mathematics

MDPI

Article

Predicting Benzene Concentration Using Machine Learning and Time Series Algorithms

Luis Alfonso Menéndez García [1], Fernando Sánchez Lasheras [2], Paulino José García Nieto [2], Laura Álvarez de Prado [1] and Antonio Bernardo Sánchez [1,*]

[1] Department of Mining Technology, Topography and Structures. Higher and Technical School of Mining Engineering, University of León, Campus de Vegazana s/n, 24071 León, Spain; lmeneg00@estudiantes.unileon.es (L.A.M.G.); laura.alvarez@unileon.es (L.Á.d.P.)

[2] Department of Mathematics, Faculty of Sciences, University of Oviedo, 33007 Oviedo, Spain; sanchezfernando@uniovi.es (F.S.L.); pjgarcia@uniovi.es (P.J.G.N.)

* Correspondence: antonio.bernardo@unileon.es; Tel.: +34-987-291-951

Received: 2 November 2020; Accepted: 8 December 2020; Published: 11 December 2020

Abstract: Benzene is a pollutant which is very harmful to our health, so models are necessary to predict its concentration and relationship with other air pollutants. The data collected by eight stations in Madrid (Spain) over nine years were analyzed using the following regression-based machine learning models: multivariate linear regression (MLR), multivariate adaptive regression splines (MARS), multilayer perceptron neural network (MLP), support vector machines (SVM), autoregressive integrated moving-average (ARIMA) and vector autoregressive moving-average (VARMA) models. Benzene concentration predictions were made from the concentration of four environmental pollutants: nitrogen dioxide (NO_2), nitrogen oxides (NO_x), particulate matter (PM_{10}) and toluene (C_7H_8), and the performance measures of the model were studied from the proposed models. In general, regression-based machine learning models are more effective at predicting than time series models.

Keywords: benzene; forecasting; air pollutant; multivariate adaptive regression splines (MLR); multivariate adaptive regression splines (MARS); multilayer perceptron neural network (MLP); support vector machines (SVM); autoregressive integrated moving-average (ARIMA); vector autoregressive moving-average (VARMA)

1. Introduction

Volatile organic compounds (VOCs) are compounds with very high volatility which are in a gaseous state under normal circumstances. They include a variety of chemical species among which are benzene, toluene, ethylbenzene and xylene [1] and are known as BETX, all of them are air pollutants, along with other compounds such us SO_2, NO_2, NO_x, PM_{10}, $PM_{2.5}$ or CO [2].

Benzene is a hydrocarbon with a ring structure made up of six carbon atoms. It is one of the products that is most used as a raw material in industrial processes in the organic chemical industry. Its usual state is liquid, highly flammable, and colorless, although it does have a very characteristic odor and very high toxicity [3].

Benzene has been used in the manufacture of several chemical products like styrene, phenols, in nylon and synthetic fibers, maleic anhydride, pharmaceuticals, detergents and dyes and explosives. It has also been used in fuel, where it is added as a lead substitute, chemical reagent and extraction agent for other chemical compounds [3]. Derivates of benzene are used primarily as solvents and diluents in the manufacture of perfumes and in intermediates in the production of dyes, although the most important industrial use is as solvents and paint thinners [4,5].

Benzene has been banned as a component of products intended for domestic use and, in many countries, it has also been prohibited as a solvent and as a component of dry-cleaning liquids [4].

Natural sources of benzene include emissions from volcanoes and forest fires [6]. On the other hand, road traffic is an important source of emissions of VOCs, such as benzene, and other compounds that mainly come from vehicle exhausts and are generated during the incomplete combustion of fuels [7].

A study [8] has reported that there is a big discrepancy between the values obtained in measurements of industrial environments compared to the homogeneous values obtained in urban areas.

The main source of benzene exposure is tobacco smoke and other outdoor sources like automobile exhaust emissions [5], combustion processes and industrial chemical vapors [8]. Indoor sources include building materials, detergents, glues, furniture varnish and products such as solvents and paints from kitchens, attached garages and fuels for housing. The concentration of benzene indoors is generally higher than it is outdoors. It is affected by weather conditions and by the type of ventilation [3,5]. However, personal exposure concentrations of benzene in and around gas stations is considerably higher than indoor and ambient concentrations [9].

Recent studies [10] indicate that, while the limits of indoor concentration have been increasingly restrictive and the sources of interior emissions have been progressively reduced, outdoor benzene has become a significant, and even in many cases dominant, contributor to indoor concentration.

Air pollution is associated with adverse health effects such as respiratory and cardiovascular diseases, cancer and even death [11]. In 2017, environmental contamination contributed to almost 5 million premature deaths in the world [12]. More than 90% of the world's population lives in areas where the WHO-established guidelines for the quality of healthy air are exceeded and about half, a total of 3.6 billion, was exposed to contamination in the home. Air pollution is related to premature deaths from ischemic heart disease (16%), chronic obstructive pulmonary disease (41%) and lung cancer (19%).

The sources of entry of benzene into the human body are the lungs, the digestive tract and the skin [5] although population exposure is mainly produced through the inhalation of air containing benzene [8].

Several factors influence benzene exposure, including mainly concentration, duration of exposure, contact pathway and also the presence of other chemical substances, as well as the characteristics and habits of people [5].

A long or repeated exposure to benzene can affect the central nervous system and the immune system. It can also affect the bone marrow, and this can cause anemia and different types of leukemia. It may also cause inherited genetic damage in human germ cells [4,13].

Benzene has been categorized as being among group I carcinogens by the International Agency for Research on Cancer (IARC). It is considered as a known human carcinogen and no safe level can be recommended [3,14,15].

Sensitive population groups such as the elderly, pregnant women, infants, children and people with previous cardiovascular and respiratory diseases are the most susceptible to poor air quality [16]. Several studies indicate that there is a strong relationship between air pollution and infant mortality [17] and diseases of the infant respiratory system, so that hospital admissions increase significantly when there are pollution peaks. Hospital admissions for other causes also increase when benzene levels and other air pollutants rise [18–20].

Epidemiological studies have shown that there is a relationship between exposure to benzene and the development of acute myeloid leukemia [14]. A Swiss pediatric oncology cohort study [21] found an increased risk of leukemia among children whose mothers had been exposed to benzene at work. Studies like [22] also indicate the existence of an increased risk of colon cancer in Nordic countries, while a study in France showed that long-term exposure to outdoor air pollution like

benzene, fine particles, sulfur dioxide and nitrogen dioxide is a significant environmental risk factor for mortality [23].

There are different limits established for exposure to benzene. The European Commission has established a concentration limit for benzene of 5 $\mu g/m^3$ as an annual averaging period in the Directive 2008/50/EC, Directive on environment air quality [24].

Regarding occupational exposure limits [13], TLV-TWA of 0.5 ppm (the threshold limit value considered as the time-weighted average for an 8 h day and 40 weekly hours) and 2.5 ppm as STEL (short-term exposure limit for periods of 15 min that should not be exceeded at any time) have been established.

OSHA regulates benzene levels in the workplace. The maximum amount of benzene in the air must not exceed 1 ppm during an 8 h day, 40 h a week [5] (Directive 97/42/EC, amendment of 90/394/EEC).

Environmental pollution processes are complex and direct measurement is not always possible. In addition, it is difficult to carry out an analysis to discover the source, the propagation and distribution models and to make predictions over time. Therefore, since the reduction of such pollution is an objective of the European Union, it is necessary to find methods that make it possible to model its behavior and predict its evolution.

In the case of pollution concentration modeling, there are two methodologies used in predicting air quality: physical models and models based on statistical data. Physical models are based on chemical dispersion and transport models, in which the input variables are often related to parameters obtained from meteorological observations, such as air temperature, dew point, relative humidity, atmospheric pressure, speed and direction of wind, UV radiation, another group of parameters related to the terrain such as vegetation, local topography, etc. and parameters related to emission sources that represent changes in traffic patterns, industrial activity and the distance to these sources. Statistical machine learning models are based on periodic parameters and these models do not need to understand physical phenomena. Models based on statistical parameters achieve better predictions than models that only used meteorological variables as input [25].

The objective of this study was to predict the concentration of benzene from other pollutants, which were collected by several measurement stations of the Community of Madrid (Spain) every day during the period indicated in Table 1 for each station, constituting voluminous information that allows their mathematical modeling and statistical learning to obtain an explanation of the dependency among the main pollutants in a geographic area based on the concentrations of other air pollutants in each station, as well as the environmental pollution existing at that time in other stations.

Table 1. Stations.

Station	Code	Name	Latitude (Degree)	Longitude (Degree)	Altitude (m)	Measurement Dates
6	28,079,006	Marañón	40.437567	−3.690759	669	2001–2009
15	28,079,015	Pz Castilla	40.465595	−3.688740	729	2001–2008
22	28,079,022	Pontones	40.406326	−3.712948	618	2001–2005
23	28,079,023	Alcalá final	40.448702	−3.609647	642	2001–2009
25	28,079,025	Sta Eugenia	40.379079	−3.602494	658	2001–2005
26	28,079,026	Embajada	40.459303	−3.580089	611	2005–2010
27	28,079,027	Barajas	40.476931	−3.580018	621	2005–2009
35	28,079,035	Pz Carmen	40.419216	−3.703165	659	2001–2006

To make this prediction, time series forecasting techniques such as autoregressive moving average vectors (VARMAs), and integrated autoregressive moving average model (ARIMA) were applied and the results compared with those obtained with machine learning methodologies such as artificial neural networks (ANNs), support vector machines (SVMs), multivariate linear regressions (MLRs) and multivariate adaptative regression splines (MARSs).

In existing literature there are several success stories of applying these methodologies as a tool for predicting contamination and dispersion of air pollutants such as SO_2, NO_2, NO_x, PM_{10}, $PM_{2.5}$ O_3 or CO.

There are multiple studies of different pollutants to make predictions with ANNs such as [25–28]. Other models like SVMs have been studied by [29–31]. MLRs by [32,33]. Time data series by [34–37] and MARSs by [38,39].

In many of these articles they analyze different models, compare the results and determine which method makes the best predictions for the pollutants that have been studied [40,41].

A previous study [28] reviewed 139 articles about pollutants examined with ANN models and the results indicate that the most analyzed pollutants are particulate matter (PM_{10}, $PM_{2.5}$) in 62% of the articles, 36% nitrogen oxides (NO, NO_2 and NO_x), 31% O_3. SO_2 and CO modeling in 13% and 16% of articles. Almost a third of them include the study of several pollutants, but only one article [42], has studied an ANN model for predicting benzene.

Therefore, models for the prediction of benzene have not yet been developed, and even less so to study the relationship between measurements obtained from other pollutants at stations with benzene. The methods used are data-oriented, so there is no need to make solid chemical-physical assumptions during the modeling process, they have the capacity to handle large amounts of data and are applied in different tasks such as studying the existing relationship between benzene and the rest of pollutants and its prediction.

2. Materials and Methods

2.1. The Database

In this study, data observed from various pollutants was used, along with four predictor variables: nitrogen dioxide (NO_2), nitrogen oxides (NOx), particulate matter with a diameter less than 10 μm (PM_{10}), and toluene (C_7H_8) were selected to perform the benzene (C_6H_6) prediction models.

These pollutants were collected by eight remote measurement stations in the Community of Madrid (Spain) every day, with hourly measurements during the periods indicated in Table 1. Figure 1 shows the geographical location of the stations used in this study on the map of Madrid.

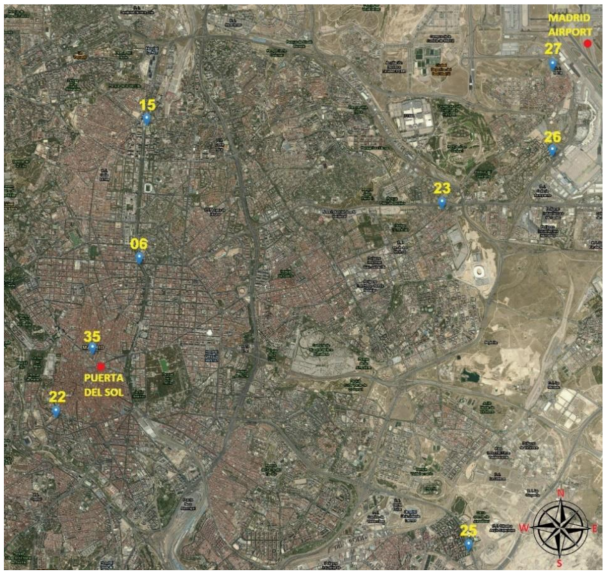

Figure 1. Location of stations in Madrid map.

Before carrying out the formation phase of the models used, an imputation of the missing data due to breakdowns and maintenance of the stations was carried out, through multiple imputations by chained equations [43].

The mean and standard deviation of the pollutants under study are shown in Table 2.

Table 2. Mean and standard deviation of pollutants in each station (BEN: benzene, NO_2: nitrogen dioxide, NO_x: nitrogen oxides, PM_{10}: particles with a diameter of 10 micrometres or less and TOL: toluene).

	St 6	**St 15**	**St 22**	**St 23**	**St 25**	**St 26**	**St 27**	**St 35**
BEN	2.989 (2.789)	2.544 (2.875)	2.544 (3.423)	1.403 (2.4472)	1.818 (2.153)	0.317 (0.395)	0.327 (0.402)	2.116 (1.999)
NO_2	79.116 (40.162)	69.170 (31.930)	56.489 (30.572)	59.861 (34.839)	62.005 (35.887)	63.932 (34.562)	40.907 (28.208)	70.052 (32.049)
NO_x	186.404 (149.716)	146.949 (107.172)	128.100 (117.860)	113.233 (126.986)	142.531 (135.603)	99.519 (88.431)	68.115 (65.472)	142.569 (120.250)
PM_{10}	39.065 (29.477)	35.9547 (27.948)	32.097 (26.910)	31.747 (28.422)	38.894 (34.107)	26.529 (23.114)	25.127 (22.961)	35.638 (29.355)
TOL	14.331 (14.249)	9.751 (10.184)	10.012 (11.220)	7.408 (10.233)	8.778 (11.519)	2.003 (4.762)	2.130 (3.716)	11.774 (11.274)

2.2. Multivariate Linear Regression

Linear regression with several independent variables is known as multivariate or multiple linear regression (MLR), which is a generalization of simple regression when the relationship of a dependent variable designated by Y is represented by the linear combination of the other k independent variables, which are denoted by $X_1,\ X_2,\ \ldots,\ X_k$.

MLR model assumes a relationship of the type $Y = \beta_0 + \beta_1 X_1 + \beta_2\ X_2 + \ \ldots + \beta_k\ X_k + \varepsilon$, where $\beta_0,\ \beta_1,\ \beta_2,\ \ldots, \beta_k$ are $(k + 1)$ parameters called regression coefficients and ε is the stochastic error associated with the regression. Regression parameters β are estimated in order to determine the best hyperplane among all of the form $y_t = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \ \ldots + \hat{\beta}_k\ x_k$. A set of n observations is available for each of the dependent and independent variables, so the model is as follows [44,45]:

$$Y_j = \hat{\beta}_0 + \hat{\beta}_1 X_{1j} + \hat{\beta}_2 X_{2j} + \ \ldots + \hat{\beta}_k\ X_{kj} + \varepsilon_j,\ j = 1, 2, \ldots,\ n \tag{1}$$

In order to be able to make predictions about the behavior of the Y variable, it is first necessary to carry out the estimations $\hat{\beta}_0,\ \hat{\beta}_1,\ \hat{\beta}_2,\ \ldots, \hat{\beta}_k$ of the model parameters by the ordinary least square method whose objective is to find the hyperplanes that minimize the sum of square error (SSE) between the observed and predicted response: $SSE = \sum_{j=1}^{n}\left(y_j - \hat{y}_j\right)^2$, where y_j is the outcome and $\hat{y}_j$ is the model prediction. That is:

$$\sum_{j=1}^{n} {\varepsilon_j}^2 = \sum_{j=1}^{n}\left(y_j - \hat{\beta}_0 - \hat{\beta}_1 x_{1j} - \hat{\beta}_2 x_{2j} - \ \ldots - \hat{\beta}_k\ x_{kj}\right)^2 \tag{2}$$

To minimize SSE, it is derived with respect to the $\hat{\beta}_0,\ \hat{\beta}_1,\ \hat{\beta}_2,\ \ldots, \hat{\beta}_k$ parameters and equating to zero gives a system of equations called normal equations.

In matrix form, the MLR model involves finding the last square solution of the linear system $X\hat{\beta} = Y$, where X is the $n \times (k + 1)$ input data matrix with each row an input vector (with a 1 in the first position), X^t its transposed matrix, Y the $(n \times 1)$ vector of output in the training set and $\hat{\beta}$ the $(k \times 1)$ parameters vector. Now the residual sum-of-squares is $\left(Y - X\hat{\beta}\right)^t\left(Y - X\hat{\beta}\right)$ and to minimize SSE is derived $^{\partial SSE}/_{\partial\hat{\beta}} = -2X^t\left(Y - X\hat{\beta}\right)$ and equating to zero (assuming rank X^tX is k and, hence, X^tX is positive definite), that is: $^{\partial SSE}/_{\partial\hat{\beta}} = -2X^t\left(Y - X\hat{\beta}\right)$; therefore, $X^t\left(Y - X\hat{\beta}\right) = 0$ and finally $\hat{\beta} = \left(X^tX\right)^{-1}X^tY$.

The fitting values at the training inputs are $Y = X\beta = X\left(X^tX\right)^{-1}X^t$ or what is the same:

$$y_j = \hat{\beta}_0 + \sum_{j=1}^{k} \hat{\beta}_j x_{ij},\ i = 1, 2, \ldots,\ n \tag{3}$$

Before modelling, stepwise regression [45] was used to choose a significant subset of independent variables for every station. This method selects the predictors by automatic iteration forward procedure dropping and including variables. It stops when the lowest Akaike Information Criterion (AIC) is reached. The models were built based on R^2 adjust when all variables were at 5% significance level and were validated by the k-fold cross validation method, which is explained later.

2.3. Multivariate Adaptive Regression Splines

Multivariate adaptive regression splines (MARSs) is a nonparametric statistical method developed by Friedman [46] and is a generalization of recursive partitioning regression (RPR) that can consider complex relationships between a set of k predictor independent variables, which are denoted by $X_1,\ X_2,\ \ldots,\ X_k$ and a dependent variable designated by Y, and does not make starting assumptions about any type of functional relationship between input and output variables. The MARS model is defined as [46–48].

$$\hat{y} = f(X) = \beta_0 + \sum_{m=1}^{M} \beta_m h_m(X) + \varepsilon \tag{4}$$

where X is a function of the independent variables and their interactions, β_0 is the intercept parameter, β is a vector of coefficients of the basis function, M is the total number of basis functions, $h(X)$ the spline basis function in the model and ε is the fitting error.

To approximate the non-linear relationships between the independent variables X and the response variable Y, basis functions (BF) are used. They consist of a single spline function or the product of two or more spline functions for different predictors. Spline functions are piecewise linear functions, that is, truncated left-hand and right-hand functions, and take the form of hinge functions that are joined together smoothly at the knots.

$$(x - t)_+ = \begin{cases} x - t,\ x > t \\ 0,\ x \leq t \end{cases} \tag{5}$$

$$(t - x)_+ = \begin{cases} x - t,\ x > t \\ 0,\ x \leq t \end{cases} \tag{6}$$

where t is a constant called a node that specifies the boundary between the regions that have continuity from the base functions of the regions from left to right and that are smoothly joined at the given node and adaptively selected from the data. The "+" sign refers to the positive part and indicates a value of zero for negative values of the argument.

As stated in [49], MARS forms reflected pairs for each predictor variable with knots of each value $x_j,\ j \in \{1, \ldots, k\}$ with knots at each observed value $x_{ij},\ 1 \in \{1, \ldots, n\}$ of that variable, where n is the sample size. The set of all possible pairs with the corresponding knots and the truncated linear basis functions can be expressed by the set $D = \left\{\left(x_j - t\right)_+,\ \left(t - x_j\right)_+ \middle|\ t \in \left\{x_{1j}.\ x_{2j} \ldots., x_{nj}\right\},\ j \in \{1, \ldots, k\}\right\}$.

An adaptive regression algorithm is taken during a recursive partition strategy to automatically select the locations of the node or breakpoints, including the two-stage process: forward-stepwise regression selection and backward-stepwise elimination procedure [50].

The first step, also called the construction phase, begins with the intercept and then sequentially adds to the model the predictor that best improves the fit; that is, when the maximum reduction in the sum-of-squares residual error occurs. The search for the best combination of variable and node is done

iteratively. Considering a current model with M base functions, the following pair will be added to the model in the form of $\beta_{M+1}h_m(X)max\left(0, X_j - t\right) + \beta_{M+2}h_m(X)max\left(0, t - X_j\right)$. The β coefficients are estimated using the least-squares method [44].

This process continues until a predetermined number of base functions (M_{max}) is reached or the R^2 changes less than a threshold [50]. A large number of BFs are added one after another and an overfitting model is created. Generally, the maximum number of BF is 2 to 4 times the number of predictor variables [46].

The second step, also called the pruning phase, begins with the full model and simplifies it by eliminating terms by applying a backward procedure to avoid oversizing. MARS identifies the basis functions that contribute the least to the model and removes the least significant terms sequentially. The final model is chosen using the generalized cross-validation method (GCV) [47], which is an adjustment of the sum-of-squares of the residuals and penalizes the complexity of the models by the number of basis functions and the number of knots.

$$GCV = \frac{\frac{1}{N}\sum_{i=1}^{N}[y_i - f(x_i)]^2}{\left[1 - \frac{M+d(M-1)/2}{N}\right]^2} \tag{7}$$

where M is the number of BF, N is the number of data sets, $f(x_i)$ denotes the predicted values of MARS and d is a penalty for each basis function, which takes the value of two for the additive model and three for an interaction model [47]. The term $(M-1)/2$ is the number of hinge functions knots [44].

Note that MARS is a trademark and is used in this document as an acronym for multivariate adaptive regression splines.

The MARS model was made with the parameters degree = 9 and thresh = 1×10^{-8}. These parameters are explained in [50].

2.4. Artificial Neural Networks

The artificial neural network (ANN) is a nonlinear and nonparametric statistical method designed to simulate the data processing of brain neurons [51]. The ANN does not make any prior assumption about the model-building process or the relationship between input and output variables. Several studies [52,53] have indicated that MLP is the most suitable and widely-used class of neural network in atmospheric pollutants. The MLP is the most common neural network and is extensively developed in the following literature [51,54,55].

The MLP network is divided into several layers: an input layer that has as many neurons as the model has independent variables, k, a single hidden intermediate layer of neurons, H, and an output layer with as many neurons as the model has dependent variables, S.

In a previous study [56] it was shown that a single hidden layer with a finite number of units is generally sufficient to fit any piecewise continuous function. In existing literature [57], several proposals have been made to estimate the number of neurons, although the number of hidden units H was determined by trial-and-error training various networks and estimating the corresponding errors in the test data set. If the hidden layer has few neurons, many training errors occur due to insufficient fit or because the network could not converge during training. In contrast, too many hidden layer neurons lead to low training error, but due to overfitting, they memorize the dataset and cause high test error and poor generalization [58].

Input variables are mapped by functions, called activation functions, into intermediate variables of the hidden layer, which are mapped to the output variables. MLP utilizes the following transformations [40]: $Y = \psi(\phi(X)) = (\psi \circ \phi)(X)$, $\phi : X \subset \mathcal{R}^k \rightarrow U \subset \mathcal{R}^h$, $\psi : U \subset \mathcal{R}^h \rightarrow Y \subset \mathcal{R}^s$, where $X = (x_1, x_2, \ldots, x_k)$ represents the inputs, $U = (u_1, u_2, \ldots, u_H)$ is the output of the hidden layer, an output layer with one dependent variable $\hat{Y}$, in this case, it is considered for generalization, $\hat{Y} = (\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_S)$ as the outputs of the network and W indicates the weight matrix between two

layers. $\phi(\cdot)$ and $\psi(\cdot)$ are transfer functions of hidden layer and output layer. Both of these are logistic functions that map the output of each neuron to the interval [0,1].

All inputs were normalized as $z_i = {}^{x_i - \overline{x}}/_{\sigma_x}$, where $\overline{x}$ is the mean and σ the standard deviation of the observation dataset for each variable.

Neural networks fit to data using learning algorithms due to an iterative training process. In this case, a supervised learning algorithm characterized by the use of a target output was compared with the predicted output and by adjusting the weights. All weights and bias were initialized with random values taken from a standard normal distribution. The main steps of this iterative training procedure are as follows:

Perform forward propagation of the first vector of input variables through the whole MLP network, which ultimately calculates an output $\hat{Y}$ for inputs X and current weights.

The input layer is just an information-receiving layer that receives the vectors of the input variables and redistributes them to the neurons in the hidden layer. This layer does not perform any type of processing on the data.

In the hidden layer, all inputs are multiplied by the weights and sum of all, taking into account the bias. The output of the i-th neuron of the hidden layer with H nodes is the one transformed by an activation function $\phi(X)$. The output of the hidden layer is

$$u_j = \phi_j\left(w_{0j} + \sum_{j=1}^{H} w_{ij}x_i\right) \tag{8}$$

where w_{ij} is the weight connecting the i-th input with the j-th hidden node, w_{0j} is a limit value known as the threshold value or bias, $i = 1, 2, \ldots, k$ and $j = 1, 2, \ldots, H$. The activation function f used in this paper is a non-linear and differentiable non-decreasing bounded function such as the logistic function $f(z) = \frac{1}{1+e^{-z}}$ In the output layer, all the outputs of the hidden layer are multiplied by the weights and the sum of all, taking into account the bias. The output of the neuron m-th of the output layer is transformed by an activation function $\psi(U)$. The output of the output layer is:

$$\hat{y}_m = \psi_m\left(w_{0m} + \sum_{m=1}^{S} w_{jm}u_j\right) = \psi\left(w_0 + W^T U\right) \tag{9}$$

where w_{jm} is the weight connecting the j-th output of the hidden layer with the m-th output node, w_{0m} is the bias, $m = 1, 2, \ldots, S$ and $j = 1, 2, \ldots, H$. The activation function used is the logistic function again.

$$\hat{y}_m = \psi_m\left(w_{0k} + \sum_{m=1}^{S} w_{0m}u_j\right) = \psi_m\left[w_{0m} + \sum_{m=1}^{S} w_{jm}\cdot\phi_j\left(w_{0j} + \sum_{j=1}^{H} w_{ij}x_i\right)\right] = (\psi \circ \phi)\left(w_0 + W^T X\right)$$

During the training process, the predicted output $\hat{y}$ will be different from the observed output y. An error function $E = \frac{1}{2}\sum_{t=1}^{n}\sum_{m=1}^{S}(\hat{y}_{lm} - y_{lm})^2$ is calculated as the sum of squared errors (SSE), where $t = 1, \ldots, n$ are the observations (input-output pairs) and $m = 1, \ldots, S$ the output nodes.

During the backward phase, the error is propagated backward through the network, but the neurons in the hidden layer only receive a part of the total error signal, which depends on the relative contribution that each neuron has made to the feed forward output. This whole forward and backward process is repeated for several iterations and stops when a given threshold is reached by all absolute partial derivatives $(\partial E/\partial w)$ of the error function with respect to the weights [59].

In this paper the Rprop algorithm was used, a resilient backpropagation with weight backtracking [59–61], which modifies the weights by adding a learning rate to find a local minimum of the error function so that when ${}^{\partial E}/_{\partial w} < 0$ the weight is augmented and when ${}^{\partial E}/_{\partial w} > 0$ the weight is reduced.

The main difference between Rprop and the backpropagation algorithm is that only the sign of the gradient is used to update the weights. In this way convergence is accelerated.

$$w_{ij}^{(t+1)} = w_{ij}^{(t)} - \eta_i^{(t)} \cdot \text{sign}\left(\frac{\partial E^{(t)}}{\partial w_{ij}^{(t)}}\right) \tag{10}$$

where t is the iteration of gradient descent and w_{ij} the weights. η_i is the learning rate that will be increased if the corresponding partial derivative keeps its sign or it will be decreased if the partial derivative of the error function changes its sign, that is [60]:

$$\eta_i^{(t)} = \begin{cases} min\left(\alpha \eta_i^{(t-1)}, \eta_{max}\right) \text{ if } \frac{\partial E^{(t)}}{\partial w_{ij}^{(t)}} \cdot \frac{\partial E^{(t-1)}}{\partial w_{ij}^{(t-1)}} > 0 \\ max\left(\beta \eta_i^{(t-1)}, \eta_{min}\right) \text{ if } \frac{\partial E^{(t)}}{\partial w_{ij}^{(t)}} \cdot \frac{\partial E^{(t-1)}}{\partial w_{ij}^{(t-1)}} < 0 \\ \eta_i^{(t-1)} \text{ otherwise} \end{cases} \tag{11}$$

where $\alpha > 1 > \beta$, usually $\alpha = 1.2$, $\beta = 0.5$.

In this study, the neuralnet library [59] of the R computer software was used. The number of neurons used was 8 for all stations. To determine this parameter, a random station was taken and was determined by trial-and-error and k-fold cross validation, k = 5, taking into account that in some studies such as [53] it is indicated that the number of neurons H could be the sum of the number of input variables plus the number of the output variables and the maximum number of neurons in the hidden layer twice the number of neurons in the input layer, although [53,58] indicate that there is no a "rule of thumb" to determine this parameter and so there should be an iterative approach to it.

The criteria used to stop the algorithm was the threshold for the partial derivatives of the error function, setting the threshold of 1 and a limit of maximum steps of 10^9. The detailed explanation of the parameters of the computer program are indicated in reference [59]. The logistic function was chosen instead of other commonly-used functions such as the hyperbolic tangent as the activation function [44,62] because the values of the dependent variable take positive values given that it is the measure of benzene concentration. The input variables were normalized as indicated. As it is a regression-based model, the output parameter was fixed as linear.

2.5. Support Vector Machines

The support vector machine (SVM) is a nonparametric machine learning method developed by Vapnik [63] for both classification as regression. In this paper, the method was used for regression, that is, support vector regression (SVR). There are two basic versions of SVM regression, epsilon-SVR and nu-SVR denoted by ε-SVR and ν-SVR, the differences between which will be discussed later.

The SVR task consists of transforming the training dataset $\mathcal{T} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ to a higher dimensional space $\mathcal{F}$ through a non-linear mapping by a kernel function $\Phi(x)$ where a linear regression can be done, with $y_i \in \mathcal{R}$, $x_i \in \mathcal{R}^k$, n being the number of samples and k the dimension of the input dataset.

The SVR model is defined as $f(x) = \sum_{i=1}^{n} w \circ \Phi(x) + b$, where b is the intercept of the model indicating the bias, $w \circ \Phi(x)$ is the dot or scalar product of weight vector $w \in \mathcal{R}$ and the kernel function.

The error function is defined in [63] by an ε-insensitive loss function that defines a tube so that if the predicted value is within the tube the loss is zero, that is:

$$\begin{cases} 0 \text{ if } \left|y_i - f(x_i)\right| < \varepsilon \\ \left|y_i - f(x_i)\right| - \varepsilon \text{ otherwise} \end{cases} \tag{12}$$

In order to solve the limitations that result from the optimization problem, introduce slack variables ξ_i^+, ξ_i^- that depend on the position in relation to the tube: above (ξ_i^+) or below (ξ_i^-). Therefore, the problem as indicated in [63,64] is stated as follows:

Minimize $C\sum_{i=1}^{n}\left(\xi_i^+ + \xi_i^-\right) + \frac{1}{2}\|w\|^2$, with the following constraints:

$$\begin{cases} y_i \le w \circ \Phi(x_i) + b + \varepsilon + \xi_i^+ \\ y_i \ge w \circ \Phi(x_i) + b - \varepsilon - \xi_i^- \\ \xi_i^+,\ \xi_i^- \ge 0,\ i = 1,\ldots,n,\ \varepsilon \ge 0 \end{cases} \tag{13}$$

where $C > 0$ is the cost parameter whose function is to control the trade-off between the complexity of the model and the maximum level of deviation above ε. If the cost parameter is large, the model becomes more flexible since the effect of the errors measured by the slack variables, and the value of ε increases. On the other hand, if C is small, the model will be tighter and less likely to overfit, since the effect of the norm of weights vector is greater and leads to a better generalization [65].

The problem can be solved in a simpler way in its dual formulation, also making it possible to extend the problem to nonlinear functions. Therefore, a standard dualization method is used using Lagrange multipliers for the optimization problem, as described in [64,66] and applying the Karush–Kuhn–Tucker (KKT) optimality conditions of the primal problem [67]. The kernel function $K\left(x_i, x_j\right)$ returns the scalar product between the pairwise in a high-order dimensional space without explicitly mapping data. [63].

$$\begin{aligned} \max L \quad = \max\Bigg(&\tfrac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\left(\xi_i^+ + \xi_i^-\right) - \sum_{i=1}^{n}\left(\eta_i^+\xi_i^+ + \eta_i^-\xi_i^-\right) \\ &- \sum_{i=1}^{n}\left(\alpha_i^+\left(\varepsilon + \xi_i^+ - y_i + w \circ \Phi(x_i) + b\right)\right) \\ &- \sum_{i=1}^{n}\left(\alpha_i^-\left(\varepsilon + \xi_i^- + y_i - w \circ \Phi(x_i) - b\right)\right)\Bigg) \end{aligned} \tag{14}$$

where L is the Lagrangian and $\alpha_i^+ > 0, \alpha_i^- > 0, \eta_i^+ > 0, \eta_i^- > 0$ are the Lagrange multipliers for all i considered. A saddle point is found by partial derivatives with respect to $b, w, \xi_i^+,\ \xi_i^-$ [63]:

$$\partial L/\partial b = \sum\nolimits_{i=1}^{n}\left(\alpha_i^+ + \alpha_i^-\right) = 0,\ \partial L/\partial w = w - \sum\nolimits_{i=1}^{n} x_i\left(\alpha_i^+ + \alpha_i^-\right) = 0,$$

$$\partial L/\partial \xi_i^+ = C - \alpha_i^+ - \eta_i^+ = 0,\ \partial L/\partial \xi_i^- = C - \alpha_i^- - \eta_i^- = 0$$

After which the dual variables η_i^+, η_i^- are eliminated after the substitution of $\eta_i^+ = C - \alpha_i^+$ and $\eta_i^- = C - \alpha_i^-$. Thus, the dual optimization is as described:

$$\max\left(\sum_{i=1}^{n} y_i\left(\alpha_i^+ - \alpha_i^-\right) - \varepsilon\sum_{i=1}^{n}\left(\alpha_i^+ - \alpha_i^-\right) - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\left(\alpha_i^+ - \alpha_i^-\right)\left(\alpha_j^+ - \alpha_j^-\right)K\left(x_i, x_j\right)\right) \tag{15}$$

$$\text{subjected to} \begin{cases} \sum\limits_{i=1}^{n}\left(\alpha_i^+ - \alpha_i^-\right) = 0, \\ 0 \le \alpha_i^+ \le C \\ 0 \le \alpha_i^- \le C \end{cases} \tag{16}$$

for all i (16)

where $K\left(x_i, x_j\right)$ is the kernel function that satisfies the Merced condition explained in [68] and can be written as $K\left(x_i, x_j\right) = \Phi(x_i) \circ \Phi\left(x_j\right)$. The radial basis function (RBF) kernel is used in this paper, that is, $K\left(x_i, x_j\right) = e^{-\lambda\|x_i - x_j\|^2}$ and λ is a parameter that regulates the behavior of the kernel.

Finally, after solving the dual problem, the prediction function $f(x)$ can be formulated in terms of Lagrange multipliers and the kernel function as:

$$f(x) = \sum_{i=1}^{n} \left(\alpha_i^+ - \alpha_i^-\right) \Phi(x) + b \tag{17}$$

ε-SVR uses parameters $C > 0$ and $\varepsilon > 0$ to apply an optimization penalty for points that were not predicted correctly. As it is difficult to select an appropriate ε, in [69] a new algorithm ν-SVR is introduced that automatically adjusts the ε parameter, which defines the tolerance margin, by introducing a new parameter $\upsilon \in [0, 1]$ that makes it possible to control the number of support vectors and training errors, establishing the relationship between the number of support vectors that remain in the solution with respect to the total number of samples in the data set. ε parameter is introduced in the optimization problem formulation and is estimated automatically. This formulation is:

$$\min C(\upsilon\varepsilon + \frac{1}{n}\sum\nolimits_{i=1}^{n}\left(\xi_i^+ + \xi_i^-\right)) + \frac{1}{2}\|w\|^2$$

which can be solved in a similar way to ε-SVR:

$$\text{subjected to} \begin{cases} y_i \leq w \circ \Phi(x_i) + b + \varepsilon + \xi_i^+ \\ y_i \geq w \circ \Phi(x_i) + b - \varepsilon - \xi_i^- \\ \xi_i^+,\ \xi_i^- \geq 0,\ i = 1, \ldots, n,\ \varepsilon \geq 0 \\ \varepsilon \geq 0,\ 0 \leq \upsilon \leq 1 \end{cases} \tag{18}$$

In [70] it is explained how to solve ν-SVR in detail and in [66] the relationship between ε-SVR and ν-SVR is discussed.

A grid search [67] was carried out to determine the parameters in one of the stations, randomly chosen, and these values were used in all stations. The study was undertaken with the following parameters: tolerance: 0.01, 0.05, 0.1, 0.5; C: 1, 10, 50; ε: 0.10, 0.11, 0.12, 0.15, 0.16, 0.20, 0.25; gamma: 0.15, 0.17, 0.20, 0.25; υ: 0.2, 0.5, 0.6, 0.75.

The parameters used for SVM algorithms are as follows:

For ε-SVR model: tolerance = 0.01, C = 10, ε = 0.11, gamma = 0.15

For ν-SVR model: tolerance = 0.01, C = 10, υ = 0.6.

where gamma is a kernel parameter that defines the influence of a single training set point and tolerance is the termination criterion.

2.6. Autoregressive Integrated Moving Average

Autoregressive integrated moving average (ARIMA) is a parametric method of univariate analysis of time series that have a stochastic nature, and whose methodology was described by Box and Jenkins [71–75].

The variable Y_t, $t = 1, 2, \ldots, n$ where n is the total number of observations, depends only on its own past and a set of random shocks, but on no other independent variables. It is; therefore, a matter of making forecasts about the future values of said variable using as information only that contained in the past values of the time series itself.

In the ARIMA model (p, d, q), p represents the order of the autoregressive process (AR), d is the number of differences that are necessary for the process to be stationary (I) and q represents the order of the moving average process (MA).

This methodology is fundamentally based on two principles [71]:

(a) Selection of the model iteratively through four steps: identification to determine the order p, d, q of the model, estimation of the parameters, validation to verify that the model fits the data and prediction.

(b) Concise parameterization or parsimony for the representation of the model with the minimum number of possible parameters.

B is defined as lag operator $BY_t = Y_{t-1}$ as the result of delaying observations by one period, in general, $B^kY_t = Y_{t-k}$, where k is the number of lags and ∇ as difference operator of the form $\nabla Y_t = Y_t - Y_{t-1}$, then as a function of B operator, is $\nabla Y_t = (1-B)Y_t$, in general, $\nabla^k Y_t = (1-B)^k Y_t$ [73].

The autoregressive or AR(p) model is based on the fact that the value of the series at a given moment t is the linear combination of the past values up to a maximum number p and a random error, that is, $Y_t = \beta_0 + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + \cdots + \beta_p Y_{t-p} + \varepsilon_t$, where $Y_{t-1}, Y_{t-2}, \ldots, Y_{t-p}$ are the past values of the series, $\beta_0, \beta_1, \beta_2, \ldots, \beta_p$ are the constants, with values different from zero, that have to be estimated with the regression and ε_t is a Gaussian white noise error random variable.

$$Y_t = \beta_0 + \left(\sum_{i=1}^{p} \beta_i B^i\right) Y_t + \varepsilon_t, \; t = p+1, \ldots, n \tag{19}$$

Another representation can be given as $\beta(B)Y_t = \beta_0 + \varepsilon_y$, where $\beta(B) = 1 - \beta_1 B - \beta_2 B^2 - \cdots - \beta_p B^p$ [73].

The moving average or MA(q) model is represented by a relationship between the variable and the present and q past values of white noise, that is, $Y_t = \alpha_0 + \varepsilon_t - \alpha_1 \varepsilon_{t-1} + \alpha_2 \varepsilon_{t-2} + \cdots + \alpha_q \varepsilon_{t-q}$, where $\alpha_0, \alpha_1, \alpha_2, \ldots, \alpha_q$ are constants, with values different from zero and $\varepsilon_{t-1}, \varepsilon_{t-2}, \ldots, \varepsilon_{t-q}$ the lagged errors.

$$Y_t = \alpha_0 - 1 - \left(\sum_{i=1}^{p} \alpha_i B^i\right) \varepsilon_t, \; t = q+1, \ldots, n \tag{20}$$

Then $Y_t = \alpha_0 + \alpha(B)\varepsilon_t$, where $\alpha(B) = 1 - \sum_{i=1}^{q} \alpha_i B^i$ [73]

When a time series is non-stationary, the integrated process I(d) is the method to make the time series near-stationary by differencing, where d is the number of differentiations necessary to make the series stationary, that is, using lag and difference operators: $\nabla^d Y_t = (1-B)^d Y_t$

Hence, an ARIMA (p, d, q) model can be written as follows, where γ is a constant:

$$\left(1 - \beta_1 B - \beta_2 B^2 - \cdots - \beta_p B^p\right)(1-B)^d Y_t = \left(1 + \alpha_1 B + \alpha_2 B^2 + \cdots + \alpha_q B^q\right)\varepsilon_t + \gamma \tag{21}$$

$\beta_p(B)\nabla^d Y_t = \alpha_q(B)\varepsilon_t + \gamma$ as indicated in [76].

A stationary model is an ARMA (p, q) process, that is an ARIMA (p, 0, q).

To adjust the best ARIMA model for each station, the auto.arima function [77] of the forecast library of the R software was used. The value of the parameters p, d, q is indicated in Table 7.

2.7. Vector Autoregressive Moving Average

Vector autoregressive moving-average (VARMA) is a parametric method of multivariate analysis time series that have a stochastic nature. They are a generalization of univariate models ARMA, with the difference that instead of a single variable there are several variables; that is, they study the relationships between several time series, without distinguishing between exogenous and endogenous variables. A detailed explanation of the model can be found in [71,78].

In the VARMA model (p, q) p represents the order of the autoregressive process (VAR) and q represents the order of the moving average process (VMA).

This methodology is fundamentally based on two principles [71]:

(a) Selection of the model iteratively. First, determination of the p, q order of the model, estimation of the parameters, validation that the model fits the data and prediction.

(b) Representation of the model with the minimum number of possible parameters.

B is defined as lag operator in the same way that in the previous section $B^kY_t = Y_{t-k}$, where k is the number of lags [73].

Vector autoregressive or VAR(p) is a model in which the value of the series at a given time t are a linear combination of the past values of the variable and of the other variables up to a maximum number p and a random error vector, that is, $Y_t = \beta_0 + \left(\sum_{i=1}^{p} \beta_i Y_{t-i}\right) + \varepsilon_t$, where Y_t is a vector of k variables to be predicted at time t, β_0 is a k dimensional vector of constants, β_i are $k \times k$ matrices for $i > 0$ and β_p not a null matrix. ε_t are the multivariate white noise vectors with a positive definite covariance matrix Σ_ε and zero mean.

Another representation with B operator can be given as $\beta(B)Y_t = \beta_0 + \varepsilon_t$, where $\beta(B) = I - \beta_1 B - \beta_2 B^2 - \cdots - \beta_p B^p$ is a degree p polynomial of matrices and I is $k \times k$ identity matrix [71].

The vector moving average or VMA(q) model is represented by a relationship between the time series and the present and q past values of white noise, that is, $Y_t = \mu + \varepsilon_t - \sum_{i=1}^{p} \alpha_i \varepsilon_{t-i}$, where α_i are $k \times k$ matrices and α_q not a null matrix, $i > 0$. μ is constant vector containing the mean of the process. ε_t are white noise series. Using B lag operator, the model becomes $Y_t = \mu + \alpha(B)\varepsilon_t$, where $\alpha(B) = I - \alpha_1 B - \alpha_2 B^2 - \cdots - \alpha_q B^q$ is a matrix polynomial of order q and I is $k \times k$ identity matrix [78].

The vector autoregressive moving-average VARMA (p,q) can be written as follows:

$$\beta(B)Y_t = \beta_0 + \alpha(B)\varepsilon_t \tag{22}$$

The VARMA parameters p and q have been determined by trial-and-error, choosing those that allow determining a smaller RMSE error. These parameters are shown in Table 8.

2.8. Performance Measurements

In order to make comparisons of the performance measures between the machined learning models, for each model the same parameters were established for all the stations, as indicated in the corresponding sections.

The different models that have been developed in this study are evaluated for their accuracy by root mean squared error (RMSE) [79,80], mean absolute error (MAE) [80,81] and bias [80] according to the following equations:

$$RMSE = \sqrt{\frac{1}{n}\sum_{j=1}^{n}\left(y_j - \hat{y}_j\right)^2} \tag{23}$$

$$MAE = \frac{1}{n}\sum_{j=1}^{n}\left|y_j - \hat{y}_j\right| \tag{24}$$

$$bias = \frac{1}{n}\sum_{j=1}^{n}\left(y_j - \hat{y}_j\right) \tag{25}$$

As a method to estimate the prediction error, a k-fold cross validation was used for machine learning algorithms. This method uses part of the data to fit the model and another part to test it [44]. The dataset has been divided into k = 5 equal parts.

To validate the model of the time series algorithms, the data was divided into two sets, 80% of the initial data to fit the algorithm. A prediction was made on 20% of the final observations, which are the values used to calculate the RMSE, MAE and bias errors.

3. Results and Discussion

In this section the performance of the forecasts performed with the MLR, MARS, MLPNN, SVM, ARIMA and VARMA models are presented. Table 3 shows RMSE, MAE and bias values for the MLR models of all the stations, while Table 4 shows the same information for the MARS models while Tables 5–8 do the same for SVM, MLPNN with a single hidden layer with eight neurons, ARIMA and VARMA models, respectively. In the case of SVM, both ε-SVR and ν-SVR regressions were tested.

Figure 2 shows a comparison of all RMSE values for all models and stations, the same is made for MAE in Figure 3 and BIAS in Figure 4.

Table 3. RMSE, MAE and bias values in each station for MLR models (RMSE: root mean squared error, MAE: mean absolute error, BIAS: bias or historical average error, MLR: multivariate linear regression).

Station	RMSE	MAE	BIAS
6	1.00118	0.52430	−0.00072
15	1.85302	1.10512	0.00001
22	1.30363	0.78811	0.00009
23	1.28901	0.61969	0.00006
25	1.06531	0.48992	−0.00006
26	0.25824	0.11533	0.00000
27	0.29791	0.13354	0.00003
35	0.89380	0.44451	−0.00008

Table 4. RMSE, MAE and bias values in each station for MARS models (RMSE: root mean squared error, MAE: mean absolute error, BIAS: bias or historial average error, MARS: multivariate adaptive regression splines).

Station	RMSE	MAE	BIAS
6	0.93551	0.49397	−0.00022
15	1.80570	1.07105	0.00286
22	1.26793	0.71869	−0.00196
23	1.13057	0.52616	0.00071
25	1.01452	0.46532	−0.00098
26	0.24430	0.10422	−0.00073
27	0.27114	0.11383	0.00004
35	0.89007	0.43271	−0.00018

For station 06, RMSE values are very similar and close to unity for the MLR, MARS and SVM models. The lowest value corresponds to MARS model with 0.936. RMSE value for MLP is slightly higher at 1.171. The highest values were reached with the time series models with 1.654 for ARIMA and 2.106 for VARMA. A similar behavior occurs with MAE, as the lowest value for the ε-SVR model is 0.478 and MLR, ν-SVR and MARS show values close to 0.5. MLP has a slightly higher value with 0.606 and the time series models reach the highest values, with 1.805 being the highest of the VARMA model. Regarding the bias error, all the models have negative values, so the prediction tends to be higher than the observed value. Again, the MLR, MARS and SVR models reached values close to zero and the time series models have a greater bias with values of 1.449 for VARMA.

Table 5. RMSE, MAE and bias values in each station for MLPNN models (RMSE: root mean squared error, MAE: mean absolute error, BIAS: bias or historial average error, MLPNN: multilayer perceptron neural network).

Station	RMSE	MAE	BIAS
6	1.17063	0.60598	−0.09609
15	2.02740	1.19491	−0.08878
22	1.21984	0.70447	−0.00451
23	1.35970	0.62052	−0.08020
25	1.13518	0.52670	−0.08030
26	0.25712	0.11539	−0.02126
27	0.30390	0.14138	−0.01653
35	0.91940	0.45993	−0.00654

Table 6. RMSE, MAE and bias values in each station for SVM models with *nu* and *eps* regression (RMSE: root mean squared error, MAE: mean absolute error, BIAS: bias or historial average error, SVM: support vector machines).

	nu-Regression			eps-Regression		
Station	**RMSE**	**MAE**	**BIAS**	**MAE**	**MAE**	**BIAS**
6	0.95816	0.47893	−0.02310	0.93916	0.47810	−0.01568
15	1.71299	1.00743	−0.08639	1.70947	1.00649	−0.09397
22	1.31154	0.72078	−0.01672	1.29252	0.73885	0.01520
23	1.09511	0.49838	−0.05577	1.09468	0.50643	−0.03502
25	1.01331	0.45760	−0.03780	0.99517	0.45572	−0.02826
26	0.24445	0.09727	−0.02911	0.24255	0.09694	−0.02778
27	0.27487	0.09955	−0.03873	0.27343	0.10312	−0.03512
35	0.88557	0.41802	−0.02636	0.88026	0.41629	−0.02398

Table 7. RMSE, MAE and bias values in each station for ARIMA (*p*,*d*,*q*) models (RMSE: root mean squared error, MAE: mean absolute error, BIAS: bias or historial average error, ARIMA: autoregressive integrated moving average).

Station	**(*p*,*d*,*q*)**	**RMSE**	**MAE**	**BIAS**
6	(5,1,1)	1.65368	0.98688	−0.64045
15	(5,1,3)	1.47438	1.13770	−0.92292
22	(5,1,2)	3.00834	1.65755	−1.64900
23	(5,1,5)	1.13321	0.55848	−0.12683
25	(0,1,0)	1.29739	0.93479	0.25096
26	(3,1,3)	0.36901	0.31947	0.24297
27	(1,1,3)	0.51699	0.29751	−0.06988
35	(4,1,3)	1.08751	0.64347	−0.31873

Table 8. RMSE, MAE and bias values in each station for VARMA (*p*,*q*) models (RMSE: root mean squared error, MAE: mean absolute error, BIAS: bias or historial average error, VARMA: vector autoregressive moving average).

Station	**(*p*,*q*)**	**RMSE**	**MAE**	**BIAS**
6	(3,1)	2.10386	1.80502	1.44976
15	(1,0)	1.32604	1.10379	0.66137
22	(1,1)	2.64159	2.16374	0.80648
23	(1,2)	1.31553	1.03732	0.67987
25	(1,1)	1.39733	1.12313	0.57590
26	(1,0)	0.27808	0.16443	−0.01519
27	(2,0)	0.54810	0.28023	−0.19499
35	(1,1)	1.47019	1.29105	1.03933

RMSE values for station 15 are higher than those for station 06 overall. In this case, the lowest RMSE value was obtained for the ARIMA model, with 1.474. The SVR, MARS and MLR models have RMSE between 1.7 to 1.8 and the highest value was obtained for the MLP model, with 2.027. The MAE values are very close to each other, from 1.006 for ε-SVR to 1.195 for MLP, including time series models. However, the worst bias error values were obtained for the ARIMA model with −0.922 followed by VARMA with 0.6613. The rest of the models have a bias error close to zero, with positive values for MLR and MARS, together with VARMA.

Again, at station 22, the RMSE values are very similar for machine learning models, ranging from 1.220 for MLP to 1.304 for MLR. The time series models present much higher RMSE values, namely 3.008 for ARIMA and 2.642 for VARMA. The same occurs with the MAE values, which all remain at 0.7, varying from 0.704 for MLP to 0.788 for MLR. The MAE value for the time series models is 1.658 for ARIMA and the highest value is for VARMA, with 2.164. The bias error has values very

close to zero for all machine learning models. MLR and ε-SVR have positive values, while the MARS, ν-SVR and MLP models have negative values of the bias error. Regarding the bias in time series models, it is 0.864 for VARMA and −1.6490 for ARIMA.

Station 23 does not have such a differentiated error pattern between machine learning models and early series models, and the error values are more clustered. The lowest value for RMSE was reached for both SVR models with 1.095 and the highest value for MLP, with 1.360. Something similar occurs with the MAE error. The lowest error was achieved for ν-SVR with 0.498 and the highest value for MLP, with 0.621. The lowest bias error was reached for MLR with 10^{-4} and the highest value for VARMA with 0.679, with positive values for MLR and MARS and negative for the rest of the models.

At station 25, the RMSE error ranges from 1.015 for the MARS model to 1.397 for the VARMA model. Time series models have slightly higher RMSE than machine learning models. Regarding MAE, the time series models have approximately twice the value of the other models, the lowest value being 0.458 for ν-SVR and the highest for VARMA, with 1.123. The same goes for the bias error. All bias errors in machine learning models are negative and close to zero with the lowest value for MLR. On the other hand, the time series models present positive values; that is, the observed value is greater than that predicted by these models, with the highest bias value for VARMA, with 0.579.

At station 26, the ARIMA model is the one that differs the most from the rest of the models, which show similar behavior for the three errors. The lowest value of RMSE was reached for the MARS and SVR models with 0.24, followed by MLP and MLR with 0.25 and 0.27 for VARMA. The ARIMA model has an RMSE of 0.36. The smallest MAE error was achieved in SVR, with 0.097. The rest of the models vary from 0.104 for MARS to 0.164 for VARMA, reaching 0.319 for the ARIMA model. The smallest bias error is for MLR, with a positive 5×10^{-6} value. In the rest of the models the values are negative, except ARIMA, which reached a value of 0.243.

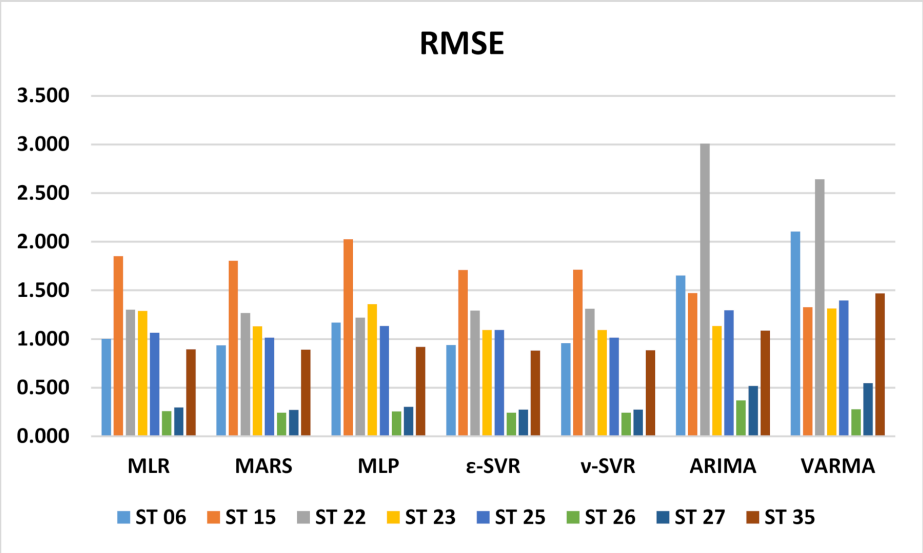

Figure 2. RMSE error for all models and stations.

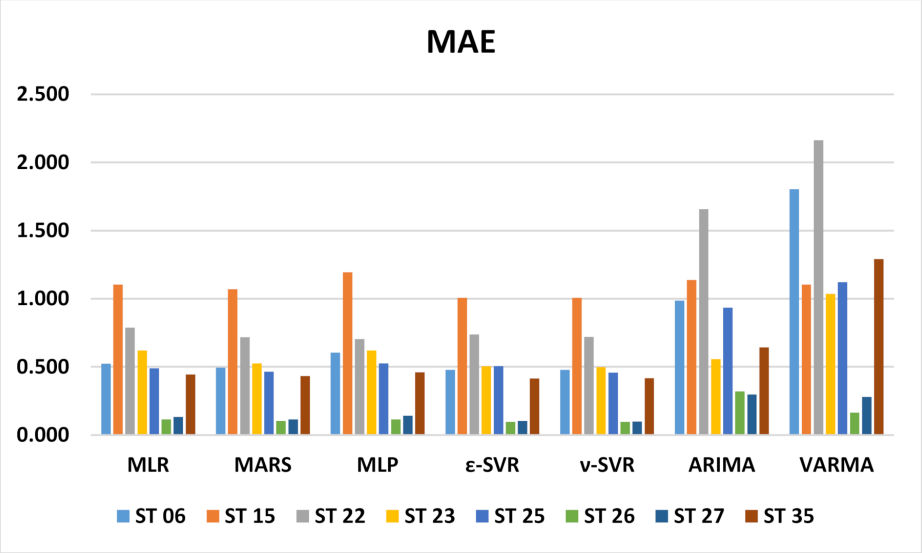

Figure 3. MAE error for all models and stations.

All machine learning models show similar behavior at station 27, with the lowest RMSE value for the MARS model being 0.271. The time series models have a somewhat higher RMSE with a higher value for the ARIMA model, with 0.517. The lowest value of MAE is 0.100 and was achieved with the ν-SVR model. The behavior is similar to that of the RMSE error. The highest value was obtained for ARIMA, with 0.298. Regarding the bias error, the models that perform better are MLR and MARS, with positive values of 10×10^{-5}. The rest of the models have a negative bias error and the highest value was obtained for the VARMA model, with −0.195.

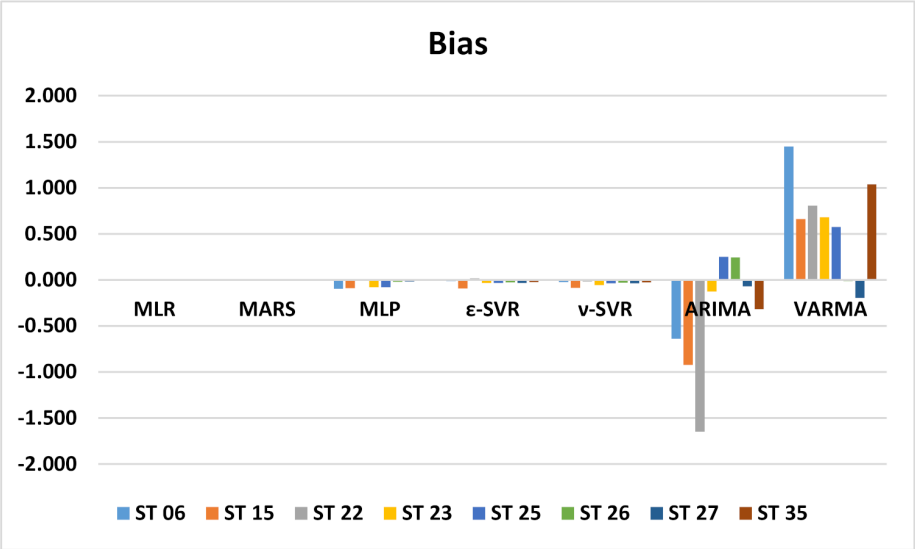

Figure 4. Bias error for all models and stations.

The behavior of the RMSE error in station 35 was very similar for all models, acquiring the lowest value for ε-SVR with 0.880 and the highest value for ARIMA, with 1.088. Machine learning models have a similar MAE error, the smallest being 0.416 for ε-SVR. Time series models have a slightly larger MAE error of 0.643 for ARIMA. The bias error is negative for all models except for VARMA, being very close to zero and the lowest value on the order of -10×10^{-4} corresponds to MLR. The RVS models show a bias error of −0.02, while the bias error of the ARIMA model is −0.318.

At stations 15 and 23, the MLP model has a very high RMSE in relation to the machine learning models, compared to the homogeneity of the RMSE error performance in the rest of the stations. Station

35 has a very homogeneous RMSE error for all models. In stations 06 and 22, the machine learning models also have a homogeneous RMSE, but there is a clear difference with respect to the time series models, whose RMSE is higher. Station 15 has the highest RMSE and MAE in machine learning models, but this does not happen in time series models. Stations 26 and 27 have clearly lower RMSE and MAE values in all models, including time series models. The worst bias error for the ARIMA model is found at station 22. This model is the one that, together with VARMA, has the highest bias error for all stations, in general. The SVM models, both ε-SVR and ν-SVR, perform in a very similar way.

4. Conclusions

This paper studies the relationship between four predictors and benzene in order to establish predictions at eight stations in the community of Madrid, Spain, using seven mathematical models: MLR, MARS, ε-SVR, ν-SVR, MLP, ARIMA and VARMA.

Stations 06, 15, 22 and 35 are stations that are located in the city center and have observations with an average concentration of benzene higher than the rest of the stations located in the east, on the outskirts of the city or near the airport.

The models were evaluated using RMSE, MAE and bias. The validation of the machine learning models was carried out using k times the cross validation with k = 5 and with 20% of the observations for the time series models.

The results showed that, in general, machine learning models are more effective at predicting than time series models, but this does not happen at all stations, since at station 15 the lowest RMSE occurs in the VARMA model. The highest error values occur for the time series models, except at station 15, where they were obtained for the MLP model.

MLR, MARS, SVR and MLP follow similar behavior patterns for all stations and stations 26 and 27 show the lowest errors.

The time series models, ARIMA and VARMA, present greater variations in the values obtained for the stations, not following the same pattern as the machine learning models.

Author Contributions: Conceptualization, L.A.M.G.; data curation, L.A.M.G., F.S.L., P.J.G.N. and L.Á.d.P.; formal analysis, L.A.M.G., F.S.L., L.Á.d.P. and A.B.S.; investigation, P.J.G.N. and A.B.S.; methodology, F.S.L., L.Á.d.P. and A.B.S.; validation, L.A.M.G., F.S.L. and A.B.S.; writing—original draft, L.A.M.G., P.J.G.N. and L.Á.d.P.; Writing—review and editing, F.S.L. and A.B.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. El-Hashemy, M.A.; Ali, H.M. Characterization of BTEX group of VOCs and inhalation risks in indoor microenvironments at small enterprises. *Sci. Total Environ.* **2018**, *645*, 974–983. [CrossRef] [PubMed]
2. Fan, Z.; Lin, L. Exposure science: Contaminant mixtures. In *Encyclopedia of Environmental Health*; Elsevier: Amsterdam, The Netherlands, 2019; pp. 805–815. [CrossRef]
3. Harrison, R.; Saborit, J.M.D.; Dor, F.; Henderson, R. Benzene. In *WHO Guidelines for Indoor Air Quality. Selected Pollutans*; World Health Organization: Geneva, Switzerland, 2010; pp. 15–53.
4. Stellman, J.M. Hydrocarbons, Aromatic. Encylopaedia of Occupational Health and Safety. Available online: https://www.iloencyclopaedia.org/part-xviii-10978/guide-to-chemicals/item/1052-hydrocarbons-aromatic (accessed on 3 July 2020).
5. ATSDR-Public Health Statement: Benzene. Published 2007. Available online: https://www.atsdr.cdc.gov/phs/phs.asp?id=37&tid=14 (accessed on 2 July 2020).
6. Sekar, A.; Varghese, G.K.; Ravi Varma, M.K. Analysis of benzene air quality standards, monitoring methods and concentrations in indoor and outdoor environment. *Heliyon* **2019**, *5*, e02918. [CrossRef] [PubMed]

7. Ndong, B.A.; Verdin, A.; Cazier, F.; Garcon, G.; Thomas, J.; Cabral, M.; Dewaele, D.; Genevray, P.; Garat, A.; Allorge, D.; et al. Individual exposure level following indoor and outdoor air pollution exposure in Dakar (Senegal). *Environ. Pollut.* **2019**, *248*, 397–407. [CrossRef] [PubMed]
8. Notario, A.; Gutiérrez-Álvarez, I.; Adame, J.A. Atmospheric benzene measurements in the main metropolitan and industrial areas of Spain from 2014 to 2017. *Atmos Res.* **2020**, *238*, 104896. [CrossRef]
9. Shinohara, N.; Okazaki, Y.; Mizukoshi, A.; Wakamatsu, S. Exposure to benzene, toluene, ethylbenzene, xylene, formaldehyde, and acetaldehyde in and around gas stations in Japan. *Chemosphere* **2019**, *222*, 923–931. [CrossRef]
10. Liu, C.; Huang, X.; Li, J. Outdoor benzene highly impacts indoor concentrations globally. *Sci. Total Environ.* **2020**, *720*, 137640. [CrossRef]
11. Rovira, J.; Domingo, J.L.; Schuhmacher, M. Air quality, health impacts and burden of disease due to air pollution (PM10, PM2.5, NO2 and O3): Application of AirQ+ model to the Camp de Tarragona County (Catalonia, Spain). *Sci. Total Environ.* **2020**, *703*, 135538. [CrossRef]
12. Health Effects Institute. State of Global Air 2017: A Special Report on Global Exposure to Air Pollution and its Disease Burden. Published 2007. Available online: https://ccacoalition.org/en/resources/state-global-air-2017-special-report-global-exposure-air-pollution-and-its-disease-burden (accessed on 8 July 2020).
13. ICSC 0015-Benzene. International Labour Office. Published 2017. Available online: http://ilo.org/dyn/icsc/showcard.display?p_lang=en&p_card_id=0015&p_version=2 (accessed on 8 July 2020).
14. Li, L.; Li, H.; Zhang, X.; Wang, L.; Xu, L.; Wang, X.; Yu, Y.; Zhang, Y.; Cao, G. Pollution characteristics and health risk assessment of benzene homologues in ambient air in the northeastern urban area of Beijing, China. *J. Environ. Sci.* **2014**, *26*, 214–223. [CrossRef]
15. Kiely, G. *Environmental Engineering*; McGraw Hill Education: New York, NY, USA, 2006.
16. Mannucci, P.M.; Harari, S.; Martinelli, I.; Franchini, M. Effects on health of air pollution: A narrative review. *Intern Emerg. Med.* **2015**, *10*, 657–662. [CrossRef]
17. Karimi, B.; Shokrinezhad, B. Air pollution and mortality among infant and children under five years: A systematic review and meta-analysis. *Atmos Pollut. Res.* **2020**, *11*, 61–70. [CrossRef]
18. Giovannini, M.; Sala, M.; Riva, E.; Radaelli, G. Hospital admissions for respiratory conditions in children and outdoor air pollution in Southwest Milan, Italy. *Acta Paediatr.* **2010**, *99*, 1180–1185. [CrossRef] [PubMed]
19. Zhou, H.; Wang, T.; Zhou, F.; Liu, Y.; Zhao, W.; Wang, X.; Chen, H.; Cui, Y. Ambient Air Pollution and Daily Hospital Admissions for Respiratory Disease in Children in Guiyang, China. *Front. Pediatr.* **2019**, *7*, 400. [CrossRef] [PubMed]
20. Sánchez Bayle, M.; Martín Martín, R.; Villalobos Pinto, E. Impacto de la contaminación ambiental en los ingresos hospitalarios pediátricos: Estudio ecológico. *Rev. Pediatr. Aten. Primaria.* **2019**, *21*, 21–29. [CrossRef]
21. Spycher, B.D.; Lupatsch, J.E.; Huss, A.; Rischewski, J.; Schindera, C.; Spoerri, A.; Vermeulen, R.; Kuehni, C.E.; Swiss Paediatric Oncology Group; Swiss National Cohort Study Group. Parental occupational exposure to benzene and the risk of childhood cancer: A census-based cohort study. *Environ. Int.* **2017**, 84–91. [CrossRef]
22. Talibov, M.; Sormunen, J.; Hansen, J.; Kjaerheim, K.; Martinsen, J.; Sparen, P.; Tryggvadottir, L.; Weiderpass, E.; Pukkala, E. Benzene exposure at workplace and risk of colorectal cancer in four Nordic countries. *Cancer Epidemiol.* **2018**, *55*, 156–161. [CrossRef] [PubMed]
23. Bentayeb, M.; Wagner, V.; Stempfelet, M.; Zins, M.; Goldberg, M.; Pascal, M.; Larrieu, S.; Beaudeau, P.; Cassadou, S.; Eilstein, D.; et al. Association between long-term exposure to air pollution and mortality in France: A 25-year follow-up study. *Environ. Int.* **2015**, *85*, 5–14. [CrossRef] [PubMed]
24. Standards-Air Quality-Environment-European Commission. Available online: https://ec.europa.eu/environment/air/quality/standards.htm (accessed on 8 July 2020).
25. Radojević, D.; Antanasijević, D.; Perić-Grujić, A.; Ristić, M.; Pocajt, V. The significance of periodic parameters for ANN modeling of daily SO 2 and NOx concentrations: A case study of Belgrade, Serbia. *Atmos Pollut Res.* **2019**, *10*, 621–628. [CrossRef]
26. Brunelli, U.; Piazza, V.; Pignato, L.; Sorbello, F.; Vitabile, S. Two-days ahead prediction of daily maximum concentrations of SO2, O3, PM10, NO2, CO in the urban area of Palermo, Italy. *Atmos. Environ.* **2007**, *41*, 2967–2995. [CrossRef]
27. Pérez, P.; Trier, A.; Reyes, J. Prediction of PM2.5 concentrations several hours in advance using neural networks in Santiago, Chile. *Atmos. Environ.* **2000**, *34*, 1189–1196. [CrossRef]

28. Cabaneros, S.M.; Calautit, J.K.; Hughes, B.R. A review of artificial neural network models for ambient air pollution prediction. *Environ. Model. Softw.* **2019**, *119*, 285–304. [CrossRef]
29. Yang, W.; Deng, M.; Xu, F.; Wang, H. Prediction of hourly PM2.5 using a space-time support vector regression model. *Atmos. Environ.* **2018**, *181*, 12–19. [CrossRef]
30. Wang, P.; Liu, Y.; Qin, Z.; Zhang, G. A novel hybrid forecasting model for PM10 and SO2 daily concentrations. *Sci. Total Environ.* **2015**, *505*, 1202–1212. [CrossRef] [PubMed]
31. Murillo-Escobar, J.; Sepulveda-Suescun, J.P.; Correa, M.A.; Orrego-Metaute, D. Forecasting concentrations of air pollutants using support vector regression improved with particle swarm optimization: Case study in Aburrá Valley, Colombia. *Urban. Clim.* **2019**, *29*, 100473. [CrossRef]
32. Abdullah, S.; Napi, N.N.L.M.; Ahmed, A.N.; Mansor, W.N.W.; Abu Mansor, A.; Ismail, M.; Abdullah, A.M.; Ramly, Z.T.A. Development of multiple linear regression for particulate matter (PM10) forecasting during episodic transboundary haze event in Malaysia. *Atmosphere* **2020**, *11*, 289. [CrossRef]
33. Gocheva-Ilieva, S.G.; Ivanov, A.V.; Voynikova, D.S.; Boyadzhiev, D.T. Time series analysis and forecasting for air pollution in small urban area: An SARIMA and factor analysis approach. *Stoch. Environ. Res. Risk Assess.* **2014**, *28*, 1045–1060. [CrossRef]
34. García Nieto, P.J.; Sánchez Lasheras, F.; García-Gonzalo, E.; de Cos Juez, F.J. Estimation of PM10 concentration from air quality data in the vicinity of a major steelworks site in the metropolitan area of Avilés (Northern Spain) using machine learning techniques. *Stoch. Environ. Res. Risk Assess.* **2018**, *32*, 3287–3298. [CrossRef]
35. Kulkarni, G.E.; Muley, A.A.; Deshmukh, N.K.; Bhalchandra, P.U. Autoregressive integrated moving average time series model for forecasting air pollution in Nanded city, Maharashtra, India. *Model. Earth Syst. Environ.* **2018**, *4*, 1435–1444. [CrossRef]
36. Cekim, H.O. Forecasting PM10 concentrations using time series models: A case of the most polluted cities in Turkey. *Environ. Sci. Pollut. Res.* **2020**, *27*, 25612–25624. [CrossRef]
37. Kumar, A.; Goyal, P. Forecasting of daily air quality index in Delhi. *Sci. Total Environ.* **2011**, *409*, 5517–5523. [CrossRef]
38. García Nieto, P.J.; Álvarez Antón, J.C. Nonlinear air quality modeling using multivariate adaptive regression splines in Gijón urban area (Northern Spain) at local scale. *Appl. Math. Comput.* **2014**, *235*, 50–65. [CrossRef]
39. Kisi, O.; Parmar, K.S.; Soni, K.; Demir, V. Modeling of air pollutants using least square support vector regression, multivariate adaptive regression spline, and M5 model tree models. *Air Qual. Atmos. Health* **2017**, *10*, 873–883. [CrossRef]
40. García Nieto, P.J.; Sánchez Lasheras, F.; García-Gonzalo, E.; de Cos Juez, F.J. PM10 concentration forecasting in the metropolitan area of Oviedo (Northern Spain) using models based on SVM, MLP, VARMA and ARIMA: A case study. *Sci. Total Environ.* **2018**, *621*, 753–761. [CrossRef] [PubMed]
41. Xu, Y.; Ho, H.C.; Wong, M.S.; Deng, C.; Shi, Y.; Chan, T.-C.; Knudby, A. Evaluation of machine learning techniques with multiple remote sensing datasets in estimating monthly concentrations of ground-level PM2.5. *Environ. Pollut.* **2018**, *242*, 1417–1426. [CrossRef] [PubMed]
42. Galatioto, F.; Zito, P.; Migliore, M. Traffic parameters estimation to predict road side pollutant concentrations using neural networks. *Environ. Model. Assess.* **2009**, *14*, 365–374. [CrossRef]
43. Van Buuren, S.; Groothuis-Oudshoorn, K. Journal of Statistical Software mice: Multivariate Imputation by Chained Equations in R. *J. Stat. Softw.* **2011**, *45*, 1–67.
44. Hastie, T.; Tibshirani, R.; Friedman, J. *The Elements of Statistical Learning. Data Mining, Inference, and Prediction*, 2nd ed.; Springer: Berlin/Heidelberg, Germany, 2009. [CrossRef]
45. Zelterman, D. *Applied Multivariate Statistics with R*; Springer International Publishing: Berlin/Heidelberg, Germany, 2015. [CrossRef]
46. Friedman, J.H. Multivariate Adaptive Regression Splines. *Ann. Stat.* **1991**, *19*, 1–67. [CrossRef]
47. Friedman, J.H.; Silverman, B.W. Flexible parsimonious smoothing and additive modeling. *Technometrics* **1989**, *31*, 3–21. [CrossRef]
48. Lasheras, F.; Nieto, P.; de Cos Juez, F.; Bayón, R.; Suárez, V. A Hybrid PCA-CART-MARS-Based Prognostic Approach of the Remaining Useful Life for Aircraft Engines. *Sensors* **2015**, *15*, 7062–7083. [CrossRef]
49. Kartal Koc, E.; Bozdogan, H. Model selection in multivariate adaptive regression splines (MARS) using information complexity as the fitness function. *Mach. Learn.* **2015**, *101*, 35–58. [CrossRef]
50. Milborrow, S. Notes on the Earth Package. 2019. Available online: http://www.milbo.org/doc/earth-notes.pdf (accessed on 8 July 2020).

51. Bishop, C.M. *Neural Networks for Pattern Recognition*; Oxford University Press: Oxford, UK, 1995.
52. Gardner, M.W.; Dorling, S.R. Artificial neural networks (the multilayer perceptron)-a review of applications in the atmospheric sciences. *Atmos. Environ.* **1998**, *32*, 2627–2636. [CrossRef]
53. Nagendra, S.M.S.; Khare, M. Artificial neural network approach for modelling nitrogen dioxide dispersion from vehicular exhaust emissions. *Ecol. Modell.* **2006**, *190*, 99–115. [CrossRef]
54. Ripley, B.D. *Pattern Recognition and Neural Networks*; Cambridge University Press: Cambridge, UK, 2014. [CrossRef]
55. Rojas, R. *Neural Networks. A Systematic Introduction*; Springer: Berlin/Heidelberg, Germany, 1996.
56. Hornik, K.; Stinchcombe, M.; White, H. Multilayer feedforward networks are universal approximators. *Neural Netw.* **1989**, *2*, 359–366. [CrossRef]
57. Sheela, K.; Deeppa, S. Review on methods to fix number of hidden neurons in neural networks. *Math. Probl. Eng.* **2013**, 1–12. [CrossRef]
58. Haykin, S. *Neural Networks and Learning Machines*, 3rd ed.; Pearson: London, UK, 2008.
59. Günther, F.; Fritsch, S. Neuralnet: Training of neural networks. *R J.* **2010**, *2*, 30–38. [CrossRef]
60. Riedmiller, M.; Braun, H. Direct adaptive method for faster backpropagation learning: The RPROP algorithm. In Proceedings of the 1993 IEEE International Conference on Neural Networks, San Francisco, CA, USA, 28 March–1 April 1993; IEEE: Piscataway, NJ, USA, 1993; pp. 586–591. [CrossRef]
61. De Juez, F.J.C.; Lasheras, F.S.; Roqueñí, N.; Osborn, J. An ANN-Based Smart Tomographic Reconstructor in a Dynamic Environment. *Sensors* **2012**, *12*, 8895–8911. [CrossRef]
62. Aggarwal, C. *Neural Networks and Deep Learning*; Springer International Publishing: Berlin/Heidelberg, Germany, 2018. [CrossRef]
63. Vapnik, V.N. *The Nature of Statistical Learning Theory*, 2nd ed.; Springer: Berlin/Heidelberg, Germany, 2000. [CrossRef]
64. Drucker', H.; Burges, C.J.C.; Kaufman, L.; Smola, A.; Vapnik, V. Support. Vector Regression Machines. 1996. Available online: https://papers.nips.cc/paper/1996/hash/d38901788c533e8286cb6400b40b386d-Abstract.html (accessed on 8 July 2020).
65. Kuhn, M.; Johnson, K. *Applied Predictive Modeling*; Springer: New York, NY, USA, 2013. [CrossRef]
66. Smola, A.J.; Schölkopf, B. *A Tutorial on Support. Vector Regression*; Kluwer Academic Publishers: Cambridge, MA, USA, 2004; Volume 14.
67. Chen, P.-H.; Lin, C.-J.; Schölkopf, B. A tutorial on nu-support vector machines. *Bus. Ind. Appl. Stoch. Model. Bus. Ind.* **2005**, *21*, 111–136. [CrossRef]
68. Steinwart, I.; Christmann, A. *Support. Vector Machines*; Springer: New York, NY, USA, 2008. [CrossRef]
69. Schölkopf, B.; Smola, A.J.; Williamson, R.C.; Bartlett, P.L. New support vector algorithms. *Neural Comput.* **2000**, *12*, 1207–1245. [CrossRef]
70. Chang, C.-C.; Lin, C.-J. Training ν-Support Vector Regression: Theory and Algorithms. *Neural Comput.* **2002**, *14*, 1959–1977.
71. Box, G.E.P.; Jenkins, G.M.; Reinsel, G.C.; Ljung, G.M. *Time Series Analysis. Forecasting and Control*, 5th ed.; Wiley: Hoboken, NJ, USA, 2015.
72. Shumway, R.H.; Stoffer, D.S. *Time Series Analysis and Its Applications with R Examples*, 4th ed.; Springer: Berlin/Heidelberg, Germany, 2017.
73. Montgomery, D.C.; Jennings, C.L.; Kulahci, M. *Introduction to Time Series Analysis and Forecasting*, 2nd ed.; Wiley: Hoboken, NJ, USA, 2015.
74. Pankratz, A. *Forecasting with Univariate Box-Jenkins Models: Concepts and Cases*; Wiley: Hoboken, NJ, USA, 1983. [CrossRef]
75. Suárez Sánchez, A.; Krzemień, A.; Riesgo Fernández, P.; Iglesias Rodríguez, F.J.; Sánchez Lasheras, F.; de Cos Juez, F.J. Investment in new tungsten mining projects. *Resour Policy.* **2015**, *46*, 177–190. [CrossRef]
76. Hyndman, R.J.; Athanasopoulos, G. *Forecasting: Principles and Practice*; OTexts: Melbourne, Australia, 2018.
77. Ohri, A. *R for Business Analytics*; Springer: New York, NY, USA, 2013.
78. Tsay, R.S. *Multivariate Time Series Analysis: With R and Financial Applications*; Wiley: Hoboken, NJ, USA, 2014.
79. Willmott, C.J.; Ackleson, S.G.; Davis, R.E.; Feddema, J.J.; Klink, K.M.; LeGates, D.R.; O'Donnell, J.; Rowe, C.M. Statistics for the evaluation and comparison of models. *J. Geophys. Res. Space Phys.* **1985**, *90*, 8995–9005. [CrossRef]
80. Ranadip, P. *Predictive Modeling of Drug Sensitivity*, 1st ed.; Academic Press: Cambridge, MA, USA, 2016.

81. Willmott, C.J.; Matsuura, K. Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assesing average model performance. *Clim. Res.* **2005**, *30*, 79–82. [CrossRef]