Multivariate Delhi Climate Forecasting

Liam Glennie, Clara Molins

June 7, 2023

Abstract

This paper includes an extension of the previously presented paper, *Delhi Climate Forecasting*. The main objective is to perform predictions of the mean temperature of a day with multiple variables. Multiple models were evaluated with VAR with daily retraining proving to be superior in terms of metrics. However, it only proves useful in predicting the temperature of the following day. Therefore, it is incapable of producing long-term predictions.

1 Introduction

This work is a continuation of the previously handed paper titled *Delhi Climate Forecasting*. As mentioned in the previous work, the sample data this project is based on is the *Daily Climate time series data* dataset provided in this Kaggle link which contains four time-series attributes, namely *meantemp*, *humidity*, *wind_speed*, and *meanpressure*, with daily data from 1st January 2013 to 24th April 2017 from the city of Delhi.

In contrast to the previous work, this paper is a detailed multivariate analysis focused on predicting *meantemp*.

First, the results on multiple relevant time-series properties that have been studied for each of the attributes in the sample, are presented. This is followed by an evaluation of the Vector Autoregression (VAR) model, which, even though the focus of the work is to predict *meantemp*, is capable of predicting multiple time-series attributes at the same time. Afterward, conclusions on applying Support Vector Regression (SVR), Gradient Boosting (GB), and Long Short-Term Memory (LSTM) to predict *meantemp* are presented.

Finally, the main conclusions of the work are summarized in the last section.

2 Data Preprocessing

The data used for this part of the project stems from the preprocessed data of the previous project mentioned in the introduction. Briefly, some preprocessing steps were applied including removing duplicate days, identifying outliers, and imputing missing values. Finally, the data was split into training, validation, and test sets with the following proportions: 0.7, 0.15, and 0.15, respectively.

3 Studying Data Properties

Most multivariate time-series models work best when the data follows specific characteristics such as stationarity. For this reason, in the following subsections, multiple properties are analyzed on all the attributes of the sample.

3.1 Autocorrelation

Figure 1 shows that the four attributes in the dataset have autocorrelation. In the four subplots, all points can be found outside the blue shadowed area.

The presence of autocorrelation indicates there is a strong and persistent relationship between the current values of the time series attributes and their past values.

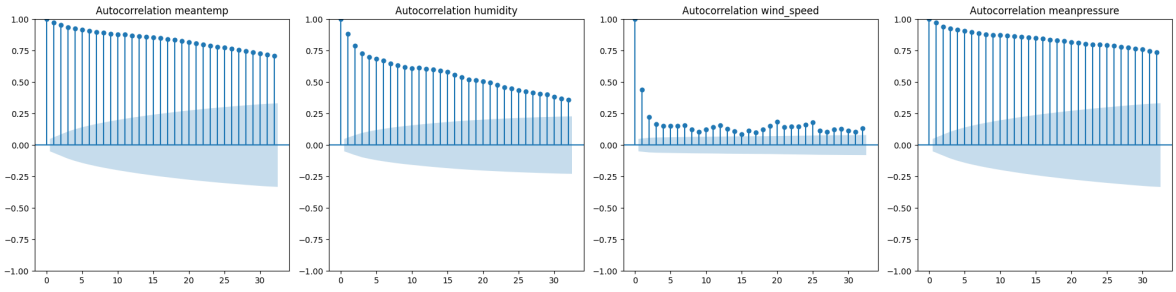

Figure 1: Autocorrelation plots per attribute. From right to left: *meantemp*, *humidity*, *wind_speed*, *meanpressure*.

3.2 Partial Autocorrelation

Figure 2 showcases the autocorrelation plots for each of the attributes. As can be observed, as one moves away from the early lags, the PACF values quickly become insignificant (they are inside the shadowed area).

This signifies that the variables have a relatively short memory or dependence on their past values. To be precise, the first ten lags always seem to be relevant, and from there, the dependence decreases. As our dataset contains daily data, the lags correspond to days. Therefore, it seems that to predict the *meantemp* or one of the other attributes in the sample, the previous 11 days seem to be meaningful.

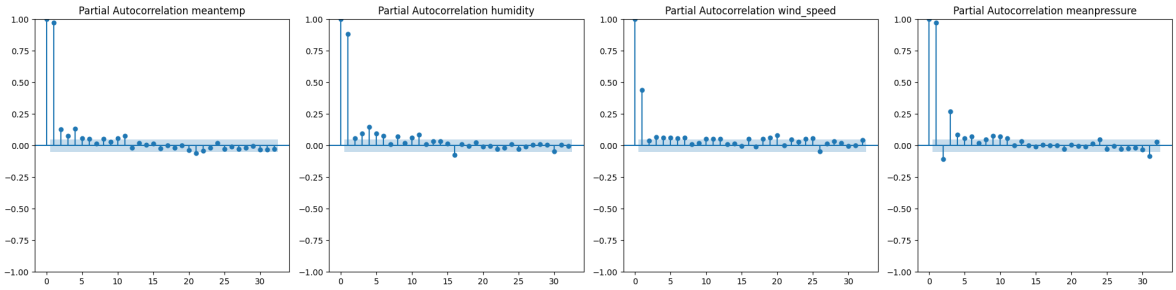

Figure 2: Partial autocorrelation plots per attribute. From right to left: *meantemp*, *humidity*, *wind_speed*, *meanpressure*.

3.3 Perfect Multicollinearity

Figure 3 shows that the correlation between two attributes in the sample that is closest to 1 or -1 is between *meantemp* and *meanpressure* with a value of -0.88. Therefore, it can be said that there is no perfect multicollinearity in the dataset.

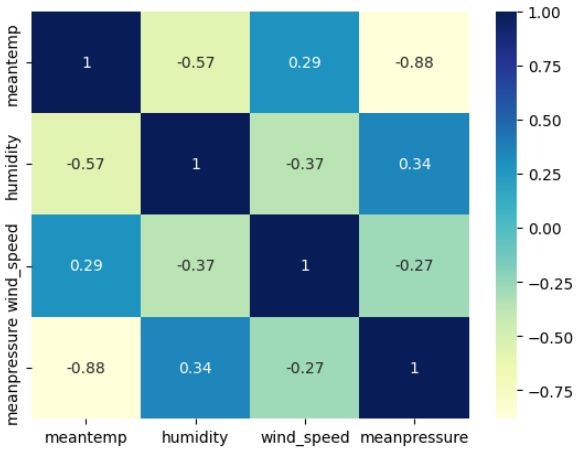

Figure 3: Correlation matrix

3.4 Augmented Dickey-Fuller Test

The augmented Dickey-Fuller test is employed to determine whether a time series has a unit root, indicating non-stationarity. Its null and alternative hypotheses are the following:

Null Hypothesis (H_0): The time series has a unit root, indicating it is non-stationary. In other words, the series follows a random walk or has a unit root in the autoregressive process.

Alternative Hypothesis (H_A): The time series is stationary, indicating the absence of a unit root. In this case, the series is considered to have a stable mean and does not exhibit a random walk behavior.

As Table 1 showcases, the null hypothesis was rejected when the test was applied to the *humidity* and the *wind_speed* attributes, indicating that they are stationary. In contrast, when applied to *meantemp* and *meanpressure* the null hypothesis fails to be rejected, indicating that they are non-stationary.

Attribute	P-value	Conclusion
meantemp	0.1488	H_0 fails to be rejected
humidity	0.005	H_0 is rejected
wind_speed	0.001	H_0 is rejected
meanpressure	0.165	H_0 fails to be rejected

Table 1: Dickey-Fuller test results.

3.5 Granger Causality Test

The Granger causality test, named after Clive Granger, is a statistical test used to determine whether a time-series variable can predict or "Granger-cause" another time-series variable. It helps assess the causal relationship between attributes. Note that this is a basic premise on which Vector Autoregression (VAR) models, whose implementation and evaluation are explained in Section 4.1, are based on.

The main idea behind this test is that if, for example, variable *meanpressure* predicts or "Granger-causes" variable *meantemp*, then the past values of *meanpressure* should provide helpful information to predict *meantemp*, beyond what can be predicted using only the past values of *meantemp* itself. Its null and alternative hypotheses are the following:

Null Hypothesis (H_0): There is no causal relationship between the variables being tested. The past values of one variable do not provide any useful information for predicting the other variable, beyond what can already be predicted using the past values of the second variable.

Alternative Hypothesis (H_A): There is a causal relationship between the variables being tested. The past values of one variable do provide additional information for predicting the other variable, beyond what can be predicted using the past values of the second variable alone.

Table 2 shows the p-values obtained. Each p-value indicates if the explanatory variable (X) can help predict the target variable (Y). The values on the diagonal are 1s because, of course, each variable clearly helps its own prediction. One should focus on the rest of the cells.

As can be observed, *humidity*, *wind_speed*, and *meanpressure* all help predict meantemp. Regarding humidity, only *meantemp* and *meanpressure* "Granger-cause" it, as the p-value for the *wind_speed* attribute is higher than 0.05. Additionally, *meantemp*, *humidity*, and *meanpressure* all have a causality relationship with *wind_speed*. Finally, *meantemp*, *humidity*, and *wind_speed* are relevant for the *meanpressure* prediction.

	meantemp (X)	**humidity (X)**	**wind_speed (X)**	**meanpressure (X)**
meantemp (Y)	1.0000	0.0125	0.0000	0.0001
humidity (Y)	0.0000	1.0000	0.1245	0.0068
wind_speed (Y)	0.0000	0.0000	1.0000	0.0000
meanpressure (Y)	0.0000	0.0003	0.0011	1.0000

Table 2: Granger test results.

3.6 Johansen Cointegration Test

The Johansen cointegration test is a statistical test, named after Søren Johansen, used to determine the presence and number of cointegrating relationships among a set of time series variables. Cointegration implies a long-term equilibrium relationship between variables, suggesting that they move together in the long run. In other words, suggests that while individual variables may be non-stationary, there exists a linear combination of these variables that is stationary.

Null Hypothesis (H_0): There are no cointegrating relationships among the variables. The variables are not jointly moving in a long-term equilibrium.

Alternative Hypothesis (H_A): There are one or more cointegrating relationships among the variables. This implies that the variables are jointly moving in a long-term equilibrium.

Table 3 showcases, for each attribute, its johansen statistic, the 95% critical value of its distribution, and the conclusion of the test. When applied to *meantemp* and *humidity* the null hypothesis is rejected as the test statistics exceed the critical values. In contrast, the null hypothesis fails to be rejected for *wind_speed* and *meanpressure* indicating the presence of cointegrating involving these variables.

Attribute	**Test Statistic**	**C(95%)**	**Conclusion**
meantemp	222.89	40.1749	H_0 is rejected
humidity	53.71	24.2761	H_0 is rejected
wind_speed	10.71	12.3212	H_0 fails to be rejected
meanpressure	0.01	4.1296	H_0 fails to be rejected

Table 3: Johansen test results.

Given the results of applying the Johansen Cointegration test on the *meantemp* variable, a long-term dependency between said attribute and the variables *humidity, wind_speed,* and *meanpressure* has been identified. Therefore, it seems reasonable to include said attributes as explanatory variables in the models to predict *meantemp*.

4 Modeling

This section will cover Vector Autoregression, Support Vector Regression, Gradient Boosting, and Long Short-Term Memory networks. The input data for each model will be discussed as well as evaluation metrics and visualization of results. The main objective of this report is to produce a model that predicts the mean temperature of the following day with multiple explanatory variables. Therefore, despite discussing briefly the performance on other attributes in Vector Autoregression, the main focus will be on *meantemp*.

4.1 Vector Autoregression (VAR)

Vector Autoregression is a multivariate time-series algorithm capable of predicting multiple features at the same time. Compared to univariate time-series models such as ARIMA or SARIMA, which are unidirectional models where the predictors influence the target and not vice versa, VAR is bidirectional. In other words, the variables influence each other. In particular, each attribute is modeled as a linear combination of past values of itself and the past values of other variables in the system.

The VAR algorithm assumes certain conditions on the data. The first and most important one is stationarity. As previously shown in Subsection 3.4, *humidity* and *wind_speed* are stationary but *meantemp* and *meanpressure* are non-stationary. In order to make all attributes stationary, the differencing technique was applied. This technique is usually only applied to the non-stationary features however, as the results of the Granger causality test shown in Subsection 3.5 indicate causality relationships between most of the attributes, it was decided to apply differencing to all of them to avoid disrupting the causality relationships and maintain consistency. Note that after applying differencing once, the Augmented Dickey-Fuller test was executed again, and it showed all variables were stationary.

Note that the fact that the Granger causality test indicates causality relationships between most attributes is very relevant for the Vector Autoregression algorithm which, as mentioned, models each attribute with a linear combination of the past values of itself and the rest of the attributes in the sample. If these relationships were not present in the data, this model would act similarly to ARIMA or SARIMA, which only work with the past values of the target variable.

Another assumption this algorithm makes on its input data is that it has no perfect multicollinearity. Luckily, as shown in Subsection 3.3 there is no perfect multicollinearity in the Delhi Climate dataset. VAR also assumes no serial correlation in the residuals. This final assumption was studied after fitting the models.

The following subsections showcase the results of the model first without using daily historical data to train it, and afterward re-training it daily with the new data. The results of the daily re-trained model without previously applying differencing are also shown.

4.1.1 VAR model without daily training

The VAR model itself, without training it daily after every prediction, showed very poor results. As can be observed in Figure 4, the first predictions seem to be more or less accurate but after the 10th day approximately, the model is incapable of properly predicting any of the features. This is probably due to the fact that to make a proper prediction, the model relies on the data of the eleven previous days. At first, it has the data it needs, but as days pass, it runs out of training data and it continuously predicts the same value.

Note that after fitting the model, the presence of serial correlation on the residuals was checked, and the results were negative. In particular, the Durbin-Watson statistic was calculated for each of the attributes and they were all very close to 2, meaning that there is no significant autocorrelation in the residuals of the model.

Table 4 showcases the mean squared error and the root mean squared error of this model per attribute:

Attribute	MSE	RMSE
meantemp	497.70	22.31
humidity	1170.58	34.21
wind_speed	21.29	4.61
meanpressure	181.17	13.46

Table 4: MSE and RMSE per attribute.

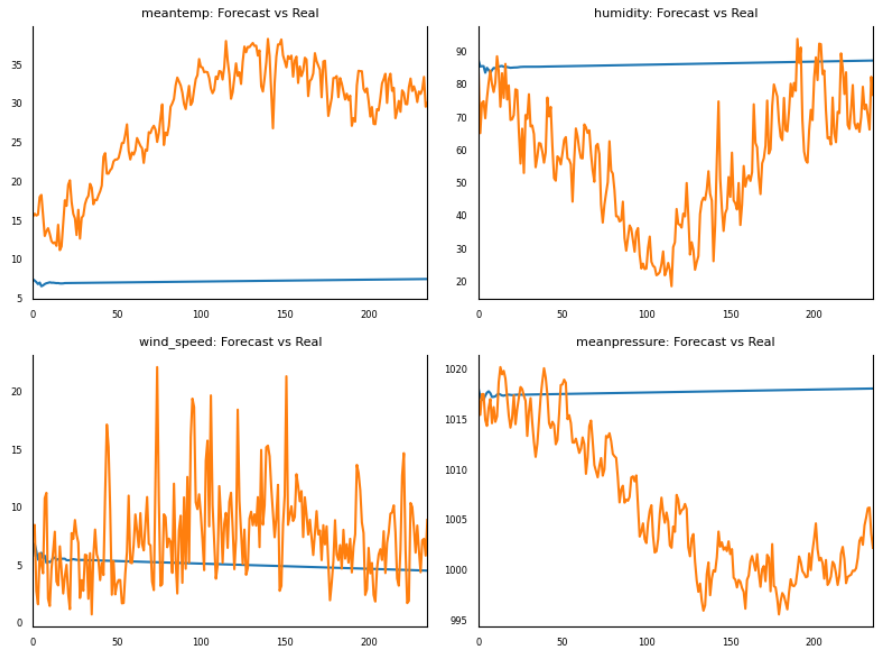

Figure 4: VAR results. Predictions vs actual values.

4.1.2 VAR model with daily training

Similarly to what was observed in the previously handed paper with univariate time-series analysis, the Vector Autoregression model improved when it was re-trained daily with the ground truth of the days it has already predicted. In other words, the model is retrained each day with the actual values of the eleven previous days.

A clear improvement can be observed in Figure 5 compared to Figure 4. This model is capable of correctly following general trends but it fails to predict accurate day-to-day changes. As can be observed, it predicts *meantemp* quite well and has more difficulties with *meanpressure*, *humidity*, and *wind_speed*. Table 5 confirms the improvement with numbers.

Finally, the same model without differencing the attributes was even better. This is strange as one would expect it to work worse knowing that stationarity is one of the assumptions VAR makes on the data, but in this case, it seems it was not decisive.

The results in Figure 6 and Table 6 visually and numerically prove the improvement.

For these models, the condition of no serial correlation in the residuals was again checked with the Durbin-Watson statistic, and it showed no significant autocorrelation in both cases.

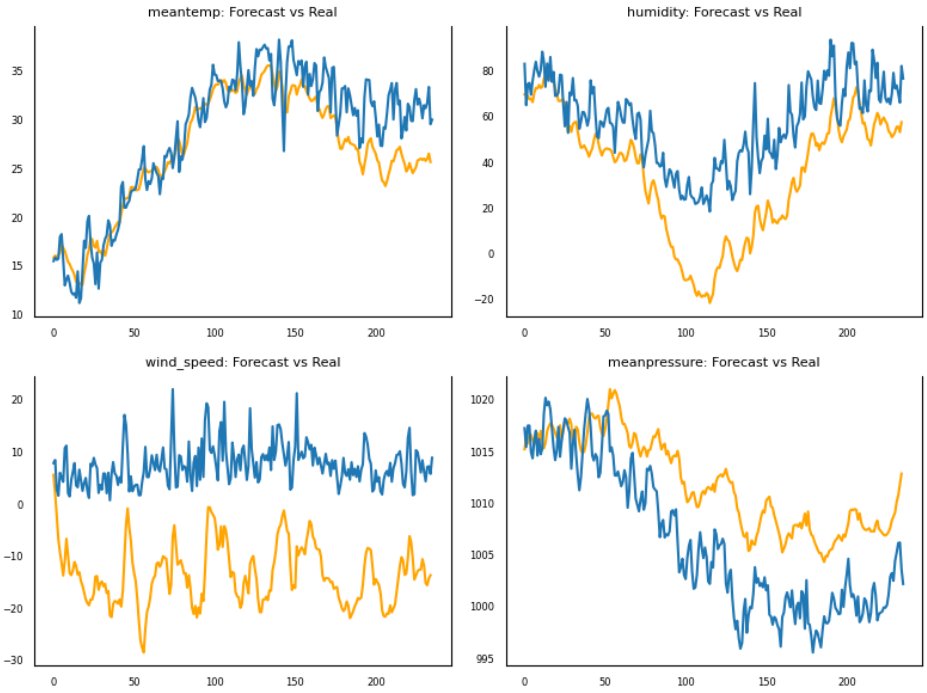

Figure 5: VAR results. Predictions vs actual values.

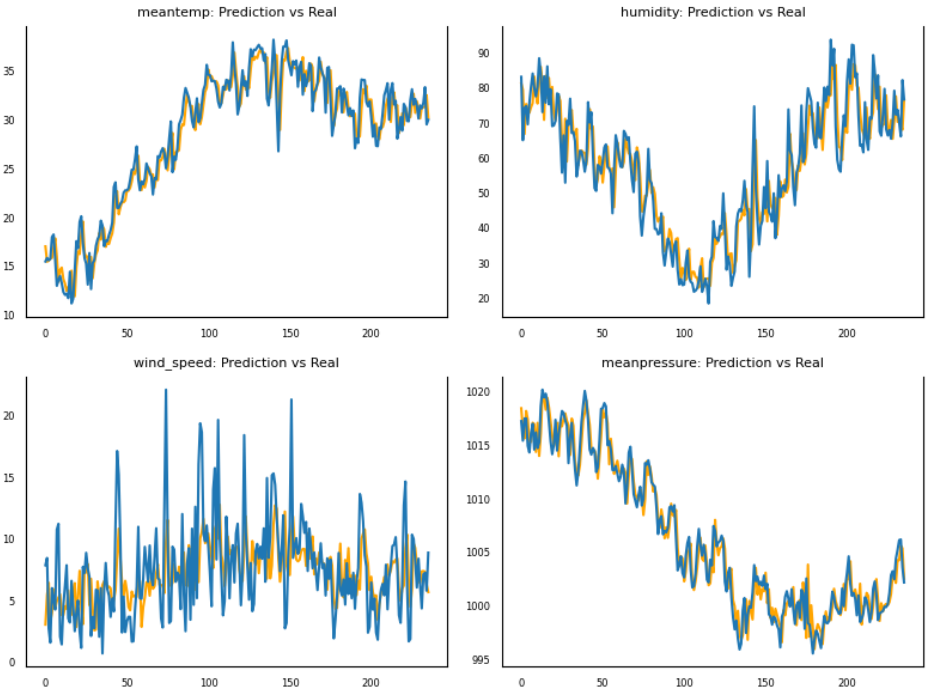

Figure 6: VAR results. Predictions vs actual values.

Attribute	MSE	RMSE
meantemp	11.82	3.44
humidity	484.21	22.00
wind_speed	734.95	27.11
meanpressure	45.63	6.75

Table 5: MSE and RMSE per attribute.

Attribute	MSE	RMSE
meantemp	2.93	1.71
humidity	11.14	3.34
wind_speed	51.92	7.21
meanpressure	2.33	1.53

Table 6: MSE and RMSE per attribute.

4.2 Support Vector Regression (SVR)

Support Vector Regression (SVR) is a multivariate machine learning algorithm that is used for regression tasks not necessarily related to time-series. It is based on the concept of Support Vector Machines (SVM) and aims to find a hyperplane that best fits the data points in a higher-dimensional space.

The main idea behind SVR is to map the input data into a higher-dimensional feature space using a kernel function. In this feature space, SVR finds a hyperplane that maximizes the margin between the predicted target values and the hyperplane. The predicted target values should fall within a certain distance (epsilon) from the hyperplane, allowing for some degree of error.

This algorithm is capable of handling time-series data if lagged values are incorporated as additional features. For this reason, it was applied to predict *meantemp*.

In addition, it is a robust algorithm as it can capture nonlinear relationships between the input features and the target variable. In fact, the kernel functions allow SVR to project the data into a higher-dimensional space, where complex patterns and dependencies can be modeled.

The SVR model usually predicts better when it is fed attributes corresponding to lags of the target. In other words, if one wants to predict *meantemp* he might want to include columns in the dataset where, for each day, those columns contain the temperature of the previous days. This helps the model give more importance to the previous days before the day of the prediction.

For this reason, three models were performed in this section. One which does not learn from lags, one which includes the temperature of the day before, and a final one with multiple lags. Note that these were all performed without applying differencing to data, and afterward, the effects of making the data stationary were studied.

SVR takes three parameters: the kernel function, the regularization parameter (C), and the epsilon-insensitive loss function (epsilon). In order to find the best parameterization a grid search was performed for each of the above-mentioned models. In particular, for each parameter, the following values were studied:

- **Kernel function**: linear, poly, rbf, and sigmoid
- **Regularization parameter**: 0.1, 1, and 10
- **Epsilon**: 0.1, 0.01, and 0.001

In the following Subsections, we specify the results of the grid-search for each of the models, and comment and compare their results.

4.2.1 SVR without differencing

The best parameterization for each of the models without differencing were the following:

- Without lags of *meantemp*: Radial Basis Function (rbf) as kernel function, 0.1 as regularization parameter, and 0.001 as epsilon.
- With one lag of *meantemp*: Radial Basis Function (rbf) as kernel function, 1 as regularization parameter, and 0.001 as epsilon.
- With multiple lags of *meantemp*: Radial Basis Function (rbf) as kernel function, 1 as regularization parameter, and 0.01 as epsilon.

Regarding their results, Table 7 showcases the mean squared error and the root mean squared error obtained from each of the models. It can be seen that adding one lag as input data clearly improves the model whereas adding multiple lags (in this case 7) did not prove to be as helpful. As expected, the temperature of the day before has a greater influence on the prediction than the temperature of the previous days.

Figure 7 visually shows the predictive abilities of the best model, i.e. the one with seven lags, by comparing its predictions with the real values. The model is quite good, and is able to follow the general trends of the *meantemp* attribute, but it is not particularly accurate. The plots corresponding to the other two models are not included here because they were very similar to the one in Figure 7.

Model	MSE	RMSE
Without lags	13.89	3.73
One lag	11.78	3.43
Multiple lags	11.33	3.37

Table 7: MSE and RMSE per model.

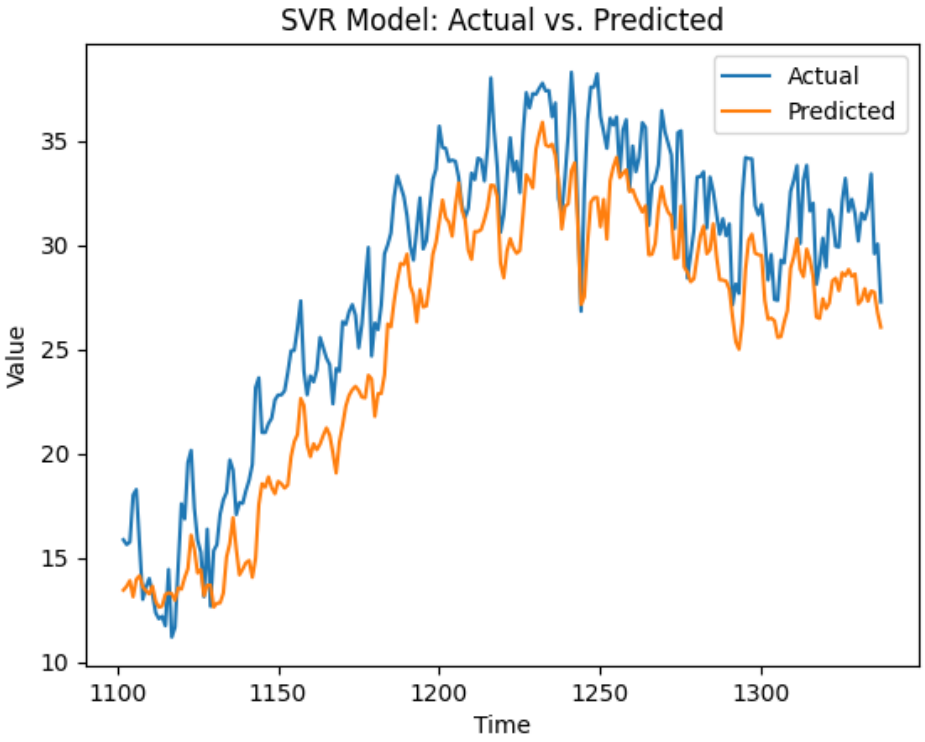

Figure 7: Results of the SVR model with 7 lags. Predictions vs actual values.

4.2.2 SVR applying differencing

Applying differencing to all the attributes in the sample did not prove to be helpful. In this case, the grid search showed the best parameters were linear kernel function, 10 as regularization parameter,

and 0.1 epsilon.

As visually proved in Figure 8 this model was less capable of following the *meantemp* trends than the SVR model with 7 lags and without differencing. In fact, its MSE and MRSE are respectively 29.95 and 5.47, which are much worse than the ones shown in Table 7.

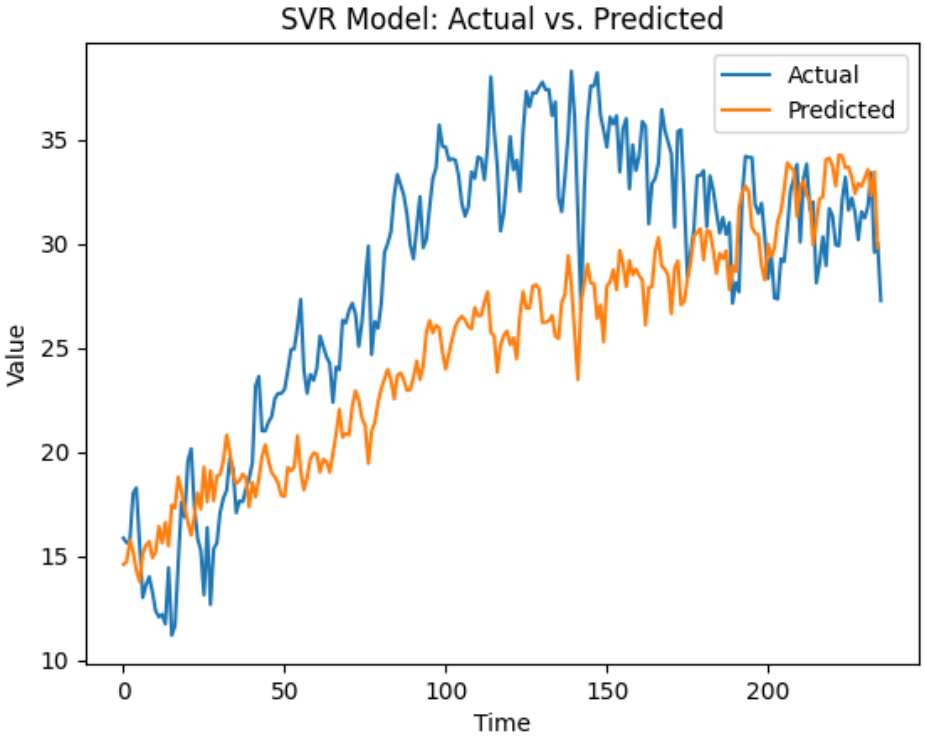

Figure 8: Results of the SVR model differencing all attributes. Predictions vs actual values.

4.3 Gradient Boosting (GB)

Gradient Boosting is an algorithm that combines multiple simple prediction models, typically decision trees, to create a strong predictive model. It works by iteratively adding weak models to the ensemble, each one focusing on reducing the errors made by the previous models. Gradient Boosting is known for its high predictive accuracy and flexibility in handling various types of data.

This algorithm naturally handles sequential dependencies in time-series data. By training on historical data and sequentially incorporating new observations, it can capture the temporal patterns and dynamics present in the multivariate time-series.

Gradient Boosting can capture complex and nonlinear relationships between the input features and the target variable in a multivariate time-series dataset. It does this by using decision trees as weak learners, which are capable of representing and capturing nonlinear interactions between variables.

Similarly to what was done with the Support Vector Regression algorithm, a grid search was applied to find the optimal hyperparameters. In this case, the hyperparameters considered were:

- **Learning Rate:** 0.1, 0.01, and 0.001
- **Number of Estimators:** 100, 200, and 300
- **Max Depth:** 3, 5, and 7

The following subsections include the hyperparameters chosen for each model and their results.

4.3.1 Without differencing

The best hyperparameters for each model without differencing are the following:

- Without lags of *meantemp*: A learning rate of 0.01, 300 estimators with a maximum depth of 3.
- With one lag of *meantemp*: A learning rate of 0.1, 100 estimators with a maximum depth of 7.
- With multiple lags of *meantemp*: A learning rate of 0.01, 300 estimators with a maximum depth of 5.

Table 8 showcases the results of the previously mentioned models. Similar to the SVR models, the model performs better with one lagged value of *meantemp* than without this variable. However, the GB's performance drops when adding multiple lags.

Figure 9 shows the predictive power of the model with one lagged value of *meantemp* by comparing the predicted values with the ground truth. This model manages to follow trends, however, it is not very accurate at detecting the exact temperature. Additionally, peaks and drops seem to be predicted a day after the actually happen. Evidently, this is understandable as the model receives the temperature from the day before as an explanatory variable.

Model	MSE	RMSE
Without lags	12.58	3.55
One lag	11.77	3.43
Multiple lags	12.50	3.54

Table 8: MSE and RMSE per model.

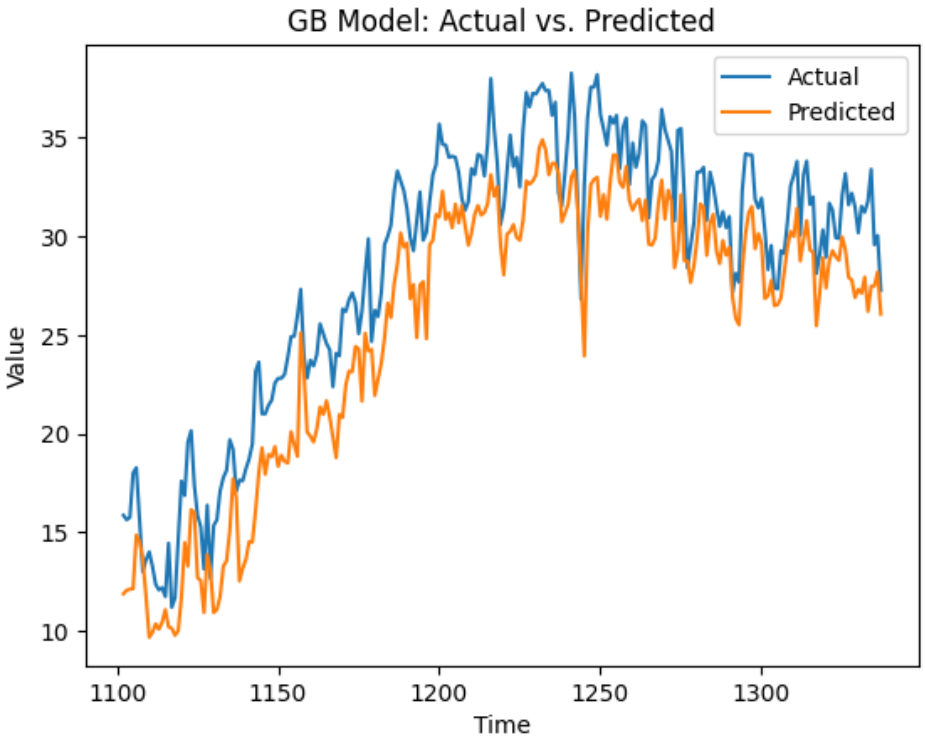

Figure 9: Results of GB with the temperature from the day before. Predictions vs actual values.

4.3.2 With Differencing

In the final experiment for GBs, the attributes *wind_speed, humidity, meanpressure,* and *meantemp_day_before* were the explanatory variables. For this approach, one degree of differencing was applied.

The grid search concluded that the best hyperparameter values were:

- **Learning Rate:** 0.1
- **Number of Estimators:** 200
- **Max Depth:** 3

Figure 10 shows a clear drop in performance compared to the other GB approaches. Nevertheless, the Gradient Boosting model using differencing follows trends better than the SVR model with differencing. The main problem this model has is the temperature values it predicts seem to be much less than the reality. This is further cemented by the MSE of 37.57 and RMSE of 6.13.

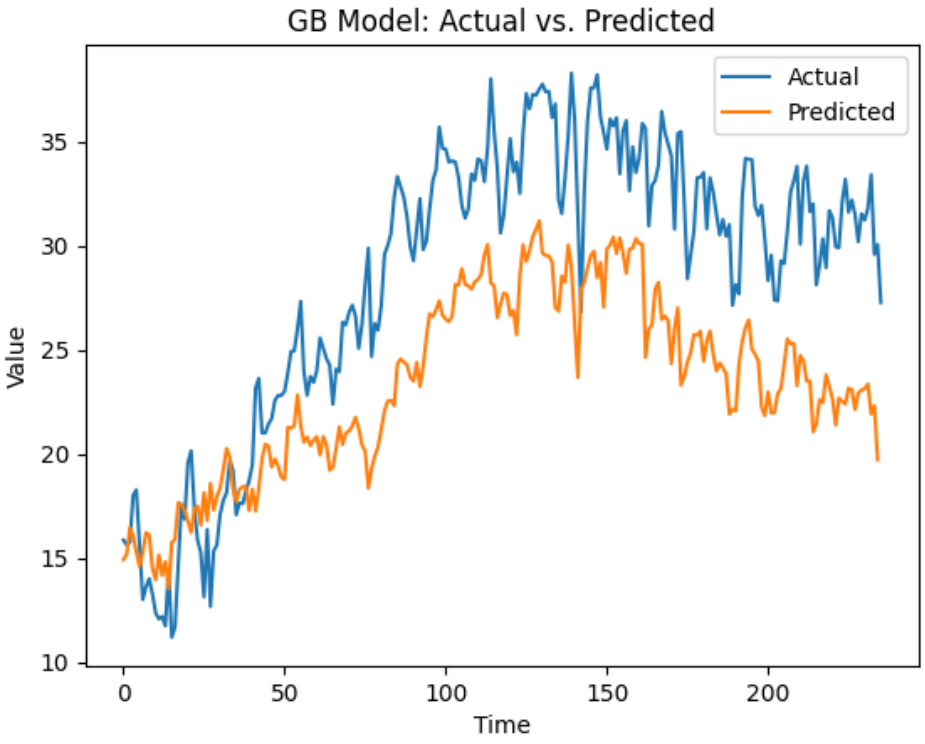

Figure 10: Results of GB with differencing. Predictions vs actual values.

4.4 Long Short-Term Memory (LSTM)

LSTMs (Long Short-Term Memory) are a type of Recurrent Neural Networks (RNNs). Now, ANNs process each input independently without maintaining a notion of time. On the other hand, LSTMs use recurrent connections to preserve information from previous time steps to use to predict current or future steps.

LSTMs use memory cells that can selectively retain or forget information. These memory cells are controlled by gate mechanisms, which are neural networks that regulate the flow of information to the memory cell. These gates include the input gate, output gate, and forget gate, which are used to control the information flow into the cell, the information flow out of the cell, and the information that is retained or forgotten in the cell, respectively.

This ability to retain past information makes LSTMs suited for time series problems. In the subsequent sections, two different models will be discussed.

4.4.1 Model with lags without sliding window

The first approach using LSTMs involved using the attributes *wind_speed, humidity, meanpressure,* and the lagged temperatures of the six previous days. This data was then fed to an LSTM with 4

units and it was trained for 45 epochs with a batch size of 1 with the Adam optimizer.

Model 1 resulted to be overfitted as it yielded an RMSE of 1.25 for the training data, whilst it produced an RMSE of 3.34 for the validation data. Figure 11 clearly shows this difference in performance between the two sets.

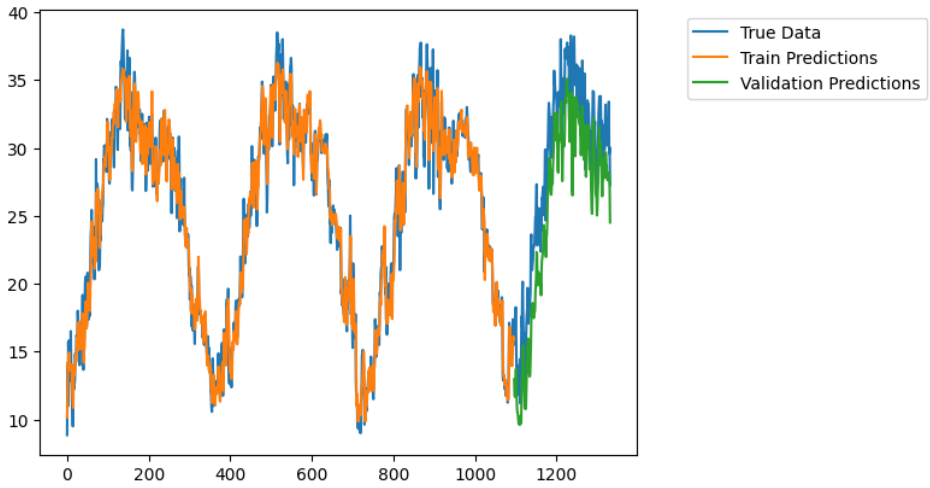

Figure 11: Results of LSTM model with no sliding window

4.4.2 Model with lags and sliding window

In the previous experiment, no sliding window was included, meaning that the power of a recurrent neural network was not being properly taken advantage of. For this approach, a sliding window of size 10 was defined. Therefore, the dataset was transformed in such a manner that the model received as input a sequence of vectors. These vectors contained the same attributes as the previous model. A grid search was performed and the following hyperparameters were considered to be the optimal:

- **Epochs:** 150
- **Batch Size:** 16
- **LSTM Units:** 32

Figure 12 showcases the performance of the model on the test set with an MSE of 2.52 and an RMSE of 1.59. Given the aspect of the graph it does not seem possible that this model outperforms other models such as VAR with daily retraining.

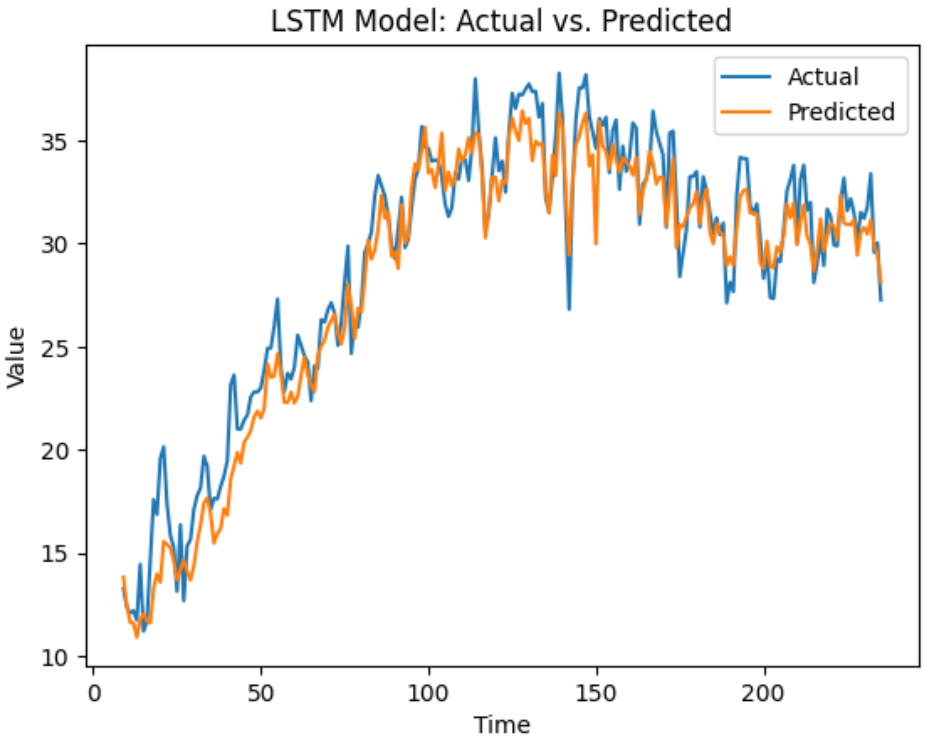

Figure 12: Results of LSTM model with sliding window of size 10

5 Testing

Of all the evaluated models, the best one was Vector Autoregression with eleven lags and re-trained daily. Knowing so, it was decided to apply it to the test dataset to see how well it performed on real unknown data.

As Figure 13 shows this model is very accurate in predicting *meantemp*. In particular, as Table 9 shows, the MSE of the *meantemp* predictions is 2.14 and the RMSE is 1.64, which is very small. In fact, these results are better than the ones found when applying the model on the validation sample.

Regarding the rest of the attributes, it also performs quite well with *meanpressure*, but has difficulties correctly predicting *wind_speed* and *humidity*.

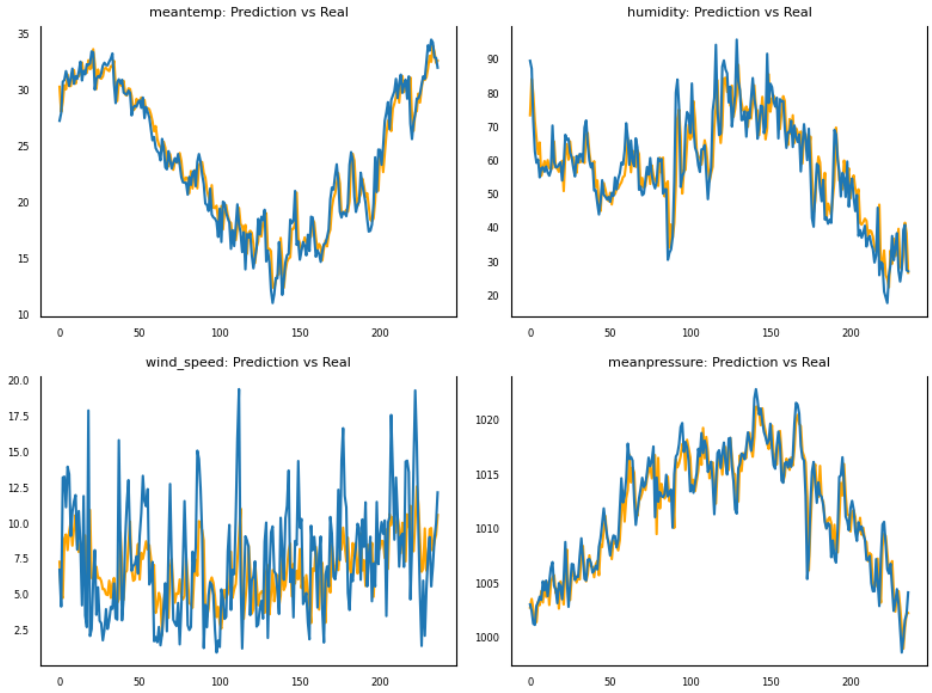

Figure 13: VAR testing results.

Attribute	MSE	RMSE
meantemp	2.14	1.46
humidity	47.87	6.92
wind_speed	12.69	3.56
meanpressure	3.01	1.73

Table 9: MSE and RMSE per attribute.

6 Conclusions

The Granger and Johansen tests provided valuable insights by demonstrating the relationship between different attributes and their ability to predict "meantemp."

The aspect that makes the VAR model so strong, retraining it daily, is also its downfall. It can only perform one day in advance, it cannot produce long-term predictions. Therefore, it is only useful if the data from the day before is available.

For the prediction of mean temperature, which is typically available daily, there is a demand for a 7-day forecast to facilitate weekly planning which cannot be provided be any of the models included in this paper.

Among the models that don't use historical data of "meantemp" for prediction, Gradient Boosting without lags or differencing can effectively capture the overall trends of the attribute. It achieves a reasonable 3.73 RMSE, compared to the best VAR model with an RMSE of 1.46. The VAR model's strength lies in its use of mean temperature history.

The final model developed in this study performs equally well as the Autoregression model trained solely on *meantemp* in the previous report, despite incorporating additional explanatory variables.