TASK Quarterly 27 (1) 2023

Application of artificial neural networks in the development of the PM10 air pollution prediction system

Aneta Wiktorzak
Lomza State University of Applied Science
1 Akademicka St., 18-400 Lomza, Poland

Andrzej Sawicki
Lomza State University of Applied Science
1 Akademicka St., 18-400 Lomza, Poland

https://doi.org/10.34808/bamk-q919

Abstract

This article presents research on the model of forecasting the average daily air pollution levels focused mainly on two solutions, artificial neural networks: the NARX model and the LSTM model. The research used an air quality monitoring system. This system includes individually designed and implemented sensors to measure the concentration of pollutants such as PM10, PM2.5, SO2, NO2 and to record weather conditions such as temperature, humidity, pressure, wind strength and speed. Data is sent to a central database server based on the MQTT protocol. Additional weather information in the area covered by pollution monitoring is collected from the weather services of the IMGW and openwethermap.org. The artificial neural network models were built in the MATLAB environment, the process of learning neural networks was performed and the results of pollution prediction for the level of PM10 dust were tested. The models showed good and acceptable results when forecasting the state of PM10 dust concentration in the next 24 hours. The LSTM prediction model were more accurate than the NARX model.

The future work will be related to the use of artificial intelligence algorithms to predict the concentration of other harmful substances, e.g. PM2.5, NO2, SO2 etc. A very important task in the future will be to frame the entire system of monitoring and predicting smog in a given area.

Keywords:

NARX, LSTM, PM10

1. Air pollution prediction models

This article presents research on the model of forecasting the average daily air pollution levels in the city of Lomza and its vicinity focused mainly on two solutions, the NARX (Nonlinear AutoRegressive eXogenous) model and the LSTM (Long Short-Term Memory) model. In the analysis of solutions, the forecasting efficiency was assumed as the main selection criterion. The results were analyzed on the basis of mean square error MSE or RMSE, correlation coefficient R, and plots of comparison of values between observed and predicted measurements. On the basis of the constructed model, a system was developed to predict the concentration of PM10 on 24 hours ahead. Due to the non-linearity of pollutants and their dependence on and between meteorological parameters, artificial neural network models proved to be useful with promising results for use in complex and non-linear systems. As a result of numerical experiments, two solutions were selected related to the use of artificial neural network algorithms, the NARX model and the LSTM model. On the basis of the presented types of artificial neural networks: NARX and LSTM and the presented structure of training vectors, network models were built in the MATLAB environment, the process of learning neural networks was performed and the results of pollution prediction for the level of PM10 dust were tested.

2. Collection of air pollution data form the city of Lomza

Data collection was based on air pollution sensors designed and built by Incontech, a company cooperating in the ongoing project. The sensors measure the following data: concentration of PM10 and PM2.5 dust, concentration of gases such as: nitrogen dioxide NO2 and sulfur dioxide SO2. Furthermore, the sensors also measure weather conditions around the sensors, such as temperature, humidity and pressure. Additionally, some meteorological data for Lomza are obtained from publicly available weather services – especially, the data concerning wind direction and its strength.

2.1. Sensor deployment

Air pollution sensors were deployed in the city of Lomza and surrounding areas in the number of about 30 devices. The deployment of the sensors took into account the following: 1) the analysis of the distribution of residential, park and industrial areas; 2) the altitude of the areas; 3) the influence of the river valley on air movement in the city; and 4) directions and strength of winds blowing in the area. A plan for the distribution of sensors in the city of Lomza and their status is shown in Figure 1.

The sensors transmit data to a central server using the Message Queueing Telemetry Transport (MQTT) protocol, creating a distributed telemetry network that extends across Lomza and the surrounding area. Transmitted data of sensor status are collected in a database, taking into account the measurement timestamp of each component. Sensors send the measurement of each examined value every 10 minutes. The collected set of source data on pollution covered the period of more than one year, from the beginning of January 2021 till the end of February 2022.

2.2. Pollution data preprocessing

While measuring the levels of air pollution in a particular location/region, one uses the term “Particle pollution” or “Particulate Matter” (PM). In the present research the main focus was on PM10 which stands for an air pollution particle with a diameter of 10 micrometers or less. In Polish standards the maximum daily PM10 dust level is set at 50 g/m3.

Preliminary analyses for data from December 2021 have shown a high correlation of measured levels of PM10 pollutants with weather conditions, such as temperature and wind speed [1], as shown in Figure 2.

To supplement the data for the study, a system was built to acquire meteorological data from publicly available weather sensors (openweathermap.org, imgw.pl), containing information on changes in: 1) wind strength and its direction, 2) temperature, 3) humidity, and 4) pressure. The operation of the weather data acquisition system was based on API (Application Programming Interface) services and weather service page parsing mechanisms, which allowed the pollution data to be supplemented with the status and forecast of weather conditions.

The collected data, represented as time series, required preliminary analysis, preprocessing and processing [2]. In this stage of the research, the accuracy and continuity of the recorded dataset and significant deviations of the data from the average values were diagnosed. Breaks in the data streams were verified and some methods for filling in missing data by adding averaged values from neighboring measurements were proposed.

Matching particular time of unsynchronized measured data and weather data to a fixed time resolution was carried out, and then sets of data vectors were prepared in csv format, adapted to the requirements of the artificial neural network learning system, containing measurement time information. The process of building

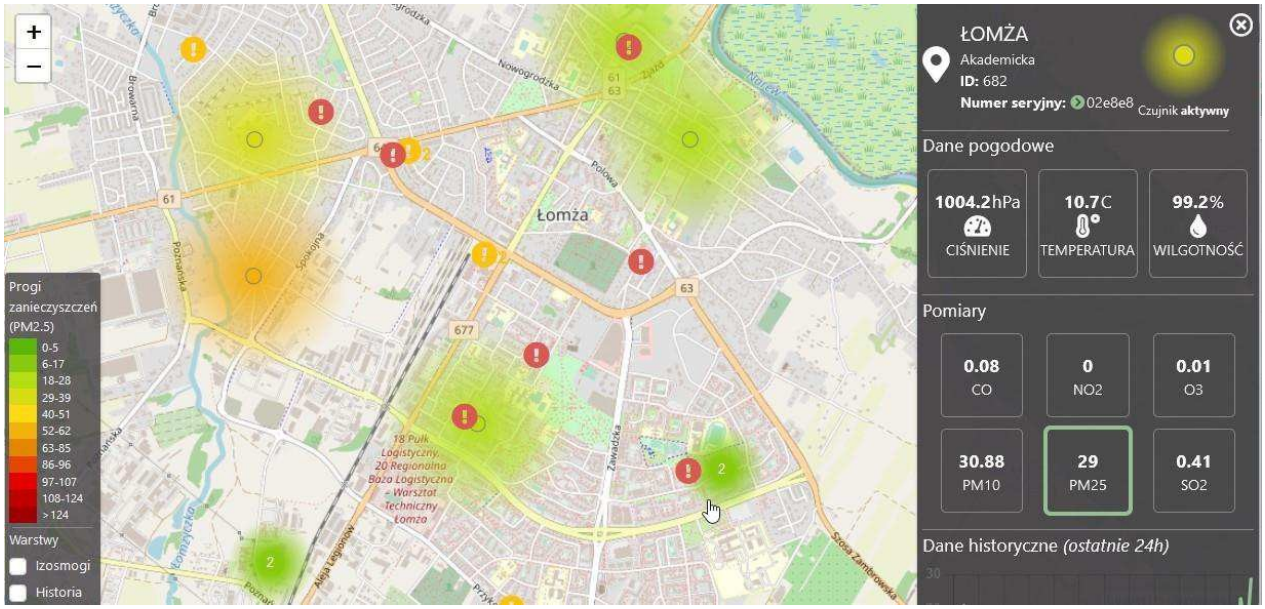

Figure 1: Distribution and status of sensors used in research in and around Lomza [source: http://sensory.incontech.pl]

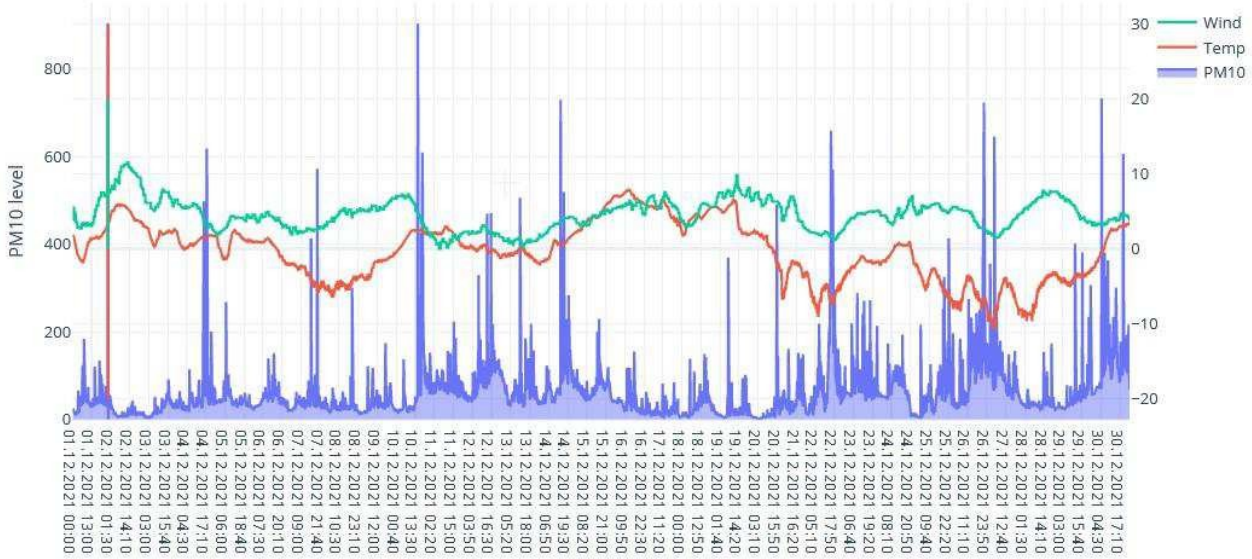

Figure 2: Correlation analysis in the pollution and weather data set (December 2021)

feature vectors began with a search through all measurements received from sensors, realized within a 10-minute time window. Next, selected data from weather services were matched to the measured data realized within the same time window. The next step involved searching and matching data for a given sensor within a +24 hour time window.

Some example sets of aggregated sensor data are presented in Table 1, and include: the date and time of measurement, PM10 concentration level at current time and 24 hours before ($PM10_{t-24}$), selected weather conditions such as: temperature, humidity, air pressure, and wind direction and its strength (abbreviated as follows: Temp, Hum, Press, Wind Deg, Wind Speed), and, in addition, the same quantities specified at time +24 hours (Temp_{t+24}, Hum_{t+24}, Press_{t+24}, Wind Deg_{t+24}, Wind Speed_{t+24}) are shown in the table. The last column represents the expected value of the neural network's response, which is the value of PM10 concentration at specific time after the next 24 hours (10_{t+24}).

Real world datasets obtained from a number of sensors and received from weather services, vary in units and range. By defining a single data set as a vector of features $\mathbf{x} = [x_1, x_2, x_3, \ldots, x_N]$, we could notice significant differences between numerical values of features, describing different physical values and differing even by orders of magnitude. This, in turn, can cause difficulties in the network learning process. Nevertheless, such difficulties can be prevented by the input data normalization proces [2]. Based on the analysis of the ranges of the collected data, feature vector normalization algorithms were used. Normalization allowed obtaining, in all observed columns

Table 1: Real input data vectors (date: 02.12.2021)

Time	PM10 t	PM10 t-24	Temp t	Hum t	Press t	Wind Deg t	Wind Speed t	Temp t+24	Hum t+24	Press t+24	Wind Deg t+24	Wind Speed t+24	PM10 t+24
00:00	42.88	24.00	3.42	89.85	97748.97	204.00	7.41	3.00	64.98	98800.14	254.00	7.19	9.63
00:10	58.13	24.00	3.46	90.09	97752.18	204.00	7.41	2.99	65.19	98793.32	254.00	7.19	11.75
00:20	51.25	23.63	3.61	90.11	97729.41	204.00	7.41	3.03	64.34	98800.54	254.00	7.19	8.00
00:30	54.63	21.88	3.74	90.12	97719.66	214.00	7.50	2.93	62.63	98836.62	254.00	7.19	5.38
00:40	50.88	23.00	3.81	90.36	97730.93	214.00	7.50	2.91	61.57	98832.62	261.00	6.13	5.00
00:50	41.25	18.00	3.95	90.42	97712.52	214.00	7.50	2.95	61.58	98846.61	261.00	6.13	5.25
01:00	55.50	13.88	4.10	90.50	97698.38	214.00	7.50	2.86	61.87	98884.53	261.00	6.13	5.88
01:20	40.50	12.38	4.34	89.68	96988.80	214.00	7.50	2.91	61.58	98407.08	261.00	6.13	5.63
01:30	36.75	14.00	4.40	90.45	97679.86	219.00	7.94	2.88	62.63	98897.98	265.00	6.10	5.50
01:40	46.25	14.00	4.58	90.23	97652.68	219.00	7.94	2.91	63.00	98917.07	265.00	6.10	4.25
01:50	24.50	11.88	4.59	90.68	97692.56	219.00	7.94	2.88	62.36	98928.69	265.00	6.10	5.00

of input data, values bounded between a fixed range of 0 and 1. Selected example sets of normalized data used as a source for machine learning algorithms are presented in Table 2.

In the conducted research, selected subsets of such prepared data vectors were tested in the process of learning, validation and testing of artificial neural network models, as presented in the following sections.

3. Nonlinear Autoregressive with External (Exogenous) Input (NARX)

In this paper, a non-linear auto-regressive neural network with exogenous inputs, NARX neural network, is used to develop models for air quality prediction. NARX neural network is represented by equation (1):

$$y(t) = f[(u(t-n_u),\ldots,u(t-1),u(t),y(t-n_y),\ldots,y(t-1)] \quad (1)$$

where: $u(t)$ - input to neural network at time t, $y(t)$ - output from neural network at time t, n_u - input order, n_y - output order.

By using previous output values as inputs to the neural network, NARX network can be modeled as a network with serial-parallel neural network structure. For NARX neural network structure, the previous output values, which are used as inputs to the neural network, are the actual output values. The backpropagation algorithm is used for training of this neural network structure [2].

3.1. NARX model test results

Nonlinear Autoregressive Network with Exogenous Inputs is a recurrent dynamic network with feedback connections that enclose several layers of the network [3]. The hourly and daily data were used in the training of artificial neural networks. Matlab Neural network time series tool is used in order to solve this problem. NARX method is applied to the modeling system. The NARX network is shown in Figure 3 with tapped delay lines and two-layer feed-forward network, a sigmoid transfer function in the hidden layer, and a linear transfer function in the output layer.

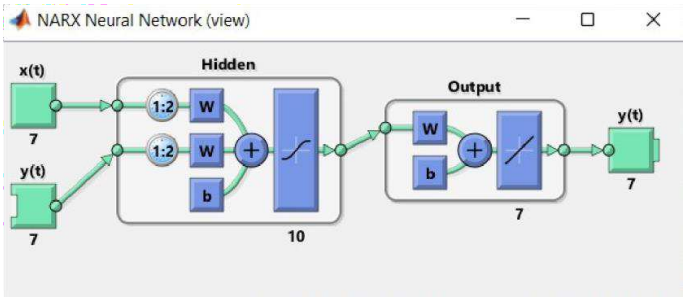

Figure 3: NARX network overview

The collected data was used in the process of learning, validation and testing of a preprepared artificial neural network model. The artificial neural network model learned was to forecast air pollution 24 hours in advance. The artificial neural network model includes an input vector: $x(t)$=$PM10(t)$, $PM10(t-1)$, humidity$(t+1)$, wind_speed$(t+1)$, wind_direction$(t+1)$, temperature$(t+1)$, pressure$(t+1)$ and an output vector $y(t) = PM10(t+1)$.

The simulated results for NARX using some variable parameters are illustrated in Table 3 and Figures 4 to 10. The Levenberg-Marquardt (LM) algorithm converged upon a resolution after maximum 40 iterations with no significant error cross-correlation or autocorrelation issues identified. Therefore, there are highly significant ($p < 0.1$) correlations between output and target data at good fit (R values that more significant than 0.9).

Graphical representations of the simulation of NARX

Table 2: Normalized input data vectors (date: 02.12.2021)

Time	PM10 t	PM10 t-24	Temp t	Hum t	Press t	Wind Deg t	Wind Speed t	Temp t+24	Hum t+24	Press t+24	Wind Deg t+24	Wind Speed t+24	PM10 t+24
00:00	0.0454	0.0244	0.4684	0.8985	0.2749	0.5655	0.3705	0.4600	0.6498	0.3800	0.7047	0.3595	0.0085
00:10	0.0623	0.0244	0.4692	0.9009	0.2752	0.5655	0.3705	0.4598	0.6519	0.3793	0.7047	0.3595	0.0108
00:20	0.0547	0.0240	0.4722	0.9011	0.2729	0.5655	0.3705	0.4606	0.6434	0.3801	0.7047	0.3595	0.0067
00:30	0.0584	0.0221	0.4748	0.9012	0.2720	0.5933	0.3750	0.4586	0.6263	0.3837	0.7047	0.3595	0.0038
00:40	0.0543	0.0233	0.4762	0.9036	0.2731	0.5933	0.3750	0.4582	0.6157	0.3833	0.7242	0.3065	0.0033
00:50	0.0436	0.0178	0.4790	0.9042	0.2713	0.5933	0.3750	0.4590	0.6158	0.3847	0.7242	0.3065	0.0036
01:00	0.0594	0.0132	0.4820	0.9050	0.2698	0.5933	0.3750	0.4572	0.6187	0.3885	0.7242	0.3065	0.0043
01:20	0.0427	0.0115	0.4868	0.8968	0.1989	0.5933	0.3750	0.4582	0.6158	0.3407	0.7242	0.3065	0.0040
01:30	0.0386	0.0133	0.4880	0.9045	0.2680	0.6072	0.3970	0.4576	0.6263	0.3898	0.7354	0.3050	0.0039
01:40	0.0491	0.0133	0.4916	0.9023	0.2653	0.6072	0.3970	0.4582	0.6300	0.3917	0.7354	0.3050	0.0025
01:50	0.0250	0.0110	0.4918	0.9068	0.2693	0.6072	0.3970	0.4576	0.6236	0.3929	0.7354	0.3050	0.0033

Table 3: Result of NARX with parameters $n = 10$ (Train and Retrain)

	MSE	R	Epoch	Time	Performance	Gradient	Validation Check
Training	6.4949e-2	8.03681e-1	17	0:00:13	0.002234	0.167	6
Validation	5.8214e-2	7.02314e-1	17	0:00:13	0.002234	0.167	6
Testing	8.9903e-2	9.14083e-1	17	0:00:13	0.002234	0.167	6

are shown in Figures 4-8.

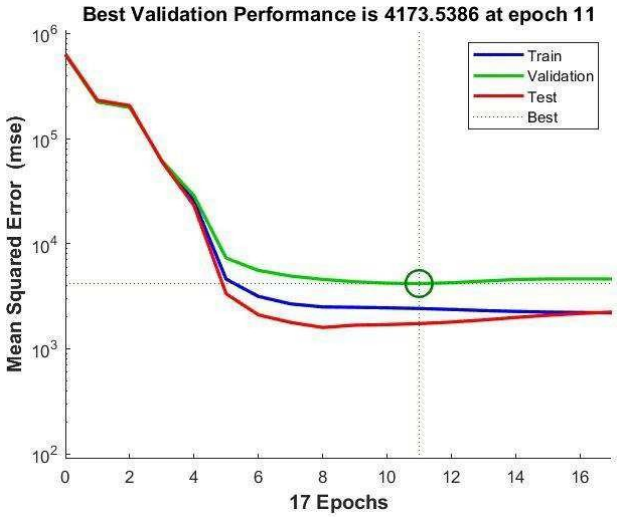

Figure 4: Performance plot of NARX (plotperform)

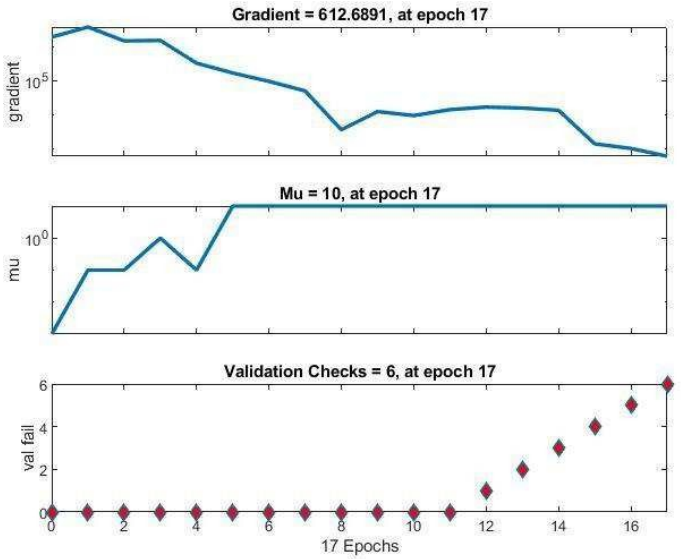

Figure 5: Training state of NARX (plottrainstate)

There are decreases in errors in training, validation, and testing as shown in Figure 2 until iteration 17 is attained which illustrated that there is no element of incidence of overfitting.

The training, validation, and testing are performed in an open-loop fashion. Likewise, the R values are also calculated based results obtained through open-loop training.

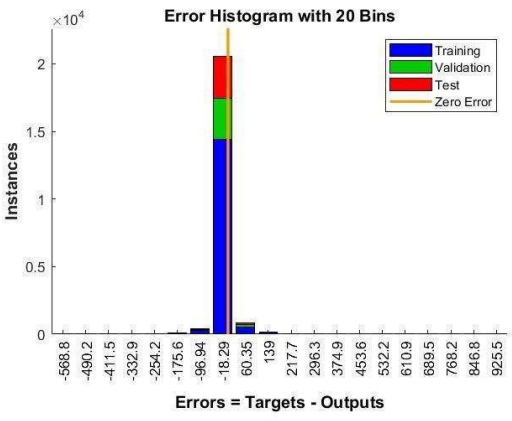

Figure 6: Error histogram of NARX (ploterrhist)

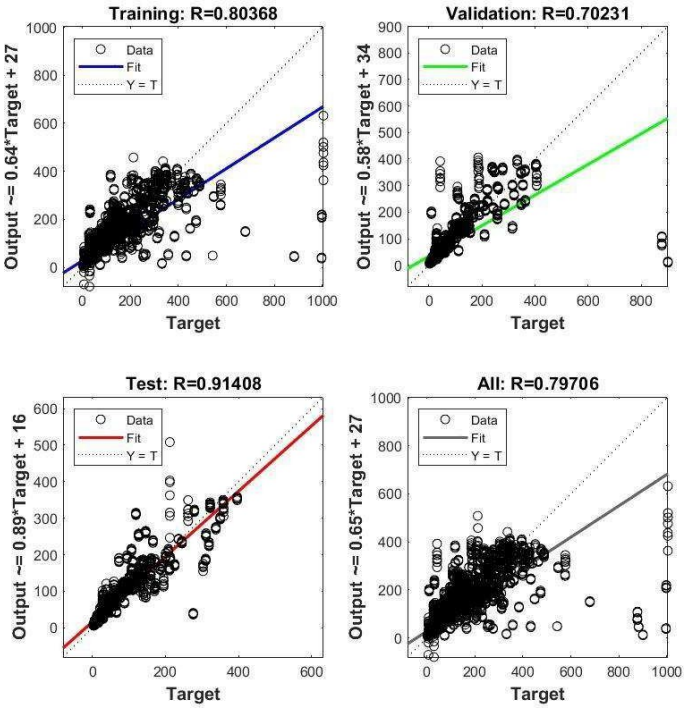

Figure 7: Regression of NARX plot (plotregression)

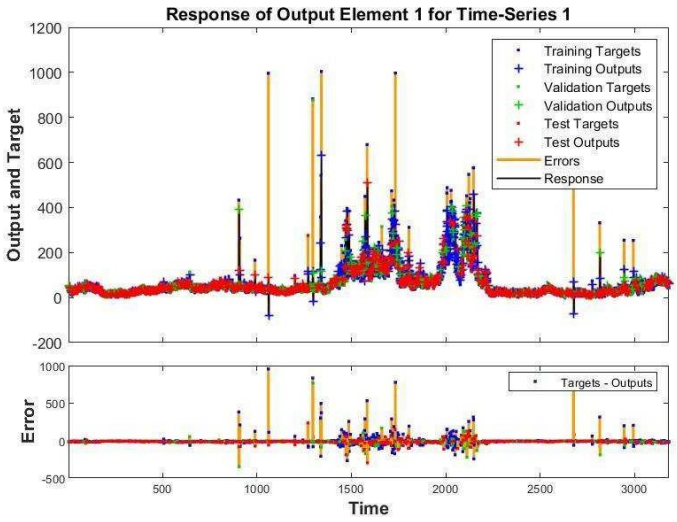

Figure 8: Time-series response of NARX (plotresponse)

Figure 7 showed the regression of the NARX plot with four different plots representing the training, validation, and testing and output data. This congested plot of NARX helps show that most of the data points are highly inter-related. The solid straight lines represent the best fit linear regression line between outputs and targets of training (blue), validation (green), testing (red), and output of all (black) while the dashed line in each plot represents the perfect result – outputs = targets. The regression (R) value indicates the relationship between the outputs and targets. If R is close to zero means there is no linear relationship between outputs and targets but, if R = 1, then there is an indication that there is an exact linear relationship between outputs and targets. In this research work, it was shown that the training data (R=0.80368) indicates a average fit, validation (R=0.70231), and test (0.91408) results also show R values that more significant than 0.79706.

Figure 8 shows the Time Series Response of NARX, which gives a clear indication where time points were selected for training, testing, and validation. The inputs, targets, and errors versus time were well displayed through time Series Response.

General measures of performance error evaluation (targetoutput) were achieved and summarized in column 2 of Table 1. These results suggested that the NARX model produces a average predictive capacity for fit and accuracy. In order to forecast the unmeasured air quality parameters, a NARX model was created. Research shows that the use of the NARX model in the forecasting of PM10 for the next 24 hours ahead gives average results; the accuracy of the model is at the level of 80%, indicating a reasonably good fit. While it may not be perfect, an 80% accuracy level is generally considered decent for many forecasting applications.

4. Long short-term memory (LSTM)

In the next part of the paper, another artificial model of the LSTM neural network used in the research is described. Long short-term memory prevents backpropagated errors from vanishing or exploding. Instead, errors can flow backwards through unlimited numbers of virtual layers unfolded in space. That is, LSTM can learn tasks that require memories of events that happened thousands or even millions of discrete time steps earlier. LSTM works even given long delays between significant events and can handle signals that mix low and high frequency components [4].

Unlike other rectilinear models, it has feedback loops that enable better communication between individual layers and neurons. An individual LSTM called a unit consists of a cell, an input gate, an output gate, and a forget gate. The cell remembers the values at any time intervals, and three gates regulate the flow of information to and from the cell. Due to the fact that it is recursive, the function is well suited for time series prediction and classification tasks. Because events can occur in time series data sets at varying intervals, the LSTM was developed as a coping framework with the problem of the disappearing gradient, which appears during the training of ordinary recursive neural networks. The error of the disappearing gradient is that in the model learning phase, the weights on the lines connecting individual neurons are constantly updated, in the case of very low updating values, there is some kind of stagnation and lack of any changes, the model becomes stuck waiting for a larger excitation signal. At worst, this can stop the neural network from train-

ing any further [5]. A visualization of a recursive neural network such as the LSTM is shown in Figure 9.

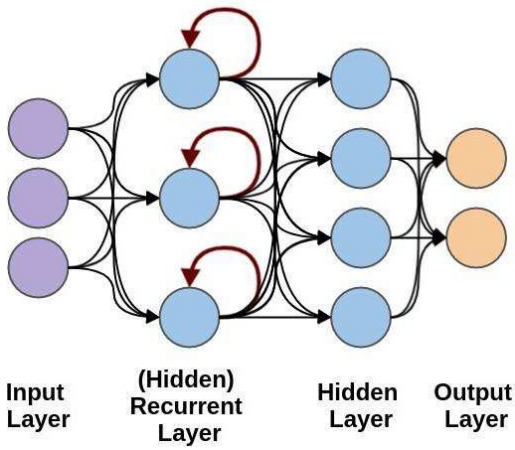

Figure 9: Recurrent Neural Network (RNN), with additional feed-forward layer

The algorithm was verified in laboratory conditions, using the Matlab programming environment [6]. The sequential regression LSTM network was trained. In each time step of the input sequence, the LSTM network learns to predict the value of the next time step. The predictAndUpdateState () function is used to predict time frames one at a time and update the network status with each forecast. An example of the LSTM network architecture for predicting PM10 pollution from the measured data is shown in the Figure 10.

4.1. LSTM model test results

A number of studies were carried out with different structures of the input vectors, i.e. with different sets of process features. The study of correlation of individual features with the baseline value - the forecast average value for the next day was carried out. The performance of the model is represented by the Loss, RMSE indicators and time series plots representing the network of forecast results against the original test series. The model adopts an iterative approach to train the network model to achieve some performance (prediction accuracy), see Figure 11.

To predict the value of multiple time slots in the future, the `predictAndUpdateState()` function was used to predict the time slots one at a time and update the network status with each forecast. For each forecast, the previous forecast was used as an input to the function. Figure 12 shows a training time series plot with predicted values.

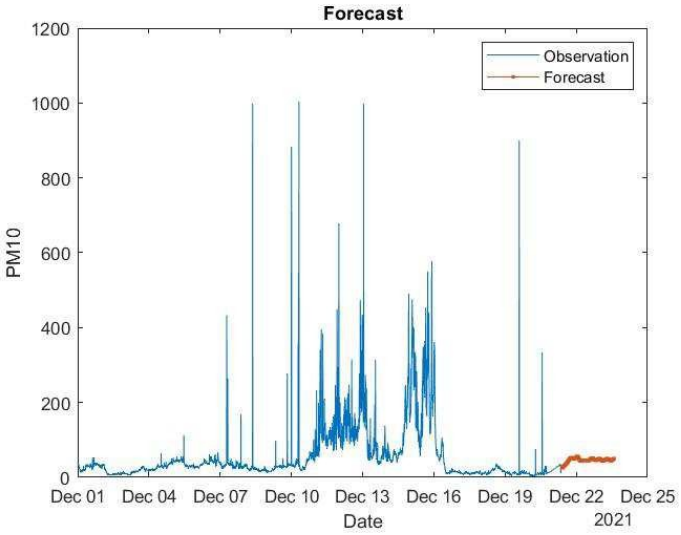

Figure 12: Graph of training time series with forecast values

If we have access to the actual values of the time steps between forecasts, we can update the network state with the observed values instead of the predicted ones. To predict the new sequence, the network state is reset with `resetState()`. Resetting the network state prevents previous forecasts from affecting the forecasts for the new data. An example of comparing the predicted values with the test data can be seen in Figure 13.

If environmental conditions or input data change over time, and the LSTM model is not sufficiently adaptable, it may have difficulty adjusting to these changes. Updating the model to account for these variations is necessary. If the data contains missing information or noise, the LSTM network may struggle to make accurate predictions. Observed values may be distorted or incomplete, affecting the quality of forecasts. The longer the forecast horizon, the more challenging it becomes to obtain accurate predictions. LSTM networks, like other models, have limitations in forecasting over extended periods because errors can accumulate. Insufficient training data can lead to underfitting of the LSTM model, resulting in low-quality predictions. In the case of recurrent networks, having a sufficiently large historical dataset is important. Finally, we evaluated the accuracy of the proposed method using the RMSE between observed and predicted values. We adjusted the learning rate, epoch, and batch size of the model to obtain optimal results. In this study, we obtained PM10 concentration data and meteorological data consisting of humidity, wind speed and direction, temperature, pressure and PM10 for use as input nodes. The output variable was predicted PM10 concentration. All data were partitioned into two sets, with 85% used for training and 15% for testing. The optimal settings for the LSTM model for both PM10 prediction were a learning rate of 0.01, epoch of 250. The RMSE values were 20.4163 for PM10, with a processing time of 5:57 min.

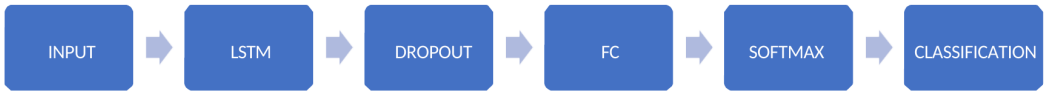

Figure 10: Sample LSTM network architecture [based on Matlab Deep Network Designer]

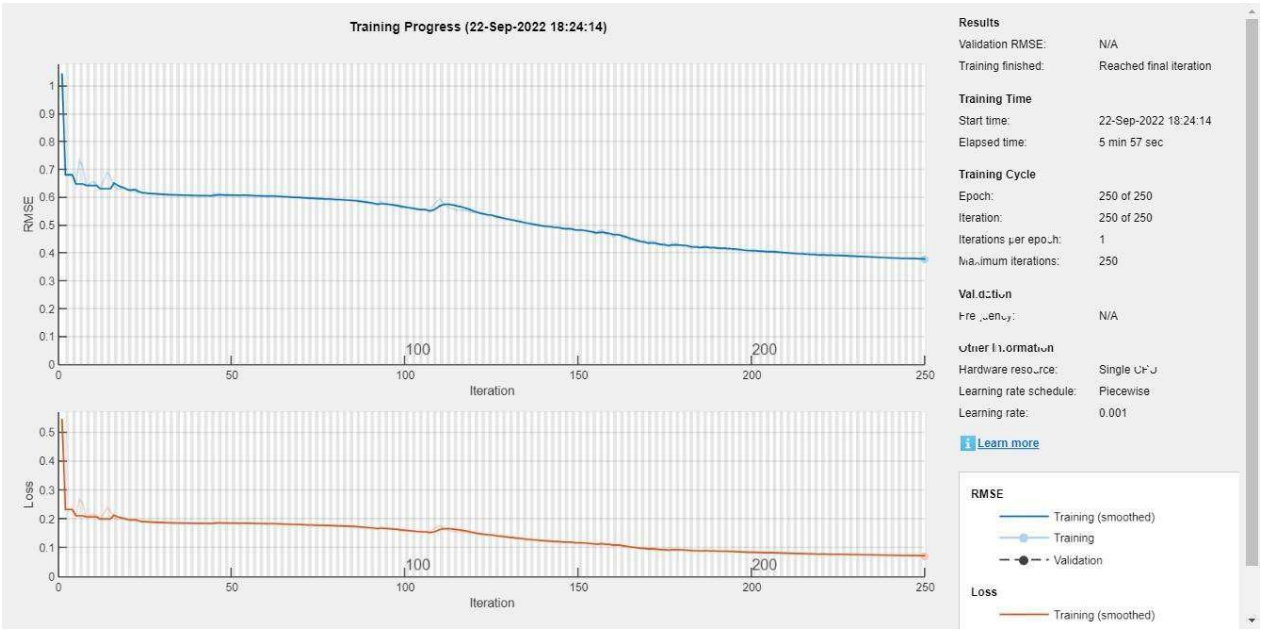

Figure 11: LSTM network training with specific options using the `trainNetwork` function

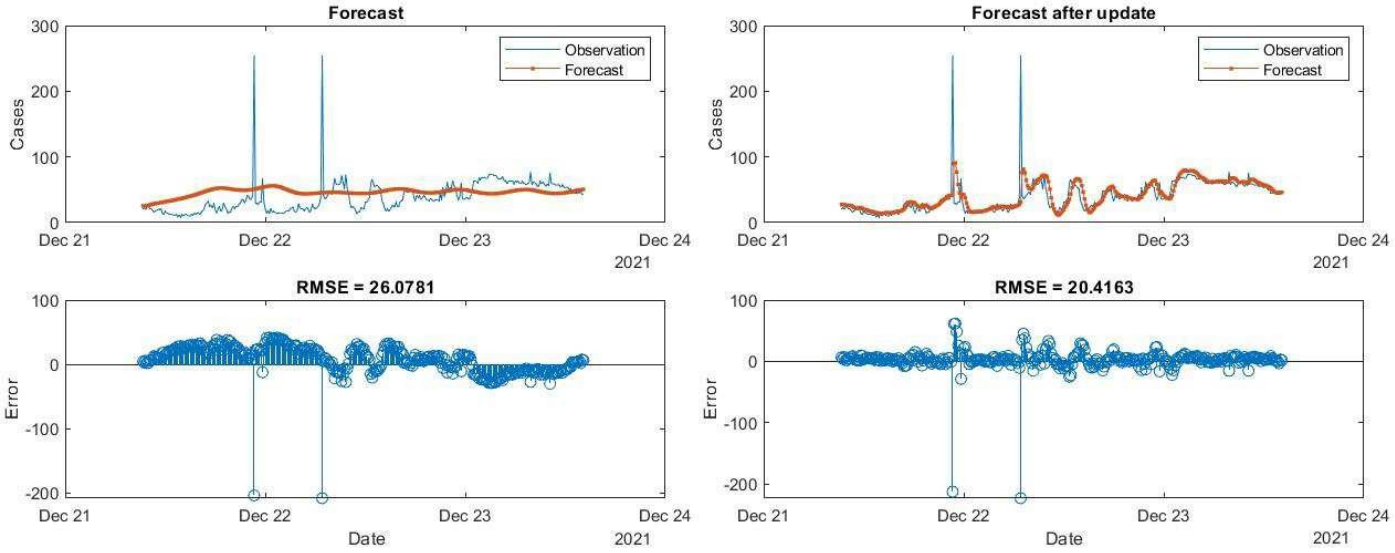

Figure 13: Comparison of the forecasted values with the test data

5. Conclusions

As a result of numerical experiments, two solutions related to the use of artificial neural network algorithms, the NARX model and the LSTM model, were selected. The models showed good and acceptable results when forecasting the state of PM10 dust concentration in the next 24 hours. We also compared the total average MSE or RMSE of prediction of PM10 the LSTM prediction model were more accurate than the NARX model.

Differences between observed and predicted values in the context of LSTM (Long Short-Term Memory) networks can result from various factors such as missing data, changing environmental conditions, a long forecasting horizon, or insufficient training data. There are many potential sources of these differences that require attention and analysis in the context of improving the quality of LSTM model predictions.

It's valuable to conduct a thorough analysis and experiments to understand which of these factors contribute to the differences between predictions and observed data and how to improve the LSTM model's forecasting qual-

ity.

The future work will be related to the use of artificial intelligence algorithms to predict the concentration of other harmful substances, e.g. PM2.5, NO2, SO2 etc. A very important task in the future will be to frame the entire system of monitoring and predicting smog in a given area.

Acknowledgement

The paper uses data collected within the framework of the project "Research work on a system to support pro-environmental measures to improve clean air", Contract: No. UDARPPD.01.02.01-20-0153/19-00, dated February 5, 2020.

References

[1] R. Hyndman and G. Athanasopoulos, "Forecasting: principles and practice," 2021.

[2] B. G. Horne, H. T. Siegelmann, and C. L. Giles, "What narx networks can compute," in *SOFSEM '95: Proceedings of the 22nd Seminar on Current Trends in Theory and Practice of Informatics*, (Berlin, Germany), pp. 95–102, Springer-Verlag, 2015.

[3] O. Omolaye and T. Badmos, "Predictive and comparative analysis of narx and nio time series prediction," *American Journal of Engineering Research (AJER)*, pp. 155–165, 2017.

[4] T. Liu, T. Wu, M. Wang, M. Fu, J. Kang, and H. Zhang, "Recurrent neural networks based on lstm for predicting geomagnetic field," in *Proceedings of the 2018 IEEE International Conference on Aerospace Electronics and Remote Sensing Technology (ICARES)*, (Bali, Indonesia), p. 1–5, Institute of Electrical and Electronics Engineers (IEEE), 20–21 September 2018 2018.

[5] J. Fan, Q. Li, J. Hou, X. Feng, H. Karimian, and S. Lin, "A spatiotemporal prediction framework for air pollution based on deep rnn," *ISPRS Ann. Photogramm. Remote. Sens. Spat. Inf. Sci*, vol. 4, p. 15–22, 2017.

[6] MathWorks, *Deep Learning Toolbox*, 2022. Design, train, and analyze deep learning networks [13.09.2022].

[7] A. Heydari, M. Majidi Nezhad, D. Astiaso Garcia, and et al., "Air pollution forecasting application based on deep learning model and optimization algorithm," *Clean Techn Environ Policy*, vol. 24, p. 607–621, 2022.

[8] S. Agarwal, S. Sharma, R. Suresh, M. Rahman, S. Vranckx, B. Maiheu, L. Blythb, S. Janssen, P. Gargava, V. Shukl, and S. Batra, "Air quality forecasting using artificial neural networks with real time dynamic error correction in highly polluted regions," 2020.

[9] G. Gennaro, L. Trizio, A. Di, J. Pey, N. Pérez, M. Cusack, A. Alastuey, and X. Querol, "Neural network model for the prediction of pm10 daily concentrations in two sites in the western mediterranean," *Sci Total Environ*, vol. 463–464, p. 875–883, 2013.

[10] D.-R. Liu, S.-J. Lee, Y. Huang, and C.-J. Chiu, "Air pollution forecasting based on attention- based lstm neural network and ensemble learning," *Expert Syst*, vol. 37, no. 3, p. 1–16, 2020.

[11] M. Zeinalnezhad, A. Gholamzadeh, and J. Kleme, "Air pollution prediction using semi- experimental regression model and adaptive neuro-fuzzy inference system," *J Clean Prod*, vol. 261, p. 121218, 2020.