

RECURRENT LSTM NETWORK VERSUS FEEDFORWARD CLASSICAL APPROACH TO THE TIME SERIES PREDICTION

Tomasz Ciechulski^{✉1)}, Stanisław Osowski²⁾

1) *Military University of Technology, Faculty of Electronics, ul. gen. Sylwestra Kaliskiego 2, 00-908 Warsaw, Poland*
(✉ tomasz.ciechulski@wat.edu.pl)

2) *Warsaw University of Technology, Faculty of Electrical Engineering, pl. Politechniki 1, 00-661 Warsaw, Poland*

Abstract

The paper presents a comparative analysis of the recurrent deep Long Short-Term Memory (LSTM) networks and the classical feedforward approach in forecasting the time series. LSTM neural networks, unlike feedforward supervised networks (Multilayer Perceptron – MLP, Radial Basis Function – RBF, Support Vector Machine – SVM), belong to the family of deep recurrent structures using a modified neuron model called the LSTM cell. The recurrence built into the cell structure allows for long memorization of the short patterns, ideally suited for forecasting processes in which there are close connections between successive time records. The work focuses on forecasting the 24-hour ahead atmospheric pollution levels for such pollutants as PM₁₀ dust, SO₂, NO₂, and ozone O₃. The advantage of the recurrent approach over classical feedforward applications will be shown.

Keywords: time series prediction, atmospheric pollution, forecasting, neural networks, LSTM networks.

1. Introduction

The time series prediction is a practical task applicable to many life problems. A typical example of it is forecasting atmospheric pollution. The high pollution level harms human health as well as the entire natural environment. Air pollution is the cause of many deaths. It is estimated that the combined or joint effects of ambient and household air pollution cause about 7 million premature deaths per year from stroke, heart disease, chronic obstructive pulmonary disease, lung cancer, and acute respiratory infections [1].

Hence, the problem is becoming increasingly important, forcing various administrative decisions to be made to reduce its impact on the environment. The significant role this process may play in the forecasting of the pollution level from day to day. The other is also the financial aspect of the problem, due to the penalties that the country must pay for exceeding the standards set by the European Union (the new European Air Quality Directive 2008/50/EC) and the European Regulation and Limits on Air Pollution 2022, defining restrictions for yearly and 24-hour average *particulate matter* (PM) 10 µm concentrations [2, 3].

For years, the feedforward approaches (either artificial neural networks or decision trees) have played a main role in forecasting the time series [4–6]. Especially, neural networks of various structures have often been used. Among them, two types of networks can be distinguished. One family uses only one hidden layer and may be called the shallow structure. An example of such a family is the *multilayer perceptron* (MLP), *radial basis function* (RBF) network, or *support vector machine* (SVM). The other solution, very popular now, is composed of many hidden layers and forms the deep network family [7]. Among them, the recurrent *long short-term memory networks* (LSTM) were found to be very efficient in time series prediction [8–10].

For proper operation of classical shallow networks, the external generation of the process features, which are based on the measurement data, is necessary. These features form the input attributes to these networks. The feature generation process is not easy and requires expert knowledge of the mechanism of pollutant formation. Much simpler in this respect is the use of deep networks. In their multilayer structure, they try to recreate the mechanism of pollutant formation automatically. An excellent example of such a network is the LSTM which can memorize past information of the analysed sequence of values and use it in creating future sequence terms [8–10].

This paper will compare the performance of both approaches to forecasting atmospheric pollutants. It will concern four pollutants: particulate matter (dust) PM_{10} , sulphur dioxide – SO_2 , nitrogen dioxide – NO_2 , and ozone – O_3 . The forecasting algorithm will use the information on the hourly changes of these processes in the past and associate them with the selected atmospheric parameters, including wind speed, humidity, and temperature.

2. Atmospheric pollution database

The numerical experiments will use a database of PM_{10} , SO_2 , NO_2 , and O_3 measured over 4 years in the Ursynów district in Warsaw. Information on wind speed, humidity, and temperature comes from the database of the *Interdisciplinary Centre for Mathematical and Computer Modelling* (ICM) of the University of Warsaw. Exemplary time series of the hourly values PM_{10} , SO_2 , NO_2 , and O_3 corresponding to 100 following days of the selected year from the database are shown in Fig. 1.

There are visible large changes in the values of their concentration, observed from hour to hour. A good view of these variabilities is provided by the statistical measure of these variabilities, especially the ratio of the standard deviation to the mean value. Table 1 presents these values for the entire database corresponding to four years of measurements. In each case, the ratio of the standard deviation to the mean value is significantly above 0.5. There are also visible large differences between the maximum and minimum values for each pollutant.

Table 1. Statistical characterization of the analysed atmospheric pollutants.

Pollutant	Mean value m [$\mu\text{g}/\text{m}^3$]	Standard deviation std [$\mu\text{g}/\text{m}^3$]	Ratio std/ m	Max value [$\mu\text{g}/\text{m}^3$]	Min value [$\mu\text{g}/\text{m}^3$]
PM10	33.56	26.24	0.7820	414.9	0.72
SO2	9.89	8.78	0.8882	157.8	0.70
NO2	24.82	17.85	0.7192	203.8	0.43
O3	48.43	31.10	0.6421	189.1	0.10

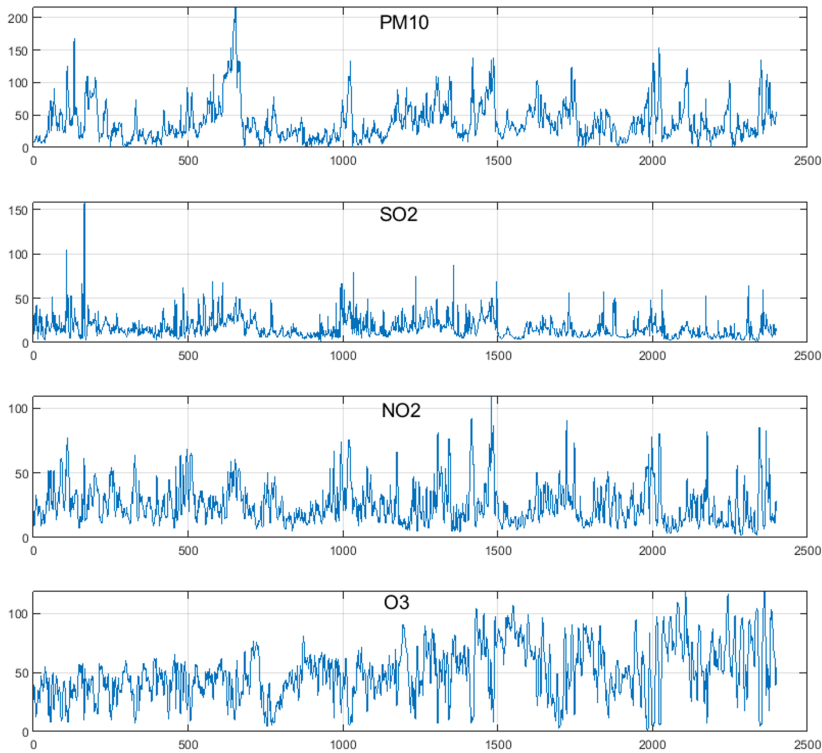


Fig. 1. Hourly changes of the measured time series of PM₁₀, SO₂, NO₂, and O₃ during the following 100 days of the year (pollutant values are expressed in µg/m³).

From the point of view of time series forecasting, absolute changes from hour to hour occurring in the values of forecasted numbers are very important. To highlight them, Fig. 2 presents graphs of subsequent changes (differences between two adjacent measurement samples ($x(k)-x(k-1)$)) for PM₁₀, SO₂, NO₂, and O₃.

Of particular importance are the maximum (absolute) differences in pollution levels in adjacent hours, as well as the ratio of the standard deviation to the absolute mean value of the forecasted values. Table 2 presents these maximum values and the ratio of the standard deviation to the absolute mean value.

Table 2. Maximum absolute values and the ratio of standard deviation to the mean absolute value of hourly changes of PM₁₀, SO₂, NO₂, and ozone.

Parameter	PM ₁₀	SO ₂	NO ₂	O ₃
Max. value [µg/m ³]	167.1	99.2	97.3	107.2
Ratio std/mean	1.6480	2.1114	1.6161	1.5511

All maximum values of differences are many times higher than the average values of pollutants, although their ranges are slightly lower than those of the original values. On the other side, the ratio of the standard deviation of differences to their absolute mean value is much higher. For all pollutants, these values are greater than 1, which indicates the high level of difficulty in their prediction.

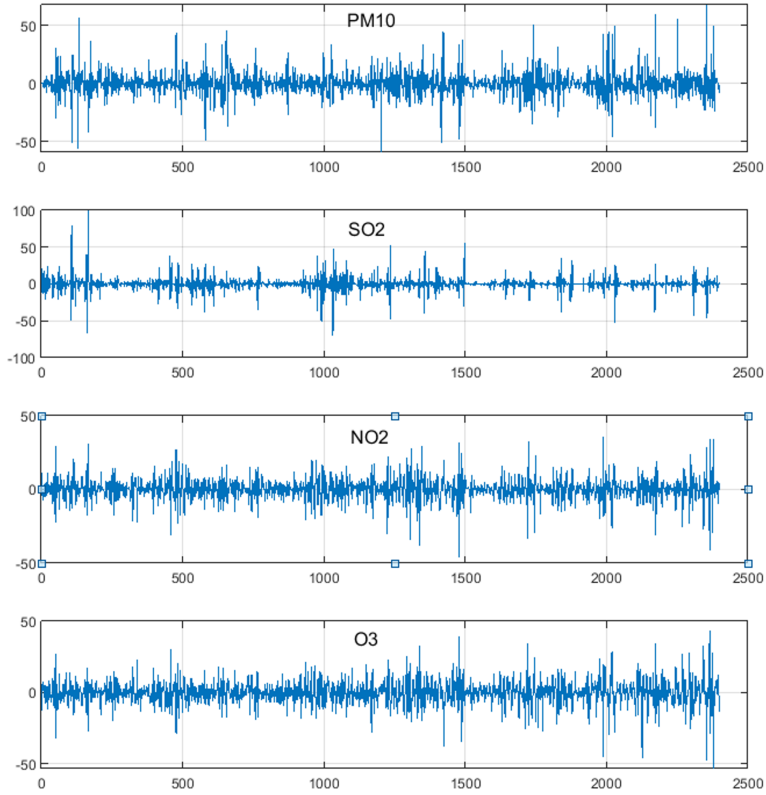


Fig. 2. Graph of differences in pollution levels between two adjacent hours in 100 days of the analysed database (values on the vertical axis are expressed in $\mu\text{g}/\text{m}^3$).

3. Feedforward networks for air pollution forecasting

Computerized solutions based on artificial intelligence play a key role in forecasting the time series corresponding to atmospheric pollution. The important thing is the application of the decision tree ensembles and the shallow neural networks containing one hidden layer, *e.g.*, MLP, RBF, SVM [4, 11]. In these cases, the challenge is the manual selection of process features. Usually, it is composed of two stages: the initial generation of the set of process descriptors defined directly from the measurement data, and then the selection of the most important components from this initial set. The selected features form the input attributes delivered to the final predictors composed of neural networks or decision trees. The use of a deep solution allows us to avoid the stage of manual generation and selection of diagnostic process features since this set is generated automatically within many hidden layers of the structure. In this way, we simplify the process of designing a predictive system, although it deprives the user of information on which of the factors plays the most important role in the prediction process.

3.1. Classical feedforward approach

In the classical solution, the user decides on the feature selection that is used in the forecasting process. Expert knowledge resulting from many years of process observations is very useful in this task. Many different proposals for the initial selection of such process descriptors can be found

in the literature. In works [4, 11], very good results were obtained by defining many different statistical parameters of the time series, which represent the descriptors of the process, treated as the first step in the process feature generation.

Initially, the minimum, maximum, and average values of atmospheric factors on individual days, including temperature, wind speed, and humidity were considered. They are defined on the values forecasted by the ICM for the next day and on the known historical values from the previous day. They also include the trend of hourly changes. The next 24 descriptors correspond to the known values of 24-hour pollution levels of the day preceding the forecast. Additionally, the type of day (working or weekend) and the season (summer, autumn, winter, and spring days) were coded due to their impact on the level of pollution.

A set of numerical process descriptors defined in this way was subjected to selection for the final set of the most important process features. Among the many existing selection methods, the best results were obtained using a genetic algorithm [12] and sequential multi-step elimination [13]. Finally, a limited set of process features that best correlate with the forecasted series was selected from the initial descriptors. The results of the selection differ for the pollution type concerning the number of selected features and their content. In the case of PM_{10} , the selected set contained 19 features, in the case of SO_2 , there were only 13, in the case of NO_2 , 15, and in the case of O_3 , 18.

The selected sets of features represent the input attributes for the executive element constituting the final solution of the prediction system. The paper [4] has analysed the application of the set of shallow neural networks and a random forest of decision trees, both arranged in the form of an ensemble. As a result, the classical approach to forecasting can be illustrated in the form of a diagram presented in Fig. 3.

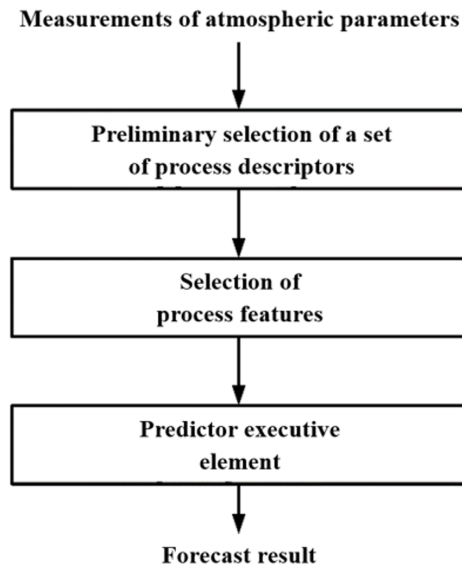


Fig. 3. Diagram of the prediction process used in the classical approach.

3.2. The recurrent LSTM predictor

Unlike the classical feedforward approach, the use of a recurrent solution using LSTM introduces many simplifications resulting from the way this recurrent network works. First, there is no manual selection of process features, because the network itself decides on their importance

automatically, based directly on the analysed time series. The LSTM network contains 3 layers: an input layer to which the original measurement signals are fed, a recurrent hidden layer using many LSTM-type memory cells (equivalent to a classical neuron), and a linear output layer of the network powered by the output signals of the hidden layer which generates the predicted values. The general structure of such a network is presented in Fig. 4.

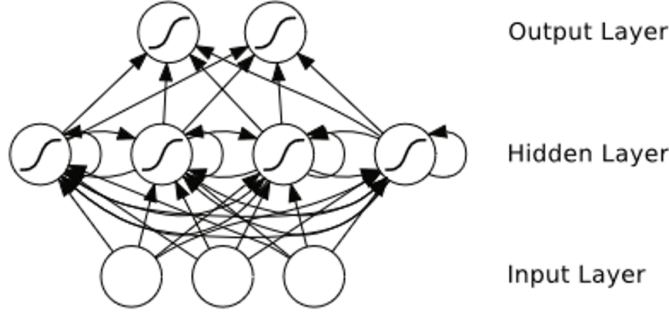


Fig. 4. General form of the recurrent neural network used in the LSTM structure [9].

A numerical sequence corresponding to the measured values of pollutants and atmospheric parameters (wind speed, temperature, humidity) in the subsequent time moments is directly fed to the input layer. The network analyses subsequent changes in these values using a cellular processing scheme in the hidden layer representing recurrence, which can be presented as a time expansion for two subsequent time moments as shown in Fig. 5 [14].

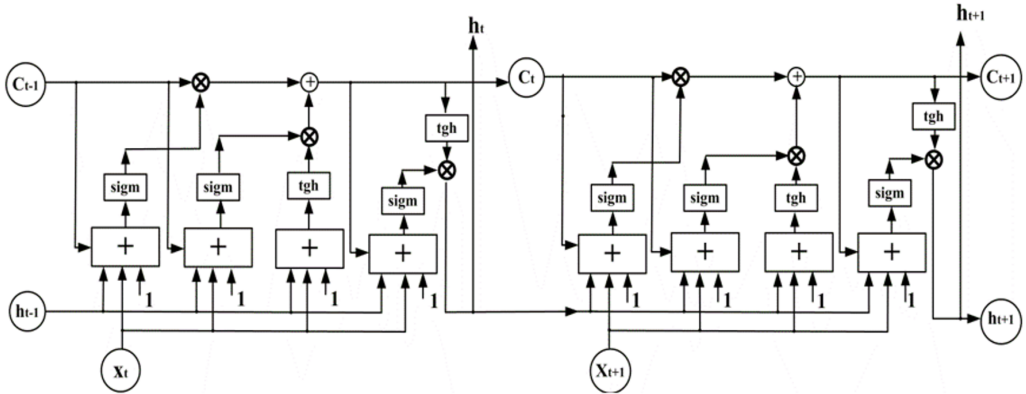


Fig. 5. Model of signal flow in two subsequent time moments in the hidden layer of an LSTM network composed of the memory LSTM cells.

Input signals representing the successive time points of the series are subjected to processing using LSTM cells. The cell is composed of four sigmoidal units supplied by the actual sample and previous signals representing the memory and its output. Each cell simultaneously processes these three input signals at the successive time points: $t, t + 1, t + 2, \dots, T$ (where T is the measurement window). It operates on the following signals: x_t – network input signal at the current time points t (input vector), C_{t-1} – cell memory signal from the previous time point $t - 1$, h_{t-1} – cell output signal from time point $t - 1$. As a result, the LSTM cell in the hidden layer produces two types of

output signals: C_t – the modified cell memory signal at the current time points t , and h_t – the cell output signal at the current time point t . This output signal of the hidden neuron (with many LSTM cells in the hidden layer, it is a vector) is fed to the output layer of the network neurons, as shown in Fig. 4. These signals create the final processing results representing the predicted values in the current time points.

An important role in LSTM networks is played by three multiplicative gates, marked with the symbol $\times$. The upper gate on the left, called the forgetting gate, decides what part of the memory C_{t-1} from the previous time point will pass to the next time point. The level of transition of this content is determined by the signal generated by the sigmoid neural layer, visible on the left side of the model. The second gate (middle symbol $\times$) is an adjustable valve of the new memory participating in the generation of the content of the C_t cell for the current time point. In result, the old content of the memory C_{t-1} is replaced with the new one – C_t . The third multiplicative gate is used to generate the output signal h_t at the current time point t . As a result of controlling the individual valves, it is possible to pass or block a specific portion of information from the previous time points, which significantly affects the generation of a new predicted value.

Network training consists of adapting the values of the weights of the individual network connections (parameters of the adders in the hidden layer and the weights of the feedforward output layer connections) by minimizing the error function of the training data in the form of pairs (input value x and the destination value at the output d) defined for the subsequent time points. The optimization process is carried out using the gradient method with gradient generation, applying the backpropagation [14]. Input signals are first sent forward in the network (within the adopted time window T), and then the error at the output in individual time points is sent in the reverse direction (from output to input). On this basis, gradient components are generated for the individual network weights, similarly as in classical neural networks [14].

4. Results of numerical experiments

The quality of the prediction process will be assessed in this work based on the following quality measures: the *mean absolute percentage error* (MAPE), *mean absolute error* (MAE), *root mean squared error* (RMSE), and correlation coefficient R between the predicted and actual sequence [11, 14]. Since the R coefficient measures only the strength and direction of the relationship between two variables, we have also included the R^2 measure (a more objective assessment of regression), defined as follows

$$R^2 = \left(1 - \frac{\sum_i (y_i - d_i)^2}{\sum_i (d_i - \text{mean}(d_i))^2} \right), \quad (1)$$

where y_i represents the predicted, and d_i the true (measured) values. This coefficient determines the proportion of variance in the dependent variable that can be explained by the independent variable. It shows the goodness of fit.

Due to the occurrence of many very small values (close to zero) of some measured pollutants (especially in the O_3 sequence), the use of the classical definition of the MAPE leads to unreliable results. Hence, most published papers focus on MAE and RMSE errors, although these measures are not very objective. In this work, a different way to avoid this problem was adopted. In such

a case, a modified MAPE measure (named MAPE_n based on the norm definition will be used (2):

$$\text{MAPE}_n = \frac{\text{norm}(\mathbf{d} - \mathbf{y})}{\text{norm}(\mathbf{d})}, \quad (2)$$

where $\mathbf{d}$ denotes the vector of measured (true) values and $\mathbf{y}$ is the vector of predicted values. This modification of MAPE is also a relative type of measure, but it avoids the traps resulting from very small target values subjected to prediction. In the presentation of the results, we will limit ourselves only to the test set, not participating in the training.

Numerical experiments using the LSTM network were conducted using the database presented in Section 2 of this paper, covering 4 years. The first three years were used in the training, and the last year was used in testing the adapted system. A separate forecasting system was created for each type of pollution, differing in the number of LSTM cells in the hidden layer as well as learning parameters. The input to the network is provided with subsequent values of time series corresponding to the pollutant analysed (PM_{10} , SO_2 , NO_2 , or O_3) and measured values of atmospheric parameters (wind speed, temperature, humidity). LSTM predictors were combined into a set consisting of 5 units differing in the number of LSTM cells. This number was selected individually for each type of pollutant. In each case, the starting value was changed randomly in the range of $\pm 20\%$. In the case of PM_{10} it was $250 \pm 20\%$, $\text{SO}_2 - 350 \pm 20\%$, $\text{NO}_2 - 400 \pm 20\%$, $\text{O}_3 - 270 \pm 20\%$.

The obtained results in the form of average values of the ensemble members and their integrated value in the ensemble are presented in Table 3. The integration was defined by averaging the hourly predicted values, and the average represents the mean of yearly averages obtained in 5 runs of the prediction process. In the case of ozone, where there were many small values (smaller than 0.01), the use of the classical MAPE definition leads to unrealistic values. Hence, for this pollutant, a modified MAPE measure using (1) based on the Euclidean norm was used.

Table 3. Values of quality measures of prediction by recurrent LSTM networks for PM_{10} , SO_2 , NO , and O_3 .

Pollutant		MAPE [%]	MAE [$\mu\text{g}/\text{m}^3$]	RMSE [$\mu\text{g}/\text{m}^3$]	R	R^2
PM_{10}	Average $\pm$ std	18.47 $\pm$ 0.48	4.67 $\pm$ 0.05	7.80 $\pm$ 0.06	0.9643 $\pm$ 0.0003	0.9114 $\pm$ 0.0020
	Ensemble	18.11	4.43	7.52	0.9648	0.9134
SO_2	Average $\pm$ std	21.15 $\pm$ 2.59	1.86 $\pm$ 0.06	3.79 $\pm$ 0.02	0.8721 $\pm$ 0.0010	0.7526 $\pm$ 0.0026
	Ensemble	21.10	1.86	3.79	0.8743	0.7532
NO_2	Average $\pm$ std	20.21 $\pm$ 0.65	4.08 $\pm$ 0.03	6.39 $\pm$ 0.01	0.9121 $\pm$ 0.0010	0.8262 $\pm$ 0.0037
	Ensemble	19.81	4.03	6.35	0.9129	0.8296
O_3	Average $\pm$ std	15.97 $\pm$ 0.21	5.68 $\pm$ 0.08	8.36 $\pm$ 0.03	0.9796 $\pm$ 0.0004	0.9194 $\pm$ 0.0006
	Ensemble	15.78 (norm)	5.55	8.23	0.9604	0.9219

The best results in terms of the MAPE measure and the correlation coefficient R correspond to the PM_{10} and ozone. The ratios of the MAE error to the average values for these pollutants are 0.139 for PM_{10} and 0.132 for ozone, respectively. The corresponding values for SO_2 and NO_2 are slightly higher, and the correlation coefficient R is lower. In the case of SO_2 , the ratio of the MAPE value to the average is 0.182, and for NO_2 it is equal to 0.164. The use of an integrated ensemble of predictors improves the results to the yearly average of its members; however, in a not significant way. The reason for this is the small diversity among the ensemble LSTM members.

The results presented above are compared to the best results obtained for the same database using the classical feedforward approach. The comparison is based on the results presented in [4] for the same database. The best-found feature selection method in this solution was the sequential multi-step elimination. The results of its application feed the set of decision trees in the form of a random forest [13] working in a regression mode (100 trees in the forest, the number of randomly selected decision variables in nodes equal to 4). The best prediction results for one year of data (not participating in the learning) are presented in Table 4. These values were obtained for randomly selected days from the whole period of years in the database. These samples do not constitute a consecutive time sequence. This is an easier task because randomly selected days are more like their neighbours used in the learning. The LSTM network approach presented in this paper solves a bit more general task of predicting the value of pollutants in the subsequent hours of the entire next year (not participating in the learning process). In such cases, the patterns of changes subjected to prediction may differ significantly from those of the previous years used in training.

Table 4. Best values of quality measures of the classical feedforward approach using a random forest of decision trees for PM₁₀, SO₂, NO₂, and O₃ prediction.

Pollutant	MAPE [%]	MAE [$\mu\text{g}/\text{m}^3$]	RMSE [$\mu\text{g}/\text{m}^3$]	<i>R</i>	<i>R</i> ²
PM ₁₀	19.96	6.04	9.17	0.933	0.879
SO ₂	18.35	1.57	2.38	0.932	0.793
NO ₂	18.27	3.94	6.06	0.931	0.839
O ₃	17.31 (norm)	7.50	10.59	0.882	0.831

Regardless of this, the prediction results of the LSTM method are better for PM₁₀: MAPE = 18.11% compared to 19.96% of the classical method, and for ozone: 15.78% compared to 17.31%. A clear advantage of the LSTM for these pollutants is also observed in other quality measures: MAE, RMSE, and the correlation coefficient *R*.

The LSTM application resulted in slightly worse measures in forecasting the SO₂ and NO₂, although the differences are relatively small. The case of SO₂ and NO₂ is very special and not easy to interpret. The different statistical characteristics of changes in the values of the time series might affect neural networks in various ways, resulting in different accuracy of prediction. The important thing is the magnitude of the direct changes of the time series values happening from hour to hour, which might influence the performance of a particular type of neural network. The higher these changes, the more difficult it is to forecast their values. This is illustrated in Table 2 in which SO₂ and NO₂ are characterized by the highest values of the ratio representing the standard deviation to the mean value of such changes. This might also, among many others, be the reason.

Moreover, we should mention different ways of selecting the testing data from the dataset, which might also influence the results for the pollutants. Observe that the testing data used in the LSTM approach are more demanding since the predicting system operates on separate year samples, which are far distant from the learning samples and might represent a different distribution.

The quality of the classical feedforward solution presented in [4] is closely related to the choice of the aggregating method in an ensemble. The use of random forest in this role [4] was found to be the best. Attempts to use the ensemble in the form of 3 neural networks (MLP, RBF, SVM) aggregated by averaging have shown a significant deterioration of the results. The MAPE values obtained in this solution have significantly increased. In the case of PM₁₀, this increase was from 19.96% to 23.93%, for SO₂ from 18.35% to 27.43%, for NO₂ from 18.27% to 29.10%, and O₃ from 17.31% to 23.91%. These facts show how difficult the optimal choice of the prediction system is.

5. Conclusions

The use of recurrent learning in the pollution prediction task significantly simplifies the process of building a system in comparison to the classical feedforward approach, since it generates the features automatically. At the same time, in most cases, it leads to improved results of quality measures. However, the price paid is the interpretability of the problem. A deep network is a “black box” type system with a complex signal processing structure and a huge number of parameters adapted in the learning process. The method of developing a decision is not transparent. In this respect, the use of the feedforward approach is much more understandable because, in the process of creating a model, the influence of individual factors on the prediction results of the system can be directly analysed. However, the problem is that different selection mechanisms might lead to different choices of the features, and there is no universal approach.

Meantime, a lot of research has been conducted to study the explainability of the recurrent networks, including the recurrent LSTM. Such methods as *SHapley Additive exPlanations* (SHAP), *Local Interpretable Model-agnostic Explanations* (LIME), and *Layer-wise Relevance Propagation* (LRP) have been tried [18–20]. The Shapley values of SHAP [18] evaluate the importance of a particular input feature by comparing the model’s predictions *with and without that variable*, considering all possible arrangements of other variables. LRP [19] is a technique to explain the predictions in terms of input variables by assigning a score to each of the input variables, indicating the extent to which they contribute to the prediction. It works by reverse-propagating the prediction through the network, recomposing the LSTM forward passing in terms of linear mapping, gated interactions, and accumulation. LIME [20] is based on perturbing other input data in a small neighbourhood of the instance being explained and observing the changes in predictions generated by the model. In future works, we will investigate these models to get a deeper interpretation of the importance of different values in the prediction process of LSTM.

Deep networks are generally associated with a prolonged learning process and the need to use high-performance graphics processors, which increases the cost of processing. To reduce this time, transfer learning based on pre-trained models is often used. Thanks to this, the training of the network may be significantly shortened. In the case of the LSTM, the situation is different. The transfer is understood as using the built-in functions controlling the learning process (generation of gradient, adapting the learning rate, *etc.*). The network is trained from scratch using the actual learning data. Still, learning of the recurrent network is a complex task, usually much more difficult than the feedforward network. Thanks to transferring the well-organized structure of the learning process included in the available MATLAB program, it was possible to shorten the training time of LSTM significantly. For example, in the case of our problem, the typical training time was around 50 minutes, applying the 64-bit laptop operating under Windows 10 Education, Intel Core i7-6700HQ processor, 2.60 GHz CPU, with 16 GB DDR4RAM, and an NVIDIA GeForce 940MX graphics card with 2 GB VRAM. In such cases, it is only slightly longer than the classical approach based on feedforward neural networks and random forests. All numerical experiments have been conducted, version R2024b [15].

A problem that requires further study is the selection of a deep model in the processing system. Although most works confirm the advantage of the LSTM network, other solutions can also be considered, such as *gated recurrent units* (GRU), *convolutional neural networks* (CNN), or the use of transformers for predicting numerical sequences [11, 16, 17].

It seems interesting to use an autoencoder in the initial processing phase (compression of results for noise reduction and returning to the original values based on the compressed signals). This type of procedure is recommended for high levels of noise in the measured signals. To obtain a proper evaluation of the proposed approach, it is necessary to increase the population of training data, especially in the context of using deep learning with a very large number of parameters subjected to training.

Acknowledgements

This work was financed by the Military University of Technology under Research Project UGB/22-060/2025/WAT.

References

- [1] World Health Organization, *Exposure & health impacts of air pollution*. <https://www.who.int/teams/environment-climate-change-and-health/air-quality-energy-and-health/health-impacts/exposure-air-pollution>
- [2] European Commission, *EU air quality standards*. https://environment.ec.europa.eu/topics/air/air-quality/eu-air-quality-standards_en
- [3] European Environment Agency, *Air pollution*: <https://www.eea.europa.eu/en/topics/in-depth-air-pollution>
- [4] Siwek, K., & Osowski, S. (2016). Data mining methods for prediction of air pollution. *International Journal of Applied Mathematics and Computer Science*, 26(2), 467–478. <https://doi.org/10.1515/amcs-2016-0033>
- [5] Kalinowski, P., Woźniak, Ł., Jasiński, G., & Jasiński, P. (2020). Evaluation of the NO₂ concentration prediction possibility based on static and dynamic responses of TGS sensors at changing humidity levels. *Metrology and Measurement Systems*, Vol. 27 (2020) No. 1, pp. 167–179. <https://doi.org/10.24425/mms.2020.131716>
- [6] Miczulski, W., & Sobolewski, Ł. (2012). Influence of the GMDH Neural Network Data Preparation Method on UTC(PL) correction prediction results. *Metrology and Measurement Systems*, Vol. 19 (2012), No. 1, pp. 123–132. <https://doi.org/10.2478/v10178-012-0011-1>
- [7] Goodfellow, I., Bengio, Y., & Courville, A. (2016a). Deep Learning. In *MIT Press eBooks*. <https://dl.acm.org/citation.cfm?id=3086952>
- [8] Greff, K., Srivastava, R. K., Koutnik, J., Steunebrink, B. R., & Schmidhuber, J. (2016). LSTM: A Search Space Odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, 28(10), 2222–2232. <https://doi.org/10.1109/tnnls.2016.2582924>
- [9] Graves, A. (2012). Supervised Sequence Labelling with Recurrent Neural Networks. In *Studies in Computational Intelligence*. <https://doi.org/10.1007/978-3-642-24797-2>
- [10] Schmidhuber, J. (2014). Deep learning in neural networks: An overview. *Neural Networks*, 61, 85–117. <https://doi.org/10.1016/j.neunet.2014.09.003>
- [11] Zhang, B., Rong, Y., Yong, R., Qin, D., Li, M., Zou, G., & Pan, J. (2022). Deep learning for air pollutant concentration prediction: A review. *Atmospheric Environment*, 290, 119347. <https://doi.org/10.1016/j.atmosenv.2022.119347>
- [12] Goldberg, D. (2003). Algorytmy genetyczne i ich zastosowania (Genetic Algorithms and their Applications), WNT, Warszawa. (in Polish)
- [13] Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32. <https://doi.org/10.1023/a:1010933404324>
- [14] Osowski, S., & Szmurło, R. (2023). *Matematyczne modele uczenia maszynowego w językach MATLAB i Python (Mathematical Models of Machine Learning in MATLAB and Python)*, Oficyna Wydawnicza PW, Warszawa. (in Polish)

- [15] MathWorks, (2024). *MATLAB User Manual*, Natick, USA.
- [16] Chen, C., Qiu, A., Chen, H., Chen, Y., Liu, X., & Li, D. (2023). Prediction of pollutant concentration based on Spatial-temporal attention, RESNET and CONVLSTM. *Sensors*, 23(21), 8863. <https://doi.org/10.3390/s23218863>
- [17] Li, D., Liu, J., & Zhao, Y. (2022). Forecasting of PM_{2.5} concentration in Beijing using hybrid deep learning framework based on attention mechanism. *Applied Sciences*, 12(21), 11155. <https://doi.org/10.3390/app122111155>
- [18] Bernardelli, M. (2025). Adding explainability to LSTM modelling of business tendency survey results. In E. W. Ziemba, W. Grzenda, & Ramsza M. (Eds.), *Explainable Artificial Intelligence for Sustainable Development*. <https://doi.org/10.4324/9781003599227>
- [19] Arras, L., Arjona-Medina, J., Widrich, M., Montavon, G., Gillhofer, M., Müller, K., Hochreiter, S., & Samek, W. (2019). Explaining and Interpreting LSTMs. In *Lecture Notes in Computer Science* (pp. 211–238). https://doi.org/10.1007/978-3-030-28954-6_11
- [20] Sen, D., Deora, B. S., & Vaishnav, A. (2025). Explainable deep learning for time series analysis: Integrating SHAP and LIME in LSTM-based models. *Journal of Information Systems Engineering & Management*, 10(16s), 412–423. <https://doi.org/10.52783/jisem.v10i16s.2627>



Tomasz Ciechulski received the M.Sc. degree in electronics and communications engineering and the Ph.D. degree in artificial neural networks from the Military University of Technology in Warsaw, Poland in 2012 and 2017 respectively. He is currently Assistant Professor with the Military University of Technology in Warsaw, Poland. His research activity focuses on forecasting of time series using neural networks and deep learning methods. He has authored or co-authored over 20

journal and conference publications.



Stanisław Osowski received the M.Sc., Ph.D., and D.Sc. degrees from the Warsaw University of Technology, Poland, in 1972, 1975, and 1981, respectively, all in electrical engineering. Currently, he is a professor of electrical engineering at the Faculty of Electrical Engineering, Warsaw University of Technology and Faculty of Electronic Engineering, Military University of Technology. His research and teaching interests are in the areas of neural networks, machine learning and their application

in different areas of electrical and biomedical engineering. He is an author or co-author of more than 200 scientific papers and ten books.