Multi-Objective Evolutionary Optimization for Time Series Lag Regression

Fernando Jiménez$^{1[0000-0001-9906-4132]}$,
Joanna Kamińska$^{2[0000-0002-0157-516X]}$,
Estrella Lucena-Sánchez$^{3,4[0000-0001-9312-1175]}$,
José Palma$^{1[0000-0003-2502-4378]}$, and
Guido Sciavicco$^{3[0000-0002-9221-879X]}$

[1] Dept. of Knowledge Engeneering and Communications,
University of Murcia (Spain)
jtpalma|fernan@um.es
[2] Dept. of Mathematics,
Wrocław University of Environmental and Life Sciences (Poland)
joanna.kaminska@upwr.edu.pl
[3] Dept. of Mathematics and Computer Science,
University of Ferrara (Italy)
estrella.lucenasanchez|guido.sciavicco@unife.it
[4] Dept. of Physics, Informatics, and Mathematics
University of Modena e Reggio Emilia (Italy)

Abstract. It is well-known that in some regression problems the effect of an independent variables on the dependent one(s) may be delayed; this phenomenon is known as lag. Lag regression is one of the standard techniques for time series explanation and prediction. However, using lagged variables to transform a multivariate time series so that a propositional algorithm such as a linear regression learner can be used requires to decide, at preprocessing time, which independent variables must be lagged and by how much. In this paper, we propose a novel optimization schema to solve this problem. We test our solution, implemented with a multi-objective evolutionary algorithm, on real data taken from a larger project that aims to construct an explanation model for the study of atmospheric pollution in the city of Wrocław (Poland).

Keywords: Regression; Lag; Multi-objective evolutionary computation; Time series explanation

1 Introduction

A *time series* is a series of data points labelled with a temporal stamp. If each data point contains a single time-dependent value, then the time series is *univariate*; otherwise, it is called *multivariate*. Time series arise in multiple contexts, for example, medical patients, who can be considered as time series in which every interesting medical value varies over time (e.g., fever, pain level, blood pressure), or environmental monitoring stations, which can also be considered time series, in which atmospheric values change over time (e.g., pressure, concentration of chemicals).

There are two main problems associated with single time series: *time series explanation* and *time series forecasting*; explanation is a necessary step for forecasting, but the latter does not necessarily follow the former in every application and context. Explaining a time series aims to construct a (possibly interpretable) model that explains the present values; forecasting a time series implies testing and using the model to predict future values. In the univariate case, a model of a time series is based uniquely on the values of the series itself. For example, a forecasting model for the stock price of a certain company would allow one to predict the future price (e.g., in the next two days) based on the prices of the same company (e.g., the price in each day of the past week). The simplest univariate forecasting approach is commonly known as *Simple Moving Average* (SMA) model: in essence, a simple moving average is calculated over the time series by considering its last n values, used to perform a smoothing process of the series, and then used to forecast for the next value. Although such an approach has some clear limitations, it is still useful to establish a baseline, against which to compare more complex solutions [3]. Based on the observation that the most recent values may be more indicative of a future trend than older ones, *Simple Exponential Smoothing* (SES) models consider a weighted average over the last n observations, assigning exponentially decreasing weights as they get older [3]. Other than this first, simple type of smoothing, it is also worth mentioning *Holt's Exponential Smoothing* (HES) models [9], which can consider an increasing or decreasing trend in the time series, and *Holt-Winters' Exponential Smoothing* (HWES) [14] models, that can take into account seasonality effects. Technically, exponential smoothing belongs to the broader *AutoRegressive Integrated Moving Average* (ARIMA) family [11], which includes models that can be fitted to time series data either to better understand the data itself or to predict future points in the series, when it shows evidence of non-stationarity. Specifically, the methods that are capable of dealing with periodical variations in the time series fall under the umbrella of *Seasonal ARIMA* [3]. Relevant to this study is also the algorithm presented in [1], in which a multi-objective evolutionary method is employed for the optimization of the parameters of an ARIMA-like model. The common aspect among all univariate models is that they make a prediction based on a weighted linear sum of recent past observations; in the multivariate case, instead, one identifies one dependent variable (time series), and aims to construct a model to explain and/or predict its future values based on the past and present values of other, independent variables (which themselves are time series): this is usually done with *lagged* models. While ARIMA-type models emerge from computational statistics, lagged models belong to the machine learning domain, and, in general, they consist of creating *lagged* version of (a subset of) the independent variable to construct a larger data set that is then used to create a model of the dependent time series using classical, propositional algorithms (such as, for example, linear regression). Among the available packages to this purpose we mention WEKA's *timeseriesForecasting* [7]. Other approaches to multivariate time series modelling include *Recurrent Neural Networks* (RNNs)[8], which have been used for time series forecasting with promising results, but at the expenses of the interpretability of the resulting model; in some recent works, neural networks for time series forecasting have been trained and optimized with multi-objective evolutionary algorithms. Autoregressive techniques can be combined with lagged methodologies; in the simplest case, it is sufficient to create, in a lagged extended data set, one or more lagged version(s) the dependent variable as well, whose values are combined with those of the independent ones.

The main limitation of multivariate lagged models is precisely the choice of lag variables and lag amounts. In some cases, it is difficult to foresee the necessary lag amount. Moreover, uncontrolled lag variable creation may lead to very large data bases which, when treated with propositional algorithms, may lead to poorer results, as unnecessary lag variables become noise. Finally, even if lagged variables increase the quality of the result, the obtained function may not be easy to interpret. In this work we present a very simple optimization schema that avoids the above problems for time series explanation using regression. The distinctive characteristics of our method are: *(i)* it is a *wrapper* algorithm based on well-known and easy-to-implement components, *(ii)* it may use any *black box* regression algorithm, and *(iii)* it includes an intrinsic feature selection mechanism. Our algorithm is an instantiation of the more general *dynamic preprocessing* mechanism, which generalizes the concept of wrapper by allowing the (possibly simultaneous) optimization of several aspects of data.

We test our model on a real data set taken from a larger project that aims to construct an explanation and prediction model for the study of atmospheric pollution in the city of Wrocław (Poland).

2 Lag Regression

2.1 Mathematical Formulation

Regression is a common statistical data analysis technique, used to determine the extent to which there is a mathematical relationship between a dependent variable and one or more independent variables, and its applications range from biology, to agriculture, to food and water resources optimization (see, e.g. [2, 12, 13]. Regression can be *univariate*, when there is only one independent variable, or *multivariate*, otherwise. Moreover, regression is usually *linear*, that is, it is usually the case that we search for a linear relationship; it becomes *non-linear*, when we search for any function (whose form is unknown) that links the independent variable(s) and the dependent one. Linear regression is not only the most common type, but it is also the one that presents the clearest mathematical formalization. In the following, and in our experiments as well, we assume that the relations that we search for are, in fact, linear; the entire optimization model, however, works for any type of regression.

Given a data set A with n independent variables $A_1, \ldots, A_n$ and one observed variable B, solving a linear regression problem consists of finding a vector $\bar{c} = (c_0, c_1, \ldots, c_n)$ of $n+1$ *parameters* (or *coefficients*) so that the equation:

$$B = c_0 + \sum_{i=1}^{n} c_i \cdot A_i + \epsilon, \tag{1}$$

where ϵ is a random value, is satisfied. Starting from a data set of observations:

$$A = \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} & b_1 \\ a_{21} & a_{22} & \ldots & a_{2n} & b_2 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ a_{m1} & a_{m2} & \ldots & a_{mn} & b_m \end{bmatrix} \tag{2}$$

the regression problem is usually solved by suitably estimating $\bar{c}$ so that, for each $1 \leq j \leq m$:

$$b_j \approx c_0 + \sum_{i=1}^{n} c_i \cdot a_{ij} + \epsilon. \tag{3}$$

The performance of such an estimation can be measured in several (standard) ways, such as *correlation, covariance, mean squared error*, among others. When A is a multivariate time series, composed by n independent and one dependent time series, then data are temporally ordered and associated to a timestamp:

$$A = \begin{bmatrix} t_1 & a_{11} & a_{12} & \ldots & a_{1n} & b_1 \\ t_2 & a_{21} & a_{22} & \ldots & a_{2n} & b_2 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ t_m & a_{m1} & a_{m2} & \ldots & a_{mn} & b_m \end{bmatrix} \tag{4}$$

Using linear regression to explain B, then, entails that, instead of (1), we are finding optimal coefficients for:

$$B(t) = c_0 + \sum_{i=1}^{n} c_i \cdot A_i(t) + \epsilon, \tag{5}$$

because we aim to explain B at a certain point in time t using the values $A_1(t), \ldots, A_n(t)$.

Lag (linear) regression consists of solving a more general equation, whose formulation is:

$$B(t) = c_0 + \sum_{i=1}^{n} \sum_{k=0}^{p_i} c_{i,k} \cdot A_i(t-k) + \epsilon. \tag{6}$$

In other words, we use the value of each independent variable A_i not only at time t, but also at time $t-1, t-2, \ldots, t-p_i$, to explain B at time t; each $A_i(t-k)$ is associated to a coefficient $c_{i,k}$, which must be estimated, along with each m_i. We work under the additional assumption that, for each i, there is precisely one lag k, denoted k_i, such that $A_i(t-k_i)$ influences the output more than any other lag. Our purpose is to devise an optimization schema that allows one to estimate both the value k_i and the coefficient c_i that corresponds to it, to obtain the best solution to the following, simpler, equation:

$$B(t) = c_0 + \sum_{i=1}^{n} c_i \cdot A_i(t-k_i) + \epsilon. \tag{7}$$

2.2 Applications Scenarios

Multivariate time series emerge in many real contexts. Consider, for example, the medical context. Each patient can be described, during the observation period, by collecting all relevant numerical values of his/her indicators: blood pressure, temperature, body weight, amount of all drugs that are administered to him/her, and so on. In this way, a patient becomes a multivariate time series. Now, if we identify one particular variable of interest (e.g., the temperature), we can approach the problem of explaining its behaviour using the values of the other variables, as in (5). Intuitively, however, changes in values (such as the amount of a certain drug that it is administered) may have a delayed effect on the temperature; thus, it is possible that the behaviour of the temperature is, in actuality, better explained by an instance of (6).

As a different example, consider an environmental study scenario. In it, we have a number of observation points, let us say underground water wells, from which, at given times, water samples are extracted. Each sample is analyzed from the chemical-physical point of view, and the amount of interesting elements is registered. Since each observation point is sampled many times during the observation period, it may be seen as a multivariate time series. As before, one particular characteristics of the samples may be of interest, for example the amount of some pollutant, and we may want to search, if it exists, for the mathematical relationship that links the amount of pollutant to the amount of the other values of each sample, possibly towards a geological explanation of its presence. In some cases, the presence of chemical elements in the water has a delayed effect on the concentration of pollutant(s), so that such a mathematical relation may be modelled by an instance of (7).

3 An Optimization Model for Lag Regression

A *multi-objective optimization problem* (see, e.g. [4]) can be formally defined as the optimization problem of simultaneously minimizing (or maximizing) a set of k arbitrary functions:

$$\begin{cases} \min / \max \;\; f_1(\bar{x}) \\ \min / \max \;\; f_2(\bar{x}) \\ \ldots \\ \min / \max \;\; f_k(\bar{x}), \end{cases} \tag{8}$$

where $\bar{x}$ is a vector of decision variables. A multi-objective optimization problem can be *continuous* or *discrete* (*combinatorial*). In combinatorial problems, 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, so that it is sufficient to consider one type only. A set $\mathcal{F}$ of solutions is *non dominated* (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 $(1 \leq i \leq n)$ that $f_i(\bar{y})$ improves $f_i(\bar{x})$, and *(ii)* for every j, $(1 \leq j \leq n, j \neq i)$, $f_j(\bar{x})$ does not improve $f_i(\bar{y})$. 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 *non-dominated* if and only if there is not other solution that dominates it. The set of non dominated solutions from $\mathcal{F}$ is called *Pareto front*.

Consider, as before, a multi-variate time series $A_1(t), \ldots, A_n(t), B(t)$ with m distinct observations, and a vector $\bar{x} = (x_1, \ldots, x_n)$ of decision variables with domain $[0, \ldots, m]$. Let M be the maximum of $\bar{x}$ (called *maximum lag* of $\bar{x}$). The vector $\bar{x}$ entails a lag transformation of (4) into a new data set with $m - M$ observations, in which the feature (time series) A_i is lagged (i.e., delayed) of the amount x_i:

$$A(\bar{x}) = \begin{bmatrix} t_M & a_{(M-x_1)1} & a_{(M-x_2)2} & \cdots & a_{(M-x_n)n} & b_M \\ t_{M+1} & a_{(M+1)-x_1)1} & a_{((M+1)-x_1)2} & \cdots & a_{((M+1)-x_1)n} & b_{M+1} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ t_{m-M} & a_{((m-M)-x_1)1} & a_{((m-M)-x_1)2} & \cdots & a_{((m-M)-x_1)n} & b_{m-M} \end{bmatrix} \tag{9}$$

The resulting data set can be used to train a classical linear regression algorithm, with the effect of learning a model as in (6). This model can be used to explain the time series $B(t)$; a more complex mechanism would be required to optimize the coefficients in order to perform forecasting, also. Let $f_1(\bar{x})$ in (8) be any performance measure of the learned model after the transformation $\bar{x}$; depending on the particular application, we can instantiate $f_2, f_3, \ldots$ as necessary, in order to optimize not only the performance of the model but also any other characteristics. For example, we can slightly improve our original formulation by allowing each x_i to take values in $[-1, 0, \ldots, m]$, and interpret $x_i = -1$ as discarding completely the i-th column (so to embed a feature selection mechanism). In this case we can instantiate $f_2()$ as:

$$CARD(\bar{x}) = \sum_{i=1}^{n} \begin{cases} 0 \text{ if } x_i \neq -1 \\ 1 \text{ otherwise} \end{cases} \tag{10}$$

In this case (8) becomes:

$$\begin{cases} \min / \max \; f_1(\bar{x}) \\ \min \; CARD(\bar{x}) \end{cases} \tag{11}$$

It is worth observing that (11) could be improved by letting $\bar{x}$ spanning over B as well (in that case, the value -1 would be forbidden in B, obviously). Solving the problem in this version, would entail searching for a linear equation similar to (6), but with the addition of terms of the type $B(t-k)$ $(k \geq 1)$, in the spirit of auto regressive models. The main drawback of such a choice is the reduced interpretability of the resulting explanation model, which would include past values of the independent variable as part of the explanation of the current one. For this reason, in this first proposal we did not include this feature.

4 Implementation and Test

4.1 Evolutionary Algorithms

Multi-objective evolutionary algorithms are known to be particularly suitable to perform multi-objective optimization, as they search for multiple optimal solutions in parallel. In this experiment, in order to solve (11) we have chosen the well-known NSGA-II (Non-dominated Sorted Genetic Algorithm) [5] algorithm, which is available as open-source from the suite *jMetal* [6]. NGSA-II is an elitist Pareto-based multi-objective evolutionary algorithm that employs a strategy with a binary tournament selection and a rank-crowding better function, where the rank of an individual in a population is the non-domination level of the individual in the whole population. As black box linear regression algorithm, we used the class *linearRegression* from the open-source learning suite *Weka* [15], run in 5-fold *cross-validation* mode, with standard parameters and no embedded feature selection. We have represented each individual solution $\bar{x}$ as an array:

$$x_1, x_2, \ldots, x_n$$

with values in $[-1, \ldots, m]$, where m is the number of observations of the data set. As performance measure for the underlying linear regression algorithm we used:

$$f_1(\bar{x}) = 1 - |CORR(\bar{x}, \bar{y}, \bar{z})|$$

where $CORR$ 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 indicated by $\bar{x}$, as explained in the previous section. The correlation varies between -1 (perfect negative correlation) to 1 (perfect positive correlation), being 0 the value that represents no correlation at all. Thus, we have designed the evolutionary computation to optimize the correlation only. We have used the standard mutation and crossover operations (suitably adapted to correctly deal with our solution representation), with probabilities (tuned with an initial experiment) of 0.3 and 0.7, respectively. Our population is composed by 100 individuals; we have set the algorithm for a total of 1000 evaluations in a single execution, and launched 5 independent executions.

Table 1. Features in the original data set.

feature	**description**
air_temp	hourly recording of the air temperature
solar_rad	hourly amount of solar irradiation
wind_speed	hourly recording of the wind speed
rel_humidity	hourly recording of the relative air humidity
air_pressure	hourly recording of the air pressure
traffic	hourly sum of vehicle numbers at the considered intersection
NO2_conc	hourly recording of the NO_2 concentration level
NOX_conc	hourly recording of the NO_X concentration level
PM25_conc	hourly recording of the $PM_{2.5}$ concentration level

Table 2. Results of the experiment.

correlation coefficient				**lags**
original (c.v.)	**lagged (c.v)**	**optim. (c.v)**	**optim. (test)**	
0.6250	0.7749	0.7180	0.7378	14,7,2,10,0,0
0.6251	0.7752	0.7184	0.7382	14,0,2,8,10,0
0.6250	0.7752	0.7187	0.7297	21,5,2,9,23,0
0.6251	0.7748	0.7208	0.7363	20,0,2,7,19,0
0.6252	0.7750	0.7039	0.7243	21,0,3,8,7,0

4.2 Data Origin and Preparation

The first environmental study that relates air pollution and meteorological variables and traffic conditions in Wrocław (Poland) is presented in [10]. The overall goal of the study was determining how the levels of specific pollutants, namely, NO_2, NO_X, and $PM_{2.5}$, are related to the values of other attributes, such as weather conditions and traffic intensity, with the purpose of building an explanation model. In it, the value of a pollutant at a certain time instant is linked to the value of the predictor attributes from the same time instant. In [10], a non-linear, non-interpretable, atemporal model has been used; the fitting ability of a non-interpretable model compensates, partially, for not using the historical values of the predictor, giving rise to a relatively good explanation model. The considered data set spans over the years 2015–2017, and it records information at one-hour granularity. The structure of reduced data set, obtained from the original one after eliminating the explicit temporal attributes (by interpreting data as a time series, the notion of time becomes implicit), and the categorical ones, can be seen in Tab. 1. The attribute *traffic* refers to the number of vehicle crossings recorded at a large intersection equipped with a traffic flow measurement system. The air quality information has been recorded by a nearby measurement station.

In this experiment we considered only one pollutant, namely NO_2, and we interpreted the data as a multivariate time series. For efficiency reasons, we

Table 3. Coefficients of the linear functions (best individuals).

air_temp	**solar_rad**	**wind_speed**	**rel_humidity**	**air_pressure**	**traffic**
-0.4037	12.6468	-4.5834	-0.3840	-0.0844	0.0083
-0.2362	-6.3024	-4.5848	-0.4956	0.0615	0.0089
-0.3769	9.9758	-4.5378	-0.4206	0.067	0.0085
-0.1853	-6.3731	-4.6274	-0.4733	0.1049	0.0087
-0.2011	-7.0963	-4.2235	-0.5036	0.0502	0.0093

considered only the 10% of the entire data set, and we have split it into a training and test set, operating the optimization on the former one only. We have used the training set in all 5 executions of the optimization model (11), and selected the best (in terms of correlation) element from each final population. Our maximum lag allowed in the optimization model is 24 hours.

4.3 Results

The first reference result is the correlation that can be obtained by training a *linearRegression* model on the training data with standard parameters and no embedded feature selection, and executing on the test data: 0.6652. Also, we consider the correlation coefficients on the training data only, in 5 experiments, varying the seed (1 to 5), in 5-folds cross validation, again in the original configuration, as shown in Tab. 2, leftmost column. Even if our data are temporal, learning (as base reference) an atemporal model (such as (5)) makes sense in some problems. Indeed it may be the case that the delayed effect of the independent variables on the dependent one falls below the temporal granularity of the data (for us, one hour), and that, at the same time, the dependent variable presents a quasi-constant behaviour in such a small interval. Should that be the case, a model such as (5) would have a relatively high performance (that is, it would be an acceptable approximation of the physical reality); in our case this is not true, which justifies the resort to temporal lag regression.

The second reference results emerges from creating a lagged version of the data set in the standard way, using WEKA's *TSLagMaker*. Because of the dimensions of the problem, we created a lagged version of each variable (excluding the class) up to 12 hours only, for a total of 79 attributes. The correlation coefficient that resulted from training a *linearRegression* model on the lagged version of the training set, and executing it on the lagged version of the test set is 0.8066, which is quite high. Unfortunately, observing the resulting model, the interpretation limits of this technique emerge clearly. For example, the resulting function shows a positive factor for the temperature at the same time, 4,5,6,7,8, and 10 hours before the observation, but negative for the temperature 1,2,3,9,11, and 12 hours before the observation. A similar behaviour is shown in almost all other variables. This makes it very hard to identify, if it exists, a cause-effect phenomenon, on top of the fact that the expert should be able to interpret a 79-variables linear function to extract a meaningful environmental model. An intermediate step of feature selection does not solve the interpretation problem; as a matter of fact, the effect of feature selection is that of selecting the best features (with an absolute measure, in the case of filters, and relatively to a learning task, in the case of wrappers), and, again, selecting, for example, the temperature at 4, 6, and 10 hours before the observation would make it very hard to construct a concrete explanation model. The results in cross-validation of the training data only, in the same conditions as before, are shown in Tab. 2, second column.

In Tab. 2, third column we can see the correlation coefficient of the best individual for each of the five execution, in cross-validation mode (that is, on training data only). Compared with those in the first column, it is possible to appreciate an improvement of about 8 points, in average. When each best individual is executed on the test set, we obtain the results shown in the fourth column, which again, compared with the original training-test experiment, show an average improvement of about 7 points. The loss in correlation coefficient of these individuals with respect to the extended (lagged) version of the data set is compensated by the intrinsic greater interpretability of the former over the latter.

4.4 Discussion

Observe, first, the coefficients of the linear functions that correspond to each individual (Tab. 3): as for five variables out of six, the coefficients present the same sign and a very similar module across the individuals (this is an indication that our proposed models are stable), and when both positive and negative coefficient appear (that is, in the variable that measures the solar radiation), the change coincides with a change in the amount of lag, maybe indicating two different physical processes.

Let us focus now on the chosen lags in each individual. Observe, to start with, that the variable that measures the hourly traffic has always lag zero: in other words, all models coincide that the amount of NO_2 is influenced by the amount of (car) traffic with no delay. This could be explained by the small distance between the point of pollution concentration measurement and the intersection where the main emission source (the cars) is located. Similarly, four out of five models agree that the speed of the wind influences the amount of pollutant with two hours of delay. This may be due to the distance between the meteorological station and the intersection. In Wrocław, North-West winds prevails; therefore, the wind generally blows from the meteorological station towards the intersection. The distance, in a straight line, is about 10km and the average wind speed is 3m/s. Taking into account the porosity of the city development area and the time needed to evacuate pollution from the built-up area around the intersection, a delay of about 2 hours in the reaction of pollution concentration to the measured wind speed is reasonable. Moreover, observe that in three out of five models the lag for the solar radiation is zero, with negative coefficient in the corresponding equation, while the remaining two is between 5 and 7, with positive coefficient. This opens the possibility of two different explanation models: for negative correlation (increasing solar radiation corresponding to a decrease in NO_2 with no delay), the physical process may be related to an intensification of photochemical reactions, while positive correlations take place with 5 to 7 hours of delay, and may indicate the reverse process.

In conclusion, our learning model produces individuals that are easier to interpret, because they identify the most relevant delays for the explanation task, so that devising a meaningful environmental model becomes possible.

5 Conclusions

In this paper we have proposed, and tested, a novel optimization model for temporal lag regression. Lag variables can be very important for the task of single multivariate time series explanation and prediction, as they allow a model to take into account possible delays in the effect of an independent variable on the dependent one. The standard approach for lag variable using consists of creating predetermined lagged artificial variables, and then using standard learning techniques on the obtained, extended data set; in a sense, this can be seen as a *brute force* approach. We proposed in this work an optimization model in which the amount of lag for each variable is decided dynamically, and we implemented it with a multi-objective evolutionary algorithm. Our learning model, that implicitly includes a feature selection mechanism, chooses the best lag for each variable, effectively providing a more interpretable, yet accurate enough, explanation model for a multivariate time series. Our schema, with minimal adaptations, can be used for multivariate time series forecasting as well. We tested our model on real data taken from a larger project that aims to construct an explanation model for the study of atmospheric pollution in the city of Wrocław (Poland).

Our model can be improved in several ways. In certain applications, for example, the same independent variable can influence the dependent one with a prolonged delay that spans more than one observation. A possible generalization, therefore, would aim to optimize the number of consecutive observations to take into account, and their algebraic combinations.

Acknowledgments

Estrella Lucena-Sánchez and Guido Sciavicco acknowledge the partial support by the project *Artificial Intelligence for Improving the Exploitation of Water and Food Resources*, founded by the University of Ferrara. Estrella Lucena-Sánchez acknowledges, moreover, the support by the Emilia-Romagna (Italy) regional project *New Mathematical and Computer Science Methods for the Water and Food Resources Exploitation Optimization.* Finally, this work was also partially funded by the Spanish Ministry of Science, Innovation and Universities under the SITUS project (Ref: RTI2018-094832-B-I00), and by the European Fund for Regional Development (EFRD, FEDER).

References

1. Al-Douri, Y., AL-Chalabi, H., Lundberg, J.: Time series forecasting using a two-level multi-objective genetic algorithm: A case study of maintenance cost data for tunnel fans. Algorithms **11**, 1–19 (2018)
2. Bijan, P.: Some applications of nonlinear regression models in forestry research. The Forestry Chronicle **59**(5), 244–248 (1983)
3. Box, G., Jenkins, G., Reinsel, G., Ljung, G.: Time Series Analysis: Forecasting and Control. Wiley (2016)
4. Collette, Y., Siarry, P.: Multiobjective Optimization: Principles and Case Studies. Springer Berlin Heidelberg (2004)
5. Deb, K.: Multi-objective optimization using evolutionary algorithms. Wiley, London, UK (2001)
6. Durillo, J., Nebro, A.: jMetal: a Java framework for multi-objective optimization. Avances in Engineering Software **42**, 760 – 771 (2011)
7. Hall, M.: Time series analysis and forecasting with WEKA (2014), *https://wiki.pentaho.com*, last accessed: May, 2019
8. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation **9**(8), 1735—-1780 (1997)
9. Holt, C.: Forecasting seasonals and trends by exponentially weighted moving averages. International Journal of Forecasting **20**(1), 5–10 (2004)
10. Kamińska, J.: The use of random forests in modelling short-term air pollution effects based on traffic and meteorological conditions: A case study in Wrocław. Journal of environmental management **217**, 164–174 (2018)
11. Poulos, L., Kvanli, A., Pavur, R.: A comparison of the accuracy of the box-jenkins method with that of automated forecasting methods. International Journal of Forecasting **3**, 261–267 (1987)
12. Salimi, A., Rostami, J., Moormann, C., Delisio, A.: Application of non-linear regression analysis and artificial intelligence algorithms for performance prediction of hard rock tbms. Tunnelling and Underground Space Technology **58**, 236 – 246 (2016)
13. S.V Archontoulis, S., Miguez, F.: Nonlinear regression models and applications in agricultural research. Agronomy Journal **107**(2), 786 – 798 (2013)
14. Winters, P.: Forecasting sales by exponentially weighted moving averages. Management Science **3**(6), 324–342 (1960)
15. Witten, I., Frank, E., Hall, M.: Data mining: practical machine learning tools and techniques, 3rd Edition. Morgan Kaufmann, Elsevier (2011)