UNIVERZA V LJUBLJANI
FACULTY OF MATHEMATICS AND PHYSICS

Financial mathematics – Master's study

Regina Blagotinšek

Local prediction of weather parameters based on historical data

Project report

Mentor: Dr. Matjaž Depolli,
Supervisor: Prof. Dr. Sergio Cabello Justo

Ljubljana, June 7, 2024

Contents

Local prediction of weather parameters based on historical data

Abstract

This project developed a short-term local weather forecast historical time series data with 5-minute intervals. The main objective was fitting statistical models $ARIMA$ (Auto-Regressive Integrated Moving Average) to the data, evaluate the errors, forecast future values and test the models on new data. The analysed weather parameters were ambient temperature, solar radiation intensity, air pressure, relative humidity, wind speed, wind direction and rain intensity.

$ARIMA$ models were trained and evaluated using absolute error and other statistical metrics. These models were compared against two simpler models that forecast based on current measurements and last 3 days average measurements. The best $ARIMA$ models were then tested on new data to assess their generalization capability.

Results indicated that that $ARIMA$ models effectively captured the essential characteristics of the data and sometimes provided better forecasts than the simple models. The next step is to explore a multivariate $ARIMA$ model,but a significant improvement of the fit and forecast is not necessarily expected, since it is a more complex model.

In summary, this project validated the use of $ARIMA$ models for short-term weather forecasting. Detailed results and code are available in the linked GIT repository.

Keywords: time series, $ARIMA$, weather forecast, error evaluations, multivariate $ARIMA$

1. Introduction

This project aims to develop a short-term local weather forecast using historical data, implemented in Python. The data is in the form of time series with 5-minute intervals which was then modelled with statistical models, analysed and used for forecasting.

In the first phase, I focused on data visualization and model fitting. Historical weather data was loaded and visualized to identify patterns and trends. *ARIMA* (Auto-Regressive Integrated Moving Average) models were employed to fit the data. Additionally, two simple models were implemented for comparison which are discussed later. Model performance was assessed using absolute error metrics and some other statistical tests.

The training involved fitting *ARIMA* models on a subset of the data. Following this, I forecast weather parameters for one day (24 steps). Then I evaluated the forecasts by comparing predicted values with actual observations.

Finally, in the testing phase, I applied the best developed model to new data to evaluate their generalization capability. The performance was tested using the error analytics that assessed how well the models fit the new data and the accuracy of their forecasts.

Here is the link to my GIT repository.

2. Theoretical Background

2.1. *ARIMA* (Auto-Regressive Integrated Moving Average).

ARIMA (Auto-Regressive Integrated Moving Average) is a statistical analysis model used to forecast future points in time series data. It combines three main components: auto-regression (*AR*), differencing (*I*), and moving average (*MA*).

2.1.1. *Components of ARIMA.*

(1) **Auto-regressive (*AR*) Component** : The AR part of the model specifies that the output variable depends linearly on its own previous values. The auto-regressive model of order p ($AR(p)$) can be written as:

$$X_t = c + \phi_1 X_{t-1} + \phi_2 X_{t-2} + \cdots + \phi_p X_{t-p} + \epsilon_t$$

where X_t is the value at time t, c is a constant, $\phi_1, \phi_2, \ldots, \phi_p$ are the parameters of the model, and ϵ_t is white noise.

(2) **Integrated (*I*) Component** : The I part of *ARIMA* indicates that the data values have been replaced with the difference between their values and the previous values to make the series stationary. The differencing of the series can be written as:

$$Y_t = X_t - X_{t-1}$$

where Y_t is the differenced series. If the series becomes stationary after differencing d times, the series is said to be integrated of order d (I(d)).

(3) **Moving Average (*MA*) Component** : The MA part of the model incorporates the dependency between an observation and a residual error from a moving average model applied to lagged observations. The moving average model of order q ($MA(q)$) can be written as:

$$X_t = c + \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \cdots + \theta_q \epsilon_{t-q}$$

where $\theta_1, \theta_2, \ldots, \theta_q$ are the parameters of the model.

2.1.2. *AIC and BIC.* For finding the best model fit, the algotithems I used were finding the best model fit by optimizing the AIC and BIC values.

(1) **Akaike Information Criterion (AIC)** AIC measures the relative quality of statistical models for a given dataset. It is a form of penalized likelihood, where lower values indicate a better-fitting model. AIC balances goodness of fit and model complexity by penalizing the number of estimated parameters. The formula for AIC is
$$AIC = 2k - 2log(L),$$
where k is the number of parameters and L is the maximum likelihood of the model. The AIC finds the best balance between a good fit and model simplicity, avoiding overfitting.

(2) **Bayesian Information Criterion (BIC)** BIC is similar to AIC but includes a stronger penalty for the number of parameters, favoring simpler models. BIC is derived from Bayesian principles and aims to find the model that best approximates the true model. The formula for BIC is
$$BIC = klog(n) - 2log(L),$$
where k is the number of parameters, n is the number of data points and L is the maximum likelihood of the model.

2.1.3. *Stationarity and Differencing.* Stationarity indicates that the statistical properties of the data do not change over time (the time series does not have a time-dependent structure). Stationarity is an essential property for many time series models, including $ARIMA$, as it ensures that the relationships observed in the historical data can be generalized to future data points.

Checking for Stationarity : To determine if a time series is stationary, we can use tests like Augmented Dickey-Fuller (ADF) and Kwiatkowski-Phillips-Schmidt-Shin ($KPSS$). The ADF test aims to reject the null hypothesis that the given time-series data is non-stationary. It calculates the p-value and compares it with a threshold value or significance level of 0.05. If the p-value is less than this level, then the data is stationary, otherwise the differencing order is incremented by one. The following table can be used for the stationarity check.

ADF **Test**	$KPSS$ **Test**	**Conclusion**
Yes	Yes	Stationary
No	No	Non-stationary
No	Yes	Trend stationary
Yes	No	Difference stationary

Table 1. Stationarity determination using ADF and $KPSS$ tests

Differencing : If a time series is not stationary, differencing can be used to transform it into a stationary series. Differencing involves subtracting the current value of the series from the previous value:
$$Y_t = X_t - X_{t-1}$$
where Y_t is the differenced series. If the series is still not stationary after the first difference, additional differencing may be required. The number of differencing steps needed to achieve stationarity is denoted by d in the $ARIMA$ model.

2.1.4. *Auto-correlation Function (ACF) and Partial Auto-correlation Function ($PACF$).*

- **Auto-correlation Function (ACF)** The ACF measures the correlation between a time series and its lagged values. It helps in identifying the moving average (MA) order q in $ARIMA$ models by showing the correlations between a series and its past values at various lags.
- **Partial Auto-correlation Function ($PACF$)** The $PACF$ measures the correlation between a time series and its lagged values while controlling for the values of the time intervals in between. It is useful for identifying the auto-regressive (AR) order p in $ARIMA$ models by showing the direct effect of past values on the series without the influence of intervening lags.

2.1.5. *Residuals.* The residuals in a time series model are what is left over after fitting a model. The residuals are equal to the difference between the observations and the corresponding fitted values. Residuals are useful in checking whether a model has adequately captured the information in the data. A good forecasting method will have residuals with the following properties:

(1) The residuals are uncorrelated. If there are correlations between residuals, then there is information left in the residuals which should be used in computing forecasts.
(2) The residuals have zero mean. If the residuals have a mean other than zero, then the forecasts are biased.

The first property is checked with Ljung-Box. The Ljung-Box test uses the following hypotheses:

$H0$: The residuals are independently distributed.

HA: The residuals are not independently distributed; they exhibit serial correlation.

Ideally, we would like to fail to reject the null hypothesis. That is, we would like to see the p-value of the test be greater than 0.05 because this means the residuals for our time series model are independent.

3. Methodology and Implementation

3.1. **Data.** The data I worked with is in the form of time series with 5 minute time steps of weather parameters' measurements. The parameters are ambient temperature, solar radiation intensity, air pressure, relative humidity, wind speed, wind direction and rain intensity. To reduce the dimension of the data set, I aggregated the 5 minute time steps into 1 hour time steps by taking the hourly means.There were also some missing measurements and since the $ARIMA$ model requires equal time intervals I filled the gaps by taking the next available measurement.

3.2. $ARIMA$ **model fitting and evaluating.** For each of the parameters I was searching for the best $ARIMA(p, d, q)$ with two different methods. First one is with the function *auto_arima* function from the *pmdarima* Python library. It searches through a range of potential models and selects the best one based on the AIC value. The second method was also done by searching for the optimal p, d and q parameters and checking AIC and BIC values.

3.3. **Errors.** There are many ways to evaluate the errors of the model. In case of modelling time series, investigating the residuals is important. I looked at ACF and $PACF$ plots of all the $ARIMA$ models. A good way to compare the models is also to compare the histograms of their residuals. I also considered the absolute errors and the mean absolute error. Another insightful comparison, especially for the solar intensity radiation, was hourly absolute error for one day, since there is considerably less solar radiation in the night, the error in the night should be lower than the error during the daytime. For evaluating the fitted $ARIMA$ models, I compared them to two simple models and compared their forecasts, which will be explained in the next subsection.

3.4. **Forecast.** I did the forecasts of all the weather parameters for one day (24 steps). I compared the absolute errors of $ARIMA$ models' predictions with the actual measurements and two simple models' predictions. The first simple model forecasts the tomorrow's weather with today's measurements. The second simple model forecast the next day's weather with the average of the last 3 days' measurements. The $ARIMA$ model is considered 'good' if it is predicting better or at least closely to this two simple models.

3.5. **Testing on new data.** The final way of evaluating the fitted $ARIMA$ models was testing them on new data. New data was again measurements of the seven weather parameters in 5 minute intervals. I aggregated it to hourly values and fit the suggested $ARIMA$ models to it. For each parameter I tested the better $ARIMA$ fitted to the first data to see if it also fits well here. I evaluated that by checking the AIC and BIC values, calculating the absolute errors and mean absolute error of fitted values and actual values and also by testing the forecasts. The forecasts were tested the similar way as in the previous step (two simple models, $ARIMA$ model and actual values).

4. Results

In this chapter I will make an overview of all the weather parameters and lastly a multivariate model for all of them combined. For purposed of this report, I will only present the main results of my work for two parameters - ambient temperature and solar radiation. Detailed results for all the parameters can be found in the GIT repository of this project.

4.1. **Ambient temperature.** The plot on the left side below is of the data in 5 minute time intervals and on the right side is are the average hourly measurements.

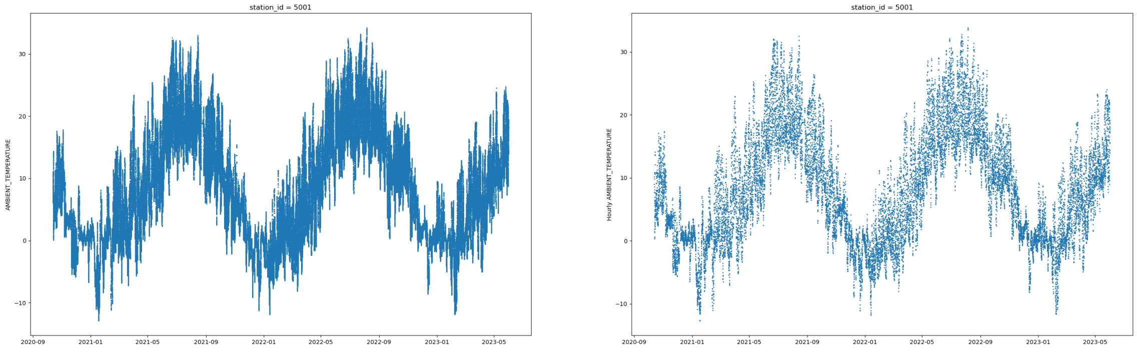

Figure 1. Ambient temperature

The best $ARIMA$ models for ambient temperature are $ARIMA(2,1,5)$ and $ARIMA(4,1,2)$. We will compare them by some statistical metrics and later with two simple models.

Based on AIC, $ARIMA(4,1,2)$ is preferred. The value is $AIC = 38849.130$. Let's look at the auto-correlation and partial-auto-correlation plots of the residuals of model $ARIMA(4,1,2)$.

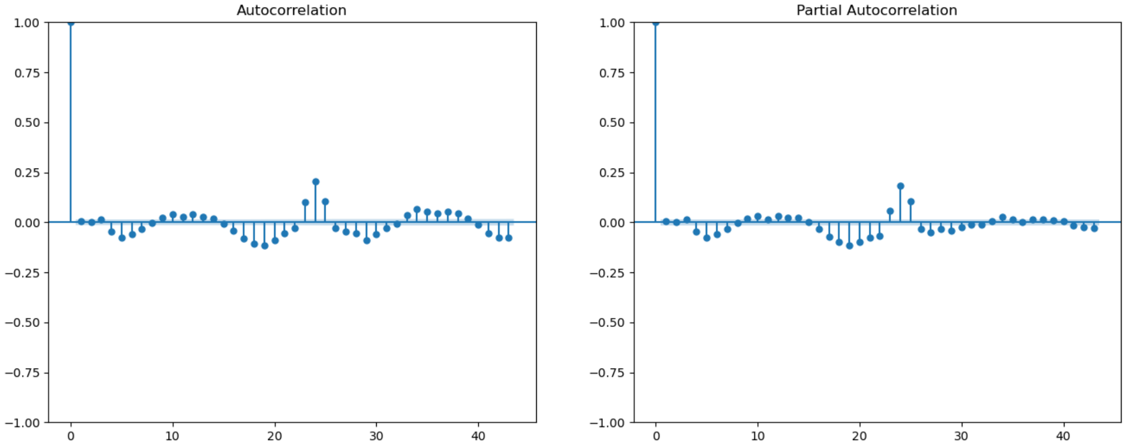

Figure 2. ACF and $PACF$ of $ARIMA(4,1,2)$ model residuals

The mean of the residuals is 0.000211676, which is close enough to 0 and the result of the Ljung-Box test is 0.28, which means the residuals are uncorrelated. Based on the residuals, the model successfully captured the information of the data.

The evaluation of the model continues with checking the absolute errors. The mean absolute error of the $ARIMA(4,1,2)$ is $0.447°C$.

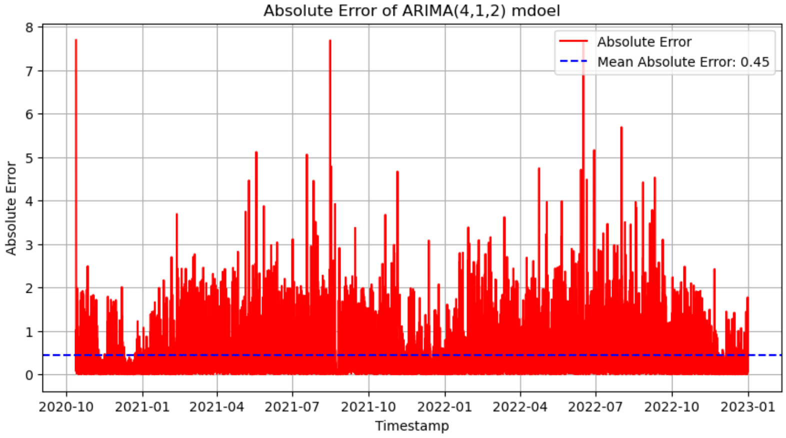

Figure 3. Absolute error of the $ARIMA$ model

Now let's investigate the results of the forecast. We will look at the forecast of the next day (24 hours) and calculate the absolute errors and the mean absolute error of each of the four models. The results are seen on the plots below.

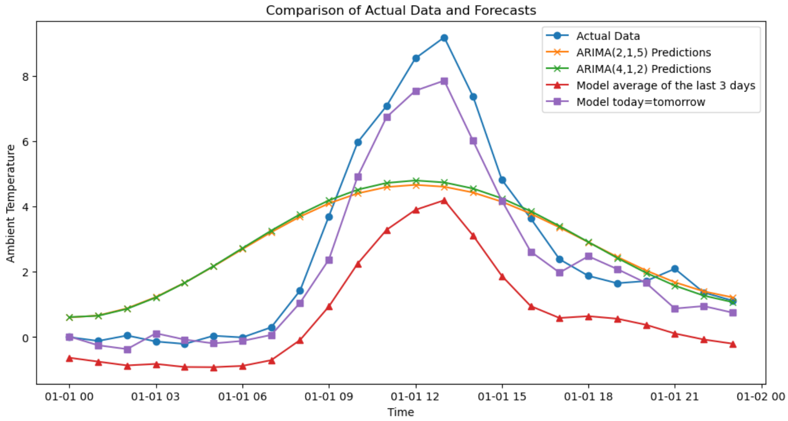

Figure 4. Models' forecasts and actual data

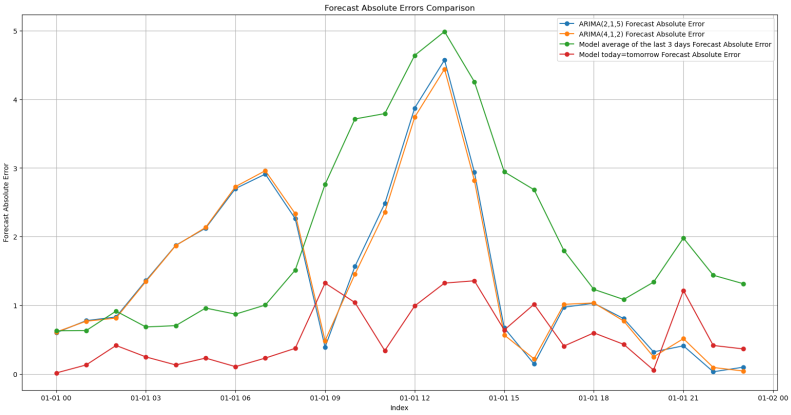

Figure 5. Absolute error of the models' forecasts

Model	**Forecast Absolute Error**$[°C]$
$ARIMA(2,1,5)$	1.490556
$ARIMA(4,1,2)$	1.474775
Model average of the last 3 days	1.995486
Model today=tomorrow	0.560069

Table 2. Forecast Absolute Errors

According to the forecast absolute errors, both $ARIMA$ models are better than the simple model that forecasts the value as the average of the last 3 days at the same time hour, but worse than the simple model, that forecasts the tomorrow's values with today's measurements.

The final step of model fit assessment is testing it on a new set of data. The mean absolute error of fitted new data compared to new actual measurements is $0.531°C$ which is close to the error of the training data $(0.447°C)$. The results of the 24-hour forecast are in the table below.

Model	**Forecast Absolute Error (new data) [°C]**
$ARIMA(4,1,2)$	2.413916
Model average of the last 3 days	1.777431
Model today=tomorrow	1.952431

Table 3. Forecast Absolute Errors (new data)

4.2. **Solar radiation intensity.** The plot on the left side below is of the data in 5 minute time intervals and on the right side is are the average hourly measurements.

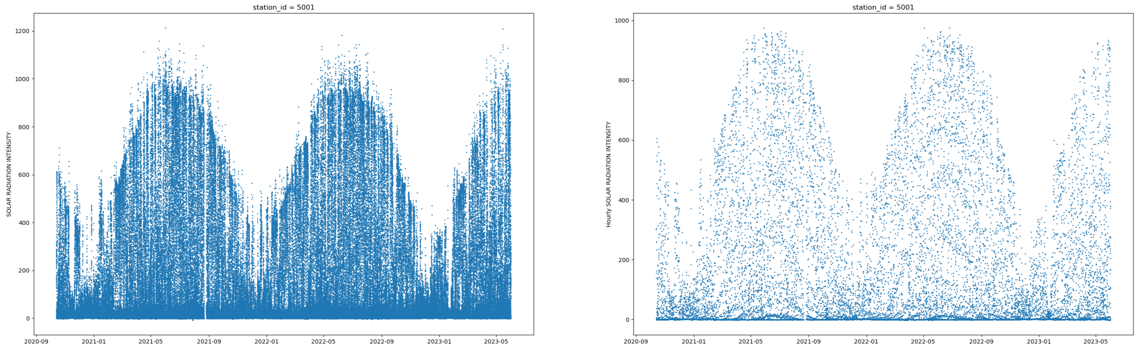

Figure 6. Solar radiation intensity

The best $ARIMA$ models for solar radiation intensity are $ARIMA(3,1,2)$ and $ARIMA(2,1,4)$. We will compare them by some statistical metrics and later with two simple models.

Based on AIC, $ARIMA(2,1,4)$ is preferred. The value is $AIC = 222463.865$. Let's look at the auto-correlation and partial-auto-correlation plots of the residuals of model $ARIMA(2,1,4)$.

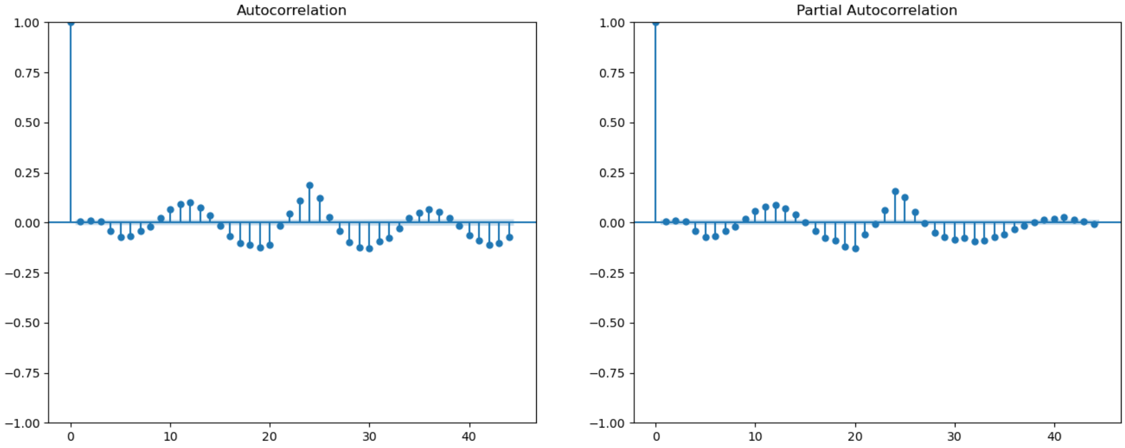

Figure 7. ACF and $PACF$ of $ARIMA(2,1,4)$ model residuals

The mean of the residuals is -0.001334192, which is close enough to 0 and the result of the Ljung-Box test is 0.12, which means the residuals are uncorrelated. Based on the residuals, the model successfully captured the information of the data.

The evaluation of the model continues with checking the absolute errors. The mean absolute error of the $ARIMA(2,1,4)$ is $44.602\frac{W}{m^2}$.

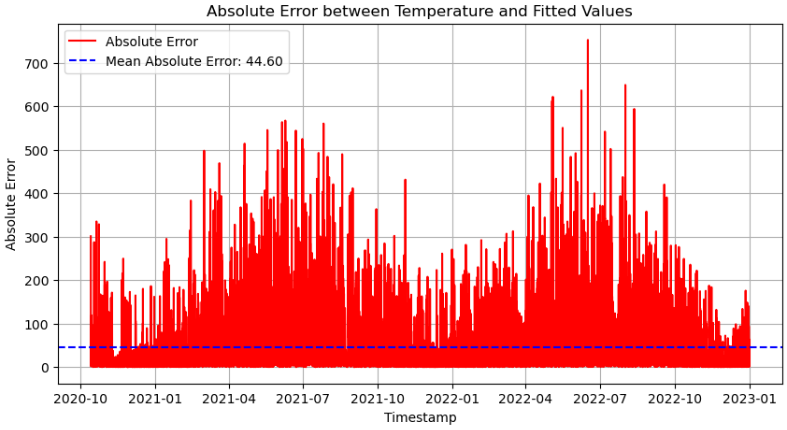

FIGURE 8. Absolute error of the $ARIMA$ model

In case of solar radiation intensity, another interesting plot is the average hourly absolute error. Since the radiation is significantly lower in the night, the average absolute errors should also be lower. This is displayed in the plot below.

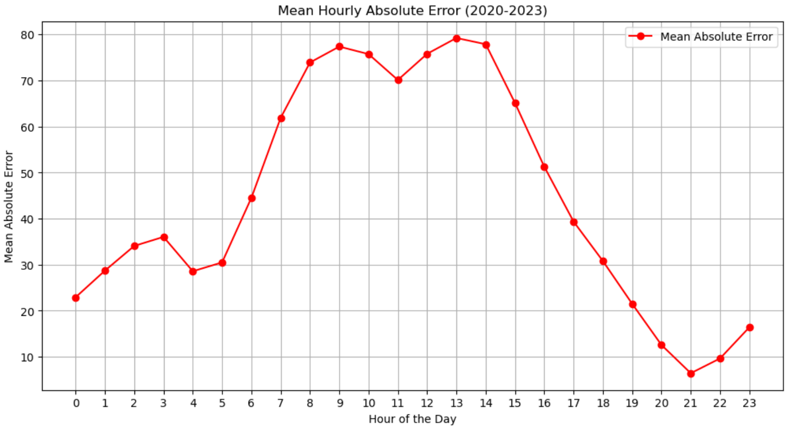

FIGURE 9. Absolute error of the $ARIMA$ model

Now let's investigate the results of the forecast. We will look at the forecast of the next day (24 hours) and calculate the absolute errors and the mean absolute error of each of the 4 models. The results are seen in the plot and table below.

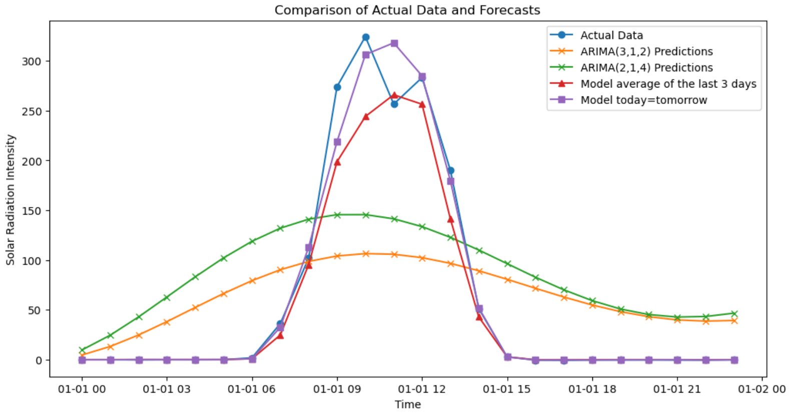

Figure 10. Models' forecasts and actual data

Model	**Forecast Absolute Error** $[\frac{W}{m^2}]$
$ARIMA(3,1,2)$	69.423578
$ARIMA(2,1,4)$	75.599554
Model average of the last 3 days	11.315625
Model today=tomorrow	6.901389

Table 4. Forecast Absolute Errors

According to the forecast absolute errors, both $ARIMA$ models are worse than both the simple models.

The $ARIMA$ fits were further assessed in the final step, where they were tested on a new dataset.The mean absolute error of fitted new data compared to new actual measurements is $44.529\frac{W}{m^2}$ which is close to the error of the training data $(44.602\frac{W}{m^2})$. The results of the 24-hour forecast are in the table below.

Model	**Forecast Absolute Error** $[\frac{W}{m^2}]$
$ARIMA(2,1,4)$	77.652954
Model average of the last 3 days	21.699653
Model today=tomorrow	9.393403

Table 5. Forecast Absolute Errors (new data)

5. VAR model

In the previous section I presented the results of modelling single-variate $ARIMA$ models. The next step is creating a multivariate model, that takes measurements of all the parameters and takes into account, that they influence each other. I choose the VAR (vector auto-regression) model. Vector auto-regression is a multivariate forecasting algorithm that is used when more time series influence each other. Since it is a much more complex model (higher dimensional dataset), it is not necessarily expected to fit the data better and give the multivariate forecast that would be better as forecasts from each respective model combined together. I trained it with the data and compare the fits and predictions to the $ARIMA$ models, two simple models and actual data. The results are in my repository.

6. Further research proposal

The data that was used to train the models were measurements of weather parameters in 5 minute intervals for 2 years. Training the models with a longer time series would definitely bring better results. I choose the suggested $ARIMA$ model, that could be improved for example by $SARIMA$ model, that also has a seasonal component or another statistical model. In the data, there is some white noise for example from faulty meteorological instruments. The white noise can be detected and removed while training the models. I had a testing data set and I tested the results, a step further would be retraining them with new data and testing them on a third dataset. The multivariate part of the research could be improved by using the multivariate $ARIMA$ instead of VAR model, because it is more complex, but at the same time that might be a problem, as more complex models do not always deliver better results.

References

[1] Bojan Basrak, *ESSENTIALS OF TIME SERIES (Lecture notes), with examples in R*, Department of Mathematics, University of Zagreb.

[2] Hyndman, R.J., & Athanasopoulos, G., *Forecasting: principles and practice, 3rd edition*, OTexts: Melbourne, Australia. OTexts.com/fpp3. Accessed on 20. 5. 2024.

[3] Param Raval , *Vector auto-regressive model in Python*,`https://www.analyticsvidhya.com/blog/2021/08/vector-autoregressive-model-in-python/`, Accessed on 1. 6. 2024.

[4] Rabi Kumar Singh , *Auto ARIMA on Multivariate Time Series*, `https://www.kaggle.com/code/jurk06/auto-arima-on-multivariate-time-series`, Accessed on 20. 5. 2024.

[5] Robert Nau , *Statistical forecasting: notes on regression and time series analysis*, Fuqua School of Business Duke University, `https://people.duke.edu/~rnau/411home.htm`, Accessed on 20. 5. 2024.

[6] Shahanaj Parvin and Murshida Khanam, *Comparison Between ARIMA and VAR Model Regarding the Forecasting of the Price of Jute Goods in Bangladesh.*

[7] Param Raval , *How to Build ARIMA Model in Python for time series forecasting?*,`https://www.projectpro.io/article/how-to-build-arima-model-in-python/544`, Accessed on 20. 5. 2024.