The Comparison of Machine Learning Algorithms for the Task of Weather and Air Pollution Forecasting

Anatoliy Doroshenko[1,2,†], Dmitry Zhora[1,†], Pavlo Ivanenko[1,†] and Olena Yatsenko[1,*,†]

[1] *Institute of Software Systems of the National Academy of Sciences of Ukraine, Glushkov Ave. 40, Kyiv, 03187, Ukraine*
[2] *National Technical University of Ukraine "Igor Sikorsky Kiev Polytechnic Institute", Peremohy Ave, 37, Kyiv, 03056, Ukraine*

Abstract

The task of weather forecasting becomes more important under conditions of global warming. Similarly, the air pollution prediction has higher value when industrial enterprises neglect environmental pollution issues. This research demonstrates how hourly weather and air pollution data can be restructured for the forecasting up to 24 hours ahead, and studies the cross-influence of parameters as all of them represent the atmosphere as single object from physical world. The parameter differences calculated for different points in time are considered as additional inputs and outputs of machine learning model. The prediction accuracy is analyzed for twelve regression algorithms using popular metrics like MASE, R2 and MAE.

Keywords

machine learning, regression algorithms, weather forecasting, air pollution forecasting

1. Introduction

In recent years, the application of machine learning algorithms has revolutionized also in the field of weather and air pollution forecasting. This article provides a comparative analysis of various machine learning techniques, including ensemble methods and neural networks, to evaluate their effectiveness in predicting meteorological and air quality conditions. By examining the accuracy metrics obtained for each algorithm, this study aims to identify the most reliable configurations, ultimately contributing to better environmental and public health strategies.

2. Weather and Air Pollution Data

The weather and air pollution data were downloaded from the website openweathermap.org. This service allows to retrieve multiple atmospheric characteristics for arbitrary GPS coordinates. The main columns of this dataset for Kyiv city are shown below in Fig. 1. This table contains hourly data and 33,863 records overall, from Nov 25, 2020 to Oct 05, 2024.

Workshop "Intelligent information technologies" UkrProg-IIT`2025 co-located with 15th International Scientific and Practical Programming Conference UkrPROG'2025, May 13-14, 2025, Kyiv, Ukraine

* Corresponding author.
† These authors contributed equally.
doroshenkoanatoliy2@gmail.com (A. Doroshenko); dmitry.zhora@gmx.com (D. Zhora); paiv@ukr.net (P. Ivanenko); oayat@ukr.net (O. Yatsenko)
0000-0002-8435-1451 (A. Doroshenko); 0009-0006-6073-7751 (D. Zhora); 0000-0001-5437-9763 (P. Ivanenko); 0000-0002-4700-6704 (O. Yatsenko)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	UtcTime	LocalDate	LocalHour	Temperatu	DewPoint	Pressure	Humidity	WindSpee	WindSine	WindCosi	WindAngle	CloudLevel	LevelCO	LevelNO	LevelNO2	LevelO3	LevelSO2	LevelNH3	LevelPM2	LevelPM10
2	1606266000	2020-11-25	3	3.29	1.01	1022	85	4.38	-0.9903	0.1392	278	99	223.64	0	4.24	49.35	3.46	0.46	3.18	5.08
3	1606269600	2020-11-25	4	3.24	1.13	1021	86	4.23	-0.9945	0.1045	276	100	226.97	0	4.33	48.64	3.73	0.47	3.1	4.89
4	1606273200	2020-11-25	5	3.43	1.79	1021	89	4.09	-0.9962	0.0872	275	100	230.31	0	4.8	46.49	4.05	0.45	3.72	5.5
5	1606276800	2020-11-25	6	3.56	1.76	1021	88	0.45	-0.6561	-0.7547	221	100	233.65	0.01	5.83	43.63	4.59	0.48	4.49	6.33
6	1606280400	2020-11-25	7	3.76	1.96	1021	88	4.1	-0.9976	0.0698	274	100	243.66	0.02	8.82	39.34	5.25	0.55	5.24	7.22
7	1606284000	2020-11-25	8	3.71	1.75	1021	87	4.44	-0.9976	0.0698	274	100	250.34	0.04	11.48	36.48	5.84	0.59	5.97	8.04
8	1606287600	2020-11-25	9	3.77	1.81	1021	87	4.78	-0.9976	0.0698	274	100	253.68	0.08	11.82	36.84	6.2	0.59	6.7	8.66
33860	1728154800	2024-10-05	22	15.27	14.96	1012	98	0.45	-0.4226	0.9063	335	100	236.99	0	8.4	43.99	3.22	0.46	7.43	8.24
33861	1728158400	2024-10-05	23	15.07	14.76	1011	98	0.45	-0.1219	0.9925	353	100	230.31	0	6.6	45.42	3.01	0.4	8.15	8.8
33862	1728162000	2024-10-06	0	15.07	14.76	1011	98	0.45	-0.6691	0.7431	318	100	226.97	0	3.98	48.64	2.86	0.34	8.6	9.07
33863	1728165600	2024-10-06	1	15.17	14.86	1011	98	4.25	0.848	0.5299	58	100	223.64	0	2.72	50.07	2.86	0.32	8.63	8.96
33864	1728169200	2024-10-06	2	15.08	14.77	1010	98	0.45	0.6157	0.788	38	100	223.64	0	2.4	50.78	2.92	0.33	8.52	8.82

Figure 1: The Kyiv city dataset used for training and validation of regression models.

Table 1: The weather and pollution parameters representing the data model.

Model Parameter	Parameter Type	Description and Measurement Unit
UtcTime	Primary key	Number of seconds elapsed since 1970-01-01T00:00:00 GMT
LocalDate	Composite key	Local date of measurement (Kyiv)
LocalHour	Composite key	Local hour from 0 to 23 (Kyiv)
Temperature	Measured	Air temperature in degrees Celsius
DewPoint	Measured	Dew point in degrees Celsius
Pressure	Measured	Atmospheric pressure in millibars
Humidity	Measured	Air humidity as percentage
WindSpeed	Measured	Wind speed in meters per second
WindAngle	Measured	Wind direction azimuth in degrees
WindSine	Calculated	Sine of wind direction angle
WindCosine	Calculated	Cosine of wind direction angle
CloudLevel	Measured	Sky cloudiness as percentage
LevelCO	Measured	CO pollution level in μg/m³
LevelNO	Measured	NO pollution level in μg/m³
LevelNO2	Measured	NO_2 pollution level in μg/m³
LevelO3	Measured	O_3 pollution level in μg/m³
LevelSO2	Measured	SO_2 pollution level in μg/m³
LevelNH3	Measured	NH_3 pollution level in μg/m³
LevelPM2	Measured	Dust pollution with particles less than 2.5 micrometers in μg/m³
LevelPM10	Measured	Dust pollution with particles less than 10 micrometers in μg/m³
SineDay	Calculated	Sine value for daily cycle
CosineDay	Calculated	Cosine value for daily cycle
SineWeek	Calculated	Sine value for weekly cycle
CosineWeek	Calculated	Cosine value for weekly cycle
SineMonth	Calculated	Sine value for monthly cycle
CosineMonth	Calculated	Cosine value for monthly cycle
SineYear	Calculated	Sine value for yearly cycle
CosineYear	Calculated	Cosine value for yearly cycle

Despite this work accounts only for data from one city, the first UTC time column in Table 1 above is helpful to synchronize records from multiple locations. Correspondingly, the local date and time columns are important for customers. The air temperature and dew point are presented in degrees Celsius. The atmospheric pressure is measured in millibars (or hectopascals). The humidity and cloudiness are both represented as percentages.

The next subset of weather-related parameters are wind characteristics. The degrees are used typically to register wind direction. However, this format is not convenient for machine learning algorithms [1] due to the representation gap between 359° and 0°. One of the popular approaches for solving this problem is the usage of the sine and cosine of the corresponding angle [5]. These columns were calculated using an algorithm written in Python. The reverse transformation is also possible when forecasted values of wind sine and cosine are properly normalized. The wind speed is measured correspondingly in meters per second.

The air pollution levels for various indicators shown in Table 1 are measured in micrograms per cubic meter (μg/m³). Carbon monoxide stands out as the most significant pollutant due to its high concentration. The parameters LevelPM2 and LevelPM10 denote dust pollution with particles up to 2.5 and 10 micrometers, respectively. It's important to note that the PM10 value includes the PM2.5 level. The particles that are 2.5 micrometers or smaller are particularly harmful as they can directly enter the bloodstream. Mid-sized particles can easily pass through the airways and settle in the lungs. Lastly, particles larger than 10 micrometers are typically filtered out by the respiratory tract and do not reach the lungs.

The accuracy of the forecast can be enhanced by incorporating cyclical parameters [7], that are presented in the lower section of Table 1. For instance, the cosine of daily cycle represents the temperature and light variations between day and night. Likewise, the cosine of the yearly cycle captures the changes between winter and summer.

3. Data Imputation and Resampling

The weather dataset included all necessary records for the specified period. At the same time, the pollution data lacked 275 records and contained several negative and outlier values, which were removed. The missing entries were subsequently recalculated using the KNNImputer class [8].

The machine learning algorithms in the scikit-learn library [9] require that all input and output parameters be represented in separate columns. However, this structure is not ideal for time series forecasting, where past and future data vary by record number and occupy the same columns. So, the dataset was restructured for training and forecasting purposes, with additional weather and pollution parameters included. The suffix notation used is detailed in the example below.

- Temperature-P1, the temperature in 1 hour
  …
- Temperature-P24, the temperature in 24 hours
- Temperature-M1, the temperature 1 hour ago
  …
- Temperature-M24, the temperature 24 hours ago

Similarly, the dataset was augmented with parameter differences, as described in the list below. Strictly speaking this information is redundant, but the layout of samples in the multi-dimensional space can be different in relation to internal computations of regression algorithm [16].

- Temperature-Diff-P1 = Temperature-P1 - Temperature
  …
- Temperature-Diff-P24 = Temperature-P24 - Temperature
- Temperature-Diff-M1 = Temperature - Temperature-M1
  …
- Temperature-Diff-M24 = Temperature - Temperature-M24

In time series slang, the two groups of parameters above are often referred to as lags and diffs. The periodic parameters do not need to be duplicated, as they precisely represent the moment in time for machine learning purposes. The dataset was divided into training and testing segments in an 80% to 20% ratio. All training data precede the testing records chronologically, with the split date being December 28, 2023.

In total, there are 8 weather parameters and 8 pollution parameters available for current hour. In particular, the feature WindAngle was excluded due to its discontinuous nature. If the past and future hours are considered then differences can be added. So, overall 16 weather and 16 pollution parameters can be used as inputs and outputs of a machine learning algorithm. When the whole 24-hour history is taken into account and periodic parameters are added the total number of inputs becomes 8 + 8 + (16 + 16) * 24 + 8 = 792. Thus, the total number of possible input combinations is 2^{792}. Clearly, this work does not attempt to explore this combinatorial space and aims to use more affordable approaches to optimize the forecasting accuracy.

4. Regression Performance Metrics

The mean absolute scaled error (MASE) is regarded as a superior alternative to the mean absolute percentage error (MAPE). A major drawback of the MAPE metric is that it can produce excessively large values when the dataset includes samples that are near zero. A classic example of this issue is temperature measured in degrees Celsius.

The main idea behind MASE metric is to compare the performance of a regression algorithm to naïve forecast approach when the current value of time series is used as a forecast for next step. This is also called as null hypothesis in the terminology of capital markets. So, here's the formula that implements this approach.

$$MASE = \frac{\frac{1}{n}\sum_{i=1}^{n}|y_i - f_i|}{\frac{1}{n-k}\sum_{i=k}^{n}|y_i - y_{i-k}|} \quad (1)$$

Here n designates the number of records in the test set, k – the number of steps the forecast is made for, y_i – the actual component output value from the test set, f_i – the predicted component output value. The numerator represents mean absolute error, and denominator represents the error of naïve forecast. As can be concluded from the formula, the MASE metric is higher than or equal to 0. The lower its value the more accurate predictions were made. The forecast can be considered as successful when MASE metric is lower than 1. Correspondingly, when MASE value is higher than 1 the forecast cannot be considered as useful, and regression algorithm performs even worse than naïve method. The algorithm that calculates MASE metric is presented in Appendix A.

Another popular metric for regression tasks is R^2 score, also called as determination coefficient. It has some similarities with a correlation coefficient in the interpretation aspects. Nevertheless, the calculation formula is different.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - f_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \quad (2)$$

Here $\bar{y}$ designates the mean value for actual component output from the test set. The higher the value of R^2 score the better, its maximum possible value is 1 for precise forecast. If R^2 score is higher than 0 the prediction can be considered as successful. If it is lower than 0 than forecast is rather harmful and its results better be avoided.

The mean absolute error (MAE) is the simplest metric. It is convenient for field engineers as its values are represented in corresponding measurement units, so that it is easy to verify if the error matches the real-world constraints. The calculation formula for MAE error is presented below.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - f_i| \tag{3}$$

As demonstrated in Table 1, up to 16 parameters can be selected as the outputs of a regression algorithm. Meanwhile, this research does not attempt to address the multi-objective optimization problem. All parameters of the machine learning algorithm are optimized solely to minimize the sum of MASE metrics for individual output parameters.

5. Prediction of Combined Outputs

The evaluation of input features was accomplished with ExtraTreesRegressor algorithm [12] from scikit-learn library [9]. It has limited number of hyperparameters to tune and provides the array of feature importances that enable individual feature selection.

The starting point of this research is to employ a single machine learning model that forecasts all 16 output parameters. The users are typically interested in all forecast ranges from 1 hour and up to 24 hours ahead. In order to reduce the computational burden and balance the quality of short-term and long-term forecasting it was desided to tune the model initially for 12-hour forecasting.

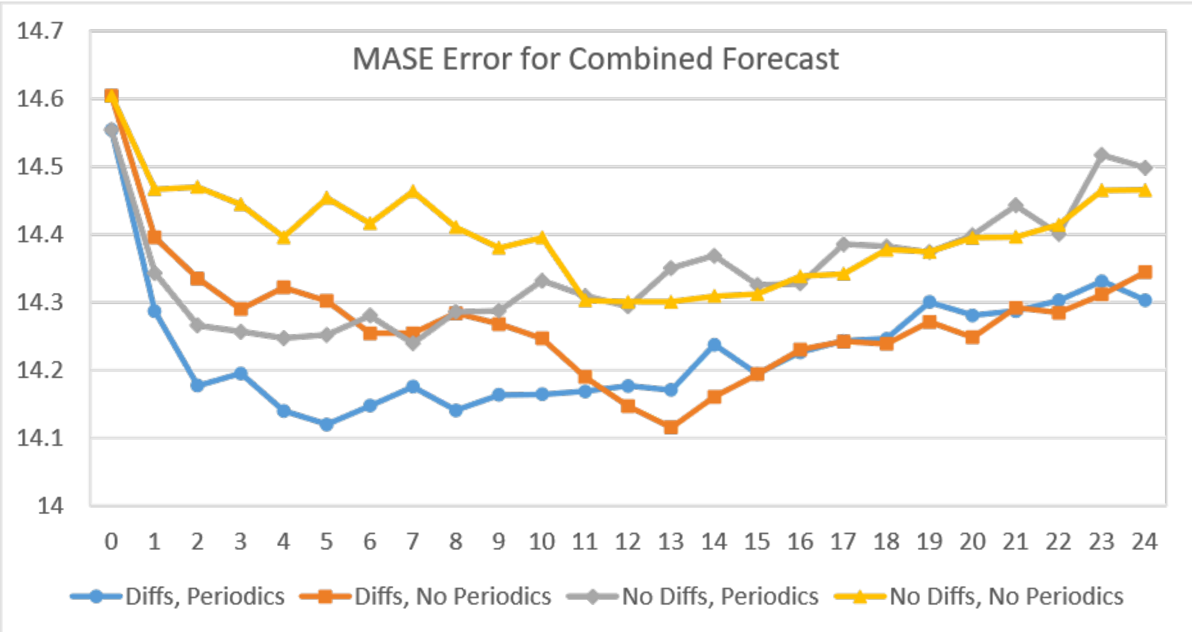

Figure 2: The sum of MASE errors for combined forecast depending on history length in hours.

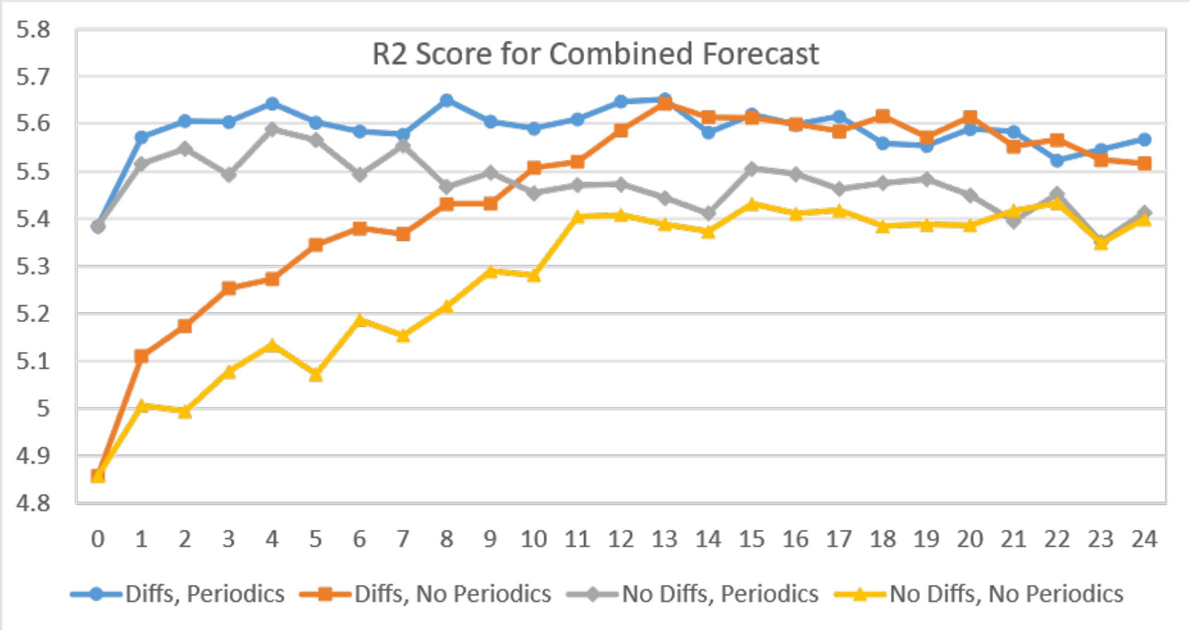

Figure 3: The sum of R2 scores for combined forecast depending on history length in hours.

The MASE metric dependencies on the history length in hours are illustrated in Figure 2. It is evident that difference inputs noticeably improve the quality of prediction. Additionally, periodic parameters are quite important for shorter history. Nevetheless, the best results were achieved with a 13-hour history and without periodic parameters. Below are the lists representing input-output configuration for this scenario (400 inputs vs 16 outputs).

Input features: `['Temperature', 'DewPoint', 'Pressure', 'Humidity', 'WindSpeed', 'WindSine', 'WindCosine', 'CloudLevel', 'LevelCO', 'LevelNO', 'LevelNO2', 'LevelO3', 'LevelSO2', 'LevelNH3', 'LevelPM2', 'LevelPM10', 'Temperature-M1', 'DewPoint-M1', 'Pressure-M1', 'Humidity-M1', 'WindSpeed-M1', 'WindSine-M1', 'WindCosine-M1', 'CloudLevel-M1', 'LevelCO-M1', 'LevelNO-M1', 'LevelNO2-M1', 'LevelO3-M1', 'LevelSO2-M1', 'LevelNH3-M1', 'LevelPM2-M1', 'LevelPM10-M1', 'Temperature-Diff-M1', 'DewPoint-Diff-M1', 'Pressure-Diff-M1', 'Humidity-Diff-M1', 'WindSpeed-Diff-M1', 'WindSine-Diff-M1', 'WindCosine-Diff-M1', 'CloudLevel-Diff-M1', 'LevelCO-Diff-M1', 'LevelNO-Diff-M1', 'LevelNO2-Diff-M1', 'LevelO3-Diff-M1', 'LevelSO2-Diff-M1', 'LevelNH3-Diff-M1', 'LevelPM2-Diff-M1', 'LevelPM10-Diff-M1', ... , 'Temperature-M13', 'DewPoint-M13', 'Pressure-M13', 'Humidity-M13', 'WindSpeed-M13', 'WindSine-M13', 'WindCosine-M13', 'CloudLevel-M13', 'LevelCO-M13', 'LevelNO-M13', 'LevelNO2-M13', 'LevelO3-M13', 'LevelSO2-M13', 'LevelNH3-M13', 'LevelPM2-M13', 'LevelPM10-M13', 'Temperature-Diff-M13', 'DewPoint-Diff-M13', 'Pressure-Diff-M13', 'Humidity-Diff-M13', 'WindSpeed-Diff-M13', 'WindSine-Diff-M13', 'WindCosine-Diff-M13', 'CloudLevel-Diff-M13', 'LevelCO-Diff-M13', 'LevelNO-Diff-M13', 'LevelNO2-Diff-M13', 'LevelO3-Diff-M13', 'LevelSO2-Diff-M13', 'LevelNH3-Diff-M13', 'LevelPM2-Diff-M13', 'LevelPM10-Diff-M13']`

Output features: `['Temperature-P12', 'DewPoint-P12', 'Pressure-P12', 'Humidity-P12', 'WindSpeed-P12', 'WindSine-P12', 'WindCosine-P12', 'CloudLevel-P12', 'LevelCO-P12', 'LevelNO-P12', 'LevelNO2-P12', 'LevelO3-P12', 'LevelSO2-P12', 'LevelNH3-P12', 'LevelPM2-P12', 'LevelPM10-P12']`

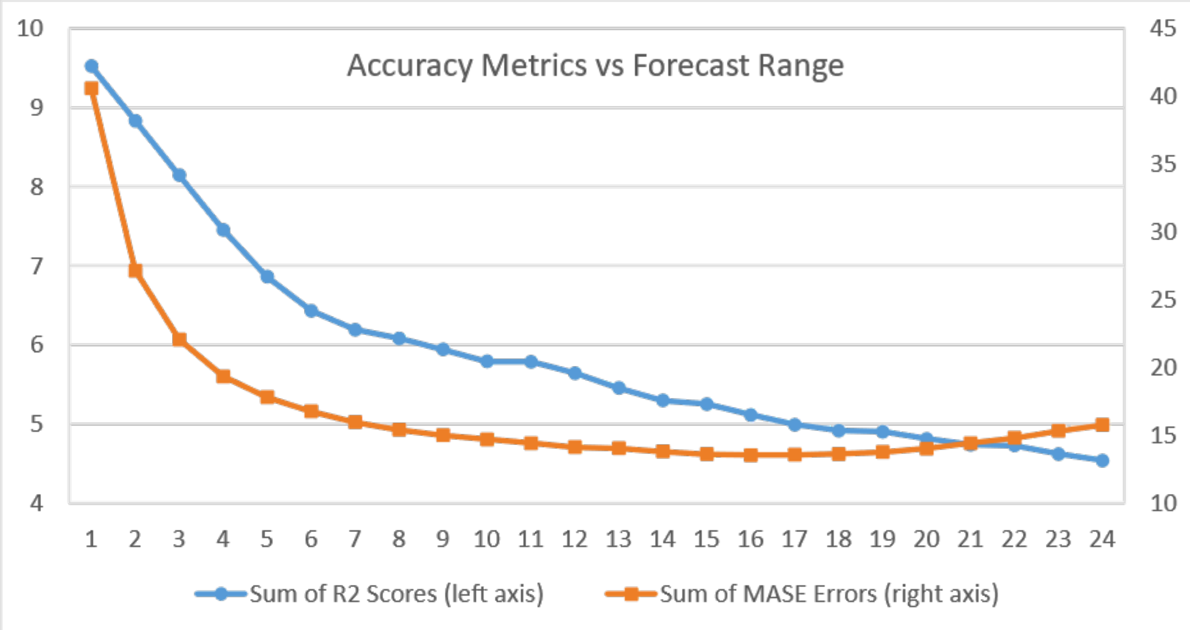

Figure 4: The prediction accuracy depending on the forecast range in hours.

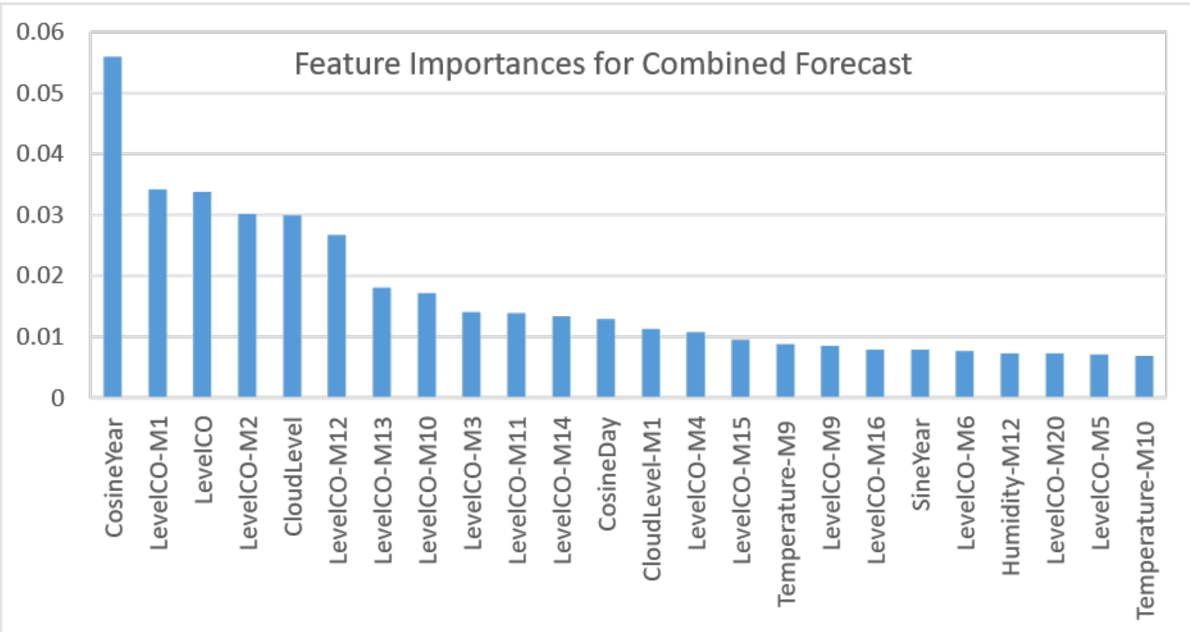

Figure 5: The feature importances for combined forecast obtained with extra trees regressor.

```
Testing mean scaled error(s) (MASE): [0.42760494 1.06906716 1.41122324 0.39518139
0.78960831 1.07409255 1.10271711 1.08403245 0.87116274 1.04628948 0.75481872 0.5116501
0.75501992 0.75901893 1.02661477 1.03704968], sum = 14.11515148
```

The performance of this input model for different forecast ranges is demonstrated in Figure 4. The R^2 score is more relevant in this case, and the best results were obtained for 1-hour forecasting. As shown in Equation 1, the MASE metric depends on the forecast range, making the comparison of nearby samples unfair. This dependency is presented here for illustrative purposes.

The feature importances calculated by ExtraTreesRegressor class for a full 24-hour history with periodic parameters are presented in Figure 5. It appears that cloudiness and CO concentration are the most predictive parameters. Additionally, the cosine representation of yearly and daily cycles are quite important.

6. Prediction of Weather Outputs

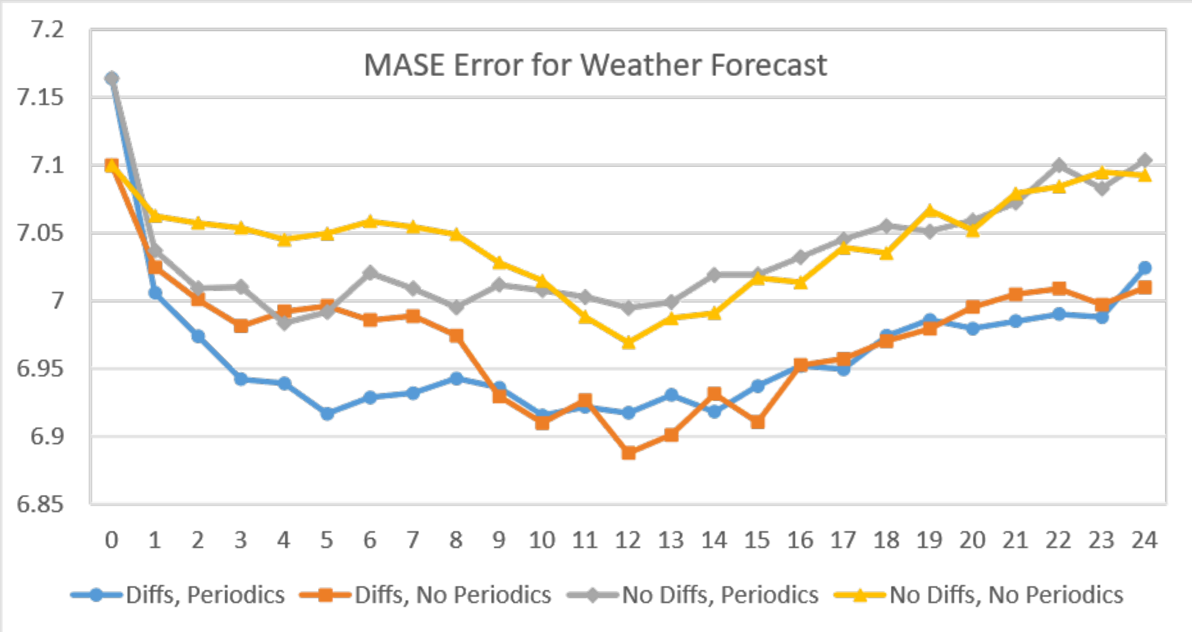

Figure 6: The sum of MASE errors for weather forecast depending on history length in hours.

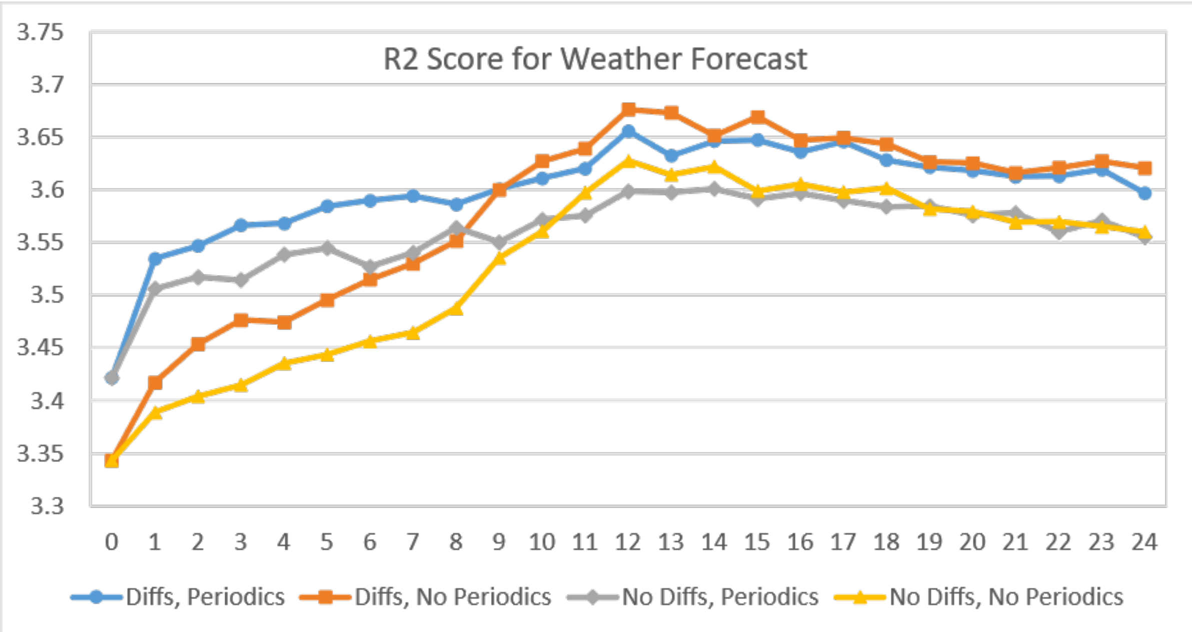

Figure 7: The sum of R2 scores for weather forecast depending on history length in hours.

While preserving the same input features there is a way to split output parameters on weather and air pollution groups. The MASE metrics for the forecasting of weather parameters are shown above in Figure 6. The best results were obtained again for 12-hour history and without periodic parameters, and this is an improvement in relation to combined forecast.

```
Testing mean scaled error(s) (MASE): [0.37630252 0.98280471 1.15055085 0.37381643
0.79749537 1.05510771 1.09868261 1.05295743], sum = 6.887717634
```

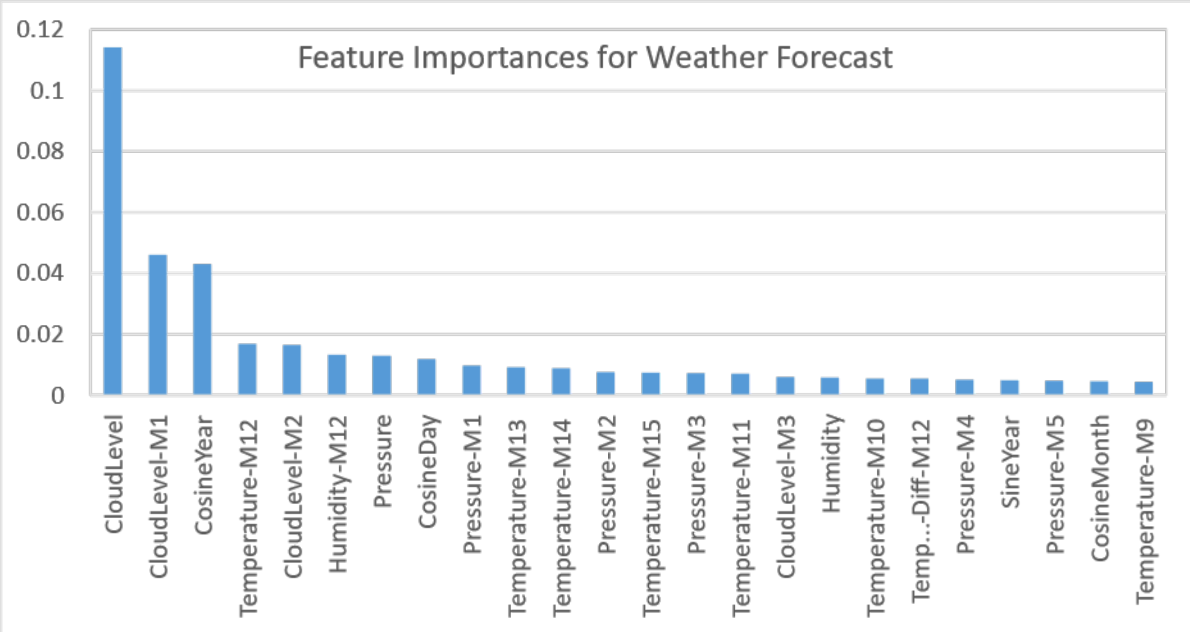

Figure 8: The feature importances for weather forecast obtained with extra trees regressor.

7. Prediction of Pollution Outputs

The MASE metrics for the prediction of pollution parameters are shown below in Figure 9. The best results were obtained for 17-hour history with differences and with periodic parameters.

```
Testing mean scaled error(s) (MASE): [0.85229725 1.02571192 0.75076816 0.49912568
0.75825543 0.76679182 1.01558379 1.0260866], sum = 6.694620651
```

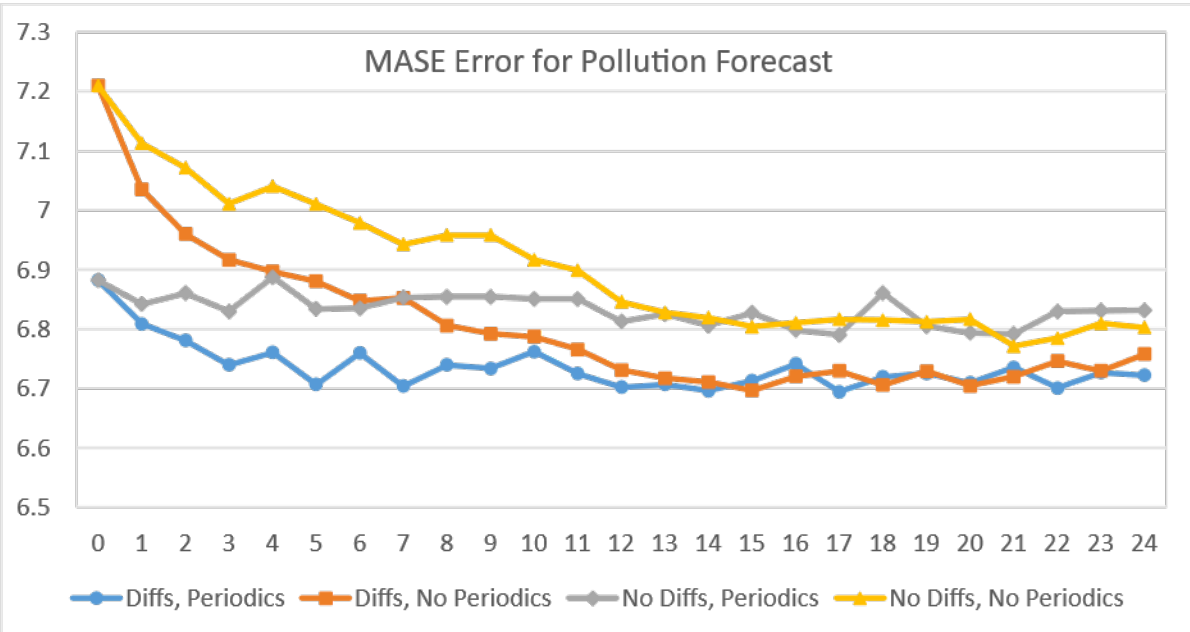

Figure 9: The sum of MASE errors for pollution forecast depending on history length in hours.

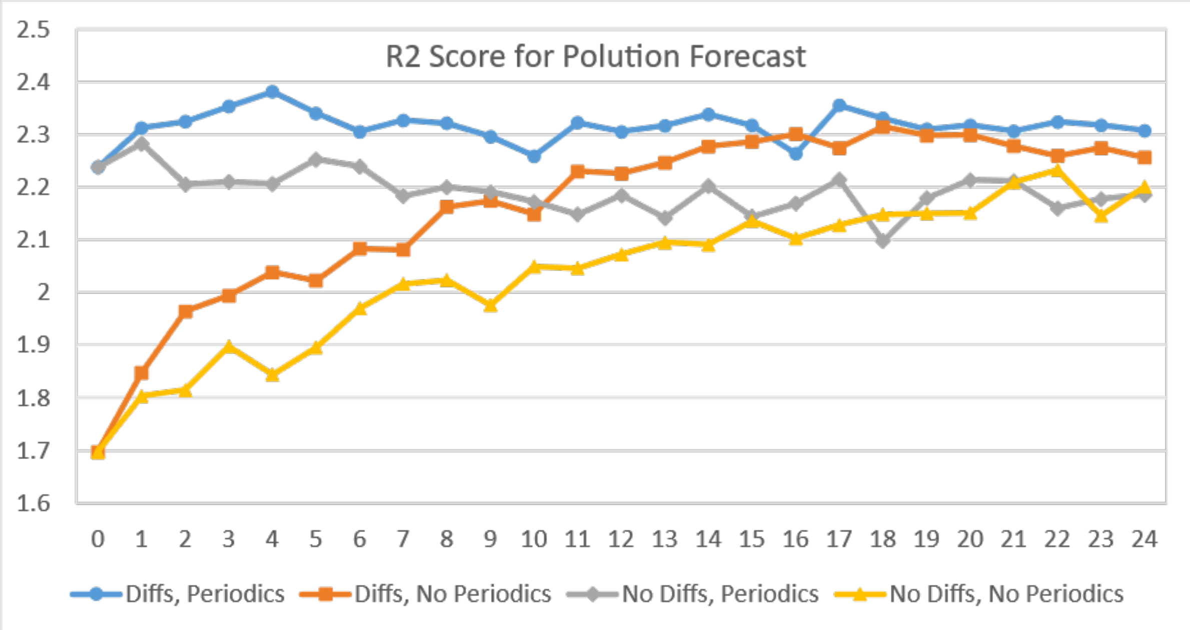

Figure 10: The sum of R2 scores for pollution forecast depending on history length in hours.

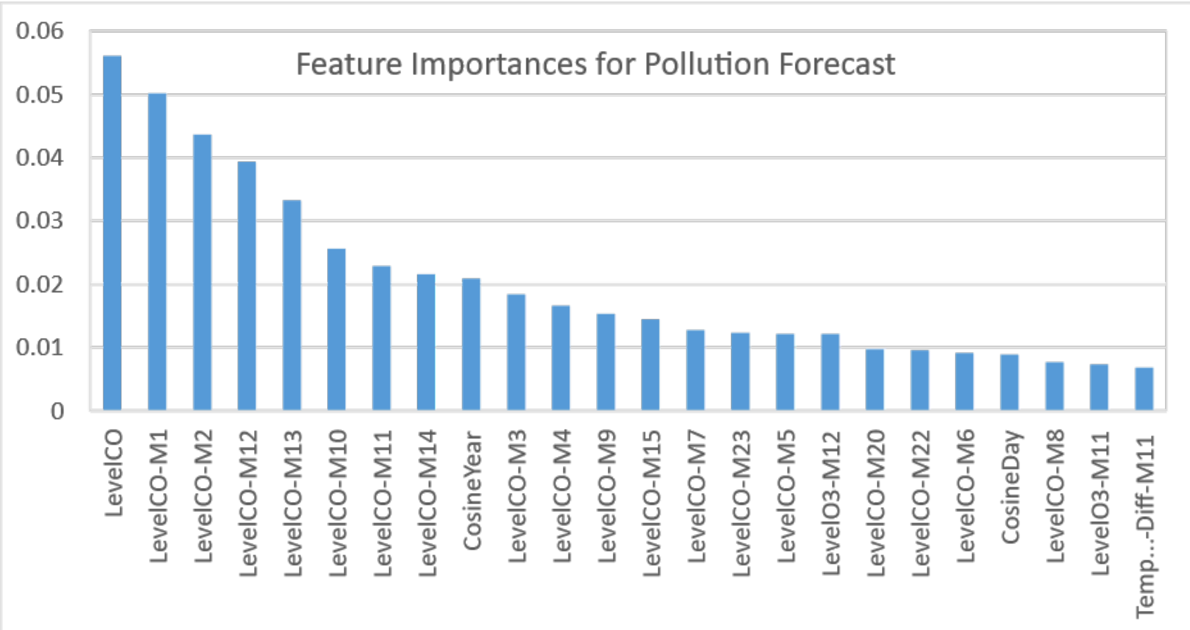

Figure 11: The feature importances for pollution forecast obtained with extra trees regressor.

And this is another improvement in comparison to the combined forecast. Regarding the shape of MASE graph, there is a general rule that initially the prediction accuracy improves when more useful information is provided to machine learning algorithm. However, when parameters become redundant or start introducing the noise into the system the forecast quality decreases.

As for input feature selection, there is a possibility to select the most important features using SelectFromModel class [15]. At the same time, this research is particularly difficult for weather and air pollution datasets and it did not become the part of this article.

8. Comparison of Regression Algorithms

Once the split of output parameters allowed to improve the prediction accuracy, it makes sense to consider forecasting of a single output. Besides, this can be done using other regression algorithms available in scikit-learn library, the MASE metrics obtained are presented in Table 2.

Table 2a: MASE error obtained for weather parameters and 12-hour forecasting.

Regression Algorithm	Temperature-P12	DewPoint-P12	Pressure-P12	Humidity-P12
Gradient Boosting	**0.290448**	0.804321	0.780185	**0.346654**
Support Vector Machine	0.323052	0.812869	0.782525	0.361426
Histo-Gradient Boosting	0.290831	**0.803297**	**0.776752**	0.348354
Extra Trees Regressor	0.309699	0.830221	0.810966	0.351747
Random Forest Regressor	0.312626	0.823599	0.816720	0.351488
Elastic Net Regression	0.344540	0.845808	0.800400	0.373180
Linear Regression	0.344543	0.845905	0.800894	0.373179
Bayes Ridge Regression	0.344551	0.846013	0.800875	0.373198
Decision Tree Regressor	0.371178	0.951572	0.943605	0.397570
Multi-Layer Perceptron	0.343359	0.856236	0.835506	0.370349
Nearest Neighbors	0.508248	1.217463	1.798733	0.421180
Ada Boost Regressor	0.423590	1.088373	1.025406	0.506579

Table 2b: MASE error obtained for weather parameters and 12-hour forecasting.

Regression Algorithm	WindSpeed-P12	WindSine-P12	WindCosine-P12	CloudLevel-P12
Gradient Boosting	0.765578	**0.868253**	**0.921366**	0.986027
Support Vector Machine	**0.753652**	0.890283	0.922716	**0.950337**
Histo-Gradient Boosting	0.775877	0.882801	0.932350	1.022101
Extra Trees Regressor	0.777338	0.906263	0.947481	1.041029
Random Forest Regressor	0.777611	0.907580	0.949287	1.046414
Elastic Net Regression	0.778504	0.924555	0.953478	1.051515
Linear Regression	0.778875	0.924119	0.953031	1.051495
Bayes Ridge Regression	0.778524	0.924355	0.953108	1.051521
Decision Tree Regressor	0.784572	0.925449	0.975530	1.056228
Multi-Layer Perceptron	0.791185	0.983977	1.043178	1.050212
Nearest Neighbors	0.801563	1.093743	1.179717	1.051063
Ada Boost Regressor	0.996020	0.963977	0.994661	1.160933

Table 2c: MASE error obtained for pollution parameters and 12-hour forecasting.

Regression Algorithm	LevelCO-P12	LevelNO-P12	LevelNO2-P12	LevelO3-P12
Gradient Boosting	**0.811747**	0.588180	**0.641686**	**0.458602**
Support Vector Machine	0.819337	**0.526212**	0.645346	0.463885
Histo-Gradient Boosting	0.850712	0.907135	0.681960	0.460109
Extra Trees Regressor	0.859329	1.103889	0.719776	0.469639
Random Forest Regressor	0.879540	1.163759	0.719972	0.470645
Elastic Net Regression	0.882887	1.175872	0.746956	0.481331
Linear Regression	0.883199	1.180683	0.748547	0.482141
Bayes Ridge Regression	0.883420	1.180751	0.748598	0.482101

Decision Tree Regressor	0.928463	1.028117	0.739561	0.503332
Multi-Layer Perceptron	0.903014	1.280309	0.756154	0.484334
Nearest Neighbors	0.943229	0.869021	0.751331	0.524242
Ada Boost Regressor	2.415821	13.002211	2.310592	0.546100

Table 2d: MASE error obtained for pollution parameters and 12-hour forecasting.

Regression Algorithm	LevelSO2-P12	LevelNH3-P12	LevelPM2-P12	LevelPM10-P12
Gradient Boosting	**0.632347**	**0.649840**	0.884449	0.888412
Support Vector Machine	0.635136	0.682333	**0.877520**	**0.885628**
Histo-Gradient Boosting	0.672324	0.673715	0.918804	0.916783
Extra Trees Regressor	0.686835	0.686608	0.915955	0.930745
Random Forest Regressor	0.687291	0.690882	0.923192	0.930449
Elastic Net Regression	0.693963	0.753811	0.924101	0.920190
Linear Regression	0.694797	0.754685	0.924244	0.920238
Bayes Ridge Regression	0.694855	0.754755	0.924272	0.920286
Decision Tree Regressor	0.728129	0.775167	0.965425	0.936466
Multi-Layer Perceptron	0.711183	0.806256	0.928723	0.927701
Nearest Neighbors	0.732473	0.769242	1.045273	1.055532
Ada Boost Regressor	2.430591	3.127918	2.956627	2.570652

The prediction accuracy has been improved again. The hyperparameters for machine learning algorithms listed in a table were manually optimized and they are available in Appendix B. As for R^2 scores and MAE metrics for the same experiments they are presented in Appendices C and D.

It was quite expected that decision tree based ensemble methods would take top of the chart. The negative surprises are that KNeighborsRegressor provided poor results and AdaBoostRegressor failed to forecast many output characteristics. The positive surprise is that Support Vector Machine (class NuSVR) took second place. However, this was achieved at the cost of high training time that takes tens of minutes on 8-core machine.

The winner algorithm for this dataset is GradientBoostingRegressor, its training time for every model takes about 5 minutes. The HistGradientBoostingRegressor provides similar results, but runs much faster, its training time is about 5 seconds per model. As for ExtraTreesRegressor, the time to train the model is also short and takes tens of seconds.

The linear methods occupy the middle of the list and this emphasizes the complexity of current task. It is quite unexpected that linear regression outperforms classic machine learning instruments like DecisionTreeRegressor and Multi-Layer Perceptron with quasi-Newton optimizer.

The prediction accuracy is not the only factor for selection of machine learning model. Other factors include the training time and the size of the serialized model on the disk. These aspects become especially important in cloud environments. Additionally, for selecting an input-output model that requires many iterations to complete, faster algorithms are preferred.

9. Prediction of Parameter Differences

So far the parameter differences were used only as inputs. At the same time, the differences can be forecasted the same way as direct parameters. The future value of a parameter can be calculated as the sum of current parameter value and difference forecasted.

The table 3 below compares these two approaches. Because of Equations 1 and 2 the MASE and R^2 metrics are not directly comparable. However, the MAE error for differences is calculated using equivalent formula, and this metric allows to compare the forecasting accuracy. It appears, that the forecast of differences provides an improvement for many weather parameters and some pollution parameters. And this happens more often for characteristics with good predictability.

Table 3a: Metrics obtained for weather parameters using gradient boosting regressor.

Prediction Type, Metric	Temperature-P12	DewPoint-P12	Pressure-P12	Humidity-P12
Direct Forecast, MASE	0.290448	0.804321	0.780185	0.346654
Difference Forecast, MASE	0.148818	0.483919	0.646530	0.175734
Direct Forecast, R2	0.952527	0.846526	0.877577	0.696013
Difference Forecast, R2	0.905270	0.226145	0.406070	0.846706
Direct Forecast, MAE	1.676653	1.752220	1.887538	6.905454
Difference Forecast, MAE	**1.666748**	**1.726670**	**1.873221**	**6.814671**

Table 3b: Metrics obtained for weather parameters using gradient boosting regressor.

Prediction Type	WindSpeed-P12	WindSine-P12	WindCosine-P12	CloudLevel-P12
Direct Forecast, MASE	0.765578	0.868253	0.921366	0.986027
Difference Forecast, MASE	0.424841	0.495768	0.534099	0.554002
Direct Forecast, R2	0.042988	0.249771	0.245715	0.334586
Difference Forecast, R2	0.489904	0.388062	0.323811	0.312515
Direct Forecast, MAE	1.210395	**0.503919**	0.494008	25.698767
Difference Forecast, MAE	**1.206469**	0.504502	**0.493705**	**25.486546**

Table 3c: Metrics obtained for pollution parameters using gradient boosting regressor.

Prediction Type	LevelCO-P12	LevelNO-P12	LevelNO2-P12	LevelO3-P12
Direct Forecast, MASE	0.811747	0.588180	0.641686	0.458602
Difference Forecast, MASE	0.474729	0.306328	0.355471	0.241845
Direct Forecast, R2	0.719361	0.032797	0.281042	0.532976
Difference Forecast, R2	0.338298	0.431516	0.518850	0.778906
Direct Forecast, MAE	**16.238159**	0.855437	**3.835031**	**14.150969**
Difference Forecast, MAE	16.365860	**0.848951**	3.934977	14.272140

Table 3d: Metrics obtained for pollution parameters using gradient boosting regressor.

Prediction Type	LevelSO2-P12	LevelNH3-P12	LevelPM2-P12	LevelPM10-P12
Direct Forecast, MASE	0.632347	0.649840	0.884449	0.888412
Difference Forecast, MASE	0.344376	0.365622	0.526062	0.543068
Direct Forecast, R2	0.269321	0.377308	0.397727	0.365168
Difference Forecast, R2	0.580179	0.512974	0.223234	0.123103
Direct Forecast, MAE	1.969536	**0.584217**	2.667819	**3.466319**
Difference Forecast, MAE	**1.940803**	0.591039	**2.619420**	3.488378

Conclusion

This work proposes modern approaches for the forecasting of weather and air pollution parameters that define input history length, output parameter configuration and selection of machine learning algorithm. The best results were obtained for GradientBoostingRegressor class.

The usage of differences both on input and output sides of the algorithm helps to improve the results. The forecasting accuracy varies a lot for different output parameters. In particular, wind, cloudiness and air pollution characteristics are quite difficult to predict.

The selection of output parameters has significant influence on the accuracy of the algorithm. And the best results were obtained when individual machine learning model was trained for every output feature. Correspondingly, the selection of single multi-output regression algorithm is not the optimal choice. As expected, better results require more computational resources.

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

References

[1] C. M. Bishop, Pattern Recognition and Machine Learning, Springer, New York, NY, 2006.
[2] S. Haykin, Neural Networks: A Comprehensive Foundation, Prentice Hall, Hoboken, NJ, 1998.
[3] V. N. Vapnik, Statistical Learning Theory, Wiley, Hoboken, NJ, 1998.
[4] C. Shah, A Hands-On Introduction to Machine Learning, 1st. ed., Cambridge University Press, Cambridge, 2023.
[5] Machine learning methods for predicting wind generation, Electricity Authority Te Mana Hiko, Wellington, 2022. URL: https://www.ea.govt.nz/documents/2385/Machine-learning-methods-for-predicting-wind-generation_MkxN3ZL.pdf.
[6] E. Levinson, Three approaches to encoding time information as features for ML models, Nvidia Developer Technical Blog, 2022, URL: https://developer.nvidia.com/blog/three-approaches-to-encoding-time-information-as-features-for-ml-models/.
[7] A. Van Wyk, Encoding cyclical features for deep learning, URL: https://www.kaggle.com/code/avanwyk/encoding-cyclical-features-for-deep-learning.
[8] Scikit-learn: imputation of missing values. URL: https://scikit-learn.org/stable/modules/impute.html.
[9] Scikit-learn: machine learning in Python. URL: https://scikit-learn.org/stable/.
[10] Skforecast: a Python library for time series forecasting. URL: https://skforecast.org/0.14.0/index.html.
[11] Mlforecast: scalable machine learning for time series forecasting. URL: https://nixtlaverse.nixtla.io/mlforecast/index.html.
[12] Scikit-learn: ExtraTreesRegressor. URL: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.ExtraTreesRegressor.html#sklearn.ensemble.ExtraTreesRegressor.
[13] B. Wohlwend, Regression model evaluation metrics: R-squared, adjusted R-squared, MSE, RMSE, and MAE, 2023. URL: https://medium.com/@brandon93.w/regression-model-evaluation-metrics-r-squared-adjusted-r-squared-mse-rmse-and-mae-24dcc0e4cbd3.
[14] Feature selection with scikit-learn library. URL: https://scikit-learn.org/stable/modules/feature_selection.html.
[15] Scikit-learn: SelectFromModel class. URL: https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectFromModel.html.
[16] S. Sande, Get started with time series forecasting in Python, 2020. URL: https://medium.com/analytics-vidhya/get-started-with-time-series-forecasting-in-python-c8ca78ee84a5.

[17] G. Ravindiran, G. Hayder, K. Kanagarathinam, A. Alagumalai, C. Sonne, Air quality prediction by machine learning models: A predictive study on the Indian coastal city of Visakhapatnam, Chemosphere 338 (2023). doi:10.1016/j.chemosphere.2023.139518.
[18] A. Samad, S. Garuda, U. Vogt, B. Yang, Air pollution prediction using machine learning techniques – an approach to replace existing monitoring stations with virtual monitoring stations, Atmospheric Environment 310 (2023). doi:10.1016/j.atmosenv.2023.119987.

A. Appendix: MASE Metric

The function to calculate the mean absolute scaled error is missing in version 1.6 of scikit-learn library, so one of the options is to implement it manually.

```
def mean_absolute_scaled_error(dataset_outputs, \
    predicted_dataset_outputs, multioutput = 'raw_values', forecast_range = 1):

    assert multioutput == 'raw_values', "Only multi-output mode is supported for now"

    if (isinstance(dataset_outputs, pandas.DataFrame)):
        dataset_outputs = dataset_outputs.to_numpy()
    if (isinstance(predicted_dataset_outputs, pandas.DataFrame)):
        predicted_dataset_outputs = predicted_dataset_outputs.to_numpy()

    if (len(dataset_outputs.shape) == 1):
        dataset_outputs = numpy.array([[number] for number in dataset_outputs])
    if (len(predicted_dataset_outputs.shape) == 1):
        predicted_dataset_outputs = numpy.array \
            ([[number] for number in predicted_dataset_outputs])

    record_count = dataset_outputs.shape[0]
    assert record_count == predicted_dataset_outputs.shape[0], \
        "The original and predicted dataset outputs should have the same record count"

    column_count = dataset_outputs.shape[1]
    assert column_count == predicted_dataset_outputs.shape[1], \
        "The original and predicted dataset outputs should have the same column count"

    assert record_count > forecast_range, \
        "The number of dataset records should be higher than forecast range"

    scaled_errors = []
    for j in range(0, column_count):

        naive_prediction_mismatch = 0.0
        for i in range (forecast_range, record_count):
            diff = dataset_outputs[i, j] - dataset_outputs[i - forecast_range, j]
            naive_prediction_mismatch += abs(diff)

        mase_denominator = naive_prediction_mismatch / (record_count - forecast_range)

        current_prediction_mismatch = 0.0
        for i in range(0, record_count):
            diff = predicted_dataset_outputs[i, j] - dataset_outputs[i, j]
            current_prediction_mismatch += abs(diff)

        mase_numerator = current_prediction_mismatch / record_count

        scaled_error = mase_numerator / mase_denominator
        scaled_errors.append(scaled_error)

    return numpy.array(scaled_errors)
```

B. Appendix: Hyperparameters

The Python-based expressions below represent the constructors of regression algorithm objects with corresponding hyperparameters, random number generation and parallelization settings.

```
ExtraTreesRegressor(n_estimators = 100, criterion = 'squared_error',
    ccp_alpha = 0.0, random_state = 1, n_jobs = 8)

RandomForestRegressor(n_estimators = 100, criterion = 'squared_error',
    max_features = 0.2, min_samples_split = 6, ccp_alpha = 0.0,
    random_state = 1, n_jobs = 8)

HistGradientBoostingRegressor(loss = 'squared_error', learning_rate = 0.1,
    max_iter = 100, min_samples_leaf = 20, l2_regularization = 0.1, random_state = 1)

GradientBoostingRegressor(loss = 'huber', learning_rate = 0.15,
    n_estimators = 100, subsample = 0.9, criterion = 'friedman_mse',
    max_depth = 5, alpha = 0.85, random_state = 1)

AdaBoostRegressor(estimator = initial_estimator,
    n_estimators = 100, loss = 'linear', random_state = 1)

DecisionTreeRegressor(criterion = 'squared_error', max_depth = 7,
    min_samples_leaf = 2, min_weight_fraction_leaf = 0.011, random_state = 1)

KNeighborsRegressor(n_neighbors = 24, weights = 'distance',
    algorithm = 'auto', p = 1, metric='minkowski', n_jobs = 8)

NuSVR(nu = 0.8, C = 1000.0, kernel = 'rbf')

MLPRegressor(hidden_layer_sizes = (200,), activation = 'relu',
    solver = 'lbfgs', alpha = 0.0000, max_iter = 1000, random_state = 1)

ElasticNet(alpha = 0.01, l1_ratio = 0.01, fit_intercept = True, precompute = True,
    max_iter = 1000, tol = 0.001, selection='cyclic', random_state = 1)

Ridge(alpha = 1.0, fit_intercept = True, solver = 'svd', random_state = 1)

LinearRegression(fit_intercept = True, n_jobs = 8)
```

C. Appendix: R2 Scores

The R2 scores below were calculated for experiments covered in section 8, when the machine learning algorithm had just one output parameter configured. The best algorithm according to this metric is still gradient boosting regressor.

Table 4a: R2 scores obtained for weather parameters and 12-hour forecasting.

Regression Algorithm	Temperature-P12	DewPoint-P12	Pressure-P12	Humidity-P12
Gradient Boosting	**0.952527**	0.846526	0.877577	0.696013
Support Vector Machine	0.941434	0.848278	0.873980	0.684538
Histo-Gradient Boosting	0.951968	**0.854775**	**0.878156**	0.694654
Extra Trees Regressor	0.945569	0.835638	0.866677	0.696572
Random Forest Regressor	0.945656	0.849587	0.865073	**0.703304**
Elastic Net Regression	0.936032	0.842633	0.873928	0.679936

Linear Regression	0.936024	0.842545	0.873843	0.679844
Bayes Ridge Regression	0.936022	0.842545	0.873847	0.679803
Decision Tree Regressor	0.923702	0.827557	0.833052	0.621333
Multi-Layer Perceptron	0.934916	0.840566	0.865753	0.679687
Nearest Neighbors	0.845698	0.756060	0.440501	0.583007
Ada Boost Regressor	0.907491	0.803719	0.814220	0.529173

Table 4b: R2 scores obtained for weather parameters and 12-hour forecasting.

Regression Algorithm	WindSpeed-P12	WindSine-P12	WindCosine-P12	CloudLevel-P12
Gradient Boosting	0.042988	0.249771	0.245715	0.334586
Support Vector Machine	0.045956	0.193935	0.200256	0.292398
Histo-Gradient Boosting	**0.054283**	**0.260373**	**0.256926**	0.335317
Extra Trees Regressor	0.051183	0.248838	0.254104	0.332687
Random Forest Regressor	0.050198	0.249483	0.255977	**0.336622**
Elastic Net Regression	0.052310	0.215894	0.231853	0.328801
Linear Regression	0.051565	0.215517	0.231558	0.328709
Bayes Ridge Regression	0.051904	0.214978	0.231439	0.328753
Decision Tree Regressor	0.028121	0.204260	0.196661	0.300969
Multi-Layer Perceptron	0.018103	0.084832	0.074604	0.311056
Nearest Neighbors	-0.040610	-0.040357	-0.120212	0.258656
Ada Boost Regressor	-0.436430	0.197250	0.213247	0.276038

Table 4c: R2 scores obtained for pollution parameters and 12-hour forecasting.

Regression Algorithm	LevelCO-P12	LevelNO-P12	LevelNO2-P12	LevelO3-P12
Gradient Boosting	**0.719361**	**0.032797**	0.281042	**0.532975**
Support Vector Machine	0.708032	0.010489	0.208164	0.520055
Histo-Gradient Boosting	0.714551	-0.033228	**0.287136**	0.528286
Extra Trees Regressor	0.717004	-0.106754	0.262620	0.514243
Random Forest Regressor	0.711519	-0.213147	0.275667	0.517114
Elastic Net Regression	0.708426	0.002286	0.260521	0.492557
Linear Regression	0.708604	-0.000199	0.260120	0.490879
Bayes Ridge Regression	0.708593	-0.000259	0.260093	0.490919
Decision Tree Regressor	0.683501	-0.106787	0.211201	0.442830
Multi-Layer Perceptron	0.692603	-0.100421	0.237774	0.491020
Nearest Neighbors	0.659045	-0.110895	0.176147	0.417633
Ada Boost Regressor	-0.201680	-31.970616	-2.111877	0.405907

Table 4d: R2 scores obtained for pollution parameters and 12-hour forecasting.

Regression Algorithm	LevelSO2-P12	LevelNH3-P12	LevelPM2-P12	LevelPM10-P12
Gradient Boosting	**0.269321**	0.377308	0.397727	0.365168

Support Vector Machine	0.204830	0.314250	**0.449990**	0.414164
Histo-Gradient Boosting	0.252145	**0.387741**	0.379408	0.354160
Extra Trees Regressor	0.240582	0.376137	0.381599	0.330860
Random Forest Regressor	0.235017	0.360005	0.392714	0.347618
Elastic Net Regression	0.208284	0.306561	0.443459	0.431940
Linear Regression	0.207736	0.305467	0.443471	**0.432371**
Bayes Ridge Regression	0.207774	0.305461	0.443459	0.432365
Decision Tree Regressor	0.114599	0.198714	0.361705	0.378636
Multi-Layer Perceptron	0.188954	0.243697	0.435550	0.426246
Nearest Neighbors	0.143160	0.221616	0.239645	0.147215
Ada Boost Regressor	-3.590727	-4.966839	-1.987259	-1.112341

D. Appendix: MAE Results

The MAE errors below were calculated for experiments covered in section 8, when the machine learning algorithm had just one output parameter configured. The measurement units correspond to original parameters listed in Table 1.

Table 5a: MAE errors obtained for weather parameters and 12-hour forecasting.

Regression Algorithm	Temperature-P12	DewPoint-P12	Pressure-P12	Humidity-P12
Gradient Boosting	**1.676653**	1.752220	1.887538	**6.905454**
Support Vector Machine	1.864859	1.770843	1.893198	7.199709
Histo-Gradient Boosting	1.678862	**1.749990**	**1.879230**	6.939328
Extra Trees Regressor	1.787780	1.808643	1.962007	7.006913
Random Forest Regressor	1.804674	1.794218	1.975927	7.001756
Elastic Net Regression	1.988903	1.842599	1.936445	7.433865
Linear Regression	1.988922	1.842810	1.937639	7.433842
Bayes Ridge Regression	1.988969	1.843047	1.937593	7.434224
Decision Tree Regressor	2.142672	2.073008	2.282906	7.919711
Multi-Layer Perceptron	1.982088	1.865318	2.021378	7.377466
Nearest Neighbors	2.933928	2.652254	4.351756	8.390043
Ada Boost Regressor	2.445231	2.371029	2.480811	10.091202

Table 5b: MAE errors obtained for weather parameters and 12-hour forecasting.

Regression Algorithm	WindSpeed-P12	WindSine-P12	WindCosine-P12	CloudLevel-P12
Gradient Boosting	1.210395	**0.503919**	**0.494008**	25.698767
Support Vector Machine	**1.191540**	0.516705	0.494732	**24.768602**
Histo-Gradient Boosting	1.226677	0.512362	0.499898	26.638973
Extra Trees Regressor	1.228988	0.525980	0.508010	27.132301
Random Forest Regressor	1.229419	0.526744	0.508979	27.272650
Elastic Net Regression	1.230830	0.536596	0.511226	27.405578
Linear Regression	1.231417	0.536342	0.510986	27.405074

Bayes Ridge Regression	1.230863	0.536479	0.511027	27.405747
Decision Tree Regressor	1.240425	0.537115	0.523050	27.528412
Multi-Layer Perceptron	1.250879	0.571083	0.559321	27.371622
Nearest Neighbors	1.267288	0.634789	0.632528	27.393801
Ada Boost Regressor	1.574728	0.559475	0.533307	30.257335

Table 5c: MAE errors obtained for pollution parameters and 12-hour forecasting.

Regression Algorithm	LevelCO-P12	LevelNO-P12	LevelNO2-P12	LevelO3-P12
Gradient Boosting	**16.238159**	0.855437	**3.835031**	**14.150969**
Support Vector Machine	16.389989	**0.765311**	3.856904	14.313992
Histo-Gradient Boosting	17.017623	1.319318	4.075729	14.197469
Extra Trees Regressor	17.189998	1.605473	4.301738	14.491541
Random Forest Regressor	17.594297	1.692547	4.302908	14.522573
Elastic Net Regression	17.661253	1.710163	4.464175	14.852329
Linear Regression	17.667499	1.717161	4.473679	14.877302
Bayes Ridge Regression	17.671916	1.717260	4.473991	14.876083
Decision Tree Regressor	18.572952	1.495272	4.419984	15.531197
Multi-Layer Perceptron	18.063861	1.862054	4.519149	14.944986
Nearest Neighbors	18.868328	1.263886	4.490325	16.176416
Ada Boost Regressor	48.326031	18.910138	13.809237	16.850893

Table 5d: MAE errors obtained for pollution parameters and 12-hour forecasting.

Regression Algorithm	LevelSO2-P12	LevelNH3-P12	LevelPM2-P12	LevelPM10-P12
Gradient Boosting	**1.969536**	**0.584217**	2.667819	3.466319
Support Vector Machine	1.978223	0.613429	**2.646918**	**3.455459**
Histo-Gradient Boosting	2.094050	0.605681	2.771447	3.577015
Extra Trees Regressor	2.139244	0.617272	2.762854	3.631489
Random Forest Regressor	2.140665	0.621114	2.784683	3.630335
Elastic Net Regression	2.161445	0.677689	2.787424	3.590307
Linear Regression	2.164044	0.678475	2.787855	3.590496
Bayes Ridge Regression	2.164224	0.678537	2.787940	3.590684
Decision Tree Regressor	2.267860	0.696888	2.912072	3.653812
Multi-Layer Perceptron	2.215080	0.724837	2.801365	3.619614
Nearest Neighbors	2.281392	0.691561	3.152924	4.118374
Ada Boost Regressor	7.570422	2.812049	8.918261	10.029923