ADVANCED DATA ANALYSIS AND MACHINE LEARNING

CLIMATE A2 - INVESTIGATE THE FORECASTING POWER FOR PREDICTING HUMIDITY DATA USING MULTIVARIATE DATA

LUT University

2025

Group: Amin Hassanzadehmoghaddam, Antonio Oliva, Nico Niemelä

https://github.com/AntoniooOliva/Climate-A2

LUT Group Confidential - Other information (3Y)

Content

LUT Group Confidential - Other information (3Y)

1 Visualization of the data and initial exploratory analysis

The dataset contains 1,462 daily observations of weather parameters including mean temperature, humidity, wind speed, and mean pressure from beginning of 2013 to the beginning of 2017.

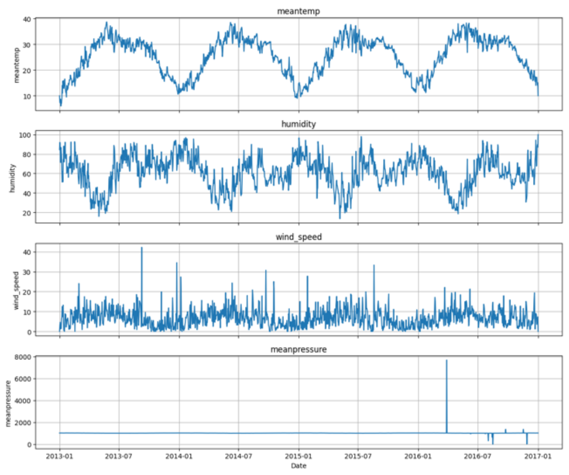

Figure 1. Data visualization

- Mean Temperature:

  The average temperature is around 25.5, with a minimum of 6 and a maximum of 38.7. The standard deviation (7.35) indicates moderate variability and clear seasonal fluctuations typical of annual temperature cycles. The cycles peak in summer and dipping in winter consistently each year.
- Humidity:

  Humidity averages 60.8 and its range is from 13.4 to 100. This large range suggests both dry and humid periods throughout the year. The plot shows that humidity follows an inverse seasonal pattern, peaking during cooler or monsoon periods.
- Wind Speed:

  The average wind speed is 6.8, with a widespread up to 42.2. This indicates occasional strong wind events or storms, and most days remain relatively calm. This parameter is highly variable but doesn’t show strong seasonality.
- Mean Pressure:

  The average pressure is 1011, but the unusually large standard deviation (180.2) and maximum value (7679) indicate the presence of outliers.

There are no missing values across any columns, ensuring data completeness for analysis.

2 Time-series decomposition analysis of long-term trend, seasonality, and residuals

For doing the decomposition analysis, because the data consists of multiple years range, 365 days are considered as period. The trend component starts around 68 humidity in early 2013 and declines to about 58 by end of 2016, indicating a steady long-term decrease of about 15% in average humidity over four years.

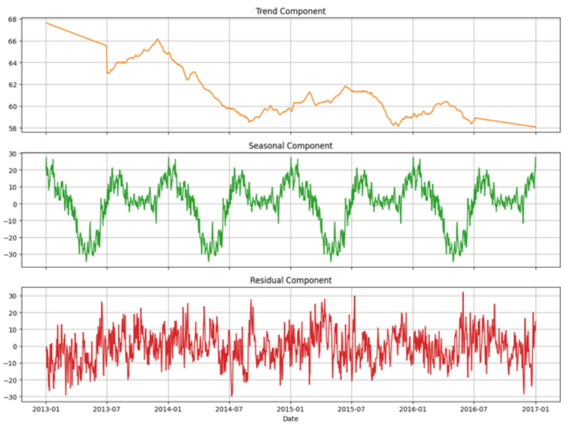

Figure 2. Long term decomposition on Humidity variable

The seasonal component oscillates between approximately 25 and –30, showing a clear annual cycle. Humidity rises during the monsoon season (June–September) and drops during dry winter months (November–February).

The residual component fluctuates within ±30 and reflects short-term daily variations due to weather anomalies, local storms, or measurement noise.

3 An autocorrelation analysis of the dataset

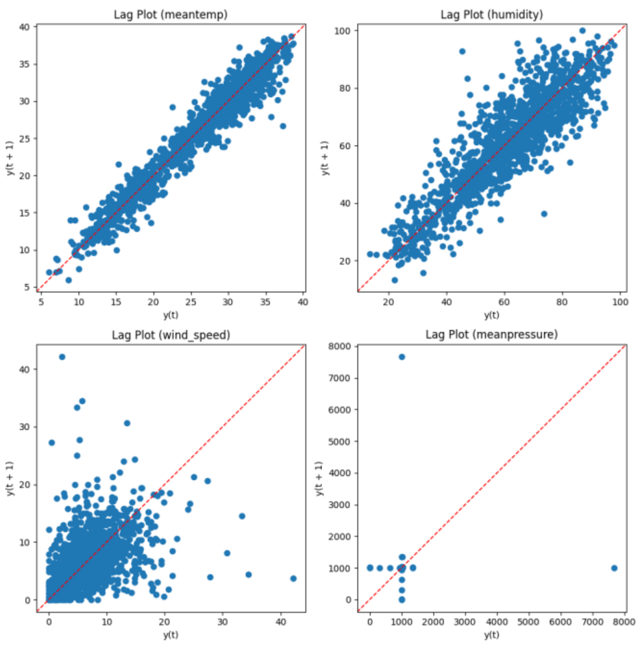

Figure 3. Lag plots for each feature

Mean temperature

	t	*t+1*	*t+2*	*t+3*
t	*1*	*0.974*	*0.956*	*0.942*
t+1	*0.974*	*1*	*0.974*	*0.956*
t+2	*0.956*	*0.974*	*1*	*0.974*
t+3	*0.942*	*0.956*	*0.974*	*1*

Humidity

	t	*t+1*	*t+2*	*t+3*
t	*1*	*0.878*	*0.783*	*0.718*
t+1	*0.878*	*1*	*0.878*	*0.783*
t+2	*0.783*	*0.878*	*1*	*0.878*
t+3	*0.718*	*0.783*	*0.878*	*1*

Wind Speed

	t	*t+1*	*t+2*	*t+3*
t	*1*	*0.436*	*0.223*	*0.171*
t+1	*0.436*	*1*	*0.436*	*0.223*
t+2	*0.223*	*0.436*	*1*	*0.436*
t+3	*0.171*	*0.223*	*0.436*	*1*

Mean pressure

	t	*t+1*	*t+2*	*t+3*
t	*1*	*0.003*	*0.011*	*0.002*
t+1	*0.003*	*1*	*0.003*	*0.011*
t+2	*0.011*	*0.003*	*1*	*0.003*
t+3	*0.002*	*0.011*	*0.003*	*1*

Table 1. Correlation matrixes for each feature

As the lag plots in Figure 3 and the correlation matrixes in Table 1 show, there is a strong autocorrelation between the values y(t) and y(t+1) in mean temperature (~ 0.974), and humidity (~ 0.878), with t representing days.

For wind speed, as expected, the correlation is not so strong, meaning the measurements of a particular day do not influence much the measurement of the following day (~ 0.437).

The mean pressure behaves differently. The lag plot reveals a strong outlier (7679) that makes the correlation difficult to analyse from the plot. There are also outliers closer to the normal behaviour of the data, and all outliers in the datasets cause a near 0 correlation between the measurements.

Just to obtain a more meaningful analysis, outliers were removed and the calculations were repeated.

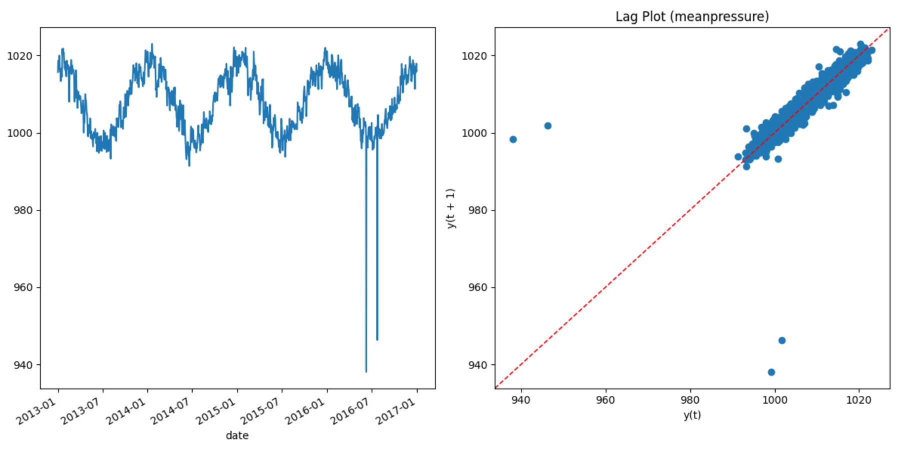

Figure 4. New lag plot for "mean pressure"

Mean pressure				
	t	*t+1*	*t+2*	*t+3*
t	*1*	*0.902*	*0.873*	*0.859*
t+1	*0.902*	*1*	*0.902*	*0.873*
t+2	*0.873*	*0.902*	*1*	*0.902*
t+3	*0.859*	*0.873*	*0.902*	*1*

Table 2. New correlation matrix for "mean pressure"

After removing most outliers, the analysis is clearer: in the mean pressure feature column, there is a strong correlation between the values (~ 0.902), behaving like a sinusoidal wave, within the range 995-1200. It is possible to conclude that these strong outliers can complicate further analyses, and a proper way to handle these values must be found.

4 Plan for partitioning the time-series data for model formation

To prevent data leakage and preserve temporal integrity, the time-series data must be partitioned so that the training data occurs before the test data in time. One method that can be used is regular Day Forward Chaining where the data is split into many different training, validation and test sets. (Chess, S. 2020) When using this method, we successively consider each day of the data set as the test set and assign all previous data into the training set. Concept of the plan is shown in table 3.

Fold	Training period	Validation period	Testing period
1	Day 1	Day 2	Day 3
2	Day 1 ➔ Day 2	Day 3	Day 4
3	Day 1 ➔ Day 3	Day 4	Day 5
4	Day 1 ➔ Day 4	Day 5	Day 6
m	Day 1 ➔ Day m	Day m+1	Day m+2

Table 3. Conceptualization of the data partitioning plan.

It should be noted that the original data partitioning plan might need to be revisited if it is computationally too expensive.

5 Data Frequency, synchronization, and missing value assessment

The dataset is checked for continuity and sampling frequency by examining the time differences between consecutive timestamps. The results illustrate that all observations are exactly one day apart. This confirms that the data is collected at a consistent daily frequency with no missing dates, so no resampling or interpolation procedures are necessary to standardize the sampling rate. Next, the variables are checked for synchrony by verifying that each column contains complete data for every timestamp. All variables show zero missing values, which shows that they are perfectly aligned and share the same time index. Therefore, the dataset is already synchronous across all variables, and no additional alignment or reindexing procedures are required.

6 STL Decomposition for outliers' elimination

A seasonal decomposition (STL) was performed on all columns of the dataset to investigate potential outliers. The range of the residuals for each feature indicates if there are extreme values in the data:

- For mean temperature, humidity, and wind speed, the residuals range between approximately -30 and 30. This shows that there are no extreme outliers that require special handling.
- For mean pressure, there is a very large spike, with a value of 7679.33 on 2016-03-28, which clearly represents an outlier.

To address outliers, an STL decomposition was applied:

1. Outliers are identified based on the residuals.
2. The outlier values are then set to NaN.
3. Missing values (NaN) are replaced using interpolation, producing a cleaned time series while preserving the overall trend and seasonal pattern.

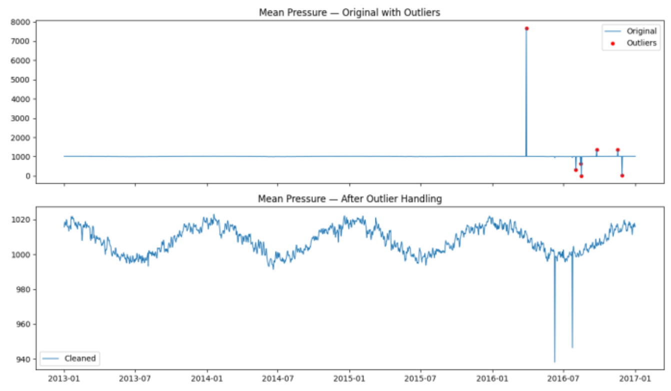

Figure 5. Mean pressure outliers' detection

7 Literature review on dividing a long time-series into sub-samplings

According to Bergmeir et. al, there are four main methods to do subsequencing when dealing with long time-series. The fixed-origin, rolling-origin-recalibration, rolling-origin-update, and rolling window evaluation. Differences between these methods are explained below. (Bergmeir, C. & Benítez, J. M. 2012).

Fixed-origin evaluation

Fixed-origin evaluation is commonly used in forecasting competitions. In this approach, forecasts for each value in the test set are generated solely based on the training set, with the forecast origin fixed at the last point of the training period. This means that for each forecasting horizon, only one forecast is produced. However, this method has some notable drawbacks: the choice of the forecast origin can significantly impact the evaluation results, and since only one forecast is available for each horizon, averaging over multiple forecasts within a series or horizon is not feasible. (Bergmeir, C. & Benítez, J. M. 2012).

Rolling-origin evaluation

In rolling-origin recalibration evaluation, forecasts for a fixed horizon are made by progressively moving data points from the test set into the training set, adjusting the forecast origin as needed. For each forecast, the model is recalibrated using the updated training data, which typically involves retraining the model from scratch. (Bergmeir, C. & Benítez, J. M. 2012).

Rolling-origin update evaluation

Rolling-origin-update evaluation is likely the most common approach in many practical applications. In this method, forecasts are made similarly to rolling-origin-recalibration evaluation, but unlike recalibration, values from the test set are not added to the training set, and no model retraining is done. Instead, past values from the test set are simply used to update the model’s input information. Both rolling-origin evaluation methods are often referred to as n-step-ahead evaluation, where *n* represents the forecast horizon. However, recalibration can be computationally intensive, and in real-world applications, the model is usually built once by experts and then used for forecasting with updated information as new data comes in, without being rebuilt. (Bergmeir, C. & Benítez, J. M. 2012).

Rolling-window evaluation

Rolling-window evaluation is similar to rolling-origin evaluation, but with a key difference: the amount of data used for training remains fixed. As new data becomes available, older data from the

LUT Group Confidential - Other information (3Y)

beginning of the series is discarded. This approach is only suitable if the model is rebuilt in every window. While it offers theoretical statistical advantages, these benefits are typically only noticeable in practice when older values interfere with model performance. (Bergmeir, C. & Benítez, J. M. 2012).

Seasonality and subsequencing and data normalization

When performing subsequencing, it is crucial to ensure that each subsequence fully encompasses a complete cycle of the seasonal pattern. This consideration is particularly important for periodic data, such as yearly or monthly patterns, where subsequences should ideally begin and end at consistent points within the seasonal cycle to accurately capture the seasonal behavior. In cases of highly seasonal data, employing overlapping subsequences that span multiple seasonal periods can enhance the model's ability to capture long-term seasonal dependencies and variations. However, if the time series is divided into smaller windows in a manner that fragments the seasonality—such as cutting through the middle of seasonal cycles—the model may fail to fully capture the seasonal structure. To address this, it is generally advisable to create windows that either begin and end within the same seasonal cycle or to use a sliding window approach that includes data from several seasonal periods. (Hyndman, R. J., 2023)

According to some studies, long Short-Term Memory (LSTM) networks are quite flexible and can be trained to capture both trends and seasonality in time series data. However, these patterns must be properly encoded and processed to enable the model to learn them effectively. (Chung, J. et al. 2014) (Kong, X. et al. 2025)

Typical standardization methods include StandardScaler (Brownlee, J., 2018.)

LUT Group Confidential - Other information (3Y)

8 Baseline autoregression model

A simple autoregression model was implemented as an initial baseline for comparing future implementations for predicting humidity in weather forecasting.

The model represents a simple autoregressive (AR-style) neural model that predicts future humidity using a sliding window of the previous T = 7 timesteps. It uses recursive prediction, so the model's outputs become future inputs.

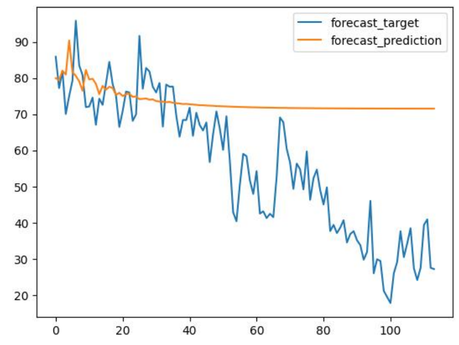

Figure 6. Autoregression model predictions

The model's performance was evaluated using the Mean Squared Error (MSE), resulting in a value of 592.0596, that will be used as the reference accuracy for more advanced implementations.

9 References

Chess, S. (2020) 'Cross-validation in time series', Medium, 6 October. Available at: https://medium.com/@soumyachess1496/cross-validation-in-time-series-566ae4981ce4 (Accessed: 8 November 2025).

Bergmeir, C. & Benítez, J. M. (2012) On the use of cross-validation for time series predictor evaluation. *Information sciences*. [Online] 191192–213.

Chung, J., Gulcehre, C., Cho, K. & Bengio, Y. 2014, *Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling*, Ithaca.

Kong, X. et al. (2025) Deep learning for time series forecasting: a survey. *International journal of machine learning and cybernetics*. [Online] 16 (7–8), 5079–5112.

Hyndman, R. J., 2023. *Forecasting: Principles and Practice* (2nd ed.). Available at: https://f0nzie.github.io/hyndman-bookdown-rsuite/seasonal-subseries-plots.html [Accessed 16 November 2025].

Heaton, J. (2018) Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep learning: The MIT Press, 2016, 800 pp, ISBN: 0262035618. Genetic programming and evolvable machines 19 (1–2) p.305–307.

Brownlee, J., 2018. *Deep Learning for Time Series Forecasting*. Machine Learning Mastery. Available at: https://machinelearningmastery.com/deep-learning-for-time-series-forecasting/ [Accessed 16 November 2025].

LUT Group Confidential - Other information (3Y)