BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 68, No. 5, 2020
DOI: 10.24425/bpasts.2020.134659

Bagging and boosting techniques in prediction of particulate matters

D. TRIANA[1] and S. OSOWSKI[1,2*]

[1] Warsaw University of Technology, Pl. Politechniki 1, 00-661 Warsaw, Poland
[2] Military University of Technology, ul. gen. Sylwestra Kaliskiego 2, 00-908 Warsaw, Poland

Abstract. The paper presents new ensemble solutions, which can forecast the average level of particulate matters PM10 and PM2.5 with increased accuracy. The proposed network is composed of weak predictors integrated into a final expert system. The members of the ensemble are built based on deep multilayer perceptron and decision tree and use bagging and boosting principle in elaborating common decisions. The numerical experiments have been carried out for prediction of daily average pollution of PM10 and PM2.5 for the next day. The results of experiments have shown, that bagging and boosting ensembles employing these weak predictors improve greatly the quality of results. The mean absolute errors have been reduced by more than 30% in the case of PM10 and 20% in the case of PM2.5 in comparison to individually acting predictors.

Key words: ensemble of predictors, bagging, boosting, PM pollution.

1. Introduction

Air pollution has become an important concern nowadays of modern societies considering its harmful implications on human beings and ecosystems [1–3]. The problem is strictly associated with the level of such pollutants as particulate matters, SO2, NOx and O3. Especially important is the particulate matter (PM) of the diameters to 10 μm (PM10) and 2.5 μm (PM2.5). The main source of PM is the vehicular traffic and dust of the streets created by the circulation. These particles have direct impact on human health via inhalation [1, 4]. To counteract their harmful effect, the monitoring of air quality and special policies have been implemented to protect public health and ensure air pollution below the maximum levels in the region.

Particulate matters (PM) are of special importance for European and American policies defining restrictions for yearly and 24-hour average PM concentrations [3, 4]. To respect the short-term limit values defined by these restrictions and reduce the concentration levels, the emission abatement actions should be planned at least one day in advance. Hence, one day ahead forecasting is needed.

Classical statistical methods (AR, ARMA, ARIMA) have been nowadays replaced by nonlinear models, for example neural networks or decision trees [5]. Their learning processes apply the pollution data from the past as well as some meteorological information. No understanding of the mechanism of pollution creation is needed. Many different solutions to this problem have been proposed in the past [5–13]. They include multilayer perceptron (MLP), radial basis function (RBF), Support Vector Machine as well as Elman network. The ensembles of many neural predictors based on averaging or dynamic integration have been also proposed [10–12] to get more accurate forecast. Nowadays, ensembles of predictors are among the most competitive forms in solving the predictive tasks [14, 15].

The paper will exploit this trend of investigations, however, by applying completely different way of building and integrating the ensemble. Two approaches will be studied. One is based on bagging and the second on boosting philosophy. Bagging is the method for generating multiple versions of predictors trained on different sets of learning data and using them to get an aggregated prediction values for unseen input samples. Boosting strategy also creates many predictors, but the next added units are learned on the samples, which were classified less accurately by previous models. The numerical experiments performed on the PM10 and PM2.5 have shown much higher accuracy in comparison to classical methods of ensemble integration.

The paper is organized as follows. Chapter 2 presents the statistical characterization of the data bases representing both types of particulate matters. Chapter 3 is devoted to generation and selection of diagnostic features. The next chapter presents the results of application of single predictors based on deep multilayer perceptron and decision tree. The following two chapters describe the idea of application of bagging and boosting in creation of ensemble. Chapter 7 presents the results of numerical experiments performed on PM10 and PM2.5 by applying bagging and AdaBoost. The concluding chapter summarizes the results and compares them with other classical methods of ensemble creation.

2. Characterization of data used in experiments

The numerical experiments will be performed on two types of particulate matters: PM2.5 and PM10. The data of the first

*e-mail: sto@iem.pw.edu.pl

Manuscript submitted 2019-12-20, revised 2020-05-19, initially accepted for publication 2020-07-06, published in October 2020

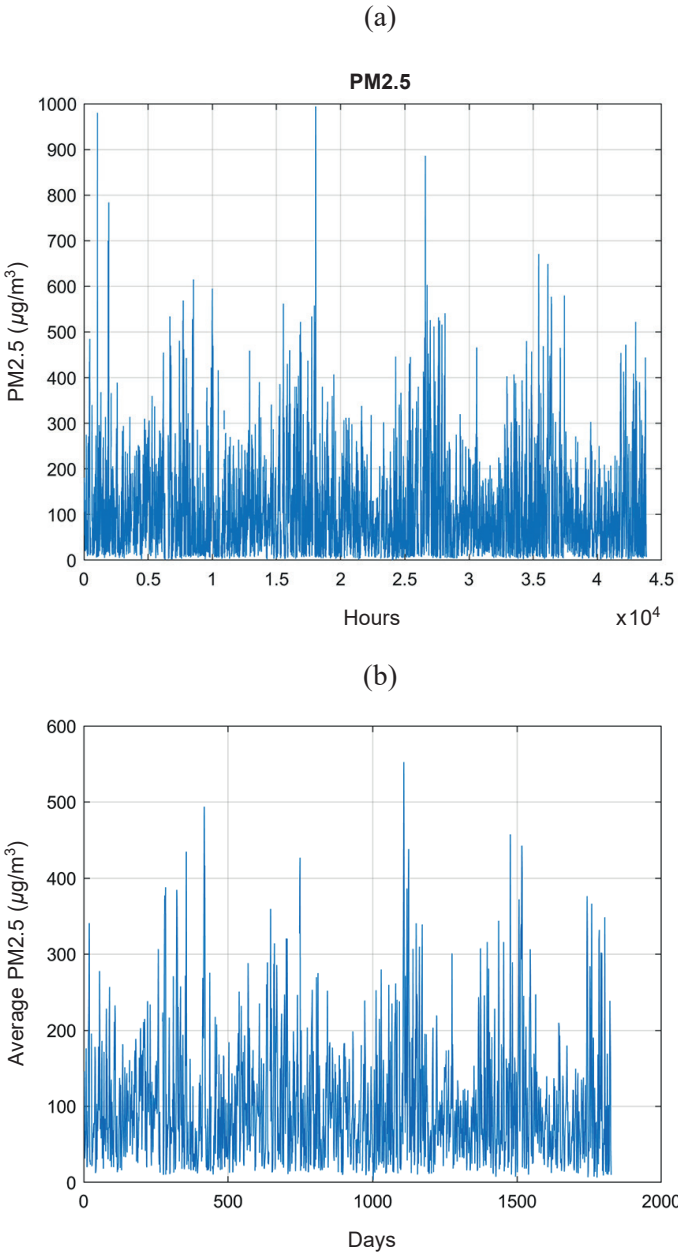

Fig. 1. The distribution of PM2.5 data: a) hourly distribution, b) daily average

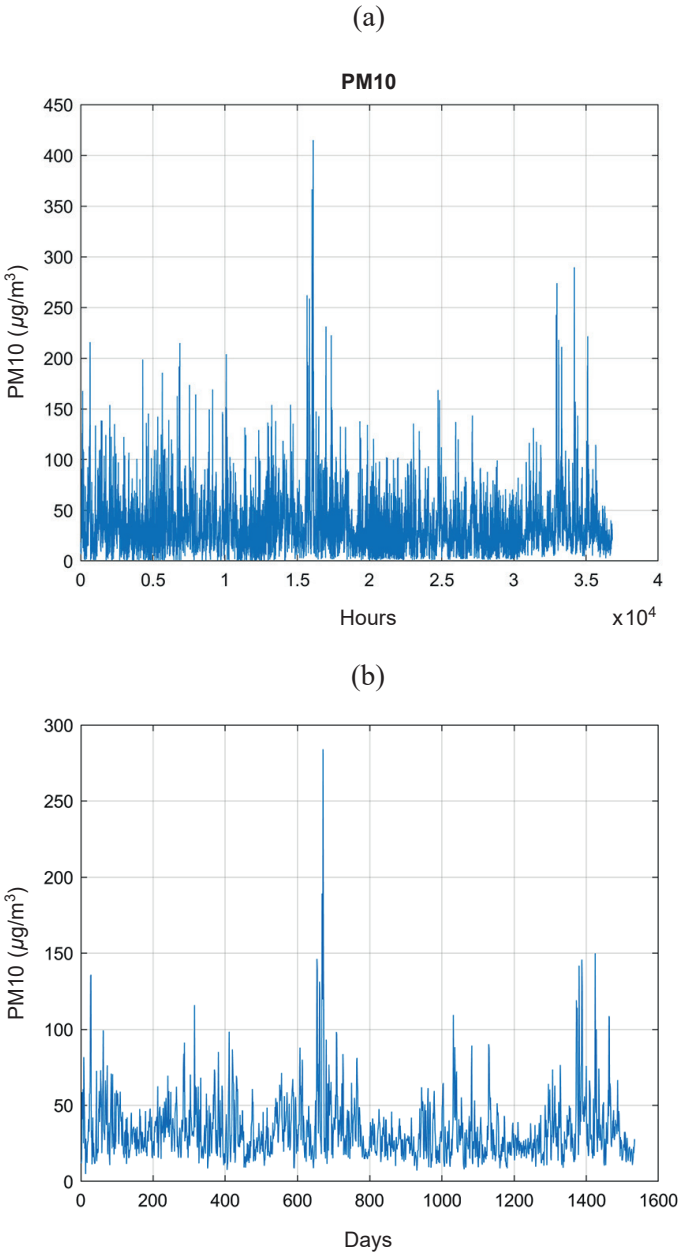

Fig. 2. The distribution of PM10 data: a) per hour, b) per day

type were taken from Chinese database representing pollution in Beijing [13] and the second type from Polish data base representing measurements made in Ursynów (Warsaw) [10]. The measurement results are given for the succeeding hours of the day. Figure 1 presents the hourly and daily average distributions of the data representing PM2.5, while Fig. 2 the distribution of PM10.

The significant differences between both types of data distribution are visible. They refer to the values, repeatability and histograms. High differences are observed in the distribution of the peak values within different seasons of the year. Figure 3 shows their dependence on the daily hours in the form of histogram of PM2.5 and PM10 in 4 seasons of the year.

The basic PM measurements have been associated with different sets of meteorological variables. In the case of PM2.5 they included: dew point, temperature, pressure, direction of wind, wind speed, cumulated hours of snow and cumulated hours of rain. In the case of PM10 the following meteorological variables were measured: wind speed, wind direction, temperature, humidity and sun radiation. Additionally, information about other pollution types was also available. They include SO2, NO2 and ozone.

The interesting thing is the correlation between the PM level and these additional variables. Table 1 presents values of Pearson correlation coefficient for both types of PM. Due to different availability of additional atmospheric variables

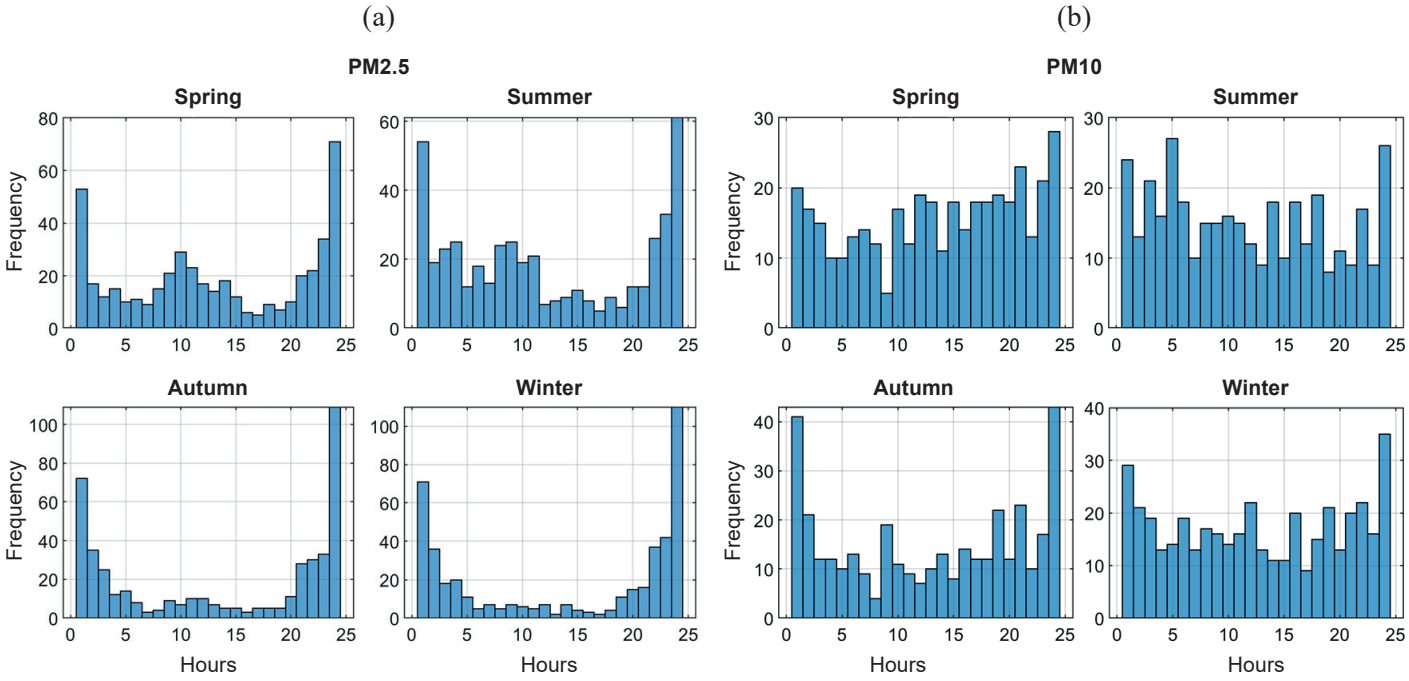

Fig. 3. The histogram of the peak hourly distribution of a) PM2.5, b) PM10 in different seasons of the year

the column contents for PM2.5 and PM10 are also different and some of them are empty due to the absence of particular measurements.

Table 1
Correlation between PM and additional variables available in measurements

Variable	PM2.5	PM10
Temperature	–0.091	–0.199
Wind speed	–0.248	–0.311
Wind direction	–0.168	–0.171
Humidity	–	0.034
Hours of rain	–0.051	–
Hours of snow	0.019	
Pressure	–0.047	–
Dew point	0.171	–
SO2	–	0.428
NO2	–	0.579
Ozone	–	–0.232

The correlation values presented in the Table 1 seem to be rather low. Remarkable is a relatively high correlation of PM concentration and other types of pollutants, especially compared to the atmospheric parameters. However, we should be aware that the correlation coefficient only contains information about linear relationships between variables and not the general nonlinear dependency. The preliminary experiments have shown that the inclusion of information on SO2, NO2 and ozone has improved the accuracy of the prediction. Taking this into account we have included them as the potential diagnostic features in the model of prediction process in PM10 prediction.

3. Selection of diagnostic features

The aim of the research is to build an efficient model of forecasting the next day average of PM based on some selected input attributes representing the diagnostic features of the process. In creation of potential features, we have considered many different aspects of the process. Analyzing autocorrelation functions, we have observed some dependence of the predicted PM level in day d on its values from the past. Three days from the past have been selected in this way. Therefore, the first set was composed of 24-hourly values of PM of the preceding days (d-1), (d-2) and (d-3). Additionally, we have included also daily average of these three days: PM(d-1), PM(d-2) and PM(d-3). The next set of descriptors was defined on the basis of mean values of all available meteorological parameters from the 3 previous days (d-1), (d-2) and (d-3). Notice, that these parameters differ for PM2.5 and PM10. Additional set of parameters referred to the maximum, minimum and their hours of appearance related to the meteorological parameters. This time their influence was limited to only the previous day (d-1). In the case of PM10 the additional set of descriptors was formed from available measurements of NO2, SO2 and ozone. Their state in only one previous day was characterized by the daily average, maximum and minimum hourly value within this day and their hours of occurrence.

To provide the information about human daily activity, we have included the code of the type of day. Working days were coded by one and non-working days by zero. Finally, the season information was also incorporated into the model. The binary code "00" was used for spring (months from 3 to 5), "01" for summer (months from 6 to 8), "10" for autumn (months from 9 to 11) and code "11" for winter (month 12, 1 and 2). As a result, the total number of potential features was equal to 115 for PM2.5 and to 151 for PM10.

Large number of automatically generated features suggests the need for their reduction. There are many selection methods [16, 17] which are used in practice. The introductory experiments have shown the stepwise fit as the most appropriate. It is a systematic algorithm which performs a series of adding or removing the variables to the set of features in order to find out which of them are necessary and which can be omitted based on their statistical significance in a regression. The F-statistic score is computed in each step to test the model with particular terms or without them and the null hypotheses is checked to decide which terms should be included or rejected in the final feature set [18]. This methodology has allowed to reduce the final diagnostic features in a significant way, leaving only the important ones.

As a result of stepwise fit procedure, 28 features were selected of PM2.5 dataset, while only 20 features were found important for PM10 dataset. The selected features have been normalized dividing the measured values by their median. We have applied median, because this statistical measure is resistant to outliers, which happen very often in measurements.

Both data sets have been collected within 4 years. Total number of available measurements exceeded 34,000 samples. Dataset records were shuffled and split to 70% for the training phase and 30% for the test. Each experiment has been performed on randomly selected set of learning and testing samples.

4. Individual predictors

The proposed bagging and boosting techniques form the ensemble composed of many units trained in a specific way, hoping to obtain better performance than could be obtained from any of the constituent member alone. In practice a machine learning ensemble consists of only a concrete finite set of alternative models that should be adjusted in a proper way.

Typically, the members of ensemble are weak predictors, which generate variety of results. The strength of ensemble is based on this variety. Application of very strong predictors in ensemble is not recommended since their verdicts will be repeatable and will not give the space for improvement. Therefore, in this work we have decided to use decision tree (DT), which is the typical weak predictor and multilayer perceptron (MLP) of many hidden layers (deep structure). Application of more than one hidden layer makes MLP alike weak predictor. We have tried also radial basis function network and support vector machine however, the results were not satisfactory. The training results of RBF and SVM in many different runs were very stable, so there was no chance for improvement by their combination.

The MLP [19] applied in the ensemble is composed of input layer of signals, 3 hidden layers of sigmoidal activation function and one linear output unit (deep structure). The final structure of MLP has been chosen after series of preliminary experiments, which tried different number of layers (1, 2, 3, 4) and different number of neurons in these layers. In the case of PM10 the best results have been obtained for MLP structure 20–32–16–16–1. In the case of PM2.5 the best MLP structure was 28–32–16–16–1. Levenberg-Marquard algorithm was used in learning procedure.

Decision tree (DT) was used as the second weak predictor [20]. Although DT is a typical classification tool it is easily adapted for regression problems by assigning classes to different ranges of signal values. This is done automatically based on special parameter *MinParentSize*, which controls the required learning accuracy.

Such approach is applied in Matlab implementation [18], which was used in this work. DT is a flowchart-like structure in which each internal node represents a "test" on an attribute, each branch the outcome of the test, and each leaf node a class label representing the range of output signal values. In our solution the decision tree was supplied by 20 (PM10) and 28 (PM2.5) input signals. Gini index was used in assessing the impurity index in every split of data.

In assessing the results, we have used different quality measures: mean absolute error (MAE), mean absolute percentage error (MAPE) and correlation coefficient R of the predicted and target series. In these definitions y_i represents the actual response of predictor, t_i – the target value and n number of samples, C_{yt} – the covariance of y_i and t_i, std – standard deviation of series of y and d.

- Mean absolute error (MAE)

$$MAE = \frac{1}{n}\left(\sum_{i=1}^{n}|t_i - y_i|\right) \tag{1}$$

- Mean absolute percentage error (MAPE)

$$MAPE = \frac{1}{n}\left(\sum_{i=1}^{n}\frac{|t_i - y_i|}{t_i}\right)\cdot 100\% \tag{2}$$

- Correlation coefficient (R) of the observed and predicted data

$$R = \frac{C_{yt}}{std(y)std(t)} \tag{3}$$

All numerical experiments have been performed with the Matlab environment. Figure 4 presents the results of MAE in learning and testing the MLP and DT achieved in 20 repetitions of experiments in the case of PM10.

Large differences between learning and testing results are visible, especially in the case of decision tree. This is typical for

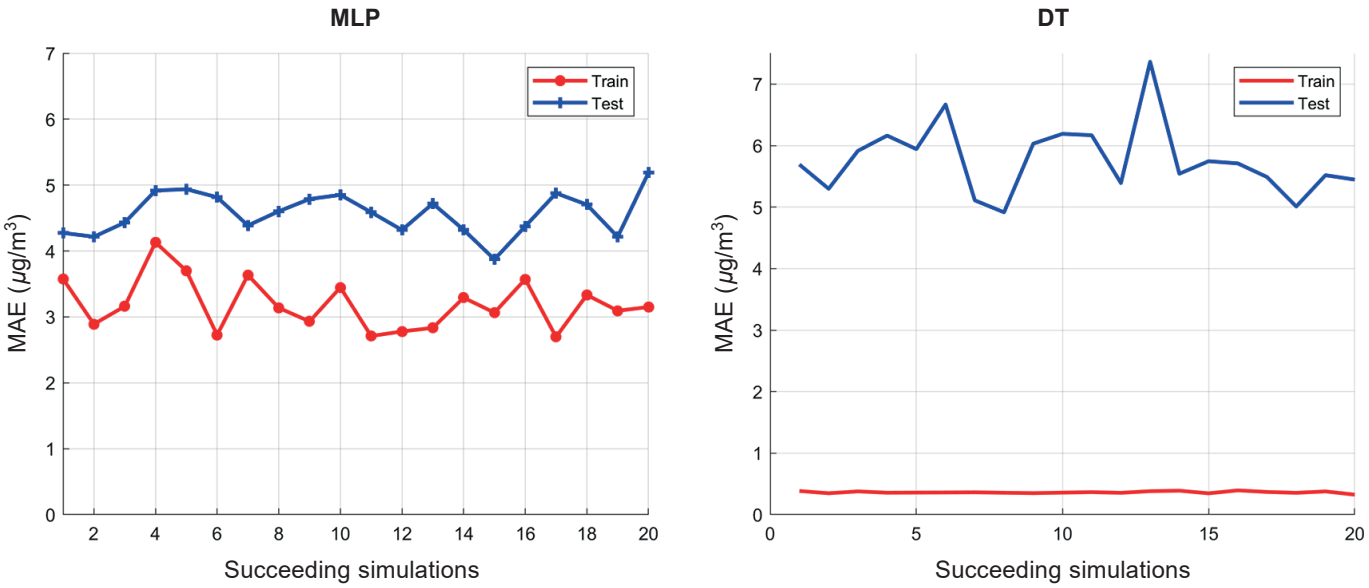

Fig. 4. The illustration of the change of MAE for PM10 prediction in learning and testing mode: a) MLP predictor, b) DT predictor

weak learners. This fact makes space for significant improvements of results by applying bagging or boosting procedures.

Table 2 depicts the average results of testing MLP and DT predictors in their individual mode of operation in 20 runs of experiments. The significant differences between the quality measures are observed for both types of pollution. MLP learner was more accurate for both PM10 and PM2.5. It is evident that the prediction model for PM10 is much more accurate with respect to all measures. It is partly since the input information is richer in this case (supported by accompanying levels of NO2, SO2 and ozone).

Table 2
The average results of testing MLP and DT, working in the mode of individual predictors

		MAE (μg/m^3)	MAPE (%)	R
MLP	PM10	4.57	14.84	0.94
	PM2.5	14.29	21.27	0.93
DT	PM10	5.76	18.22	0.94
	PM2.5	18.15	25.64	0.92

To check the influence of the feature selection procedure we have repeated experiments applying full set of generated descriptors (115 for PM2.5 and 151 for PM10). However, this time the results were much worse. In the case of MLP the MAPE was equal to 17.95% for PM10 and to 25.47% for PM2.5. Even worse results were in the case of DT (MAPE 23.21% for PM10 and 29.6% for PM2.5). This is a natural effect of oversizing the network and including input signals that are not representative of the process under consideration.

The next experiments have been directed to improve the prediction results of ensembles by applying bagging and boosting principles.

5. Bagging technique in prediction problems

Bootstrap aggregating, called simply bagging, is very popular technique used in ensemble of predictors [17, 20, 21]. It helps to increase the accuracy of prediction result and at the same time also reduces variance and allows to avoid overfitting. It is a method for generating multiple versions of predictors and using them to get an aggregated prediction values for unseen input data.

Let us assume the original training set D containing M learning pairs (**x**, **d**). In bagging approach, we apply many predictors trained by using smaller subset of m training samples $(m < M)$, randomly selected from the original set. The training samples are drawn from D uniformly and with replacement. By sampling with replacement, some observations may be repeated in the training sets. Then T predictors are trained using different sets of m bootstrap samples. Their results are combined by averaging (in regression) or majority voting (in classification).

Thanks to the aggregation process, bagging procedure reduces the variance of an individual base learner. However, bagging does not always lead to the improvement of the best individual learner, participating in the ensemble. It works especially well for weak learners, for example unstable, high variance base learners, for which we observe major changes in response to small changes in the training data. Such situation is typical for decision trees or some types of neural networks. However, for more stable predictors the bagging procedure offers less improvement on predicted outputs since there is less variability. Therefore, application of support vector machine is not recommended for this type of systems.

The important question is how deep (i.e. how complex) the predictors used in an ensemble should be. If the structure of predictor is too complex, there might be tendency to over fit the training data, which results in poor generalization performance. Consequently, there is a balance to be achieved in the complexity of the predictor structure (decision tree or neural

network) to optimize predictive performance on future unseen data. The introductory numerical experiments are needed to solve this dilemma.

6. AdaBoost for regression

Boosting technique applied in machine learning is another, very efficient way of obtaining good results of classification or regression by using many weak learner models forming an ensemble [22]. The fundamental assumption of boosting, is that it is possible to create strong and accurate model of the process by combining many weak models paying higher attention to less accurately predicted samples.The most important implementation of this approach is adaptive boosting, called shortly AdaBoost. It was proposed by Freund and Schapire [21]. AdaBoost applies many weak learners of relatively low accuracy to make their integrated results stronger. The learning algorithm starts from full original set of training data. Then creates a second model, which tries to correct the errors committed by the first model.

The succeeding models are added to ensemble until the training set is predicted perfectly or a maximum number of models have been reached. In the succeeding iteration of the training process, the weight is assigned to each training sample. The values of these weights inform the weak learners to prefer the samples with high values when selecting the next learning setAny learner network can be used in solution, for example neural networks or decision trees. At the beginning, the weights associated with the learning samples are set equal. Therefore, the probability of selecting each sample is the same. The chosen prediction model is trained by using the set of training samples and the errors committed at particular samples are calculated and stored to be used in the next bootstrap selection. The algorithm prefers the samples for which the model is unable to perform prediction task accurately. These samples are more likely to be selected in the next stage of learning. The method used in adjustment of weights might use different mathematical formulas, which are called loss functions. Once the loss function is calculated for each training vector, it is possible to associate the weight with this vector. The larger the loss, the weight is more increased.

The basic AdaBoost algorithm was defined for classification task. We will use it in a slightly different arrangement for regression [22]. The applied AdaBoost in regression can be presented as follows:

1) Associate each learning pair $(\mathbf{x}_i, \mathbf{d}_i)$ for $i = 1, 2, ..., M$, where $\mathbf{x}_i$, represents the input vector and $\mathbf{d}_i$ the destination, with the weight w_i. Initially, to each training pair we assign weight $w_i = 1$ for $i = 1, 2, ..., M$. The probability that ith sample will be selected in the bootstrap training set is then

$$p_i = \frac{w_i}{\sum_{j=1}^{M} w_j} \tag{4}$$

where the summation is over all members of the training set. Select N_1 samples (with replacement) to create the training set. Usually N_1 is smaller than M, the total number of samples.

2) Train the machine using the selected set of training samples.
3) Pass every member of the data set through this machine to obtain a prediction $y(\mathbf{x}_i)$ for all $i = 1, 2, ..., M$.
4) Calculate the loss for each training sample $L_i = L\big(y(\mathbf{x}_i) - d_i\big)$, where the loss function L may take any form, providing its value in the range [0, 1]. Let us assume the normalization factor in the form $D = \sup\big(y(\boldsymbol{x}_i) - d_i\big)$ then different forms of the loss function can be associated with each pair, for example
   - linear function

$$L_i = \frac{|y(\mathbf{x}_i) - d_i|}{D} \tag{5}$$

   - exponential function

$$L_i = 1 - \mathrm{e}^{-\frac{|y(\mathbf{x}_i) - d_i|}{D}} \tag{6}$$

   Then calculate the average loss of the whole set

$$L_m = \sum_{i=1}^{M} p_i L_i \tag{7}$$

5) Calculate confidence measure c of the actual predictor

$$c = \frac{L_m}{1 - L_m} \tag{8}$$

   Small value of c means high confidence of this predictor in processing the learning data and high value small confidence.
6) Update the weights associated with the individual ith training sample

$$w_{i+1} = w_i c^{\frac{(1 - L_i)}{Z}} \tag{9}$$

   The normalization factor Z is needed to ensure the sum of all instance weights is equal to one. Its value was defined here as the sum of all weights. The lower the loss of a particular learning pair, the higher the reduction in weight value. It makes less likely that this pattern will be chosen as a member of the training set in the next prediction phase.
7) Repeat T times the training processes on new bootstrap of selected data sets and keep all trained predictor parameters in memory.
8) Each of T trained predictors makes individual prediction $y(\mathbf{x}_t)$ for a particular testing vector $\mathbf{x}_t$. Final prediction of the whole ensemble made by T units is defined as the weighted median of all predictions (considering their respective confidence levels associated with each predictor). It means that value $y_i(\mathbf{x}_t)$ generated by ith predictor is associated with its confidence measure c_i. Next, they are relabeled in a way: $c_1 y_1(\mathbf{x}_t) < c_2 y_2(\mathbf{x}_t) < \ldots < c_T y_T(\mathbf{x}_t)$. The middle element in this series is the final response $y(\mathbf{x}_t)$ of the ensemble. In the case, when all confidence measures c_i are equal, the result of ensemble is ordinary median of the responses of all members of ensemble.

7. Results of numerical experiments

7.1. Application of bagging. Bagging algorithm may include as many predictors as necessary to outperform the single model. It is composed of two main procedures: bootstrap (selecting the learning samples) and aggregation (generating final verdict of ensemble) [17]. Bootstrap is a random selection of subsets of samples with replacement of vectors from the training data. However, the bags should contain fewer samples than the whole training dataset to have diversity between learners. The correct choice of bag size is very important. If the number of samples in the bag is too small, the real distribution of the data is not well represented. On the other hand, too large a size results into a higher degree of uniformity of the data in certain bags, which leads to a loss of independence between the members of the ensemble.

It is necessary to carry out some experiments that provide a suitable number of predictors. This was done by increasing the size of bags and learners iteratively. Once a new bag is defined, the learner is trained with such subset and the result is added to the current predictions, hence the next forecast will be the average of all past predictions made before.

Figure 5 shows the graphical results of such experiments at changing size of the bag and at different number of predictors. Figure 5a corresponds to MAE for PM2.5 and Fig. 5b for PM10. All of them represent testing results. The dashed line represents the result of the best individual MLP predictor at application of the whole data set. In both cases the advantage of bagging procedure is evident. However, its results depend on the size of bags. The optimal size of bags corresponds here to application of 80% of available data in learning process and the number of predictors over 100.

Similar results are also characteristic for DT. Figure 6 shows the appropriate testing results for this predictor. This time the profit from bagging application is even more visible with

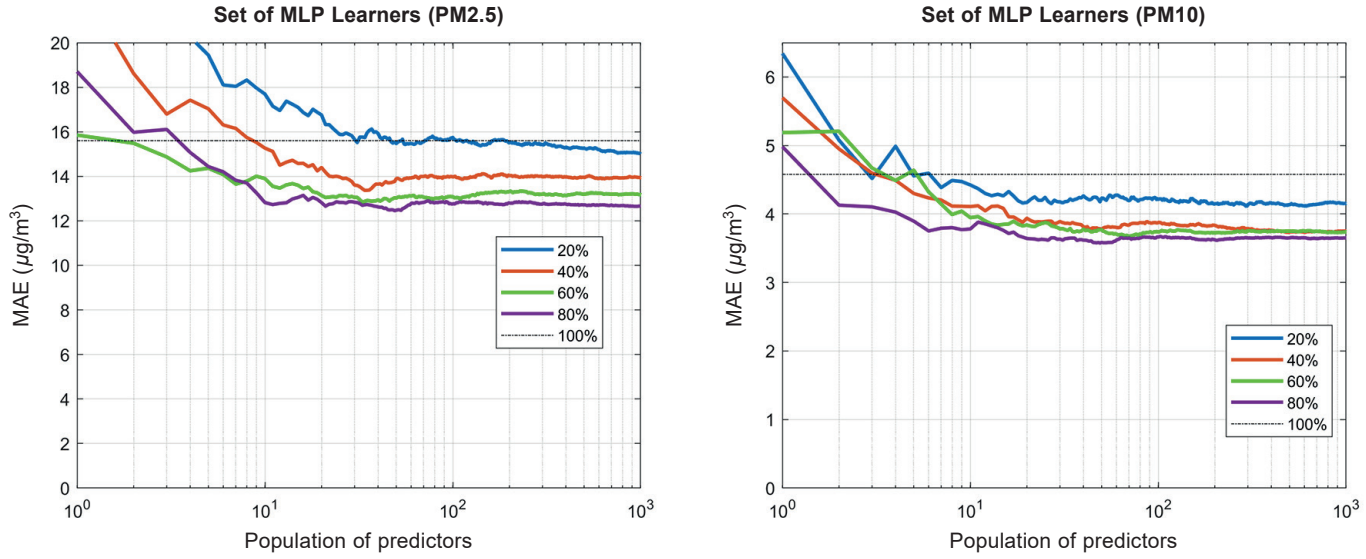

Fig. 5. The MAE changes with the number of MLP learners at different sizes of the bags

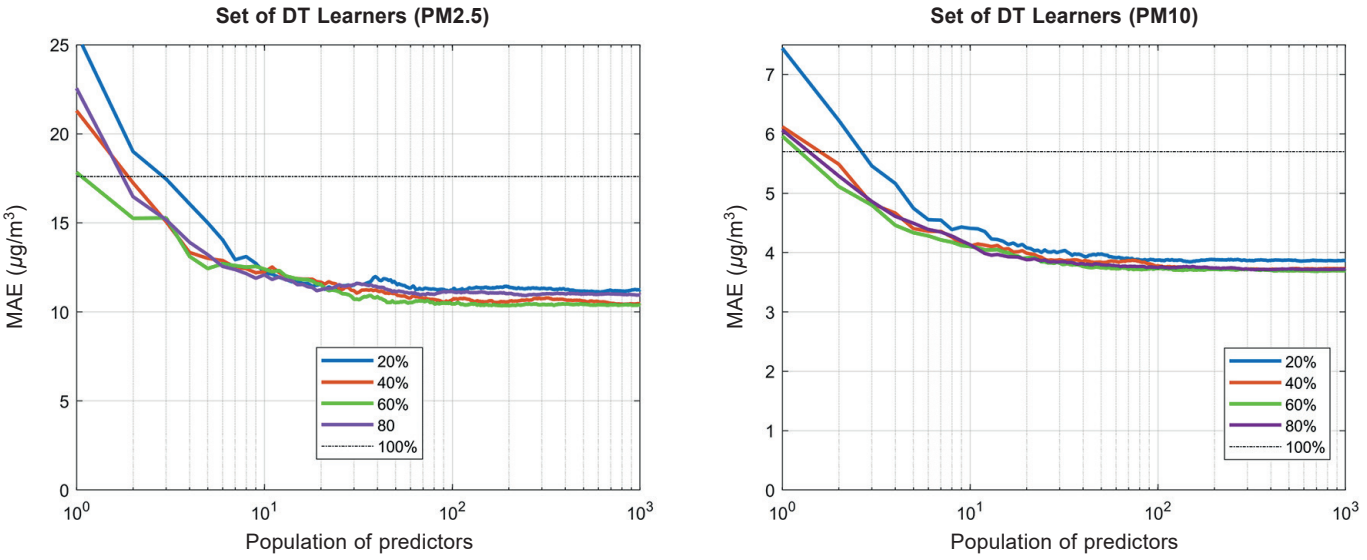

Fig. 6. The MAE changes with the number of DT learners at different sizes of the bags

respect to the best DT predictor trained on the whole data set. The best results in this case correspond to application of 60% of available learning data in the bags. Similarly to the previous case, the number of predictors in ensemble should be over 100.

Twenty repetitions of experiments performed at the optimal size of bags and application of 110 predictors in the ensemble have led to the results presented in Table 3.

Table 3
The average results of testing MLP and DT at application of bagging ensemble

		MAE $(\mu g/m^3)$	MAPE (%)	R
MLP	PM10	3.31	11.13	0.98
	PM2.5	11.47	15.26	0.98
DT	PM10	3.67	11.55	0.98
	PM2.5	12.21	14.77	0.97

These quality factors (MAE, MAPE and correlation coefficient R) outperform the best individual application of both predictor types.

7.2. Application of AdaBoost. The first decision in AdaBoost is to select proper form of loss function. Loss function represents the measure based on which we select the population of data set in the next generation of algorithm. The preliminary experiments have shown that exponential loss is not a very good choice. In most cases its application resulted in higher regression error than linear one. Therefore, in the experiments we have applied linear loss function.

Similarly to the bagging procedure, the first task is to find the optimal number of predictors and optimal size of bags used in experiments. Once again, the experiments have been repeated at changing values of the bag size and different population of weak learners. Contrary to bagging, it happened that increasing the predictors beyond some value did not lead to the improvement of results. Fig. 7 depicts such situation of MLP learner in the case of PM2.5.

In this case around 50 learners seem to be optimal with respect to MAE. Once again, the best size of the bag contained 80% of available learning data. The performed experiments have shown that in the case of DT the optimal number of predictors was 90 at the bag size equal to 60% of the available learning data set. These sets of parameters have been applied in final experiments of predicting PM10 and PM2.5.

Twenty repetitions of experiments performed at the optimal size of bags and by applying the optimal number of predictors in the ensemble have led to the results presented in Table 4.

Table 4
The average results of testing MLP and DT at application of AdaBoost

		MAE $(\mu g/m^3)$	MAPE (%)	R
MLP	PM10	3.82	12.34	0.98
	PM2.5	12.03	17.05	0.98
DT	PM10	4.18	12.25	0.98
	PM2.5	13.19	16.90	0.96

As it is seen application of AdaBoost has improved the results of individual best predictors, however, its results are slightly worse than these obtained in bagging. The important advantage of AdaBoost is high repeatability of results. Figure 8 shows the MAE results for PM10 obtained in 20 repetitions of experiments with MLP in the role of predictor. Standard deviation of results was equal to std = 0.1119 at average value of MAE equal to 3.82.

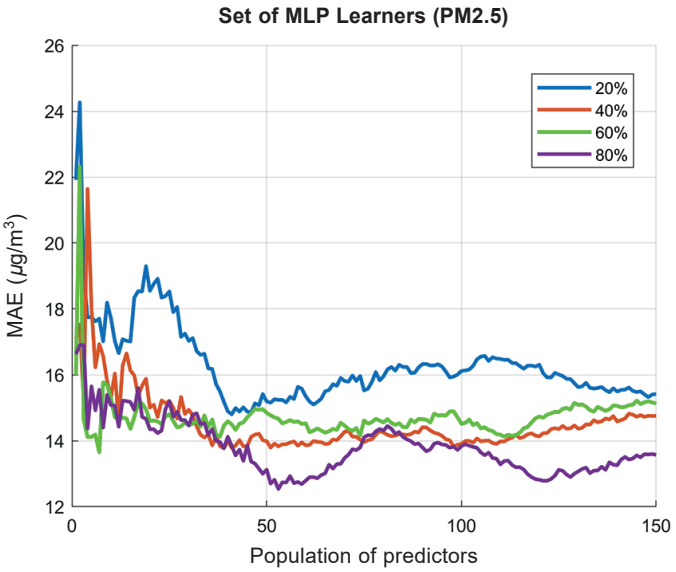

Fig. 7. The change of MAE for PM2.5 at application of MLP learners in boosting application

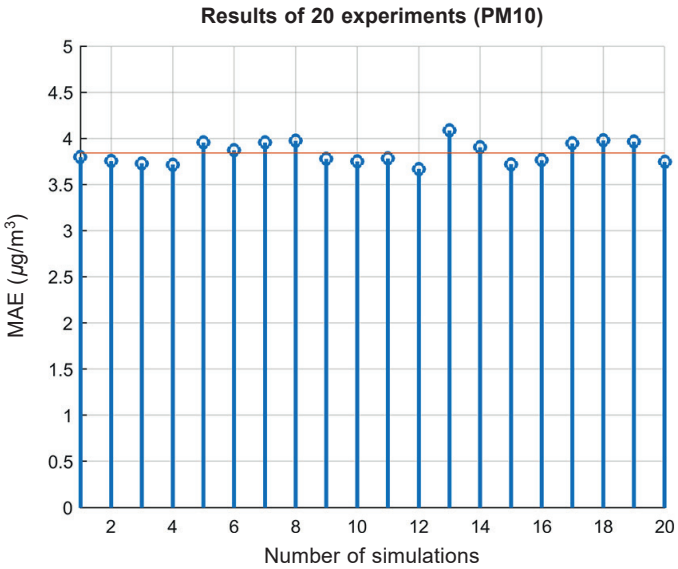

Fig. 8. The change of MAE results in 20 repetitions of AdaBoost experiments at application of MLP predictor for PM10

8. Conclusions

The paper has studied the application of bagging and boosting ensembles to predict daily average of particulate matters (PM10 and PM2.5) for the next day. The experiments have proved that such form of ensemble creation provides high repeatability and better performance in comparison to either individual predictors or typical ensemble integrated by averaging. The numerical results related to the quality measures have shown very similar performance of both investigated methods, although, slightly better results have been obtained in bagging. On the other side, the AdaBoost ensemble is more stable and, in most cases, requires smaller number of predictors to achieve an improvement of quality measures. The experiments have shown that deep multilayer perceptron and decision tree represent good candidates for the implementation in bagging and boosting ensembles.

The complexity of the calculations should also be commented on. To obtain the improved results, we must use a high number of ensemble members (usually more than one hundred). This increases the cost of the calculation. Therefore, this technique is not suitable for online learning. Note, however, that the typical learning procedure is off-line. Therefore, the calculation time is not critical. On the other hand, the test mode is very fast and practically does not depend on the number of regressors in the ensemble.

Compared with previous studies, our model provided more accurate predictions of the daily level of pollution, including all quality measures. This is well seen on the example of PM10. The results presented in the paper [12] for the same data base by using dynamic integration of ensemble have reported following values MAE = 5.79 $\mu g/m^3$, MAPE = 18.62% and R = 0.935. The corresponding results obtained by applying complex ensemble [10] using many different predictors and selection methods were as follows: MAE 5.31 $\mu g/m^3$, MAPE=17.83% and R = 0.924. The best results of this paper for the same data are MAE = 3.31 $\mu g/m^3$, MAPE = 11.13% and R = 0.980.

It is more difficult to compare our results for PM2.5 with those presented for the same database in [13], since this work only presented the mean square root error (RMSE) of the succeeding months.

The results presented in this paper were not stable and have been changing in the range [4.55–41] $\mu g/m^3$ at application of non-parametric regression model and in the range [2.92–26.47] $\mu g/m^3$ applying partial linear regression model of prediction. No results regarding MAE or MAPE have been presented. Our best result of RMSE obtained for the whole data base was 18.45/ $\mu g/m^3$ at application of bagging ensemble built based on MLP.

Future work will include a larger exploration of this topic by covering wider horizon of data used in prediction process and application of larger population of predictors cooperating with each other.

References

[1] M. Martuzzi, F. Mitis, I. Iavarone, and M. Serinelli, "Health impact of PM10 and ozone in 13 Italian cities", *WHP report* (2005).

[2] https://www.epa.gov/criteria-air-pollutants, *National Ambient Air Quality Standards (NAAQS*, *Air criteria Air Pollutants*, US EPA. US Environmental Protection Agency, last updated on March 8 (2018).

[3] N. Saliba, R. Massoud, F. Zereini, and C. Wiseman, *Urban airborne particulate matter: origin*, *chemistry*, *fate and health impacts*, *environmental science and engineering*, Springer-Verlag Berlin Heidelberg (2011).

[4] J. Cao, J. Chow, F. Lee, and J. Watson, "Evolution of PM2.5 measurements and standards in the USA and future perspectives for China", *Aerosol Air Qual. Res.* 13, 1197–1211 (2013).

[5] H Taheri Shahraiyni, and S Sodoudi, "Statistical modeling approaches for PM10 prediction in urban areas; A review of 21st-century studies", *Atmosphere* 7, 1–24, doi:10.3390/atmos7020015 (2016).

[6] G. Gennaro, L. Trizio, A. Gilio, 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.* 463–464, 875–883 (2013).

[7] J. Anderson, J. Thundiyil, and A. Stolbach, "Clearing the Air: A review of the effects of particulate matter air pollution on human health", *American College of Medical Toxicology* (2011).

[8] M. Oprea, M. Popescu, E. Dragomir, and S. Mihalache, "Models of particulate matter concentration forecasting based on artificial neural networks", in *The 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing*, *Technology and Applications*, Bucharest, Romania (2017).

[9] R. Chandra and M. Zhang, "Cooperative coevolution of Elman recurrent neural networks for chaotic time series prediction", *Neurocomputing* 86, 116–123 (2012).

[10] K. Siwek and S. Osowski, "Data mining methods for prediction of air pollution", *Int. J. Appl. Math. Comput. Sci.* 26(2), 467–478 (2016).

[11] C. Carnevale, E. De Angelis, G. Finzi, E. Turrini, and M. Volta, "An integrated forecasting system for air quality control", in *18th European Control Conference* (*ECC*), Napoli, Italy, 25–28 (2019).

[12] S. Osowski, and K. Siwek, "Local dynamic integration of ensemble in prediction of time series", *Bull. Pol. Ac.: Tech.* 67(3), 517–525 (2019).

[13] X. Liang, T. Zou, B. Guo, S. Li, H. Zhang, S. Zhang, H. Huang, and S. X. Chen. "Assessing Beijing's PM2.5 pollution: severity, weather impact, APEC and winter heating", *Proc. R. Soc. A-Math. Phys. Eng. Sci.* 471, 20150257 (2015), https://doi.org/10.1098/rspa.2015.0257.

[14] L. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*, Wiley, New York (2004).

[15] Zhi-Hua Zhou, *Ensemble Methods Foundations and Algorithms*, Chapman and Hall, London (2012).

[16] I. Guyon., and A. Elisseeff, "An introduction to variable and feature selection", *J. Mach. Learn. Res.* 3, 1157–1182 (2003).

[17] P.N. Tan, M. Steinbach, and V. Kumar, *Introduction to data mining*, Pearson Education Inc., Boston (2006).

[18] *Matlab user manual*, *MathWorks*, Natick, USA, (2016).

[19] S. Haykin, *Neural networks*, *a comprehensive foundation*, Macmillan College Publishing Company, New York (2000).

[20] L. Breiman, "Random forests", *Mach. Learn.* 45(11), 5–32 (2001).

[21] R. Schapire and Y. Freund, *Boosting: Foundations and Algorithms*, The MIT Press (2012).

[22] H. Drucker, "Improving regressors using boosting techniques", in *Proceedings of the 14 International Conference on Machine Learning*, 107–115, Morgan Kaufmann Publishers, San Francisco (1997).