USE OF NON-INDUSTRIAL ENVIRONMENTAL SENSORS AND MACHINE LEARNING TECHNIQUES IN TELEMETRY FOR INDOOR AIR POLLUTION

Gomez Carlos[1], Fonseca Valeria[2] and Valencia Guillermo[1]
[1]Universitaria Agustiniana, Facultad de Ingeniería, Colombia
[2]Universidad Nacional de Colombia, Facultad de Ciencias, Colombia
E-Mail: carlos.gomezr@uniagustiniana.edu.co

ABSTRACT

Classic telemetry systems are usually limited to taking a large variety and number of measurements focused on outdoor air, large concentrations of automobiles and factories, but indoor air pollution has not been addressed with the same intensity. Any of the telemetry techniques generates a large amount of data that implies a great challenge for its analysis; this work demonstrates the application of machine learning techniques in telemetry systems focused on the study of indoor air pollution. A telemetry system has been developed which collects data from the environment which are concentrated in a centralized storage unit and are analyzed by automatic learning techniques that can predict the historical behavior of the CO_2 concentration based on the variables of the environment, past records of CO_2 and independent variables such as time of day, which creates an important tool to detect anomalous behavior in air pollution by $CO_{2.}$ Results of the development of several data prediction models based on Kernel methods to estimate a regression are presented.

Keywords: telemetry, CO_2, environment, server, machine learning, kernel methods, least squares support vector machines (LS-SVM) LSSVM, Gaussian processes.

INTRODUCTION

The development of smart cities requires more and more tools to capture all types of data and the development of analytical techniques to obtain the greatest amount of knowledge about these data. Regarding the behavior of air pollution, it is a difficult to control situation and one of major changes, besides, these changes can occur on a small scales and not in a generalized manner, this means, it is possible to have high levels of air pollution few meters from areas with normal contamination levels.

The vast majority of air quality monitoring approaches have focused on monitoring the outdoor environment near places of high pollution such as in areas with industrial factories, however given the particular characteristics of air pollution, since the vast majority of the time humans remain in closed indoor environments, it is necessary to complement this approach with a simple indoor analysis, using techniques with affordable sensors and a data centralization structure that allows their storage and analysis in order to apply techniques which permit to automatically learn from them.

LITERATURE REVIEW

The technological development oriented to greater automation processes has had to develop in the last decades different techniques to make measurements of physical variables in a non-physical and automatic way.

[1] exposes one of the first works on environmental monitoring, but oriented to the sampling and analysis of the environment in nuclear plants under normal conditions and in nuclear accidents. The author addresses general issues of environmental monitoring such as sampling, accuracy and operational difficulties. This work takes as a framework the importance of maintaining a monitoring system that guarantees *Rem's* allowed levels of exposure to people in the nuclear power industry. The author emphasizes his work on the importance of quantifying measurers and samples to achieve more accurate estimates.

In the work of [2], he analyzes the development of environmental monitoring from 3 perspectives: the social / political, the point of view from the implementation (standardization) and the technical point of view. Environmental monitoring is a strong area of work for governments and international regulatory bodies, which act from international organizations such as the United Nations. Manufacturers, government technical organizations, laboratories and others strive to implement environmental data monitoring and analysis models for product development or policy support from other government entities.

There are a lot of examples of these activities that also join their efforts on pollution remediation programs. In terms of technological development, environmental monitoring is also a work area that constantly produces advances in sensors and in a great variety of techniques and variables to achieve greater and better analyzes of different factors of environmental pollution. [2]

With the development of different communication networks, various systems for remote monitoring of different variables have begun to be deployed, [3] specifically shows the monitoring of renewable energy sources, using 3G mobile network infrastructure and a centralized private server scheme.

In Brazil, a working group on hydroelectric reserves has set up an environmental monitoring system - EMS, capable of measuring a large number of variables associated with water reserves, such as: chlorophyll-a, water surface temperature (ºC), temperature of the water column (ºC), turbidity, pH, dissolved oxygen concentration (mg L-1), electrical conductivity (µS cm-1), etc., the author has documented the great difficulty of maintaining active telemetry systems in water systems by variables associated with the degradation of sensors and components, and the difficulty of transmitting data from some remote stations by satellite, however, it highlights how the data collected and their disposition over time of large periods is a very important source of information to understand the dynamics of aquatic systems. [4]

[5] proposes a telemetry system for private use in measurements of environmental conditions (temperature and relative humidity) in different locations of a house. A regression analysis with a decision tree is performed in order to obtain the CO_2 value according to the temperature and humidity values of the different parts of the house as well as the date and time. From the analysis options the so-called "Random Forest" has been used, with which they have achieved a close approximation to the real value of CO_2.

In [6] the authors address the need to monitor pollutant gases inside buildings, given the complexity with which air pollution develops and the major changes that can occur even at small scales of space. For the above, they propose the development of an autonomous monitoring system consisting of a robot that makes measurements of different gases and conducts itself autonomously thanks to the use of several positioning systems. This project has achieved a mobile and autonomous platform that can measure contaminating gases such as methane (CH4), ethylene (C2H4), ammonia (NH3), benzene (C7H8), LPG (C4H10), carbon dioxide (CO2), carbon monoxide (CO) and nitrogen oxide (NOx) in a certain area, with which it manages to map the different states of pollution in small spaces.

With the emergence of smart city paradigms and the internet of things IoT, new telemetry applications have been addressed. [7] addresses the issue of monitoring in the context of smart cities and smart buildings. They work on the problems that exist in constructions made with wood. [7] shows the possibility of carrying out a monitoring system that detects and registers the moisture level of the wood, and thus creates a system of early warnings and historical information of this variable, given the great impact that wood has on the changes of the environment leading to the deterioration of the quality of structures and possible risks of integrity of the same.

Environmental measurement techniques

[8] created in 2016 a device that allows the measurement of concentrations of CO and HCHO in internal environments, if the concentration of these gases exceeds certain ranges, the device will notify through text messages and emails. In addition, the measurements are constantly transmitted through the internet so that it can be consulted from any computer or cell phone that has access to the internet. A Marvell 88MW302, sensors and the Amazon Web services (AWS) cloud platform were used to create the device.

When you want to know the air quality in external environments, it is necessary to use several devices that perform measurements simultaneously. An example of this is the system proposed by Joy Dutta [9] which is based on several devices reporting air quality measurements to a cloud service. With the collection of this information it creates a map in which you can see pollution levels in real time.

The system proposed by [9] is composed of an air quality monitoring device (AQMD), an application for mobile devices with an Android operating system and a service in the cloud. For the AQMD he used the air quality sensor MQ-135 and the bluetooth module HC-05 to communicate with a smartphone. The mobile application is responsible for receiving the information, sending it to the cloud service and displaying the air quality map. The service in the cloud is called thing speak and it allows to receive the information from the different devices, analyze it, generate graphs and the map that can be seen from the smartphone application.

Taking into account the energy consumption of a device such as the one proposed by [9] and the need for it to perform measurements constantly, Mitar Simic [10] proposes a system supported by a battery system and solar panels. Its design, like that of Joy Dutta, involves the MQ-135 sensor. These projects differ in that the last sends the information directly to the cloud and measures environmental variables such as the presence of volatile organic compounds (VOC), relative humidity, water temperature and PH levels. For the above, it uses sensors SHT11, LM35 and a thick resistive film based on TiO2.

With the previously mentioned as basis, we can think of a device that has air quality sensors, powered by solar panels and that can report the measurements made using Wi-Fi networks as proposed by [11] in a development called Clean WiFi, which would give citizens access to information regarding the air quality of the network to which they are connected, raising awareness about the importance of reducing environmental pollution.

This development has enabled the measurement of variables remotely and automated, however it is necessary to apply statistical analysis techniques that can learn from these data and create greater properties for their use.

Statistical learning techniques

The theory of statistical learning is based on the modeling of the existing but unknown relationship between input type variables and output type variables. On one hand, we assume the existence of a link function f such that y = f(x)+e, where y is the output variable, x refers to the vector of input variables, and e is a random error term . After assuming the existence of this function, the objective is to find it, or said within this context, to learn it based on the data collected. For the purposes of the present problem, regression estimation techniques are addressed, since we need to find a function that relates the environmental variables of input x with CO_2 as the output variable y, whose values are real [13].

Among the most used methodologies nowadays for estimating a regression are artificial neural networks (ANN) and Kernel methods, which include different types of support vector machines (SVM). There are other methods to address the same problem, such as classical linear regressions, decision trees, as presented in [5], or closer neighbors. However, ANN and the types of SVM have shown a highly favorable performance not only in adjusting to the behavior of observed data, but especially to the quality of prediction that can be obtained. Among the multiple applications in which these two methods have taken place, problems of digit recognition, of images or modeling of radioactive activity in the human body are found [12].

The models corresponding to ANN are parametric models that propose a structure of multiple layers that make it possible to approximate functions f of high complexity. For the estimation of the regression by means of these models there are algorithms such as the backward propagation that allow working with non-linear models in the parameters to be estimated. The optimization problem for estimating neural network models is based for starters on minimizing the mean square error, where the error is given by the difference between the output variable *y*, and the function that approximates it f.

On the other hand, the models given by the Kernel methods and in particular by the SVM methodology are non-parametric models that also allow f functions of high complexity to be approximated with computational and theoretically more efficient algorithms. Given a nonlinear relationship between x and y, the basic idea that describes the support vector method is to transform the input variables x from the original space to a larger space (even of infinite dimension) where the relationship can be established with a linear equation to later estimate it. Based on the above description, the optimization problem for this method is given by the minimization of the coefficient standard of the linear function subject to such function providing a minimum adjustment error. The most recent version of SVM is the least squares support vector machines (LSSVM) where the adjustment error is a quadratic error. This methodology allows a computationally simpler optimization with equal performance in prediction and has a direct equivalence with squared minimums regularized in ridge regression or with maximizing the likelihood of the predictive distribution in a Gaussian process [16].

The greatest contrast that exists for the choice of ANN or SVM as estimation methods for the regression lies in the ease and computational efficiency for each one. For the case of ANN, the estimation algorithms make use of the gradient of the function to be estimated, and in this case only local optima are assured. For its part, in the case of SVM, optimization is possible through linear or quadratic programming, which ensures convergence to global optima [15].

The implementation of any of these methods involves the study of the compensation between the adjustment in the observed data and the quality of prediction in new observations. The exercise of finding function f for the present work applies techniques known as cross-validation for training models which allow finding a model that compensates for these two characteristics.

MATERIALS AND METHODS

For the development of a system of automatic measurement of environmental variables and analysis of statistical learning for the prediction of environmental contamination, it is necessary to develop a device to measure variables and transport data.

The designed device is based on a microcontroller system, gas sensors, light, temperature and relative humidity of non-industrial type; in addition to communication modules. The description of these elements can be found in Table-1.

Table-1. Device components.

ELEMENT	DESCRIPTION
Microcontroller	ATmega328
Gas sensor	MQ-135
Humidity and temperature sensor	DHT11
Light sensor	Phototoresistor
Ethernet communications module	ENC28J60

The MQ-135 sensor is capable of measurements of NH3, NOx, alcohol, benzene and CO_2; the prototype designed uses this sensor to make measurements of carbon dioxide ($C0_2$). To carry out this measurement, the sensitivity characteristics to different gases of this sensor were taken into account, which the manufacturer specifies through the Figure-1.

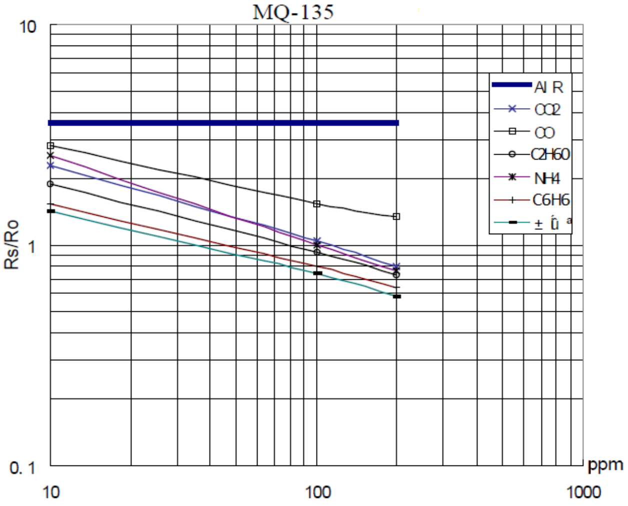


Figure-1. MQ-135 sensor characteristic curves.

Rs is the resistance of the sensor which varies depending on the concentration of the gas and Ro is the internal resistance of the sensor when the concentration of NH3 in clean air is 100 ppm. To obtain the CO_2 concentration, from the previous image the values were taken from the straight line for CO_2 and an exponential regression was made obtaining equation 1.

$$C02 = 219,2e^{-0,24Rs/Ro} \tag{1}$$

Rs can be calculated by knowing the load resistance connected to the sensor (RL) and the voltage delivered by the sensor (vin), since Rs and RL are connected as a voltage divider and the voltage delivered by the sensor is the RL voltage. Taking into account the above, we would obtain equation 2.

$$R_S = R_L\left(\frac{5}{vin} - 1\right) \tag{2}$$

Ro is a resistance that is constant and according to the sensor manufacturer it must be calculated after the sensor is on for 24 hours. Once Ro is known, it is possible to proceed with $C0_2$ measurements using the MQ-135 sensor.

The measurements of the temperature and humidity sensor, with reference DHT11, give measurements in degrees Celsius and relative humidity. To perform the light measurement, a photoresistor in series with a load resistance was used to form a voltage divider, so in order to find the value of the photoresist; the above equation can be used. To calibrate the light measurements these were compared with the values thrown by a luxometer.

The microcontroller function is to receive the information from the sensors and transmit the measurements made to the database. For this the ethernet module ENC28J60 was used.

Transport and storage of data platform

The microcontroller devices have communication capabilities through the TCP / IP protocol, with which they are able to establish communications with remote devices, protocol servers compatible with TCP/IP. A solution based on the client server architecture was implemented, as shown in Figure-2, with the following functions:

- Interface for receiving communications from the nodes, based on the HTTP protocol.
- Storage of measurement data in a relational database server.
- Access to data by predictive analytics software.

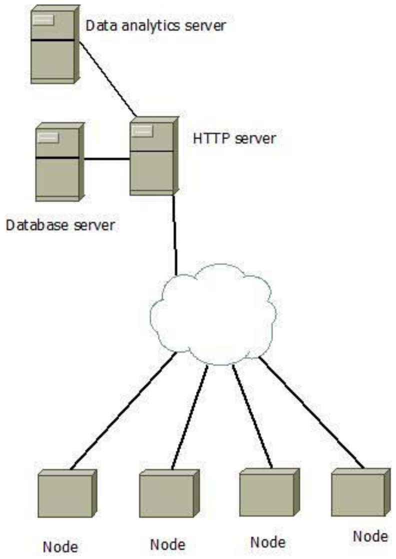


Figure-2. Diagram of the data reception and storage system.

Data analysis

The component of the data analysis of this project is the construction and use of a prediction model of CO2 levels. The model was built using the input variables Temperature, Humidity, Light and Hour. The objective is to obtain a model that allows a user to indicate certain levels or values of the input variables and obtain a prediction of the CO_2 value. For this, Kernel methods (LSSVM and Gaussian processes) were used [16].

The problem of estimating the regression is to find a function *f* such that $y = f(x)+e$, where y is the output variable, x refers to the vector of input variables, and e is a random error term. To find this function, we start with an optimization problem where the main objective is to minimize an error function (-loss function-) that measures the discrepancy between observed values of y and values of y predicted by $f(x)$.

Kernel methods have shown a highly favorable performance in the prediction quality of the values of y and based on x. Within this family are the developments of support vector machines (SVM) in different modalities, which differ essentially in the type of error function within the minimization problem. In this work we consider the least squares support vector machines (LSSVM) and their equivalence with the Gaussian processes, where the error function L is the quadratic error of the form:

$$L = \sum_{i=1}^{n} e^2{}_i,\ e = y - f(x) \tag{3}$$

where the subscript i corresponds to the data set that supports the training of the regression model.

In the following section we present the formulation, training and use of the trained model. The training of the model is done taking a subset of the data that is called training data and the use of the model is done in the remaining subset of data called evaluation or test data.

Regression model formulation in Kernel methods

Given a series of observations*{(x₁, y₁),...,(xₙ, yₙ)}*, where *x* is of dimension *p* corresponding to the number of input variables, and *y* is a real value, the regression model provided by the Kernel methods for a new input vector *x* is of the form:

$$f(x) = \sum_{i=i}^{n} \alpha_i K(x_i, x) = \alpha' K(x) \qquad (4)$$

where i are called weights or regression coefficients and $K(x_i,x)$is a function called Kernel. The essence of this regression model lies in the possibility of training models with good adjustment when there is a nonlinear and unknown relationship between *x* and *y*. For the SVM models in estimation of a regression there are several Kernel functions from which can generate a prediction model. Among the best known are the polynomial kernel and the radial base kernel (RBF) [12, p 460]. The functional form of the last is given by:

$$K(x_i, x) = exp(-\gamma(||x_i - x\,||)^2) \qquad (5)$$

where ‖.‖ is the rule function in a space of finite dimension *d* and *y* is the only parameter that is needed to define an RBF Kernel. Note that this kernel expresses the degree of discrepancy that exists between point xi and *x* by means of the difference rule. When the difference tends to 0, the value of the Kernel tends to 1 while when the difference increases, the value of the Kernel decreases exponentially. This behavior is crucial within the functional form of the prediction model given in equation (3) especially when the unknown functional relationship between *x* and *y* is non-linear. In the present work, this Kernel is used. In applications of prediction models, their use is justified especially by fulfilling certain theoretical conditions within the estimation problem which in turn contributes to a high quality of prediction of the model in practice [16, p 43].

Solution of the optimization problem

Once a Kernel is available to determine the prediction model, the problem is to determine the weights i. When the optimization problem is determined by a quadratic error function as presented above and it is established that the relationship between *x* and *y* is of the form *y = f(x) e*, the solution of the weights and corresponding prediction for the model of Equation (3) based on the set of observed data is given by [16, p108]:

$$\alpha = (\Omega + I/\sigma)^{-1} y_{obs}\,; \qquad \hat{f}(x) = \hat{\alpha}' K(x) \qquad (6)$$

where *α=(α1,...,αn)', Yobs=(y1,...,yn)',K(x)=(K(x1,x),...,K(xn,x))', Ω* is a matrix of size nxn with entries *Ωjk=K(xj,xk)* and *I* is the identity matrix of the sames size.

The parameter is a parameter that determines the importance of the minimization of the loss function L with respect to the control over the magnitude of the values of the weights. In the LSSVM theory this parameter is known as a regularization parameter [16, p 108] and in the Gaussian processes it corresponds to the a priori variance of the error*e* [21].

Autoregressive type prediction model

One of the extensions for prediction models are nonlinear models of autoregressive type with exogenous variables [16, p 24]. In this paper we propose the construction of the CO_2 prediction model based on environmental variables (exogenous) as well as lagged values (autoregression) of the output variable. It is proposed to model yt (values of CO_2 in time t), with base in the entry vector of form

$$x_t = (y_{t-1,} y_{t-2}, \ldots, y_{t-q}, u_t^1, u_t^2, \ldots, u_t^p) \qquad (7)$$

Where q is the order of auto regression and p is the number of environmental variables represented in time t such as ut1, ut2,...,u_t^p.

Training the prediction model

Once the problem has been formulated, the practical work consists in estimating or training the prediction model and then using it with new data. The training of the model consists mainly in calculating the weights, whose solution is given in equation (5). However, for the model proposed in this paper, it is necessary first to completely define not only the Kernel function, but also the autoregressive order of the model. In the case of the RBF Kernel, the definition is subject to the value of the parameter. These two parameters (, q) are known as tuning parameters and it is possible to determine them during the training of the model by means of cross validation. The description of this technique can be found in [16, p12]. This procedure allows calculating the prediction error *(CVerror)* based on the formula:

$$CVerror = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_{(-i)})^2} \qquad (8)$$

where y(-i) is the prediction of and when the data i has been excluded during the cross-validation. This procedure provides *CVerror* values as a function of the tuning parameters, with which it is possible to determine its optimum values.

Additionally, the a priori variance parameter is also considered in various applications as a tuning parameter. In Gaussian process applications this parameter usually takes initial values that oscillate between 0 and 1.

In this work we propose to set this value to 0.001 to perform a search of (ɣ, q) computationally faster [19].

Variable standardization

The standardization of the scale of variables is an additional aspect in the training process of the present prediction model. When the input variables and the output variable are measured in different units, a normalization is done previously, which consists of transforming them linearly to a mean scale of 0 and variance 1 by means of the formula:

$$u^* = \frac{u - m(u)}{sd(u)} \tag{9}$$

where *m(u)* represents the average of the variable *u* and *sd(u)* its standard deviation. This same transformation is done for the variable *y*. This new scale allows the measure of discrepancy present in the Kernel function not to be influenced by the units of measurement of the variables. This preprocessing is common in multiple machine learning problems [16, p91]

Measuring the prediction quality of the model

After training a final model with optimal values of the tuning parameters, several measures are calculated to determine the prediction quality of the model. The measures used here are the average prediction error and the R2 statistic. These measurements are obtained by making use of the prediction from the chosen model in the evaluation data set. The average prediction error (*TEST error)* is given by:

$$TESTerror = \sqrt{\frac{1}{m}\sum_{j=1}^{m}(y_j - \hat{y}_j)^2} \tag{10}$$

where *m* is the number of observations in the evaluation set. The R^2 statistic is determined by the correlation between the observed values and the predicted values in the same set:

$$R^2 = corr(y, \hat{y}) = \frac{cov(y, \hat{y})}{sd(y)\, sd(\hat{y})} \tag{11}$$

where *cov* is the covariance and *sd* is the standard deviation. The R^2 is a measure between 0 and 1 that determines the alignment between vectors of values. A good prediction quality is given by values of R^2 close to 1. The prediction error is in the same units of the variable *y* and therefore it can be interpreted as the average of the deviation between the two values.

Prediction models Voting method

In this paper, it is proposed to integrate voting methods to obtain a more robust prediction in new observations. Voting methods is one of the ways of assembling models [20]. These consist of deciding which prediction to choose from the predictions given by several trained models for the same input observation and adopted when there are several data sets with different characteristics and therefore the models can lead to different predictions. In the present work there are measurements made in different spatial locations under different instruments. Therefore, it is convenient to apply a voting method for models trained in several data sets.

The voting strategy proposed in this paper is based on the similarity between the mean and standard deviation of the variable and for a set of new observations with respect to each training set. Given a training set

$$D_g = \left\{(x_1, y_1), \ldots, (x_{n_g}, y_{n_g})\right\} \tag{12}$$

where n_g is the number of observations in the set *g(g=1,...G),* the mean is denoted $m_g(y)$ and the standar deviation $sd_g(y)$ from the variable *y*in the group as:

$$m_g(y) = \frac{1}{n_g}\sum_{i=1}^{n_g} y_i \;;\; sd_g(y) = \sqrt{\frac{1}{n_g - 1}\sum_{i=1}^{n_g}(y_i - m_g(y))^2} \tag{13}$$

Being $f_g(x)$ the model trained with the set of data D_g. The measure of similarity between mean and standard deviation is defined as an Euclidean distance of the form:

$$dist_g = \sqrt{(m_g(y) - m(y))^2 + (sd(y) - sd(y))^2} \tag{14}$$

where *m(y)* and *sd(y)*are the mean and standard deviation of the new data set to be predicted. In the case of predictions in the future, the input data of the variable *y* are the previous values observed yt-1, yt-2,...,yt-q and from them it is calculated m(y) y sd(y). The winning model is the one for which the distgis minimum. The prediction for a new data set is the value given by the winning model.

Use of the prediction model

The use of the prediction model is the calculation of the prediction for a new data set. This new set of data can be in two forms: New set of evaluation or vector of lags of and to predict future values with values of the fixed variables. The first form corresponds to the use of the model that is done during cross validation and evaluation of the model and the second corresponds to new observations to predict the future. Since the training of the model is done with standardized variables, the final prediction is the inverse transformation of the standardized prediction with the values of mean and standard deviation of the training set. The use of the model in both modalities is described below.

Use of the model in cross validation and evaluation data

Being x_t the input vector of an observation in time *t* given by$x_t = (y_{t-1}, y_{t-2}, \ldots, y_{t-q}, u_t^1, u_t^2, \ldots, u_t^p)$ and $f_g(x)$the winning model to predict in this data set, the final prediction for x_t is given by:

$$y(x_t) = m_g(y) + sd_g(y) * f_g(x_t) \tag{15}$$

Use of the model for a future value

To make predictions of time $t+1$, the input data are fixed values of the variables$u^1,u^2,...,u^p$ and a vector of at least$q$lags of the variable$y$ $(y_t,y_{t-1},...,y_{t-q+1})$. The input vector to perform the prediction and the corresponding prediction are given by:

$$x_{t+1} = (y_t, y_{t-1}, ..., y_{t-q+1}, u^1, u^2, ..., u^p);$$

$$y(x_{t+1}) = m_g(y) + sd_g(y) * f_g(x_{t+1}) \tag{16}$$

Predictions of time greater than $t+1$ can be established recursively by taking predictions from $t+1$. In this case, it is advisable to make only a small number of predictions with the fixed values of u^1, u^2,...,u^p. For more distant times, it is possible to restore the values of these variables to continue evaluating the predictions.

Software

The development of model training was done in R version 3.4.1 [18] making use of the kernlab library [19].

RESULTS AND DISCUSSIONS

A protocol for testing and data collection was designed where 4 environmental measurement nodes were implemented in residential interior locations, far from obvious sources of contamination such as industrial zones, in the city of Bogotá, Colombia. Data collection was performed throughout the month of September. Table-2 presents the general data collected:

Table-2. Summary of obtained data.

Number of monitored nodes	**4**
Number of variables monitored	5
Time range of the samples	30 días
Sampling date	September/2017
Average number of samples per node	50000

After compiling the data, they were used to design prediction models and its evaluation.

Data analysis

To train the prediction models, three sets of data were available. Each data set contains measurements of the CO_2 variables, temperature, humidity, light and the time of measurement, which takes values between 1 and 24. In each data set the measurements are given with intervals of 5 minutes. To train the model in each one, a division was made between training data and evaluation data of 70% and 30%, respectively. The number of observations for training, the mean and the standard deviation of each set are listed in Table-3:

Table-3. Number of observations for training, the mean and standard deviation of each set.

	Data1	**Data2**	**Data3**
N	2628	1557	484
Mean	209.36	371.35	760.84
SD	128.66	109.05	201.72

These summary measures show the dissimilarity that exists in the average and dispersion of each data set, which proposes different distributions for each one of them. These measures are used during the voting process described in the previous section.

Training the models

To carry out the training of each of the models, all the variables were standardized and ranges were established to refine the parameters (γ, q). For the lag order parameter q, it was proposed to search the values 10, 20, 30, 40 corresponding to past CO_2 values of up to 50, 100, 150 and 200 minutes. For the parameter $\gamma$of the RBF Kernel, different ranges were established, including values between 0.0001 and 10. The range that presented the best results in cross validation was the one with values between 0.0001 and 0.1 for the three data sets. The results for this range are presented here. Cross-validation was done by combining each lag order with each value in theγ parameter range. The cross-validation results for the data set1 are shown in Figure-3.

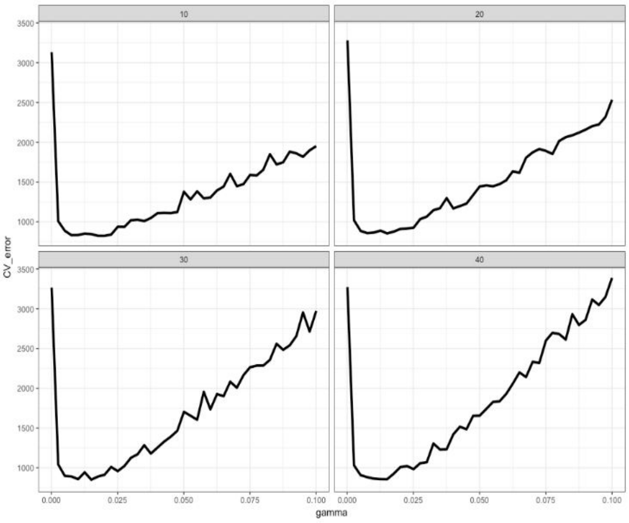


Figure-3. Results of cross validation.

The behavior of the cross-validation error was similar in the 4 established orders. The minimum that the graphs present corresponds to the optimum value of the parameter *γ*in each case. The increase that occurs from this value corresponds to the phenomenon of over fitting, where the model tends to adjust only to the training data and therefore produces bad predictions in new data. The results of the cross-validation error in the second data set were similar. For the case of the third data set there was an almost constant behavior after the greater decrease of the curve.

Based on the cross validation, the optimal values of (*γ*, q) were established for each data set. The optimal values of *γ* for each data set in each lag order and the corresponding cross-validation error are shown in Tables 4 and 5.

Table-4. Optimal values of *γ*.

OPTIMUS	Data1	Data2	Data3
10	0.0201	0.0176	0.0476
20	0.0151	0.0126	0.0550
30	0.0151	0.0176	0.0476
40	0.0151	0.0126	0.0376

Table-5. Cross validation error values.

CVerror (OPTIMUS)	Data1	Data2	Data3
10	846.38	1449.85	1397.13
20	852.87	1495.56	1565.78
30	850.04	1487.65	1573.56
40	855.92	1468.11	1554.99

Table-3 shows the stability of the parameter for the different lag orders. In the first two sets, the optimum value was between 0.01 and 0.02, while for the third, it was approximately 0.05. Given these values of γ, Table-4 shows the magnitudes of the prediction error in cross validation. In all three cases, the minimum error values were presented for 10 lags. Based on these results, the structure of the three prediction models is a model of autoregressive order equal to 10 with the respective values of γ. It is observed that the structure of the prediction model in the three cases is the same and the differences between the three data sets fall, in effect, on the measures of centrality and dispersion (mean and standard deviation) that likewise influence the prediction final.

Evaluation of trained models

For the evaluation of performance of the models, summary measures presented in the previous section were calculated. The results for each data set are shown in Table-6.

Table-6. Evaluation of trained models.

	Nobs train	CR Error	R2 train	Nobs test	Test error	R2 test
Data1	2628	846.38	0.98	1126	922.19	0.95
Data2	1557	1449.85	0.95	667	6031.88	0.88
Data3	484	1397.13	0.99	208	1762.71	0.90

The summary measurements report the prediction error when using the models trained in the assessment set. As a reference measure, the R^2 statistic is presented, which in all cases was above 0.88, which shows a high quality of prediction of the models. The first model presented more favorable results than the other two models, especially due to the presence of potential outliers in the evaluation set.

For the case of model 3, two extreme values were excluded in the training set. Figure-4 shows the adjustment and quality of prediction in the training and evaluation data for the case of model 1, where good prediction quality is observed. Similar results were presented for models 2 and 3.

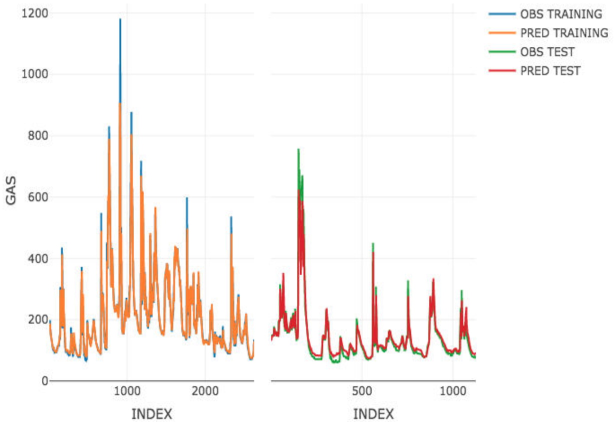


Figure-4. Adjustment and prediction quality.

Use of the model in a new data set

In addition to the evaluation of the models presented above, the model was applied in a set of new data obtained in which the sturdiness of the models could be clearly determined thanks to the voting strategy and obtaining quality prediction measurements corresponding to this type of assembly of the models. The summary measures are shown in Table-7.

Table-7. Summary of measures of new data set.

Nobs test	Test error	R2 test
3194	11004.78	0.79

In this data set there was presence of potential outliers whereby a greater prediction error was obtained. Despite this, the quality measures showed an R^2 of 0.79 which still corresponds to a good alignment between observed and predicted values. The performance of the prediction with voting can be observed in the correspondence of the predictions with the values observed in Figure-5.

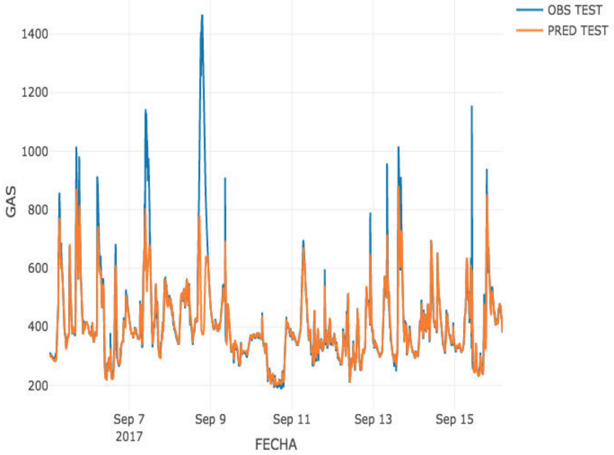


Figure-5. Prediction performance through voting.

Use of the model for future time predictions

As a final step, the exercise of obtaining predictions in future time was applied. In this respect, the established input values and the predictions given by the model for 5 future times are shown in Table-8:

Table-8. Prediction values.

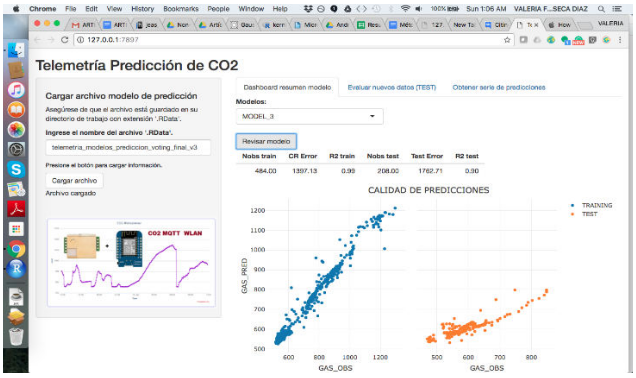
Date	Gas_Pred	Light	Temperature	Humidity
2017-09-16 4:21:00	366.98	7.2	20	19
2017-09-16 4:26:00	361.55	7.2	20	19
2017-09-16 4:31:00	357.87	7.2	20	19
2017-09-16 4:36:00	352.61	7.2	20	19
2017-09-16 4:41:00	348.15	7.2	20	19

Interactive application development

With the aim of making the methodology developed in the present work available to users, an interactive application was developed to evaluate the trained models, apply them in new evaluation sets and obtain predictions in future times. The application was developed in the Shiny extension for R [22]. Below are the sections that make up the application.

The application allows you to enter a file that contains the trained prediction models. Any file that contains the same type of models can be loaded, allowing the user to evaluate as many available models. It consists in three main environments that correspond to the visualization of the characteristics of the models loaded as a dashboard, the evaluation in a new set and the calculation of predictions in future times.

The dashboard presents the prediction quality summary measures of the model with the training data and the available evaluation data, as presented in Figure-6.


Telemetría Predicción de CO2

Cargar archivo modelo de predicción

Asegúrese de que el archivo está guardado en su directorio de trabajo con extensión '.RData'.

Ingrese el nombre del archivo '.RData'.

telemetria_modelos_prediccion_voting_final_v3

Presione el botón para cargar información.

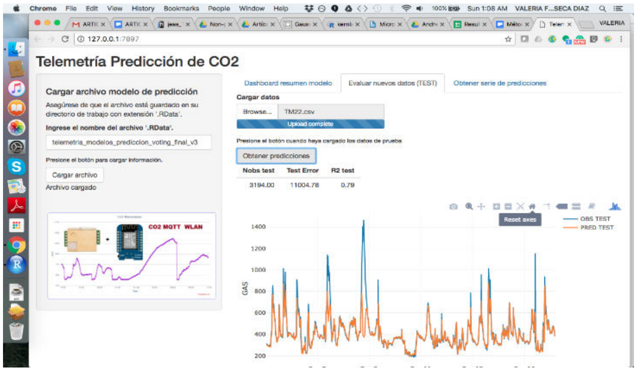
Archivo cargado

Modelos: MODEL_3

Nobs train	CR Error	R2 train	Nobs test	Test Error	R2 test
484.00	1397.13	0.99	208.00	1762.71	0.90

Figure-6. Dashboard with presentation of model evaluation prediction.

The second environment allows the user to enter their own data on which the prediction applies with the voting strategy and provides a summary of the prediction quality in said set, as shown in Figure-7.



Telemetría Predicción de CO2

Cargar archivo modelo de predicción

Asegúrese de que el archivo está guardado en su directorio de trabajo con extensión '.RData'.

Ingrese el nombre del archivo '.RData'.

telemetria_modelos_prediccion_voting_final_v3

Presione el botón para cargar información.

Archivo cargado

Cargar datos

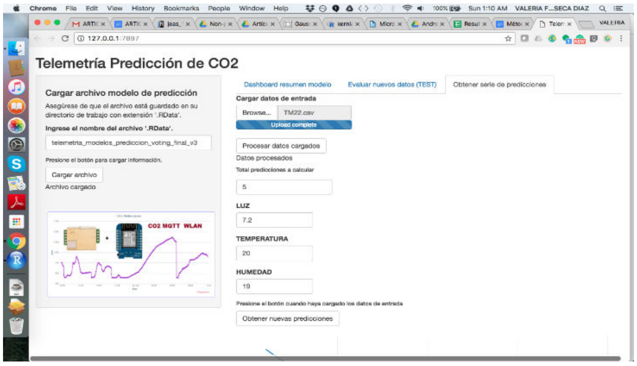
TM22.csv

Presione el botón cuando haya cargado los datos de prueba

Nobs test	Test Error	R2 test
3194.00	11004.78	0.79

Figure-7. Dashboard with evaluation of new data sets.

Finally there is the environment where the predictions are made in future times. Here the user must enter the necessary lags of the variable to be predicted and establish input values for the environmental variables. The program returns a graph and a table with calculated predictions, in addition to providing downloadable files of these elements to give the possibility of external use of the results obtained, as shown in Figure-8.


Chrome File Edit View History Bookmarks People Window Help
Sun 1:10 AM VALERIA F...SECA DIAZ
127.0.0.1:7897
Dashboard resumen modelo
Evaluar nuevos datos (TEST)
Obtener serie de predicciones
Browse...
Upload complete
Procesar datos cargados
Obtener nuevas predicciones
Cargar archivo
CO2 MQTT WLAN

Telemetría Predicción de CO2

Cargar archivo modelo de predicción

Asegúrese de que el archivo está guardado en su directorio de trabajo con extensión '.RData'.

Ingrese el nombre del archivo '.RData'.

telemetria_modelos_prediccion_voting_final_v3

Presione el botón para cargar información.

Archivo cargado

Cargar datos de entrada

TM22.csv

Datos procesados

Total predicciones a calcular

5

LUZ

7.2

TEMPERATURA

20

HUMEDAD

19

Presione el botón cuando haya cargado los datos de entrada

Figure-8. Dashboard for the prediction CO_2.

CONCLUSIONS AND ACKNOWLEDGEMENTS

The feasibility of using data acquisition systems with non-industrial and low-cost equipment was demonstrated to massify the monitoring of environmental variables to indoor environments of any type of structure, and may even be linked to the infrastructure of mass networks, such as Wifi networks. It was possible to develop a centralized data acquisition structure based on the low data processing capacity of remote micro-controlled devices, but high processing capacity of centralized servers.

Se consiguió desarrollar algoritmos de machine learning aplicables a sistemas de telemetría de variables ambientales.

The methodology adopted in the present work to construct a model that would allow obtaining CO2 predictions based on historical values of the same variable and on environmental variables showed highly satisfactory results in the quality of the predictions. Given the availability of different data sets obtained under different factors, it was possible to implement more sturdy statistical modeling strategies in order to obtain the predictions. The development of the application added an important contribution of the present work by providing the use of the models for external users.

Los resultados obtenidos no reemplazan los enfoque clásicos de monitoreo de fuentes de combustión para el estudio de la contaminación del aire, sino que los complementan para entender el comportamiento de esta contaminación en interiores.

REFERENCES

[1] Cline J. 1983. Environmental Monitoring and Sampling Overview. IEEE Transactions on Nuclear Science (Volume: 30, Issue: 1, Feb. 1983) DOI: 10.1109/TNS.1983.4332321.

[2] Gard M. 2002. A status report on environmental monitoring. IEEE Transactions on Instrumentation and Measurement (Volume: 51, Issue: 4, Aug 2002) Print ISSN: 0018-9456.

[3] Lee P, Lai L. 2009. A practical approach of smart metering in remote monitoring of renewable energy applications. Power & Energy Society General Meeting, 2009. PES '09. IEEE. ISSN: 1932-5517.

[4] AlcântaraI E, *et al.* 2013. A system for environmental monitoring of hydroelectric reservoirs in Brazil. Revista Ambiente & Água. 8(1). ISSN 1980-993X.

[5] Vanus J *et al.* 2016. New method for accurate prediction of CO_2 in the Smart Home. Instrumentation and Measurement Technology Conference Proceedings (I2MTC), 2016 IEEE International. ISBN: 978-1-4673-9220-4

[6] Gugliermetti L, Sabatini M, Palmerini G, Carpentiero G. 2016. Air quality monitoring by means of a miniaturized sensor onboard an autonomous wheeled rover. Smart Cities Conference (ISC2), 2016 IEEE International. ISBN: 978-1-5090-1846-8.

[7] Včelák J, Vodička A, Maška M, Mrňa J. 2017. Smart building monitoring from structure to indoor environment. Smart City Symposium Prague (SCSP), 2017. ISBN: 978-1-5386-3825-5.

[8] Tapashetti A, Vegiraju D, Ogunfunmi T. 2016. IoT-Enabled Air Quality Monitoring Device. IEEE Global Humanitarian Technology Conference. ISBN 978-1-5090-2432-2.

[9] Dutta J, Gazi F, Roy S, Chowdhury C. 2016. AirSense: Opportunistic Crowd-Sensing based Air

quality monitoring system for smart city. IEEE Sensors. ISBN 978-1-4799-8287-5.

[10] Simic M, Stojanović G, Zaraska K. 2016. Multi-Sensor System for Remote Environmental (air and water) quality monitoring. 24th Telecommunications forum TELFOR 2016. ISBN 978-1-5090-4086-5.

[11] Gomez A. 2016. CLEANWIFI: the wireless network for air quality monitoring, community internet access and environmental education in smart cities. Kaleidoscope 2016. ISBN 978-92-61-20441-9.

[12] Vapnik V. 1998. Statistical Learning Theory. Ed John Wiley & Sons, Inc, p. 736. ISBN 978-81-265-2892-9.

[13] Vapnik V. 1999. An Overview of Statistical Learning Theory. IEEE transactions on neural networks. 10(5).

[14] Suykens J. 1998. Nonlinear modeling. Advanced black-box techniques. Ed Springer-science+business media, B.V. p 98. ISBN 978-1-4613-7611-8.

[15] Vapnik V. 2000. The Nature of Statistical Learning Theory. Second Edition, Ed Springer. ISBN 978-1-4419-3160-3.

[16] Suykens J., *et al*. 2003. Least Squares. Support Vector Machines. Ed. World Scientific. p. 10. ISBN 9812381511.

[17] Karatzoglou A, Smola A y Hornik K. 2004. kernlab – An S4 Package for Kernel Methods in R. Journal of statistics.

[18] R Core Team. 2017. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

[19] Alexandros Karatzoglou, Alex Smola, Kurt Hornik, Achim Zeileis. 2004. kernlab - An S4 Package for Kernel Methods in R. Journal of Statistical Software. 11(9): 1-20. URL http://www.jstatsoft.org/v11/i09/.

[20] Opitz D, Maclin R. 1999. Popular ensamble methods: An empirical study. Journal of Artificial Intelligence Research. 11: 169-198.

[21] Williams C, Barber D. 1998. Bayesian Classification with Gaussian Processes. IEEE transactions on patern analysis and machine learning. 20(5).

[22] Chang W, Cheng J, Allaire JJ, Xie Y and McPherson J. 2017. Shiny: Web Application Framework for R. R package version 1.0.5. https://CRAN.R-project.org/package=shiny.