the continuous case, where H is the set of arbitrary differentiable functions, the model is updated following the equations (Hastie et al., 2003; Taieb & Hyndman 2014; Döpke et al., 2017):

$$F_m(x) = F_{m-1}(x) - \gamma_m \sum_{i=1}^{n} \nabla_{F_{m-1}} L\left(y_i, F_{m-1}(x_i)\right) \tag{7}$$

$$\gamma_m = \arg\min_{\gamma} \sum_{i=1}^{n} L\left(y_i, F_{m-1}(x_i) - \gamma \frac{\partial L\left(y_i, F_{m-1}(x_i)\right)}{\partial F_{m-1}(x_i)} \right) \tag{8}$$

where the derivatives are obtained with respect to the functions F_i for $i \in \{1, 2, ..., m\}$. If we are treating a discrete case, where the set H is finite, the candidate function h that is closest to the gradient of L will be chosen and the coefficient γ can then be calculated using line search in equations (7) and (8). This is a heuristic approach and will not give an exact solution to problem, but a good approximation.

The generic gradient boosting method can be described by a pseudocode (Friedman, 2002; Hastie et al., 2003; Taieb & Hyndman, 2014; Döpke et al., 2017):

- Input: differentiable loss function $L(y, F(x))$, training set $\{(x_i, y_i)\}_{i=1}^{n}$ and iteration number *M*.
- Algorithm:
  1. Initialize model using a constant value:
     $$F_0(x) = \arg\min_{\gamma} \sum_{i=1}^{n} L(y_i, \gamma)$$
  2. For $m = 1$ to M:
     - Compute so-called *pseudo-residuals*:
       $$r_{im} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)} \quad \text{for } i = 1, ..., n$$
     - Fit a weak learner $h_m(x)$ to the pseudo-residuals using the training set $\{(x_i, r_{im})\}_{i=1}^{n}$.
     - Calculate the multiplier γ_m solving the one-dimensional optimisation problem:
       $$\gamma_m = \arg\min_{\gamma} \sum_{i=1}^{n} L(y_i, F_{m-1}(x_i) + \gamma h_m(x_i))$$
     - Update the model:
       $$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x)$$
  3. Output $F_M(x)$.

Gradient boosting can be used with decision trees, in particular with CART, of a given fixed size as weak learners. For this particular situation, Friedman (Friedman, 2002) proposes a variation of the gradient boosting method that improves each weak learner quality of fit (Friedman, 2002; Ridgeway, 2007; Hastie et al., 2003; Taieb & Hyndman, 2014; Döpke et al., 2017).

In the *m*-th step a generic gradient boosting fits a decision tree $h_m(x)$ to the pseudo-residuals. If J_m is the number of its leaves, the tree model splits the input space into J_m separated regions $R_{1m},...,R_{J_m m}$ and obtains a constant value for each region. $h_m(x)$ for input x is written as the sum (Bühlmann & Hothorn, 2007; Hastie et al., 2003; Taieb & Hyndman, 2014; Döpke et al., 2017):

$$h_m(x) = \sum_{i=1}^{J_m} b_{jm} I(x \in R_{jm}) \tag{9}$$

where b_{jm} is the constant value calculated for the region R_{jm}. These coefficients b_{jm} are multiplied by some value γ_m, calculated using line search that minimises the loss function, and then the model is updated:

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x); \quad \gamma_m = \arg\min_{\gamma} \sum_{i=1}^{n} L\left(y_i, F_{m-1}(x_i) + \gamma h_m(x_i)\right) \tag{10}$$

Friedman (Shapire, 2003; Bühlmann, & Hothorn, 2007) proposed a modification of this algorithm that chooses a different optimal γ_{jm} for each of the regions, instead of only one γ_m for the whole tree. This modified algorithm is called TreeBoost. Then, the model is updated (Bühlmann & Hothorn, 2007; Hastie et al., 2003; Taieb & Hyndman, 2014; Döpke et al., 2017):

$$F_m(x) = F_{m-1}(x) + \sum_{i=1}^{J_m} \gamma_{jm} I\left(x \in R_{jm}\right); \quad \gamma_{jm} = \arg\min_{\gamma} \sum_{x_i \in R_{jm}} L\left(y_i, F_{m-1}(x_i) + \gamma\right) \tag{11}$$

where the size of trees, *J*, is the number of terminal nodes in trees and it is a parameter that can be adjusted for the training data set. It controls the interaction level between variables in the model. If $J = 2$ (decision stumps), there is no interaction between variables. With $J = 3$ the model can allow interactions between up to two variables, and so on. Typically a value between 4 and 8 works well and the results are quite insensitive to *J* for these values. $J = 2$ is usually not enough for many applications, and $J > 10$ is often unnecessary.

Overfitting the training set can lead to a poor prediction ability. The *regularisation* techniques are intended to reduce this overfitting effect controlling the training process.

There are different approaches to attain this aim (Bühlmann and Hothorn 2007; Hastie et al., 2003; Taieb & Hyndman 2014; Döpke et al., 2017). In particular, the technique used by the function GBRT is to include in the loss function the so called penalty function whose aim is to limit the overfitting:

$$L(x) = E(x) + \Omega(x) \tag{12}$$

where *E* can be, for instance, the mean squared error, and Ω is the penalty function that controls the model complexity, aiding to avoid overfitting by means of increasing the value of the loss function when the complexity of the model grows, thus penalising it.

It is also well known that the GBRT technique depends strongly on the following hyperparameters (Chen, & Guestrin, 2016; Chen et al., 2017):

- *Nrounds*: is the maximum number of iterations performed by the algorithm.
- η : it controls the learning rate, that is to say, scales the contribution of each tree by a factor η when it is added to the current approximation. Used to prevent overfitting by making the boosting process more conservative. Lower value for η implies larger value for *Nrounds*.
- γ : It is the minimum loss reduction required to perform another partition on a leaf node of the tree. As it grows, the algorithm is more conservative.
- Minimum child weight: this parameter avoids the splitting of a node once its sample size has gone under a certain threshold.
- Maximum Δ step: is a cap value over each tree's weight estimation.
- Subsample ratio: is the ratio between the training and testing instances.

Therefore, it is convenient to use some technique that adjusts these parameters. Usually, the traditional way of performing hyperparameter optimisation has been *grid search*, or a parameter sweep, which is simply an exhaustive searching through a manually specified subset of the hyperparameter space of a learning algorithm. Indeed, the grid search is a brute force method and, as such, almost any optimisation method improves its efficiency. In this study, in order to avoid these problems associated with the grid search method, the differential evolution (DE) metaheuristic technique described below was used (Storn & Price, 1997; Price, Storn, & Lampinen, 2005; Simon, 2013; Yang et al., 2013) with success.

2.3.2. The differential evolution (DE) algorithm

In evolutionary computation, differential evolution (DE) is a metaheuristic method that optimises a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality. DE is used for multidimensional real-valued but does not require for the optimisation problem to be differentiable. Therefore, DE can also be used for optimisation problems that are not continuous, are noisy, and change over time, etc. DE optimises a problem by maintaining a population of candidate solutions and creating new candidate solutions by combining existing ones according to its simple formulae, and then keeping whichever candidate solution has the best fitness on the optimisation problem at hand (Storn & Price, 1997).

The algorithm assumes that the variables of the problem to be optimised are encoded as a vector of real numbers. The length n of these vectors is equal to the number of variables of the problem, and the population is composed of NP vectors (number of parents). A vector $\mathbf{x}_p^g$ is defined, where p is the index of the individual in the population $(p = 1, \ldots, NP)$ and g is the corresponding generation. Each vector is composed in turn by the variables of the problem $x_{p,m}^g$, where m is the index of the variable in the individual $(m = 1, \ldots, n)$. It is assumed that the domain of the problem variables is constrained between minimum and maximum values $\mathbf{x}_m^{\min}$ and $\mathbf{x}_m^{\max}$, respectively. Hence, DE technique is basically composed of four steps:

- Initialisation;
- Mutation;
- Recombination; and
- Selection.

Initialisation is performed at the beginning of the search, and the mutation-recombination-selection steps are performed repeatedly, until a termination condition or stopping criterion is satisfied (number of generations, elapsed time, or quality of solution reached, etc.).

Initialisation

The population is initialised (first generation) randomly, considering the minimum and maximum values of each variable:

$$\mathbf{x}_{p,m}^{1} = \mathbf{x}_{m}^{\min} + rand\left(0,1\right)\cdot\left(\mathbf{x}_{m}^{\max} - \mathbf{x}_{m}^{\min}\right) \text{ for } p = 1,\ldots,NP \text{ and } m = 1,\ldots,n \tag{13}$$

where $rand\left(0,1\right)$ is a random number in the range $\left[0,1\right]$.

Mutation

Mutation is the construction of *NP* noisy random vectors, which are created from three individuals chosen at random, called target vectors $\mathbf{x}_a$, $\mathbf{x}_b$ and $\mathbf{x}_c$. The noisy random vectors $\mathbf{n}_p^t$ are obtained as follows:

$$\mathbf{n}_p^g = \mathbf{x}_c + F\cdot\left(\mathbf{x}_a - \mathbf{x}_b\right) \text{ for } p = 1,\ldots,NP \tag{14}$$

with *p*, *a*, *b* and *c* different from each other. *F* is a parameter that controls the mutation rate, and is in the range $\left[0,2\right]$.

Recombination

After obtaining the *NP* noisy random vectors, the recombination is performed in a random manner, comparing them with the original vectors $\mathbf{x}_p^g$, obtaining the trial vectors $\mathbf{t}_m^g$ as follows:

$$t_{p,m}^{g} = \begin{Bmatrix} n_{p,m}^{g} & \text{if} \quad rand(0,1) < GR \\ x_{p,m}^{g} & \text{otherwise} \end{Bmatrix} \text{for } p = 1,\ldots,NP \text{ and } m = 1,\ldots,n \tag{15}$$

GR is a parameter that controls the recombination rate. Note that the comparison is carried out variable by variable, so that the test vector will be a mixture of the noisy random vectors and original vectors.

Selection

Finally, the selection is made simply by comparing the test vectors with the original ones, so that the vector of the next generation will be the one that has the best value of the fitness function fit:

$$\mathbf{x}_{p}^{g+1} = \begin{Bmatrix} \mathbf{t}_{p}^{g} & \text{if} \quad fit\left(\mathbf{t}_{p}^{g}\right) > fit\left(\mathbf{x}_{p}^{g}\right) \\ \mathbf{x}_{p}^{g} & \text{otherwise} \end{Bmatrix} \tag{16}$$

2.4. The goodness–of–fit of this approach

The operation physical input variables considered in this research work are shown in Table 1. Therefore, the total number of predicting variables used to construct the hybrid DE–GBRT-based model was eight. The output predicted variable is the pressure drop per unit length $\left(\Delta p / \Delta L\right)$ and the input variable for filter media type was a category.

Table 1 - Set of operation physical input variables used in this study and their names along with their mean and standard deviation.

To predict the pressure drop per unit length $\left(\Delta P / \Delta L\right)$ from other operation parameters, it is necessary to choose the model that best fits the experimental data. To determine the goodness–of–fit, the two criteria considered here were the coefficient of determination R^2 and the root mean square error (*RMSE*), respectively (Freedman et al., 2007). A dataset takes values t_i, each of which has an associated modelled value y_i. The former are termed the observed values and the latter are often referred to as the predicted values. The dataset variability is measured through different sums of squares as follows (Freedman et al., 2007):

- $SS_{tot} = \sum_{i=1}^{n} (t_i - \bar{t})^2$: the total sum of squares, proportional to the sample variance.
- $SS_{reg} = \sum_{i=1}^{n} (y_i - \bar{t})^2$: the regression sum of squares, also termed the explained sum of squares.
- $SS_{err} = \sum_{i=1}^{n} (t_i - y_i)^2$: the residual sum of squares.

Note that in the previous sums, $\bar{t}$ is the mean of the n observed data:

$$\bar{t} = \frac{1}{n}\sum_{i=1}^{n} t_i \qquad (17)$$

Taking into account the above sums, the coefficient of determination is defined via:

$$R^2 \equiv 1 - \frac{SS_{err}}{SS_{tot}} \qquad (18)$$

so that a coefficient of determination value of 1.0 points out that the regression curve fits the data perfectly.

Similarly, the second ratio used in this research work to measure the goodness–of–fit is the root mean square error (*RMSE*). It indicates the sample standard deviation of the differences between predicted values and observed values. The *RMSE* is defined for n different predictions as follows (Freedman et al., 2007):

$$RMSE \equiv \sqrt{\frac{SS_{err}}{n}} \tag{19}$$

2.3.3. The hybrid DE-GBRT-base model

Additionally, as previously mentioned, the GBRT technique is greatly dependent on the GBRT hyperparameters such as the maximum number of iterations (*Nrounds*), learning rate, minimum loss reduction, minimum child weight, maximum step and subsample ratio. Some methods frequently used to determine suitable hyperparameters are (Hastie, Tibshirani, & Friedman, 2003): grid search, random search, Nelder-Mead search, heuristic search, genetic algorithms, pattern search and so on. Usually, the traditional way of performing hyperparameter optimisation has been *grid search*, or a *parameter sweep*, which selects sets of parameters from a chosen grid and studies the performance of the model for each set. Indeed, the grid search is a brute force method and, as such, almost any optimisation method improves its efficiency. In this study, in order to avoid these problems associated with the grid search method, the differential evolution (DE) metaheuristic technique was used (Price, Storn, & Lampinen, 2005; Simon, 2013; Yang et al., 2013).

The DE optimisation technique was selected as ait appeared to be an appropriate, effective and simple tool for tuning the GBRT parameters. A hybrid model, specifically a novel hybrid DE–GBRT–based model, was constructed taking as its dependent variable the pressure drop per unit length (output variable) from the other eight remaining variables (input variables) found in granular filters (Bové et al., 2015a), studying their effect in order to optimise its calculation through the analysis of the coefficient of determination R^2 with success. Fig. 1 shows the flowchart of this new hybrid DE–GBRT–based model implemented in this research work.

Fig. 1 - Flowchart of the new hybrid DE–GBRT–based model.

Furthermore, cross-validation was the standard technique utilised here for finding the real coefficient of determination (R^2) (Picard & Cook, 1984; Freedman et al., 2007). Indeed, in order to assessment the predictive capacity of the DE–GBRT–based model, a thorough 10-fold cross-validation algorithm was implemented in this study (Picard & Cook, 1984). To this end, the regression modelling has been performed with the Extreme Gradient Boosting algorithm, using the *Xgboost library* (Chen, He, Benesty, Khotilovich & Tang, 2017) along with the DE technique with the DEoptim package (Ardia, Mullen, Brian, & Peterson, 2016) from the R Project. The initial intervals of the space of solutions used in DE technique are indicated in Table 2.

It should be noted that sixty population members were used in the DE optimisation. The process stopped if the value of the relative tolerance (10^{-8}) could not be reduced after 30 steps or a maximum number of 200 iterations. Under this conditions, the tuning of the parameters required 89 iterations in order to get convergence.

Table 2 - Search space for each of the GBRT parameters in the DE tuning process.

In order to optimise the GBRT parameters, the DE module was used. In this way, the DE looks for the best parameters (maximum number of iterations (rounds), learning rate, minimum loss reduction, minimum child weight, maximum step and subsample ratio) by using the comparison of the cross-validation error in every iteration. The search space is six-dimensional, with one dimension per each parameter. Hence, the objective function or main fitness factor is the coefficient of determination (R^2) in this problem.

3. Results and discussion

Table 3 points out the optimal hyperparameters of the best fitted DE–GBRT–based model found with the differential evolution (DE) technique.

Table 3 - Optimal hyperparameters of the best fitted GBRT model found with the DE technique.

Table 4 shows the determination and correlation coefficients for the hybrid DE–GBRT–based model fitted for the pressure drop per unit length in this article.

Table 4 - Coefficients of determination (R^2), correlation coefficients (r) and root mean square errors (RMSE) for the hybrid DE–GBRT–based model fitted in this study for the pressure drop per unit length.

According to these previous statistical calculations, the GBRT technique in combination with the DE optimization is an excellent model for estimating the pressure drop per unit length in granular filters, since the fitted GBRT model with DE has a coefficient of determination R^2 equals 0.77 and a correlation coefficient equals 0.88, respectively.

These coefficients are similar to those obtained by García-Nieto et al. (2017) using an ABC-MARS model, altough the RMSE was slightly smaller with the DE-GBRT model. So, these results show a trustworthy goodness of fit, that is to say, a good agreement is obtained between our model and the observed data.

An iMac with a processor 3.2 GHz Intel Core i5, with 8GB RAM and Maverick 10.9.5 as operating system was used to perform the computation. A time of 773.984 s, approximately 12 min, was necessary for the tuning and construction of the model.

The importance measure are relative and the addition of all the values for each criteria amounts to one. They are:

- *Gain*: it is computed taking into account each variable contribution to each tree that appears in the model.
- *Cover*: it is the relative number of observations of the variable in the model.
- *Frequency*: it is the relative number of times an independent variable appears in the trees of the obtained model.

The most significant measaure is *Gain* and thus it has been used to contruct the graph of the relative importance of the variables.

As an additional result of these calculations, the significance ranking for the three input variables predicting the pressure drop per unit length (output variable) in this complex study is shown in Table 5 and Fig. 2. Therefore, for the DE–GBRT model the most significant variable in pressure drop per unit length prediction is the Flow surface velocity, followed by Average grain size, and finally equivalent diameter.

Table 5 - Significance ranking for the variables involved in the best fitted DE–GBRT–based model for the pressure drop per unit length prediction $(\Delta p / \Delta L)$ according to criteria *Gain*, *Cover* and *Frequency*.

Fig. 2 - Relative importance of the input operation variables to predict the pressure drop per unit length $(\Delta p / \Delta L)$ in the fitted DE–GBRT–based model.

Bearing in mind that the flow surface velocity and average grain size are two variables easy to determine experimentally and the GBRT model indicates that they are the two most important variables, a *simplified* GBRT model was built using only these two variables. The curve predicted with this model is compared with the observed one in Fig. 3. The determination coefficient and correlation coefficient for this simplified model were 0.78 and 0.88, respectively.

In conclusion, this work was able to estimate the pressure drop per unit length in agreement with the actual experimental values observed using the DE–GBRT–based model with great accurateness as well as success. Therefore, it was appropriate to use a

GBRT model with a DE–based optimisation technique in order to achieve the best effective approach in this regression problem. Because these results agree with the outcome criterion of 'goodness of fit' (R^2) the DE–GBRT–based model was an excellent fit to the experimental.

Fig. 3 - Comparison between the pressure drop per unit length values observed and predicted, by type of filter, using the DE–GBRT–based simplified model ($R^2 = 0.78$).

Finally, the residual errors for each observation of the predicted model, calculated as the difference between the predicted and the observed pressure drop per unit length values, by type of filter, using the DE–GBRT–based simplified model, is represented in Fig. 4.

Fig. 4 – Residuals for the predicted pressure drop per unit length values, by type of filter, using the DE–GBRT–based simplified model.

4. Conclusions

Taking into account the experimental and numerical results, the main findings of this study can be summarised as follows:

- The new hybrid DE–GBRT–based model used in this work can accurately predict the pressure drop per unit length in different granular media used in sand filters without using as input variable the sphericity, which is a parameter difficult to obtain experimentally.

- A reasonable coefficient of determination equal to 0.78 was obtained when this hybrid DE–GBRT–based model was applied to the experimental pressure drop dataset.
- The significance order of the input variables involved in the prediction of the pressure drop per unit length in granular filters was set. This is one of the main findings in this work. Specifically, input variable Flow surface velocity could be considered the most influential parameter in the prediction of the pressure drop per unit length. In this regard, it is also important to highlight the influential role of the Average grain size in the dependent variable pressure drop per unit length.
- Taking into account the results of the previous point, and for practical reasons, as the two most important variables are relatively easy to obtain, a simplified GBRT model that used only these variables was developed with a comparatively very good coefficient of determination.
- The influence of the hyperparameters involved in the GBRT approach to predict pressure drop per unit length regression performance was established.
- The results verified that the hybrid DE–GBRT–based regression method significantly improved the generalisation capability achievable with only the GBRT–based regressor. Thus, input data from other filtered materials used in microirrigation can be processed to predict the pressure drop measuring only a few key variables.

In summary, this innovative methodology presented could be applied to other filtration processes with similar or distinct filter media types with success, but it is always necessary to take into account the characteristics of each filter and experiment. Consequently, an effective DE–GBRT–based model is a good practical solution to the problem of predicting the pressure drop in sand media filters that are used in microirrigation systems.

Acknowledgements

Authors wish to acknowledge the computational support provided by the Department of Mathematics at University of Oviedo as well as financial support of the Spanish Ministry of Economy and Competitiveness through Grant AGL2015-63750-R. Additionally, we would like to thank Anthony Ashworth for his revision of English grammar and spelling of the manuscript.

References

Arbat, G., Pujol, T., Puig-Bargués, J., Duran-Ros, M., Montoro, L., & Barragán, J. (2013). An experimental and analytical study to analyze hydraulic behaviour of nozzle-type underdrains in porous media filters. *Agricultural Water Management*, 126, 64–74.

Ardia, D., Mullen, K. M., Peterson, B. G., & Ulrich, J. (2016). DEoptim: Differential evolution in R. version 2.2-4.

Barreiros, F. M., Ferreira, P. J., & Figueirido, M. M. (1996). Calculating shape factors from particle sizing data. *Particle & Particle Systems Characterization*, 13(6), 368–373.

Bové, J., Arbat, G., Duran-Ros, M., Pujol, T., Velayos, J., Ramírez de Cartagena, F., & Puig-Bargués, J. (2015a). Pressure drop across sand and recycled glass media used in micro irrigation filters. *Biosystems Engineering*, 137, 55–63.

Bové, J., Arbat, G., Pujol, T., Duran-Ros, M., Ramírez de Cartagena, F., Velayos, J., & Puig-Bargués, J. (2015b). Reducing energy requirements for sand filtration in microirrigation: improving the underdrain and packing. *Biosystems Engineering*, 140, 67–78.

Breiman, L., Friedman, J.H., Olshen, R.A., & Ston, C.J. (1984). *Classification and regression trees*. Monterey, CA, USA: Wadsworth and Brooks/Cole.

Bühlmann, P., & Hothorn, T. (2007). Boosting algorithms: regularization, prediction and model fitting. *Statistical Science*, 22(4), 477–505.

Burt, C.M. (2010). Hydraulics of commercial sand media filter tanks used for agricultural drip irrigation. ITRC Report No. R 10001. San Luis Obispo, California: Irrigation Training and Research Center.

Chen, T., & Guestrin, C. (2016). XGBoost: a scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785–794). San Francisco, California, USA: ACM Publishers.

Chen, T., He, T., Benesty, M., Khotilovich, V., & Tang, Y. (2017). xgboost: Extreme gradient boosting. R package version 0.6-4. https://CRAN.R-project.org/package=xgboost.

Clark, G. A., Haman, D. Z., Prochaska, J. F. & Yitayew, M. (2007). General system design principles. In F.R.Lamm, F.R., J.E.Ayars, & F.S.Nakayama. (Eds.), Microirrigation for Crop Production (pp.161–220). Amsterdam: Elsevier.

Clerc, M. (2006). *Particle swarm optimization*. London, UK: Wiley-ISTE.

Döpke, J., Fritsche, U., Pierdzioch, C. (2017). Predicting recessions with boosted regression trees. *International Journal of Forecasting*, 33, 745–759.

Dorigo, M., & Stützle, T. (2004). *Ant colony optimization*. Cambridge, Massachusetts, USA: Bradford Publisher, The MIT Press.

Droste, R. L. (1997). *Theory and practice for water and wastewater treatment*. New York: John Wiley & Sons.

Duran-Ros, M., Puig-Bargués, J., Arbat, G., Barragán, J. & Ramírez de Cartagena, F. (2009). Performance and backwashing efficiency of disc and screen filters in microirrigation systems. *Biosystems Engineering*, 103(1), 35–42.

Eberhart, R. C., Shi, Y., & Kennedy, J. (2001). *Swarm intelligence*. San Francisco: Morgan Kaufmann.

Elbana, M., Ramírez de Cartagena, F., & Puig-Bargués, J. (2013). New mathematical model for computing head loss across sand media filter for microirrigation systems. *Irrigation Science*, 31(3), 343–349.

Feoktistov, V. (2006). *Differential evolution: in search of solutions*. New York: Springer.

Freedman, D., Pisani, R., & Purves, R. (2007). *Statistics*. New York: W.W. Norton & Company.

Friedman, J. H., Hastie, T., & Tibshirani, R. (2000). Additive logistic regression: A statistical view of boosting. *Annals of Statistics*, 28(2), 337–407.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29(5), 1189–1232.

Friedman, J. H. (2002). Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4), 367–378.

García-Nieto, P.J., García-Gonzalo, E., Bové, J., Arbat, G., BovéM. & Puig-Bargués, J. (2017). Modeling pressure drop produced by different filtering media in microirrigation sand filters using the hybrid ABC-MARS-based approach, MLP neural network and M5 model tree. *Computers and Electronics in Agriculture*, 139, 65–74.

Hastie, T., Tibshirani, R., & Friedman, J. H. (2003). *The elements of statistical learning*. New York: Springer-Verlag.

Horan, N., & Lowe, M. (2007). Full-scale trials of recycled glass as tertiary filter medium for wastewater treatment. *Water Resources*, 41(1), 253–259.

Johnson, N. E., Ianiuk, O., Cazap, D., Liu, L., Starobin, D., Dobler, G., & Ghandehari, M. (2017). Patterns of waste generation: A gradient boosting model for short-term waste prediction in New York City. *Waste Management*, 62, 3–11.

Lamm, F.R. & Rogers, D.H. (2017). Longevity and performance of a subsurface drip irrigation system. *Transactions of the ASABE*, 60(3), 931–939.

Landry, M., Erlinger, T. P., Patschke, D., & Varrichio, C. (2016). Probabilistic gradient boosting machines for GEFCom2014 wind forecasting. *International Journal of Forecasting*, 32(3), 1061–1066.

Mayr, A., Binder, H., Gefeller, O., & Schmid, M. (2014a). The evolution of boosting algorithms: From machine learning to statistical modelling. *Methods of Information in Medicine*, 6(1), 419–427.

Mayr, A., Binder, H., Gefeller, O., & Schmid, M. (2014b). Extending statistical boosting: An overview of recent methodological developments. *Methods of Information in Medicine*, 6(2), 428–435.

Mesquita, M., Testezlaf, R., & Ramirez, J.C.S. (2012). The effect of media bed characteristics and internal auxiliary elements on sand filter head loss. *Agricultural Water Management*, 115, 178–185.

Natekin, A., & Knoll, A. (2013). Gradient boosting machines, a tutorial. *Frontiers in Neurorobotics*, 7, 1–21

Nielsen, D. (2016). Tree Boosting with XGBoost. Master of Science in Physics and Mathematics. Norwegian University of Science and Technology.

Olsson, A. E. (2011). Particle swarm optimization: theory, techniques and applications. New York: Nova Science Publishers.

Persson, C., Bacher, P., Shiga, T., & Madsen, H. (2017). Multi-site solar power forecasting using gradient boosted regression trees. *Solar Energy*, 150, 423–436.

Picard, R., & Cook, D. (1984). Cross-validation of regression models. *Journal of the American Statistical Association*, 79(387), 575–583.

Price, K., Storn, R.M., & Lampinen, J.A. (2005). *Differential evolution: a practical approach to global optimization*. Berlin: Springer.

Ridgeway, G. (2007). Generalized boosted models: a guide to the GBM package. http://www.saedsayad.com/docs/gbm2.pdf. Accessed 3 August 2007

Ridgeway, G. (2017). gbm: Generalized boosted regression models. R package version 2.1.1. http://CRAN.R-project.org/package=gbm. Accessed 21 March 2017

Rocca, P., Oliveri, G., & Massa, A. (2011). Differential evolution as applied to electromagnetics. *IEEE Antennas and Propagation Magazine*, 53 (1), 38–49.

Simon, D. (2013). *Evolutionary optimization algorithms*. New York: Wiley.

Schapire, R. E. (2003). The boosting approach to machine learning an overview. In D. D. Denison, M. H. Hansen, C. C. Holmes, B. Mallick, & B. Yu (Eds.), *Nonlinear estimation and classification* (vol. 171, pp. 149–171), Lecture notes in statistics. Berlin: Springer.

Storn, R., & Price, K. (1997). Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11, 341–359.

Taieb, S. B., & Hyndman, R. J. (2014). A gradient boosting approach to the kaggle load forecasting competition. *International Journal of Forecasting*, 30(2), 382–394.

Vapnik, V. (1998). *Statistical learning theory*. New York: Wiley-Interscience.

Wu, W., Chen, W., Liu, H., Yin, S., Bao, Z., Niu, Y. (2014a). A dimensional analysis model for the calculation of head loss due to disc filters in drip irrigation systems. *Irrigation and Drainage*, 63(3), 349–358.

Wu, W., Chen, W., Liu, H., Yin, S., Niu, Y. (2014b). A new model for head loss assessment of screen filters developed with dimensional analysis in drip irrigation systems. *Irrigation and Drainage*, 63(4), 523–531.

Yang, X.–S., Cui, Z., Xiao, R., Gandomi, A.H., & Karamanoglu, M. (2013). *Swarm intelligence and bio-inspired computation: theory and applications*. London: Elsevier.

Yurdem, H., Demir, V., & Degirmencioglu, A. (2008). Development of a mathematical model to predict head losses from disc filters in drip irrigation systems using dimensional analysis. *Biosystems Engineering*, 100(1), 14–23.

Yurdem, H., Demir, V., & Degirmencioglu, A. (2010). Development of a mathematical model to predict clean water head losses in hydrocyclone filters in drip irrigation systems using dimensional analysis. *Biosystems Engineering*, 105(4), 495–506.

Zhong, Q., Zheng, T.; Liu, H., & Li, C. (2015). Development of head loss equations for self-cleaning screen filters in drip irrigation systems using dimensional analysis. *Biosystems Engineering*, 133, 116–127.

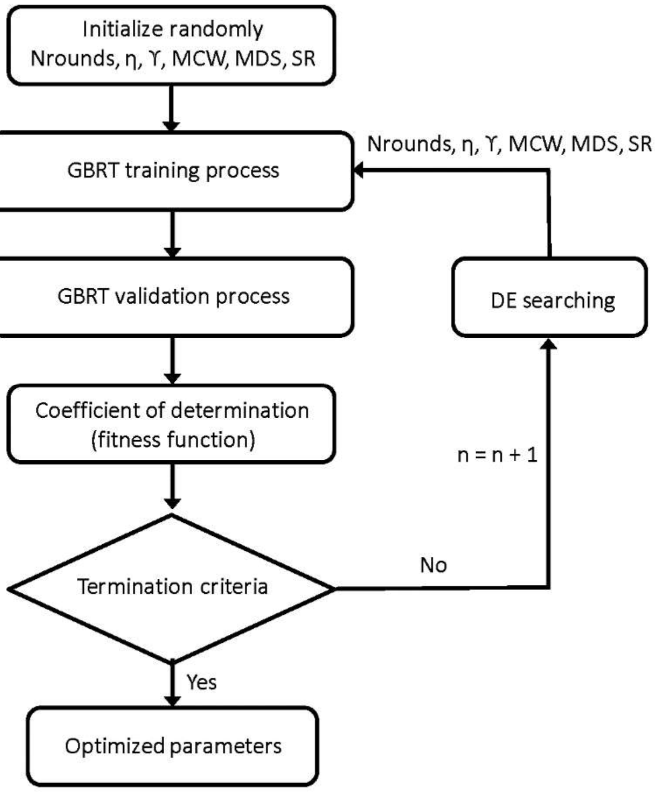

Fig. 1 - Flowchart of the new hybrid DE–GBRT–based model.

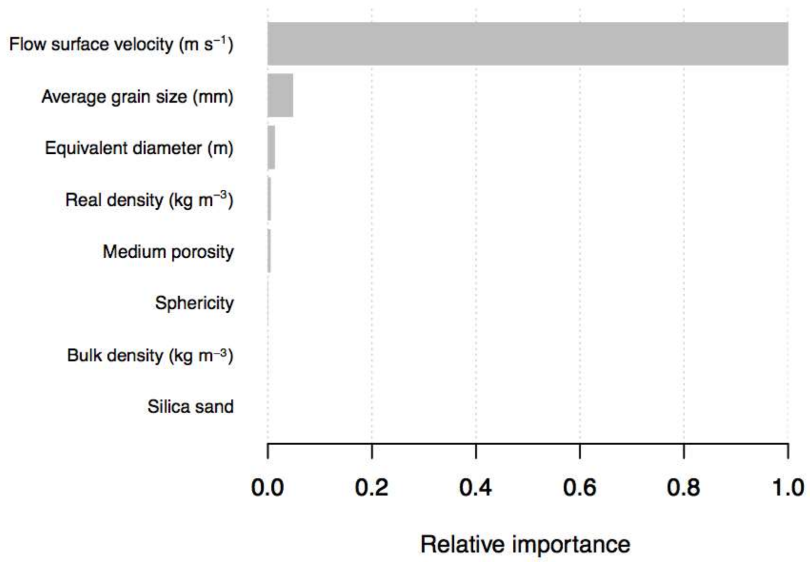

Fig. 2 - Relative importance of the input operation variables to predict the pressure drop per unit length $(\Delta p / \Delta L)$ in the fitted DE–GBRT–based model.

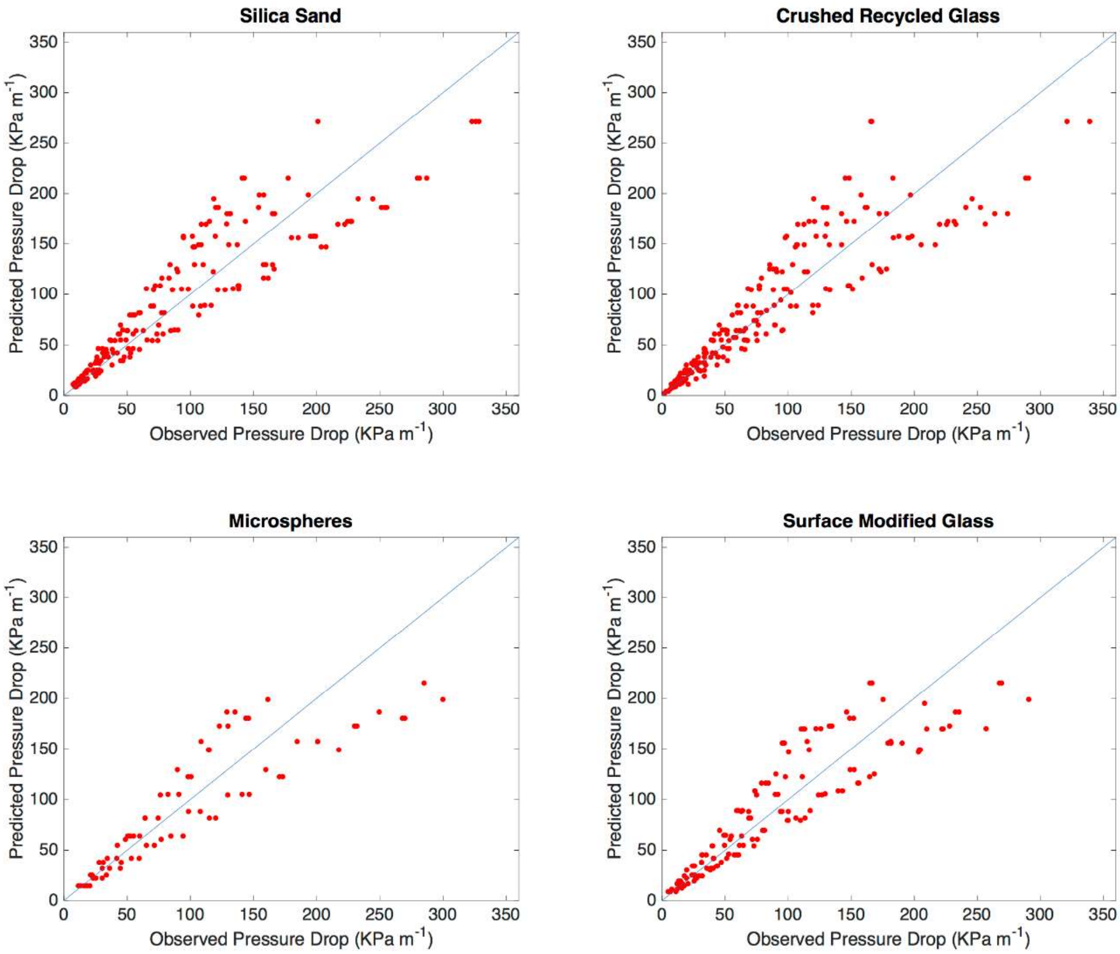

Fig. 3 - Comparison between the pressure drop per unit length values observed and predicted, by type of filter, using the DE–GBRT–based simplified model ($R^2 = 0.78$).

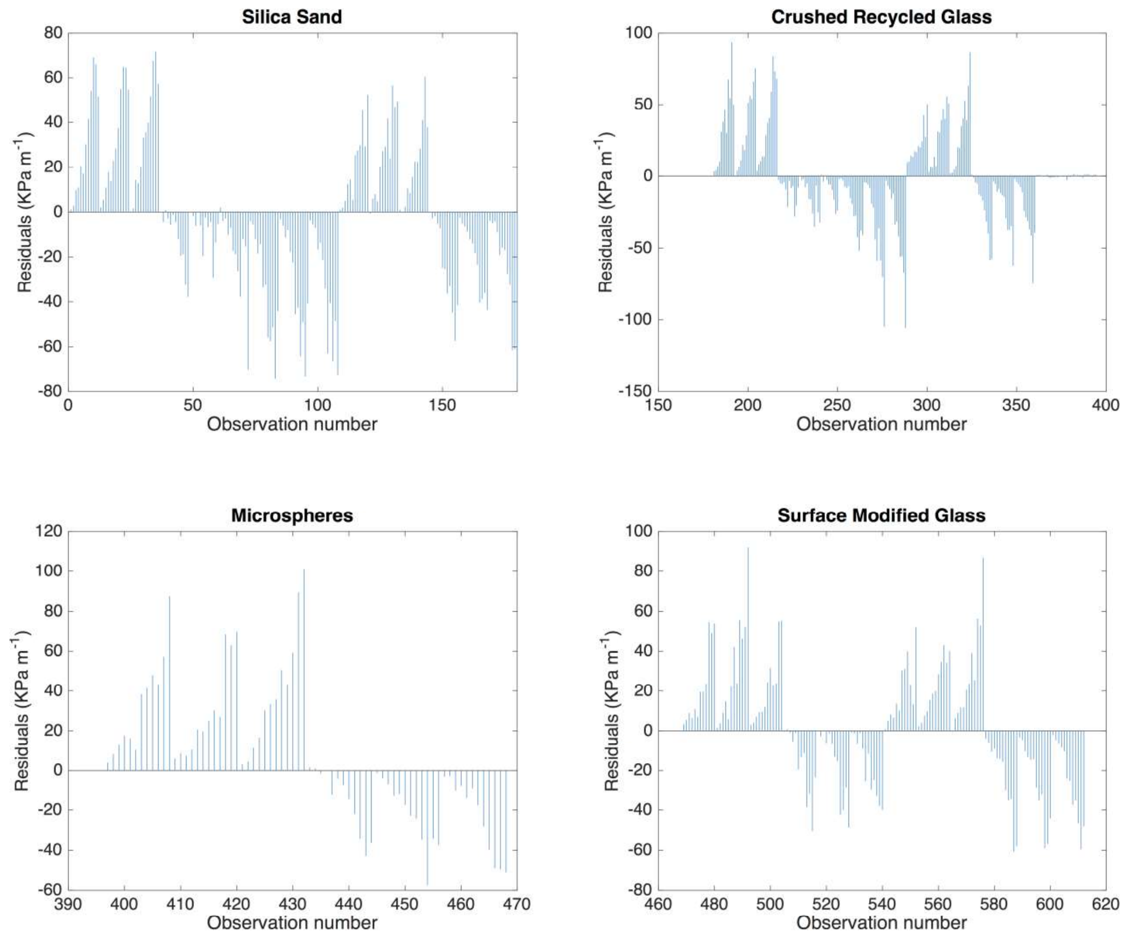

Fig. 4 – Residuals for the predicted pressure drop per unit length values, by type of filter, using the DE–GBRT–based simplified model.

Table 1 - Set of operation physical input variables used in this study and their names along with their mean and standard deviation.

Input variables	Name of the variable	Mean	Standard deviation
Filter media type	Filter_type	--	--
Bulk density (kg m^{-3})	Density_b	1397	88.77
Real density (kg m^{-3})	Density_r	2471	62.58
Medium porosity	Porosity	0.4324	0.03119
Equivalent diameter (m)	Diameter	0.7935	0.1761
Sphericity	Sphere	0.7637	0.1314
Flow surface velocity (m s^{-1})	Velocity	0.01421	0.006594
Average grain size (mm)	Grain_size	0.7576	0.1176

Table 2 - Search space for each of the GBRT parameters in the DE tuning process.

GBRT hyperparameters	Lower limit	Upper limit
Rounds	1	100
η	0.1	1
γ	0	30
Minimum child weight (MCW)	1	30
Maximum Δ step (MDS)	0	30
Subsample ratio	0.5	1

Table 3 - Optimal hyperparameters of the best fitted GBRT model found with the DE technique.

GBRT hyperparameters	Optimal values
Rounds	60
η	0.51
γ	0.17
Minimum child weight (*MCW*)	4.7
Maximum Δ step (*MDS*)	15
Subsample ratio (*SR*)	0.87

Table 4 - Coefficients of determination (R^2), correlation coefficients (*r*) and root mean square errors (RMSE) for the hybrid DE–GBRT–based model fitted in this study for the pressure drop per unit length.

Model	Coef. of determination (R^2)/correlation coef. (*r*)	RMSE
DE–GBRT	0.7741/0.8798	30.15

Table 5 - Significance ranking for the variables involved in the best fitted DE–GBRT–based model for the pressure drop per unit length prediction $(\Delta p / \Delta L)$ according to criteria Gain, Cover and Frequency.

Input variable	*Gain*	*Cover*	*Frequency*
Flow surface velocity	9.32×10^{-1}	0.7025	0.5593
Average grain size	4.50×10^{-2}	0.0606	0.1282
Equivalent diameter	1.22×10^{-2}	0.0643	0.0459
Real density	4.96×10^{-3}	0.0658	0.0763
Medium porosity	4.66×10^{-3}	0.0603	0.1467
Sphericity	5.79×10^{-4}	0.0266	0.0280
Bulk density	1.41×10^{-4}	0.0176	0.0137
Silica sand	3.11×10^{-6}	0.0021	0.0017