Advances in Computer Science and Information Technology (ACSIT)
p-ISSN: 2393-9907; e-ISSN: 2393-9915; Volume 4, Issue 5; October-December, 2017, pp. 280-284

http://www.krishisanskriti.org/Publication.html

Time Series Forecasting of Air Pollutant Concentration Levels using Machine Learning

Subhra Rani Patra

Narsee Monjee Institute of Management Studies
E-mail: patra.subhra@gmail.com

Abstract—*This study presents a time series forecasting for the prediction of atmospheric pollutant concentration levels using artificial intelligence techniques. The feasibility of artificial intelligence (AI) algorithms as a forecast tool for the one month ahead estimation of CO and NO2 concentrations is discussed here. A conventional air pollution monitoring system is used to provide the reference data. In this case, the results are assessed by means of root mean square error characterization throughout one month long interval and discussed. Performances of the algorithms are also investigated. The results demonstrate that the proposed (AI) methodology achieves a fair prediction of the presented pollutant time series.*

1. INTRODUCTION

Pollution is one of the most relevant problems of metropolitan areas. With population growth and economical increases leading to new industry, environmental health problems have captured society's interest. Problems that affect the ecosystem, such as noise pollution; garbage and its disposal; and, in particular, air pollution, have a direct effect on people's health. There are numerous air quality indicators that show effects of pollution on people's health. Some of the most important ones include carbon monoxide (CO), sulphur dioxide (SO2), and nitrogen dioxide (NO2). When concentration level of an indicator exceeds an established air quality safety threshold, severe health problems might affect humans. There are many environmental agencies around the globe that develop their own policies and have established air quality standards and indicators regarding allowed atmospheric pollutant levels. Environmental agencies use the indicators as a monitoring measure, using a network of pollution and atmospheric sensors. The measurement results are observations equally spaced and ordered in time (e.g., hourly, daily, and monthly), resulting in a time series of pollutant concentrations. Methods used for time-series prediction are native to the statistics field, such as the autoregressive (AR) model and the autoregressive moving average (ARMA) model. There are some studies in the literature that use these ideas, where an intelligent search method is combined with an ANN to enhance a predictive system. The ARIMA models are almost the most widely used methods. The ARIMA models are described using three basic time series models (1) autoregressive (AR), (2) moving average (MA), and (3) autoregressive moving average (ARMA) [1-2]. In recent years, machine learning based time series models such as artificial neural networks have been successfully applied for modelling infectious disease incidence time series [3]. Support vector machines (SVMs) are a new type of machine learning methods based on statistical learning theory [4]. The focus of this study is to employ three methods: ARIMA, Artificial Neural Network and Support Vector Machine for forecasting the pollutant concentrations.

2. MODEL SELECTION

The literature survey presents that for a daily basis forecast we can use models for example Traditional Time Series Models and the Artificial Intelligence Models. With a specific end goal to have a more extensive thought for them their advantages and downsides have been listed below.

2.1. ARIMA

- Widely acknowledged by business analysts.
- Not costly computationally
- Widely utilized in the literature.
- Difficult to catch non-linear patterns.
- Their performance depends on few parameter settings.

2.2. Neural Networks

- Able to follow both linear and non-linear patterns.
- Computationally more expensive.
- Not equally acknowledged by economists in regard with the traditional time series traditional approach
- Their performance depends on a large number of parameter settings.

2.3. Support Vector Machine

- It utilizes the kernel trick, so one can prepare the expert knowledge about the problem by engineering the kernel.
- kernel models can be very sensitive to over-fitting the model selection criterion

It is clear that each category of models has its strong and weak points. In my endeavor to compare them I did not manage to choose one and neglect the other. Rather I am going to use the three of them and compare their productivity on the attempted task. All the more particularly, at the first stage I will use traditional time series prediction models and I will inspect if they figure out how to capture all the patterns that exist in our data, if not I will consider neural network models and support vector machine model to attempt to capture these patterns. The case study I have examined clearly displays that there are non-linear relationships in the data sets used. Thus our instinct is that in my case study too the traditional time series prediction models won't have the capacity to exploit all the patterns that exist in my data sets.

3. TIME SERIES ANALYSIS

A time series is a time dependent or chronological sequence of observations on a considered variable. Examples include (i) sales of a particular product in successive months, (ii) the temperature at a particular location at night on successive days, and (iii) electricity consumption in a particular area for successive one-hour periods. However, time-series data presents an excellent opportunity to look at what is called out-of-sample behaviour. A time-series model will provide forecasts of new future predictions which can be checked against what is actually observed. If there is good agreement between each other, it will be argued that this provides a more convincing verification of the model than in-sample fit

3.1. ARIMA Model

ARIMA models are considered to be very well known class of models for estimating a time series. The models can be made to be stationary by differencing or they might be in conjunction with nonlinear transformations. A random variable with time series nature is considered to be stationary if its statistical properties are all consistent over time. A stationary series is supposed to have no pattern. That means its variations around its mean will have a steady amplitude. It swings in a constant predicted fashion, showing its short-term random time patterns appear to be same in a statistical sense. This implies that its autocorrelation will be constant over time, or comparably, that its power spectrum remains constant over time. A random variable of this form can be viewed as a mixture of signal and noise. The signal may be thought of as a pattern of fast or slow mean reversion, or sinusoidal oscillation, or rapid alternation in sign, and it could also have a seasonal component. The equation of ARIMA forecasting is a linear equation for a stationary time series. In ARIMA the predictors comprise of lags of the dependent variable or lags of the forecast errors. It means predicted estimation of Y equal to a constant or a weighted sum of one or more recent values of Y or a weighted sum of one or more recent values of the errors. Whereas if the predictors comprise only of lagged values of Y, it is a pure autoregressive model. That means it is just a unique case of a regression model and it could be fitted using standard regression software. If some of the predictors are slacks of the errors, an ARIMA model cannot be considered as a linear regression model. Because the error of last period cannot be specified as an independent variable. The errors has to be estimated on a period-to-period basis once the model is fitted to the data. The acronym ARIMA stands for Auto-Regressive Integrated Moving Average. A nonseasonal ARIMA model is presented as an ARIMA(p,d,q) model. Where p represents the number of autoregressive terms, d represents the number of nonseasonal differences needed for stationarity, and q represents the number of lagged forecast errors in the prediction equation.

3.2. Artificial neural network

ANN is one of the most valuable artificial intelligence techniques for data mining tasks, for instance classification as well as regression problems. An extensive measure of research showed that ANN is able to deliver good accuracy in forecasting of parameters. However, this strategy has couple of limitations. In ANN algorithm, some parameters should to be tuned in the beginning of training process: number of hidden layer and hidden nodes, learning rates, and activation function. Numerous efforts had been made to accomplish the solutions of limitations of neural network. [Huang and Babri1998] presented the Single Hidden Layer Neural Networks (SFLN) with utilization of tree steps extreme learning technology called as ELM was able to should take care of the issues with exactness. The backpropagation algorithm for accessing parameters in neural networks has been the most well known in the medical literature [Reggia, 1993]. In some biomedical applications, the pattern of interest is precisely the one that is uncommon, and backpropagation-based neural networks may have problems in learning this pattern. The challenges related to learning infrequent patterns in neural networks have driven some investigators to create algorithms for preprocessing the data and to develop modifications of the backpropagation algorithm. The answer for the issue of dealing with infrequent patterns in backpropagation-based neural networks has been the change of the weight update function used in the backpropagation algorithm.

The processing units or neurons of an ANN consists of three main components; synaptic weights connecting the nodes, the summation function within the node and the transfer function (see Fig. 1). Synaptic weights are known for their strength which corresponding to the importance of the information coming from each neuron. Which means the information is encoded in these strength-weights. The summation function is

used to estimate the total input signal by multiplying with their synaptic weights and summing up all the products.

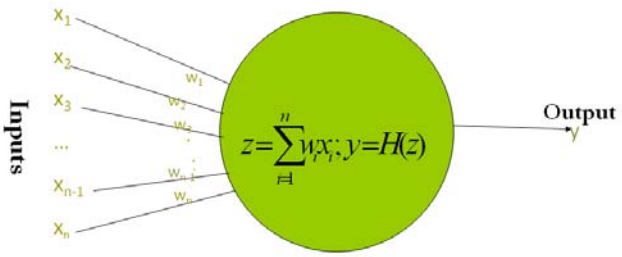

Figure 1. Schematization of artificial neuron

Activation function transforms the summed up input signal, received from the summation function, into an output. The activation function can be either linear or non-linear. In this study, the neural networks used have the sigmoid function and the linear function as the activation functions of the hidden and output layers, respectively. The dataset used for training, validating, and testing the neural network is divided into three groups. First, the training set, which usually consists of half or more of all data gathered. It is used by the ANN to adjust its weights and biases. Second is the validation set, which is used for validating the network training. It checks the network's capability to generalize a series of input data. Finally, the test set is used to evaluate the network's performance [5]. The latter two sets consist of data that have not been previously presented to the network. The training process is performed until any stop criterion is achieved. The errors associated with the stop criteria are the validation (generalization) and training errors. The test error is the network's performance measurement based on the test set. The training, validation, and test errors are evaluated by comparing the actual observed data to the predicted data.

3.3. Support Vector Machine

SVM is a learning framework utilizing a high dimensional feature space. It generates the prediction functions that are developed on a subset of support vectors. SVM has the ability to generalize complex structures with help of a very few support vectors hence provides a new mechanism for image compression. A different form of a SVM for regression has been proposed in 1997 by [6]. This strategy is called support vector regression (SVR). The model generated by support vector classification is only dependent on a subset of the training data, because the cost function for building the model does not care about training points that goes beyond the margin. Similarly, the model generated by SVR only depends on a subset of the training data, on the grounds that the cost function for building the model overlooks any training data that is close (within a threshold ε) to the model prediction. Support Vector Regression (SVR) is the most well-known application type of SVMs. A review of the essential ideas presenting the support vector (SV) regression with function estimation has been presented in [7]. Moreover, it has incorporated an outline of currently used algorithms for training SVMs, covering both the quadratic (or convex) programming part and advanced methods for managing extensive datasets. At last, a few alterations and extensions have been applied to the standard SV algorithm.

Assume we are given training data $\{(x_1, y_1), \ldots, (x_l, y_l)\} \subset \chi \times R$, where χ means the space of the input patterns (e.g. $\chi = R_d$).

In ε – SV regression, the objective has been to find a function $f(x)$ that has at most ε deviation from the actually obtained targets y_i for all the training data at the same time as flat as it can be expected. The case of linear function f has been described in the form as

$$f(x) =< w, x) + b \text{ with } w \in N, b \in R \quad (1)$$

Where<.,.> presents the dot product in N. Flatness in (1) means small w . For this, it is required to limit the Euclidean norm i.e. $\| w \|^2$ .Formally this can be represented as a convex optimization problem by requiring

$$\text{Minimize} \quad \frac{1}{2} \| w \|^2$$

$$\text{Subject to} \begin{cases} y_i - < w, x_i > -b \le \varepsilon \\ < w, x_i > -y_i \le \varepsilon \end{cases} \quad (2)$$

The above convex optimization problem is achievable in cases where *f* actually exists and approximates all pairs (x_i , y_i) with ε precision. Once in a while, some errors are allowed. Introducing slack variables ξ, ξ_i to cope with otherwise infeasible constraints of the optimization problem (2), the formulation becomes

$$\text{Minimize } \frac{1}{2} \| w \|^2 + C \sum_{i=1}^{l} (\xi_i + \xi_i^*)$$

$$\text{Subject to} \begin{cases} y_i - < w, x_i > -b \le \varepsilon + \xi_i \\ < w, x_i > -y_i \le \varepsilon + \xi_i^* \\ \xi, \xi_i^* \ge 0 \end{cases} \quad (3)$$

The constant $C > 0$ decides the trade-off between the flatness of *f* and the amount up to which deviations larger than ε are tolerated.

4. DATA SELECTION

The objective of this research is to provide a means for enhancing the ability of machine-learning methods to perceive past data and thereby forecast the future parameters. The dataset utilized in this experiment is Air Quality, UCI Dataset

acquired from the University of California, CA, Department of Information and Computer Science. The dataset contains 390 instances of daily averaged responses from metal oxide chemical sensors embedded in an Air Quality Chemical Multi sensor Device. The device was found on the field in a considerably polluted area, at road level, within an Italian city. Data were recorded from March 2004 to February 2005 (one year) representing the longest freely available recordings of on field deployed air quality chemical sensor devices responses. Hourly averaged concentrations for CO and Nitrogen Dioxide (NO2) were provided by a co-located reference certified analyzer.

Conventional fixed station provided reference concentration estimation for CO (mg/m3), NO_2 (mg/m3). It was sampled recording hourly averages of the concentration values. The multi sensor device was sampled to provide the hourly average of the resistivity expressed by CO, NO_2 plus the commercial temperature and relative humidity sensors. Measurement campaign took place using as testing site one of the main street in the centre of an Italian city, characterized by heavy car traffic [8].

5. RESULTS AND DISCUSSION

In this study, the following tow real world time series, corresponding to natural phenomena, were considered, gaseous concentrations of CO and NO_2. The dataset has been divided in to training set with 90% of the time-series data, validation set with another 10% of the time-series data. The measurements available for the CO and NO_2 pollutant were collected between the years 2000 and 2001. The dataset consists of 390 averaged daily observations.

ANN with maximum architecture of 12 – 10 – 1, which makes reference to an MLP network, which denotes 10 units in the input layer, 10 units in the hidden layer, and 1 unit in the output layer (prediction horizon of multi-step forward). For each time series, 10 experiments were performed with the combined algorithms, where each algorithm with the greatest fitness function was chosen as the representative of the respective model for a particular time series. The error performance of CO series shows (4-8-1) to be the optimum architecture and NO2 series shows (10-2-1) to be the optimum architecture.

6. PERFORMANCE MEASURES

For the problem of time-series forecasting, there is no single metric universally adopted by researchers to evaluate a model's predictive adequacy. In the present study, the root mean square error (RMSE), which is one of the most common performance measures applied to neural networks is considered to allow a better appreciation of the forecasting system performance. Root Mean Square Error (RMSE) is defined by the standard deviation of the residuals. Residuals (prediction errors) are a representation of how far from the regression line sample points are. RMSE is represented as a measurement of how spread out these residuals are. In other words, it will tell how concentrated the data is around the line of best fit.

$$RMSE = \sqrt{\frac{1}{2}\sum_{i=1}^{n}(y - y_i)^2}$$

Where y = forecasts (expected values or unknown results),

y_i = observed values (known results)

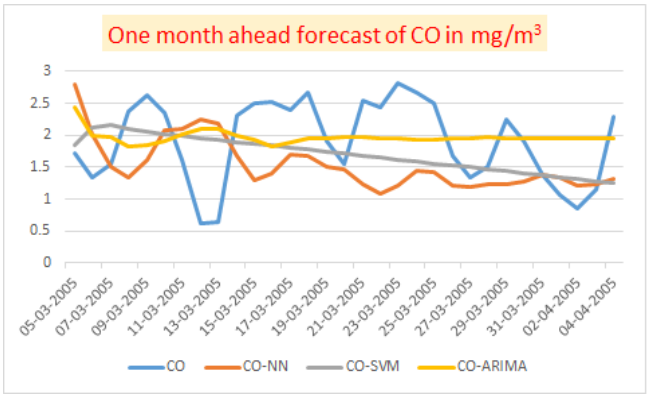

Figure 2. One month forecast of CO using time series models

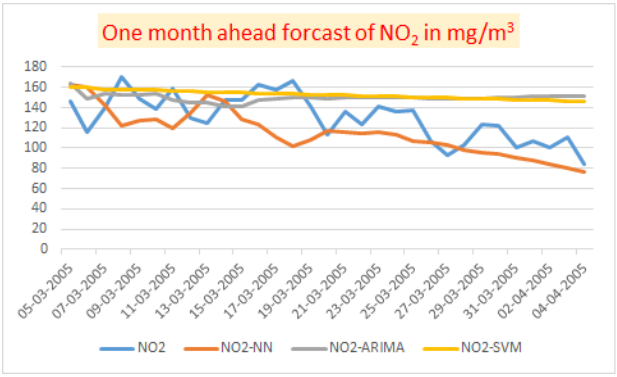

Figure 3. One month forecast of CO using time series models

The Table 1 presents the root mean square values for both the pollutants for all the algorithms. The results show that for both the pollutants the RMSE value is least in case of artificial neural network algorithm.

Table 1: RMSE values for all the algorithms

Pollutant	RMSE-NN	RMSE-ARIMA	RMSE-SVM
CO	1.58	1.969	1.724
NO2	117.28	149.823	152.877

In this study, artificial intelligent systems for time-series forecasting of concentration levels of air pollutants was evaluated. The system consists of composed of ARIMA, an

intelligent hybrid model- ANN algorithm and also SVM. The results were presented in terms of measures of RMSE. Among the proposed models, ANN found to be the best configuration for both (CO and NO2) of the time series addressed. The results depict the comparison of the forecast of three algorithms. From the graphs also it can be seen that artificial neural network performs better compared to the ARIMA and support vector machine. Fig. 2 gives the forecast of CO and Fig. 3 presents the forecast of NO_2.

7. CONCLUSION

In this study, the time-series forecasting of concentration levels of air pollutants was evaluated using artificial intelligence systems. The system consists of an intelligent model composed of ANN algorithm and SVM and also the conventional ARIMA model. An optimized structure of a neural network is presented in terms of input units, hidden processing units, initial weights, and biases. The results were presented in terms of RMSE. Among the proposed combinations, ANN is found to be the best configuration in both (CO and NO2) forecast. The artificial intelligence techniques combines exploitation and exploration characteristics. The gradient descendent algorithm can exploit the search space locally, performing the search and achieving a local minimum within that specific area in less time. In particular, the results show that this approach may be an interesting tool to predict the concentration levels of pollutants.

8. 8. ACKNOWLEDGEMENTS

This work was supported by the data taken from machine learning repository, UCI.

REFERENCES

[1] Zhang GP (2003) Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing* 50: 159–175.

[2] Pai P.F, Lin C-S (2005) A hybrid ARIMA and support vector machines model in stock price forecasting. *Omega* 33: 497–505.

[3]Chang C.C, Lin C-J (2011) LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology* (TIST) 2: 27.

[4] Thissen U, Van Brakel R, De Weijer A, Melssen W, Buydens L (2003) Using support vector machines for time series prediction. *Chemometrics and intelligent laboratory systems* 69: 35–49.

[5] Haykin, S.; Neural Networks: *A Comprehensive Foundation, 2nd ed., Prentice Hall*: New Jersey, 1999.

[6] V. Vapnik, S. Golowich and A. Smola, (1997), "Support Vector Method for Function Approximation, Regression Estimation, and Signal Processing", in M. Mozer, M. Jordan, and T. Petsche (eds.), Neural Information Processing Systems, Vol. 9. MIT Press, Cambridge, MA.

[7] A.J. Smola, and B. Schölkopf, (1998), "A Tutorial on Support Vector Regression", NeuroCOLT, Technical Report NC-TR-98-030, Royal Holloway College, University of London, UK.

[8] S. De Vito, E. Massera, M. Piga, L. Martinotto, G. Di Francia, On field calibration of an electronic nose for benzene estimation in an urban pollution monitoring scenario, Sensors and Actuators B: Chemical, Volume 129, Issue 2, 22 February 2008, Pages 750-757, ISSN 0925-4005