UNIVERSITÀ DEGLI STUDI DI BERGAMO
Dipartimento di Ingegneria Gestionale, dell'Informazione e della Produzione

Air quality forecasting using statistical and machine learning models in Schivenoglia area

Silviu Filote - 1059252
Jonathan Bommarito - 1068755

Statistics for High Dimensional Data (S4HDD) and CompStat Lab a.a. 2023/2024 (2nd edition)

DATA Febbraio 2024

Outline

1. Introduction
2. Dataset description
3. Methodology
4. Statistical analysis
5. Machine learning analysis
6. Conclusions
7. Packages
8. References

Introduction

Air pollution, stemming from various sources including industrial activities, transportation, and agriculture, contributes to the degradation of air quality through the emission of primary pollutants and volatile organic compounds (VOCs). These emissions represents a serious threat to public health and the environment, necessitating a comprehensive understanding of its sources, dynamics, and impacts to formulate effective mitigation strategies. Our research purpose is to investigate Ammonia (NH3) compound using the daily observations in the Agrimonia dataset. We will apply a time series analysis approach and a machine learning one to see the differences between the 2 methods.

Ammonia facts

- Ammonia (NH_3) is one of the most produced inorganic chemicals worldwide, with a total global production of 109 million metric tons (Mt) in 2009 **[1]**
- Ammonia is a widely used chemical with various industrial and agricultural applications. About 80% of the ammonia produced in industry is used in agriculture as fertilizer. Ammonia is also used as a refrigerant gas, to purify water supplies, and in the manufacture of plastics, explosives, fabrics, pesticides, dyes and other chemicals.
- When NH_3 enters the body because of breathing, swallowing or skin contact, it reacts with water to produce ammonium hydroxide. This chemical is very corrosive and damages cells in the body on contact.
- Ammonia (NH_3) serves as a key precursor gas for secondary particulate matter (PM), including both PM_{10} and $PM_{2.5}$. **[2]**
- The agricultural sector, particularly livestock and fertilizers, is widely recognized as a significant source of ammonia emissions, with up to 90% originating from this sector in Europe and an impressive 97% in the Lombardy region, Italy. **[2]**

Dataset description

Agrimonia overview

- The Lombardy region, Italy, stands out as one of Europe's most polluted areas, primarily due to limited air circulation and elevated emission levels.
- There exists a substantial scientific consensus attributing the agricultural sector as a significant contributor to the region's air quality issues.
- The dataset includes daily observations from 2016 to 2021
- The daily metrics includes air quality, weather patterns, emissions, livestock statistics, land and soil characteristics, providing a comprehensive view of environmental Lombardy's dynamics.
- The dataset is constructed by the AgrImOnIA project and contains 43 variables. NH3 will be used as response variable and the others as covariates in the further analysis **[3]**
- We will take into the account only the data related to "Schivenoglia", which comprehend 2192 observations from 1st January 2016 to 31st December 2021

Methodology

- Implementing imputation robust techniques as well as the uncertainty to fill missing data within the variables' time series
  - **A) Block bootstrap**
  - **B) Kalman Smoother**
- **Objective:** analyze the Ammonia phenomenon using statistical and a machine learning approaches. Also understand which are the most significant covariates that influence Ammonia concentrations in the Schivenoglia location
- **Splitting the dataset** into training (80%) and validation (20%)
- Evaluate model dynamics by plots
- Employing **tuning processes** and **model selection techniques** in the training phase to get the best model possible
- Implement the best model and check the forecast performances and validation statistics: **RMSE**

- Residual analysis and testing phase
  - **Shapiro and Wilk test** – H0: normality distribution
  - **Breusch-Pagan Test** – H0: homoscedasticity
  - **Ljung-Box test** – H0: no correlation
- Statistical Analysis:
  - **A) Linear regression**
  - **B) ARIMA models**
- In the machine learning analysis multiple models will be developed and compared with the others to see which one fits the best to the current case study. We will see:
  - **A) XGBoost**
  - **B) Prophet, facebook's model**
  - **C) LSTM (Long short-term memory)**

Statistical analysis

Kalman Smoother

- **Introduction to Kalman Smoother**
- An algorithm used in the analysis of dynamic systems and signal processing.
- Extension of the Kalman filter, providing optimal estimates of the system state.

- **Difference between Kalman Filter and Kalman Smoother**
- Kalman filter provides "one-step-ahead" estimates based only on observed data up to that point.
- Kalman smoother offers retroactive estimates using the entire sequence of available observations.

- **Utility of Kalman Smoother for Missing Values**
- Treatment of missing or incomplete values in time series.
- Utilizes information from past and future observations to improve estimates of missing states.

Block Bootstrap

- **“N”:** Specifies the total number of samples (in this case, 2192).
- **“L”:** Indicates the number of blocks size (in this case, 90).
- **“B”:** Indicates the number of time series generated (in this case, 100).

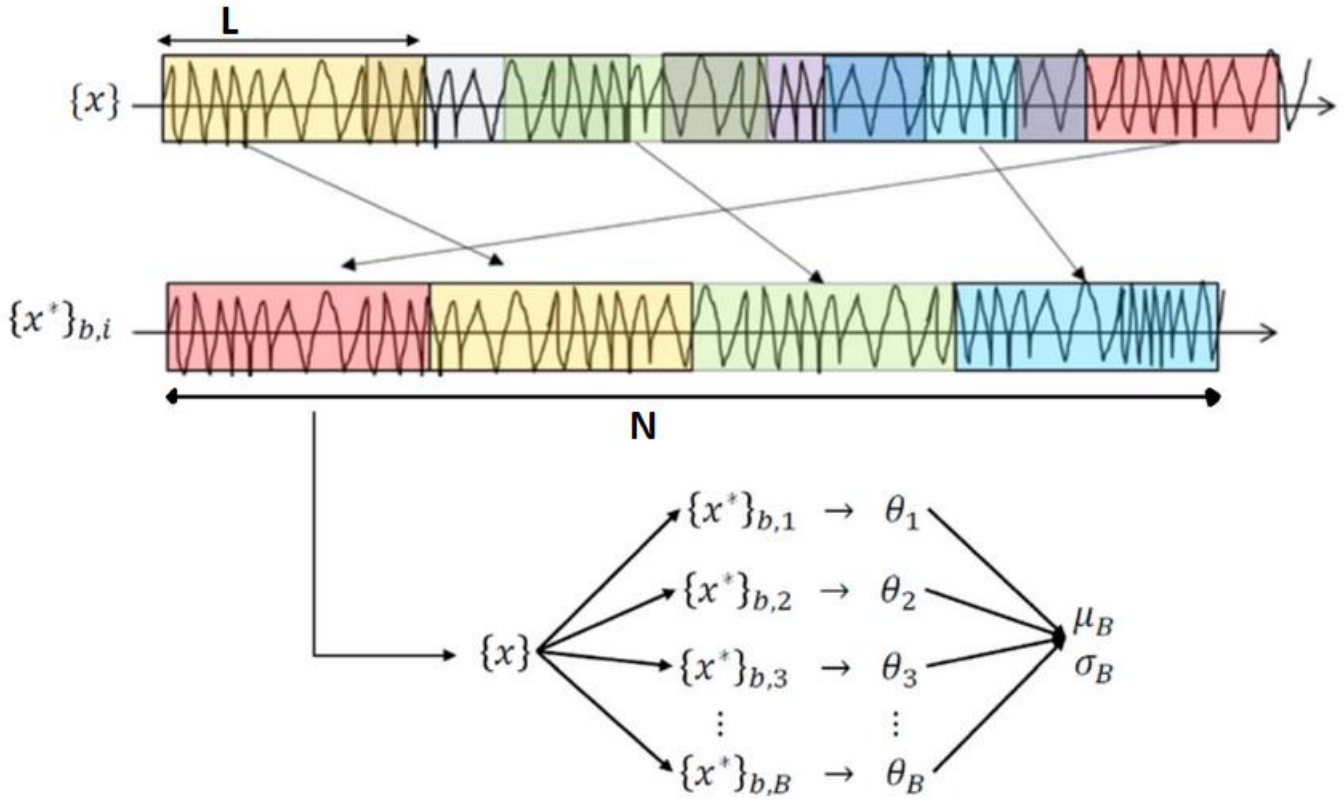


Figure: Block Bootstrap **[4]**

Linear regression

Linear regression model is defined with given equation:

$$y(t) = \beta_0 + x(t)'\beta + \mu(t)$$

Where:

- y(t) represents the time-series data.
- β_0 is the intercept term
- x(t) is a vector of predictor variables at time t.
- β vector of coefficients corresponding to each predictor variable in x(t).
- μ(t) the error or disturbance term at time t. y(t) that is not explained by the predictors in x(t)

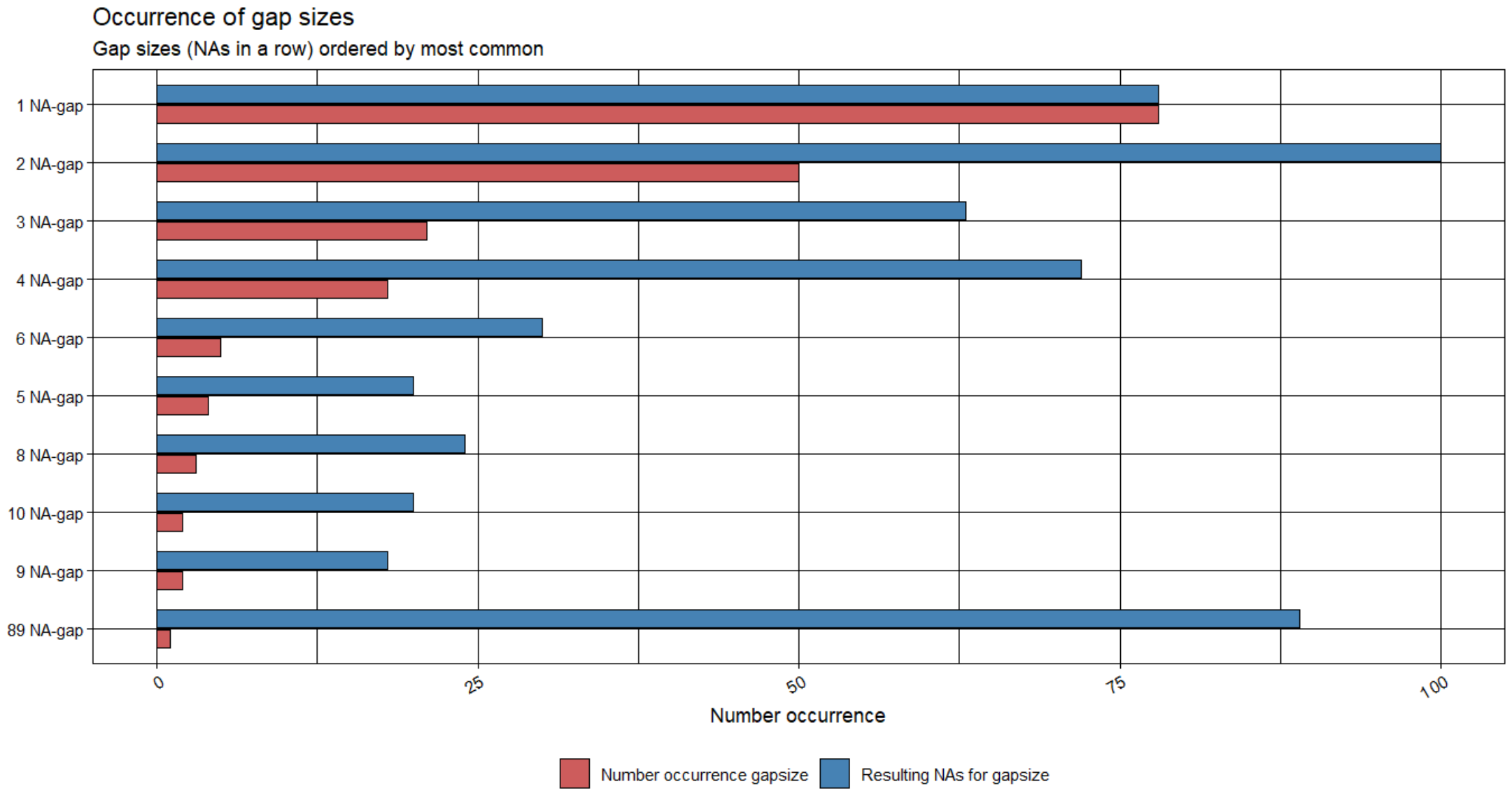
ARIMA model

ARIMA model is defined with given equation:

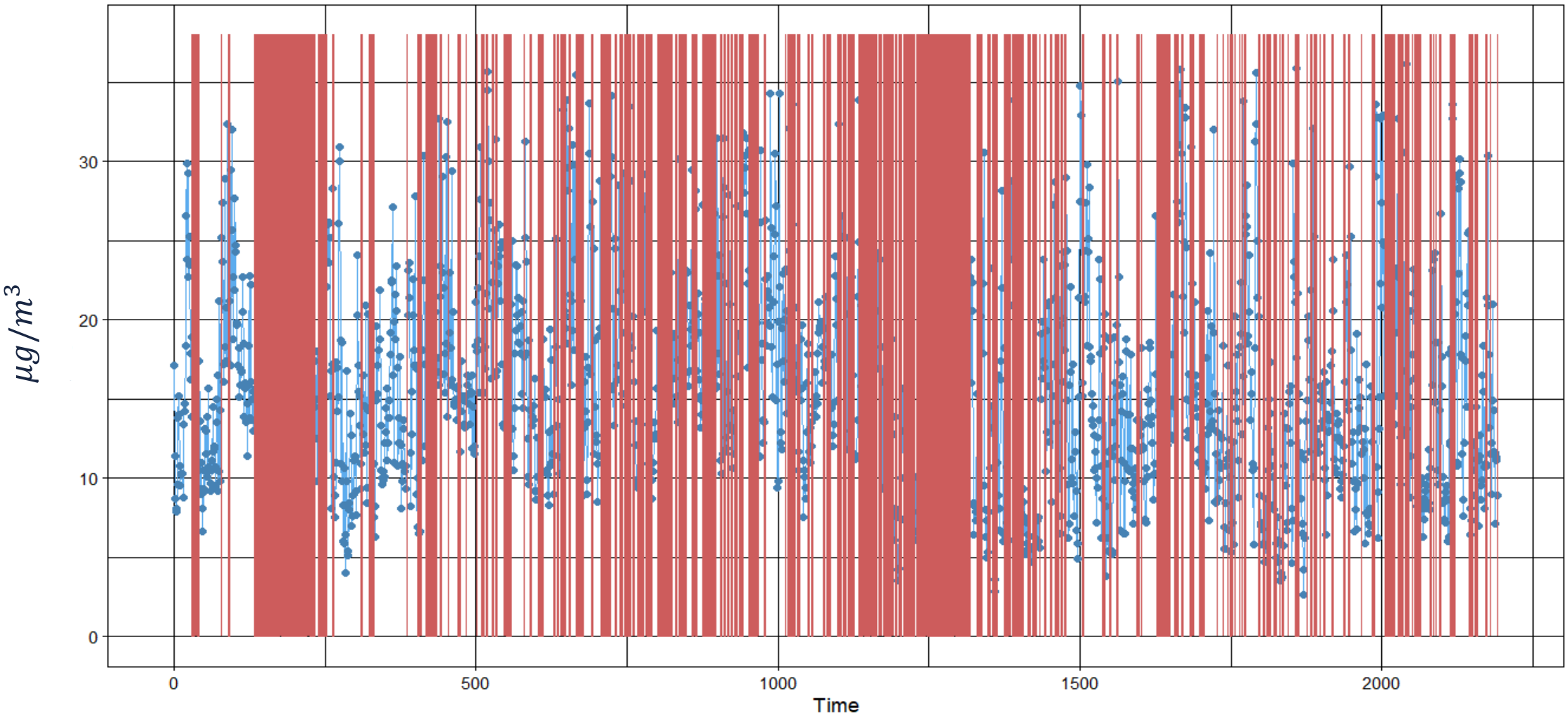
$$\Delta^d \mu_t = c + \varepsilon_t + \sum_{i=1}^{p} \phi_i \mu_{t-i} + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j}$$

Where:

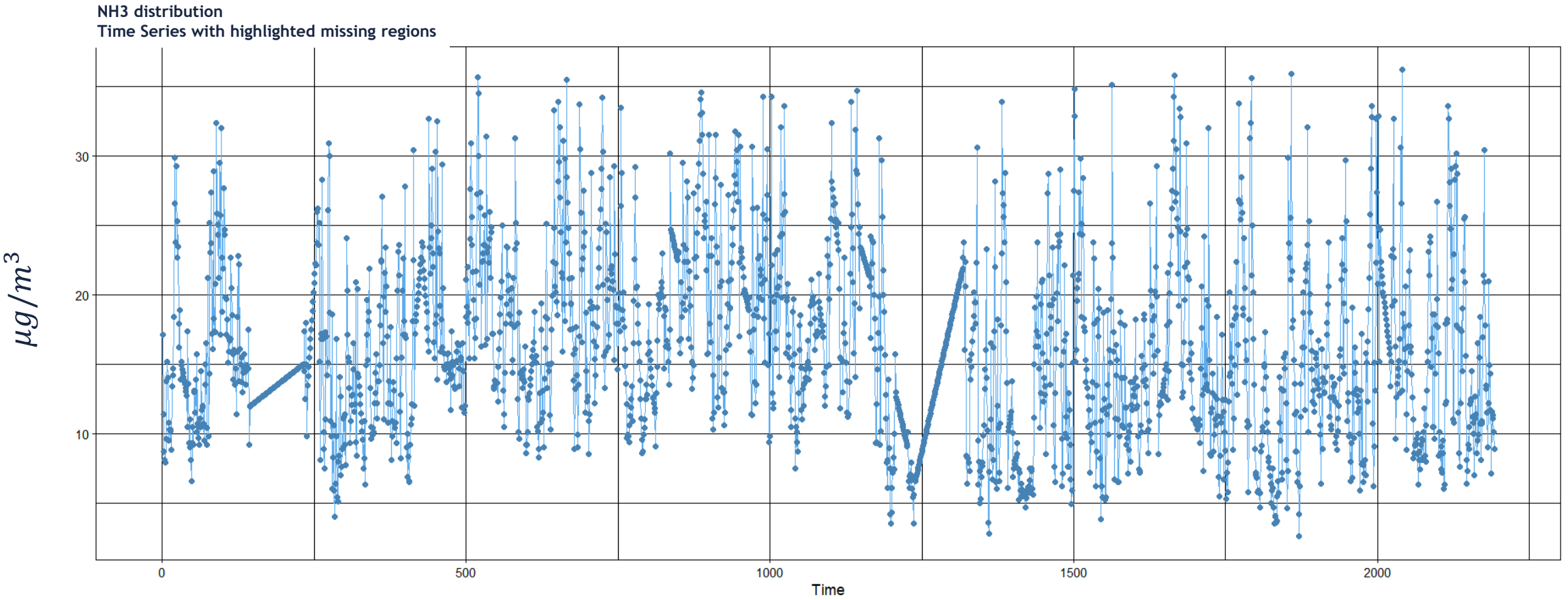
- Δ^d denotes the differencing operator of order d,
- μ_t represents the value of the residuals time series at time t.
- c is a constant term.
- ε_t is the error term at time t
- ϕ_i are the autoregressive parameters for lagged values of μ.
- p is the order of the autoregressive (AR) component.
- θ_j are the moving average parameters for lagged values of ε.
- ε_{t-j} represents the error term at time t lagged by j periods.
- q is the order of the moving average (MA) component.


Occurrence of gap sizes
Gap sizes (NAs in a row) ordered by most common
1 NA-gap
2 NA-gap
3 NA-gap
4 NA-gap
6 NA-gap
5 NA-gap
8 NA-gap
10 NA-gap
9 NA-gap
89 NA-gap
0
25
50
75
100
Number occurrence
Number occurrence gapsize
Resulting NAs for gapsize

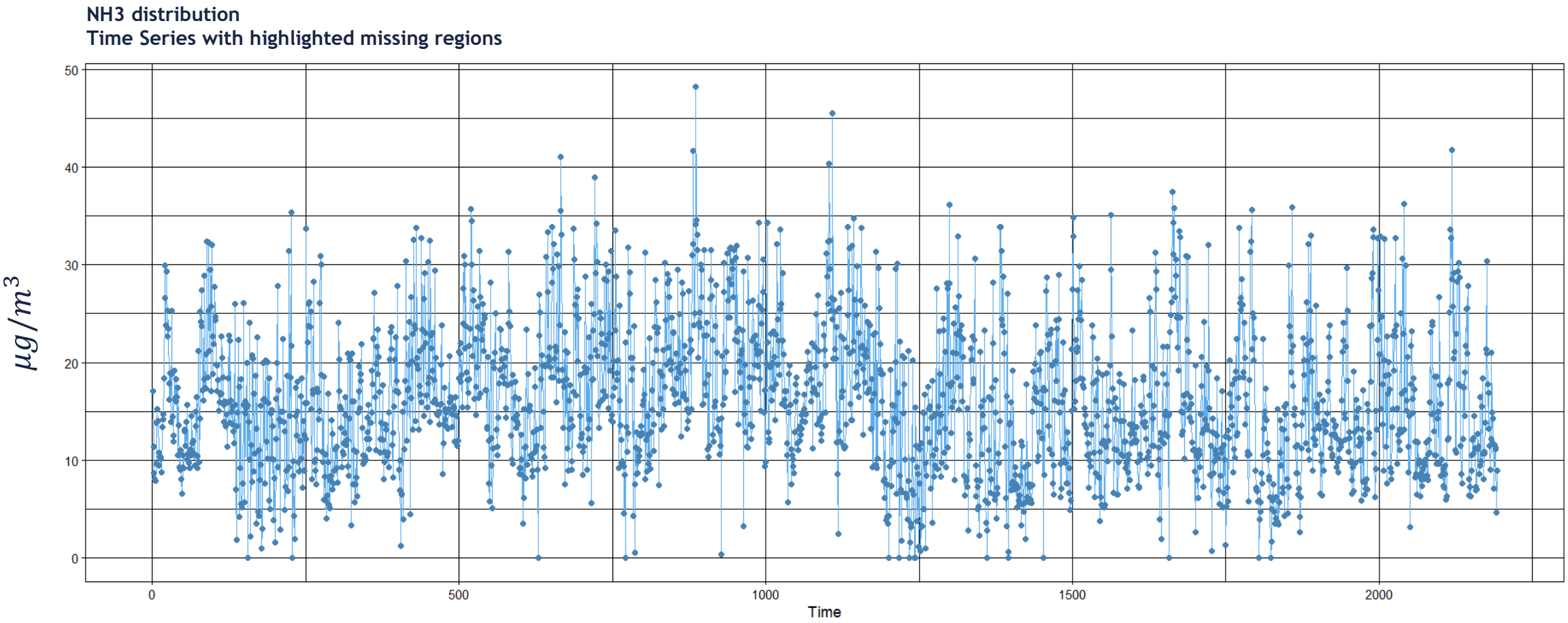
NH3 distribution
Time Series with highlighted missing regions



Kalman Smoother implementation



Kalman Smoother + Block Bootstrap



Linear model

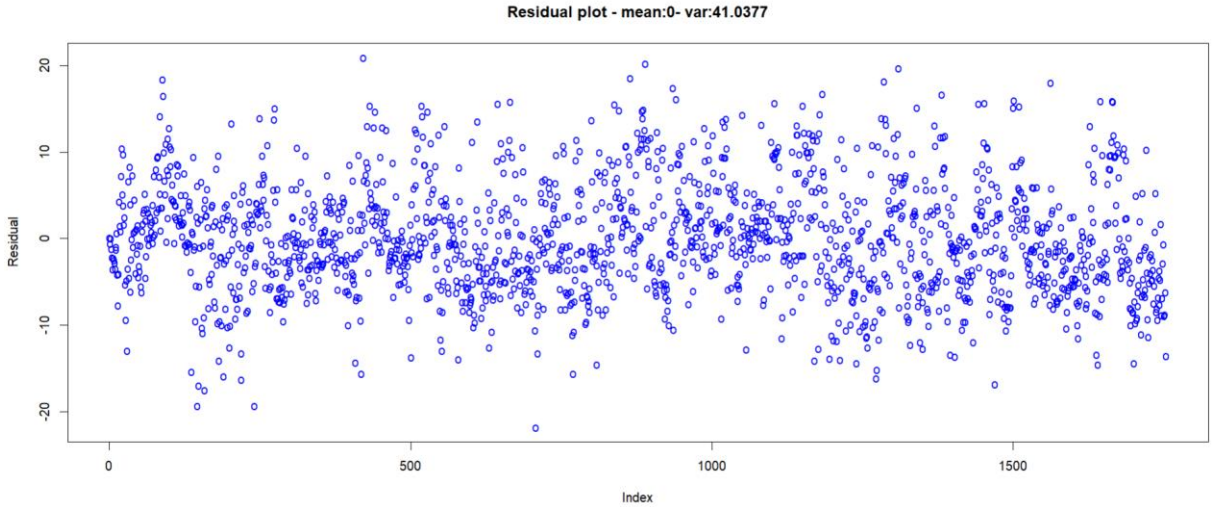
- Backward Stepwise algorithm applied to obtain the best model
- RMSE metric used for evaluation

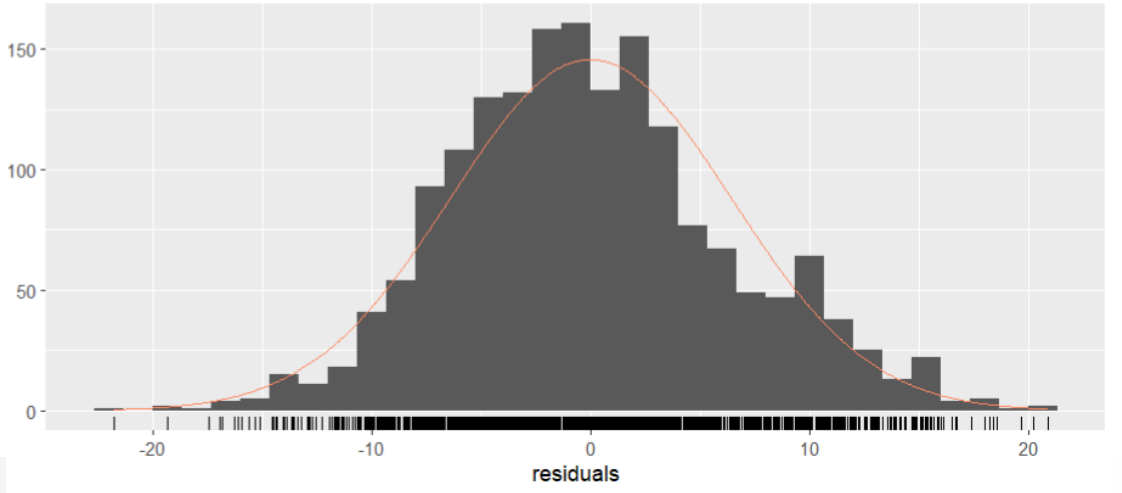
	train RMSE	validation RMSE
model	6.272	6.945
backward_model	6.404	6.928

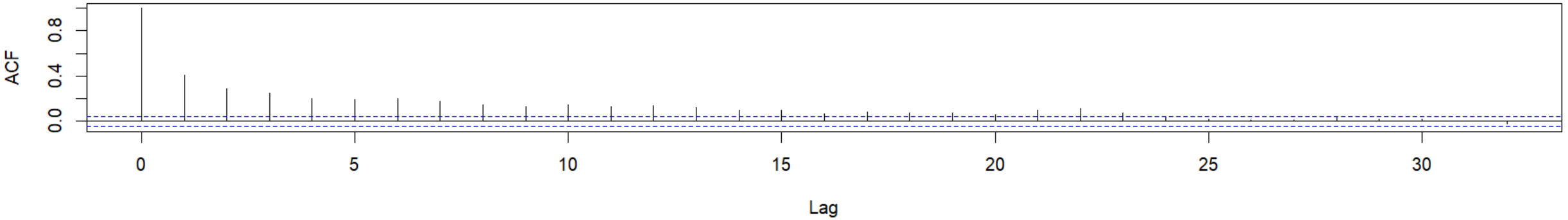
backward_model	Estimate	Std. Error	t value
AQ_nox	4.645e-02	1.032e-02	4.499
AQ_pm25	1.477e-01	1.397e-02	10.574
AQ_co	1.876	5.962e-01	3.147
AQ_so2	-2.938e-01	1.132e-01	-2.596
WE_temp_2m	2.570e-01	3.534e-02	7.272
WE_solar_radiation	1.851e-07	5.606e-08	3.303
WE_rh_min	-2.027e-01	3.149e-02	-6.435
WE_rh_mean	1.825e-01	3.520e-02	5.185
WE_blh_layer_max	-1.375e-03	5.273e-04	-2.608
LI_pigs	1.839e-02	1.770e-03	10.389
LI_bovine	-1.319	1.964e-01	-6.714

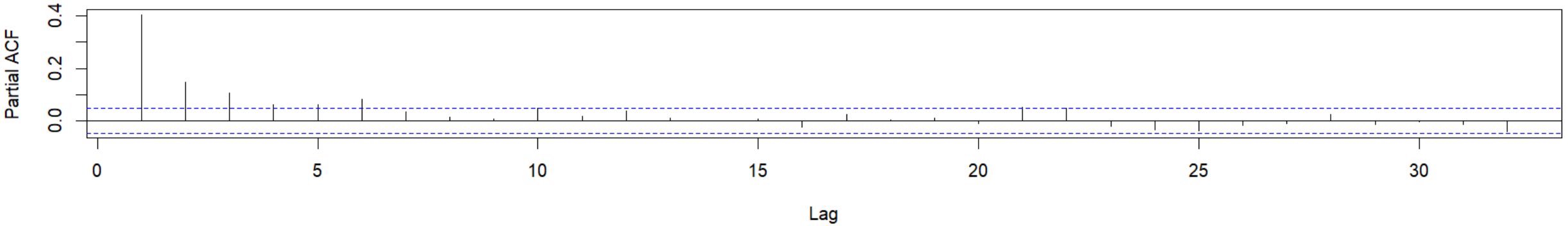
Residuals Analysis

test	H0	p-value
Shapiro-Wilk	Normality	8.742e-08
Breusch-Pagan	homoschedasticity	5.036e-10
Ljung-Box	independently distributed	< 2.2e-16






ACF
0.0
0.4
0.8
0
5
10
15
20
25
30
Lag

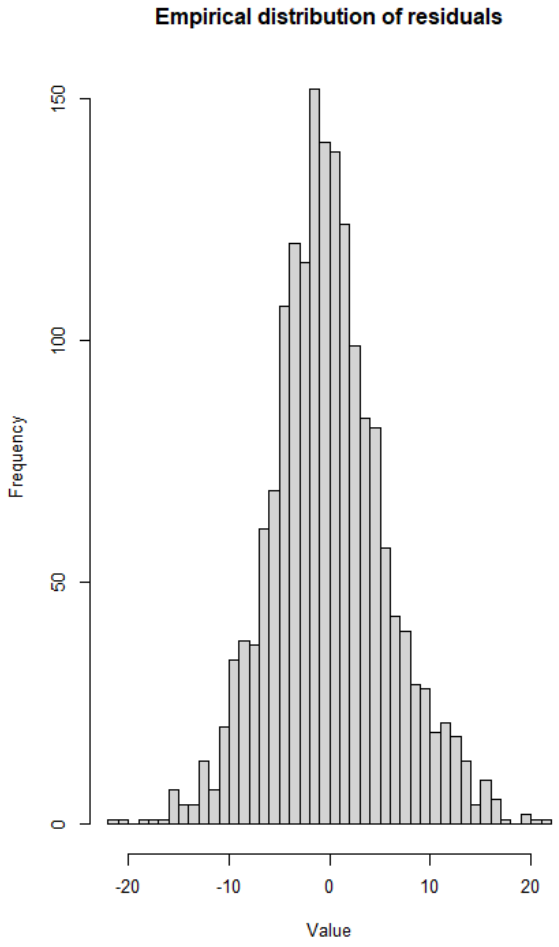

Partial ACF
0.0
0.2
0.4
0
5
10
15
20
25
30
Lag

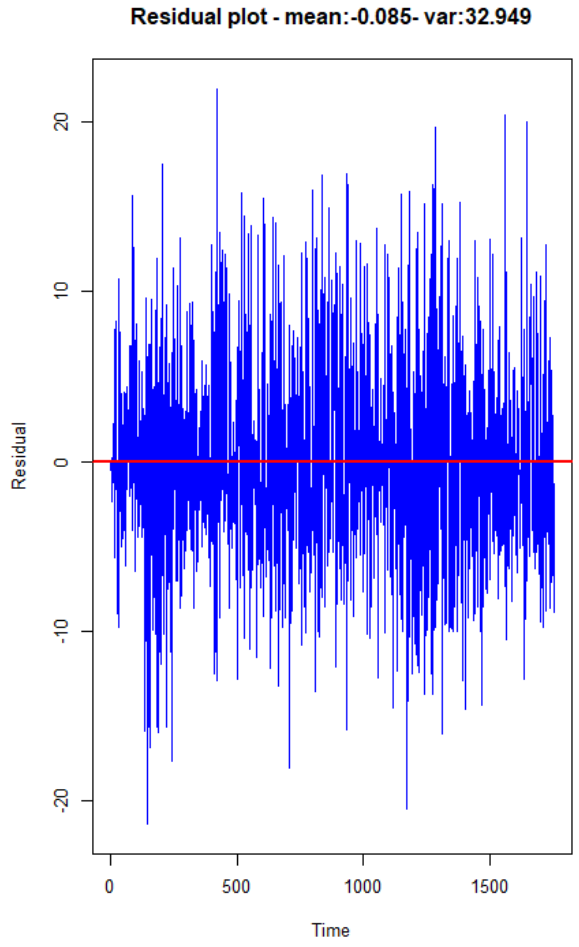
ARIMA Implementation

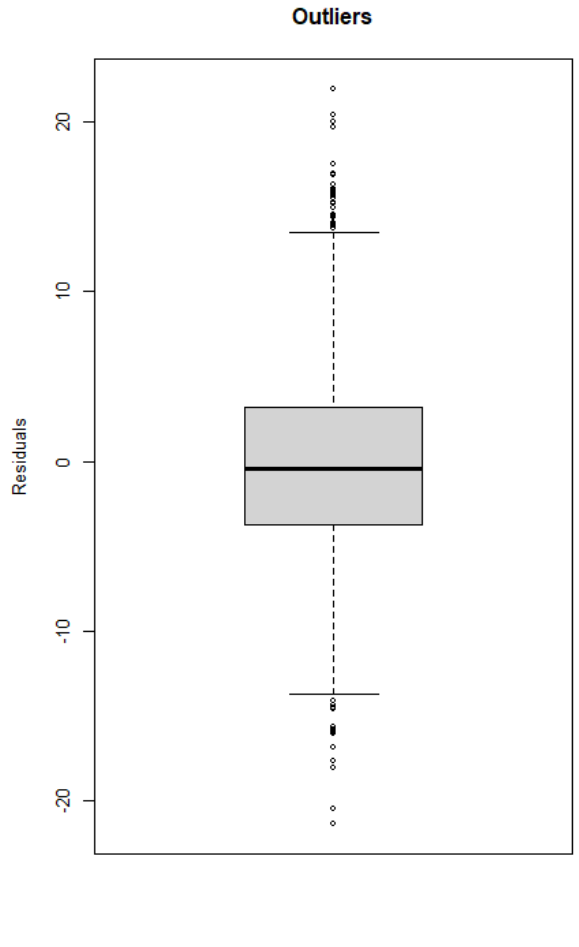
- Various combinations of AR (AutoRegressive), I(Integretion), MA (Moving Average) values were tested for the ARIMA model.
- After evaluating the performance of each combination, considering RMSE on the validation data, Ljung-Box test, and Complexity, the ARIMA model with the `AR1_I0_MA3` combination was selected as the best.

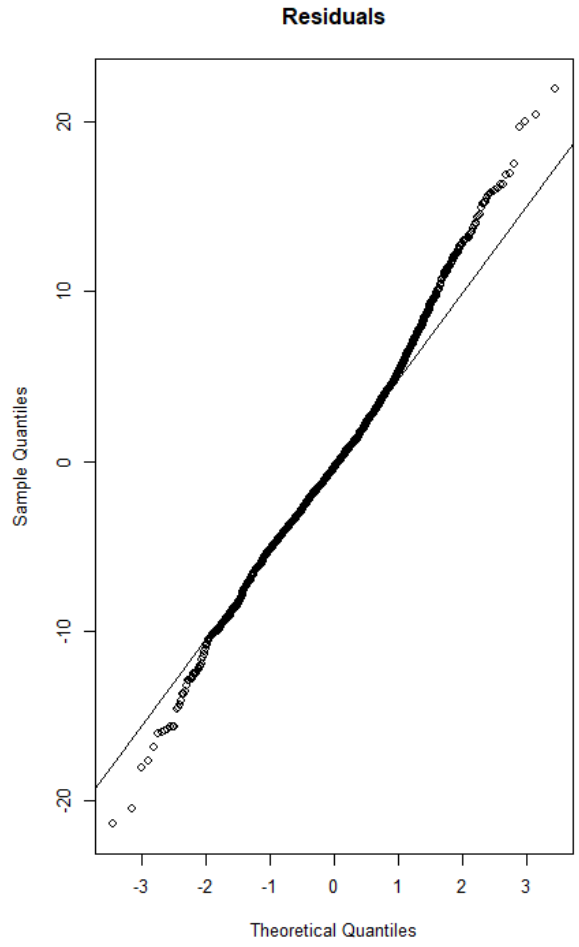
models	train RMSE	validation RMSE	Shapiro-Wilk test	Ljung-Box test
AR1_I0_MA1	5.7368	6.9027	8.418e-09	0.0851
AR1_I0_MA2	5.7111	6.9063	3.755e-09	0.8964
AR1_I0_MA3	5.7088	6.9062	3.765e-09	0.9730
AR1_I1_MA1	5.7687	7.9367	2.971e-09	0.6525
AR1_I1_MA2	5.7392	6.6422	5.880e-09	0.1513
AR1_I1_MA3	5.7129	6.8722	2.934e-09	0.8258
AR2_I0_MA1	5.7086	6.9108	3.697e-09	0.9362
AR2_I0_MA2	5.7079	6.9076	3.825e-09	0.9902
AR2_I0_MA3	5.7103	6.9125	3.628e-09	0.9225
AR2_I1_MA1	5.7606	7.7315	3.158e-09	0.9023
AR2_I1_MA2	5.7110	6.8415	3.298e-09	0.9735
AR2_I1_MA3	5.7153	6.8498	2.975e-09	0.9435
AR3_I0_MA1	5.7080	6.9019	3.809e-09	0.9979
AR3_I0_MA2	5.7084	6.9108	3.700e-09	0.9569
AR3_I0_MA3	5.7079	6.9061	3.828e-09	0.9901
AR3_I1_MA1	5.7528	7.2905	4.815e-09	0.9215
AR3_I1_MA2	5.7133	6.7485	3.310e-09	0.9624
AR3_I1_MA3	5.7090	6.9029	3.013e-09	0.9299

ARIMA Residuals Analysis









Machine learning analysis

XGBoost

- XGBoost (eXtreme Gradient Boosting) is a decision tree-based **ensemble algorithm** that leverages the concept of **gradient boosting** to create a strong model
- XGBoost works by creating a set of decision trees iteratively, each tree attempting to **correct the mistakes** of the previous tree. The algorithm employs a **gradient descent** algorithm to **minimize a cost function**, which is the sum of the errors of each tree in the ensemble.
- The final model is a **weighted combination of all the decision trees**, with each tree assigned a weight based on its contribution to the cost function.

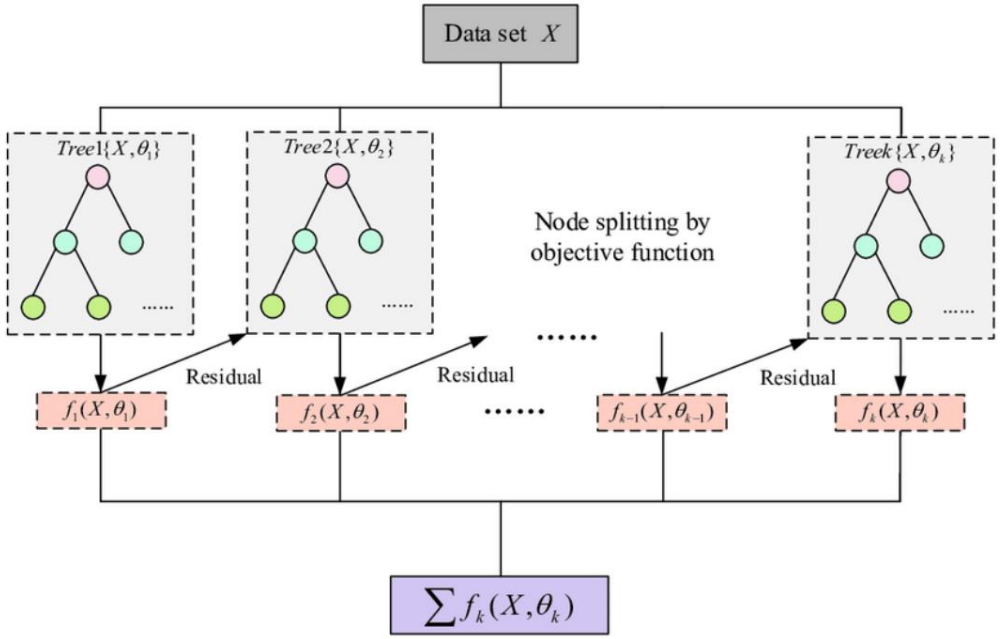


Figure: XGBoost model schematics **[5]**

- **Algorithm steps:**
  1. Initialize the model
  2. Fit the first tree
  3. Compute the loss
  4. Fit the next tree
  5. Make predictions
- **Advantages of XGBoost:**
  - Building decision trees as weak learners and performing tree pruning
  - Implementing Shrinkage technique in the Gradient Descent algorithm
  - Calculates feature importance by measuring its contribution to the loss reduction
  - Scalibility, flexibility, regularization and speed
- **Disadvantages**
  - Overfitting problems
  - Hyperparameter Tunings
  - Trees can be very non-robust

Prophet

- Prophet is a model for forecasting time series data based on an **additive regression model** where non-linear trends are fit with yearly, weekly, and daily seasonality, plus holiday effects
- It works best with time series that have **strong seasonal effects** and several seasons of historical data
- Prophet is **robust** to missing data and shifts in the trend, and typically handles outliers well.
- Prophet **automatically detects** and models various components of time-series data including trend, seasonality, and holiday effects.
- Prophet automates much of the forecasting process and involves statistical techniques that may be complex to some users

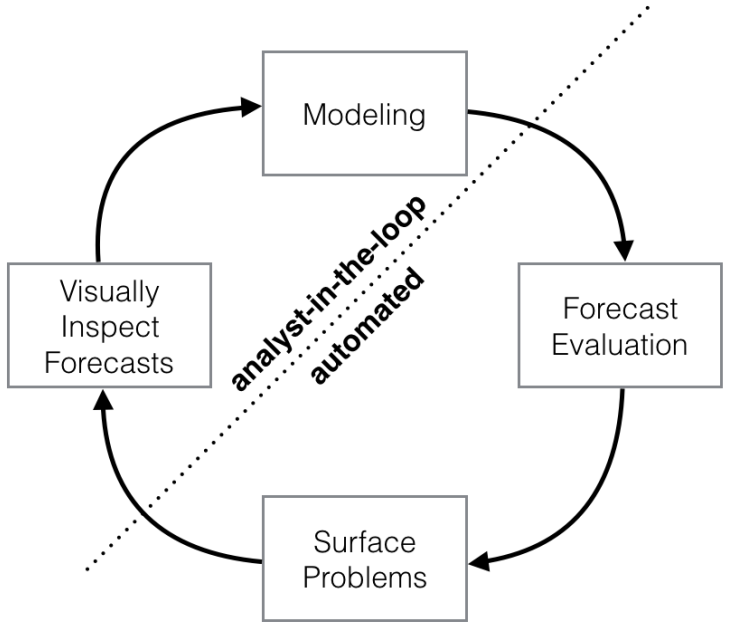


Figure: Prophet model logic **[6]**

- **Prophet procedure is an additive regression model with four main components:**
  1. Trend component
  2. Seasonal seasonal component modeled using Fourier series/dummy variables.
  3. Holiday effect, user can provide list
  4. Random error
- **Advantages of Prophet:**
  - Prophet makes it much more straightforward to create a reasonable, accurate forecast
  - Uncertainty Estimation
- **Disadvantages**
  - Limited Flexibility in the customization options
  - seasonal patterns are only additive

LSTM

- A neural network is a collection of algorithms designed to learn the patterns in observed
- Training phase aims to adjust the weight between neurons
- **Recurrent Neural Network (RNN)** allows to propagate the information among nodes creates long-term dependencies therefore, to process sequences.
- Training the RNN with long term temporal dependencies leads to **vanishing gradient problem**
- **LSTM (Long Short Term Memory network)** networks fix the vanishing gradient problem, implementing a more complex structure
- LSTM networks has the same propagation logics as RNN, however the information is processed in a different way

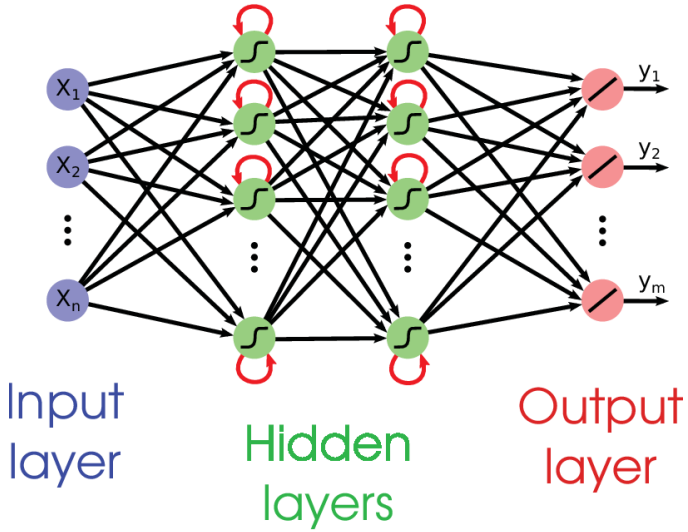


Figure: Example of a RNN architecture **[7]**

- The module structure of the LSTM network is composed by:
  - Input gate
  - Forget gate
  - Output gate
- The LSTM network is composed by chain formed by repeating the same module
- Advantages:
  - Long-term dependency modeling
  - Robustness to noise and irrelevant information
- Disadvantages:
  - Complexity
  - Computationally intensive
  - Overfitting

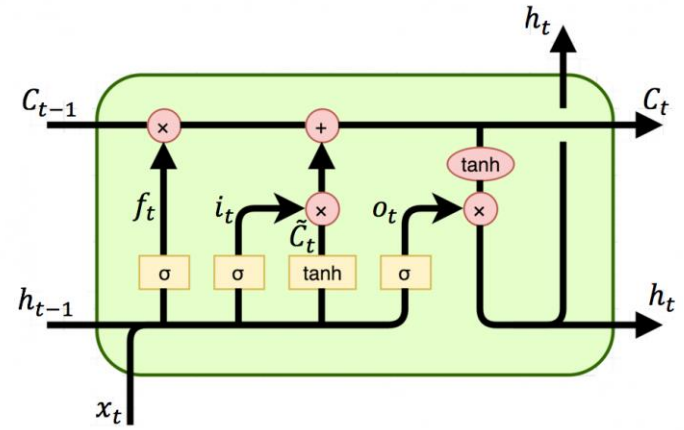


Figure: One module of the LTSM network **[7]**

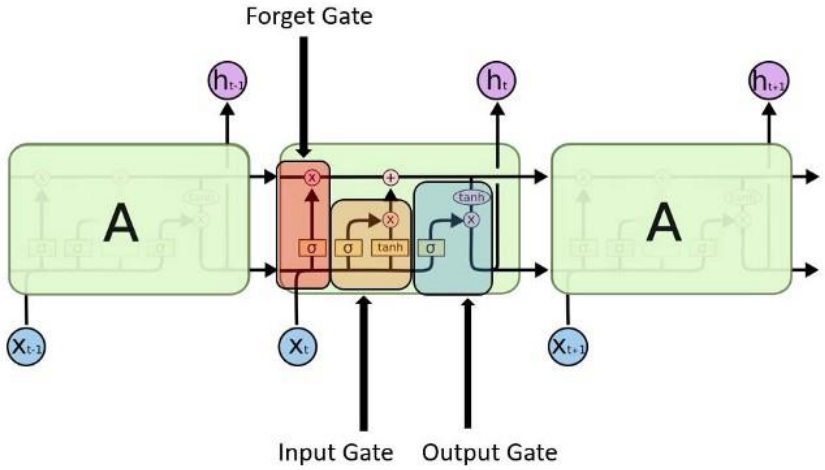


Figure: repeating module in an LSTM **[7]**

XGBoost implementation

- XGboost is widely used in classification or a regression problem
- Training 80% and validation 20%
- After 1000 iterations overfitting is detected, we set that as the max boundary of our iterations
- max_depth indicates the max number of splits per decision tree, higher value means low bias but high variance
- Tuning on the max_depth of the decision trees to pick up the best model
- The shrinkage parameter η that controls the learning rate of the model is set as 0.01 to prevent overfitting

Max deapth	min iterations	train $RMSE$	validation $RMSE$
max_deapth_1	998	6.088	7.325
max_deapth_2	997	5.349	7.132
max_deapth_3	745	4.552	7.097
max_deapth_4	324	3.684	7.047
max_deapth_5	336	2.758	6.943
max_deapth_6	347	1.914	7.950

Figure: Tuning on `max_depth` parameter

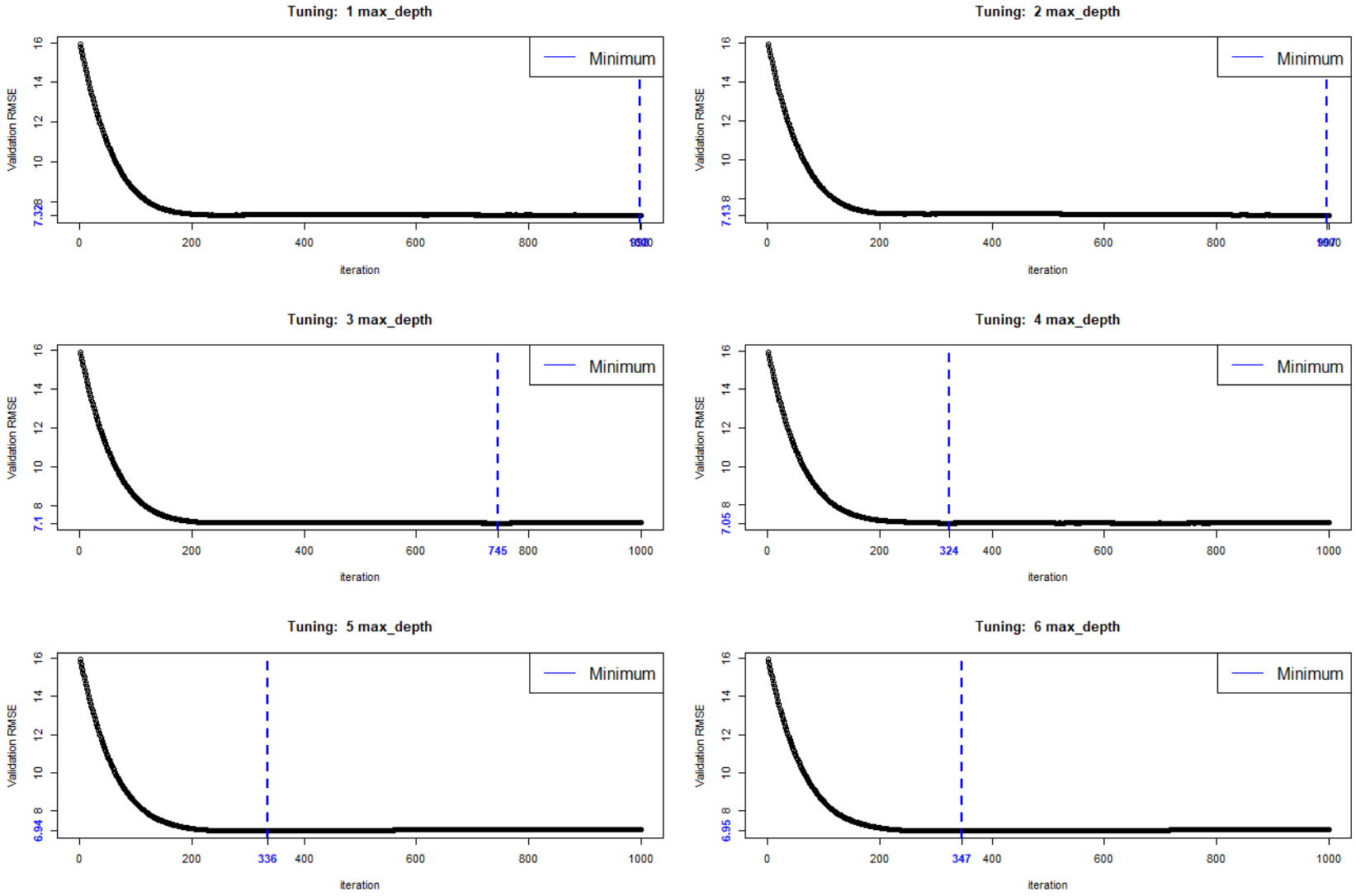


Figure: Tuning plots of `max_depth` parameter

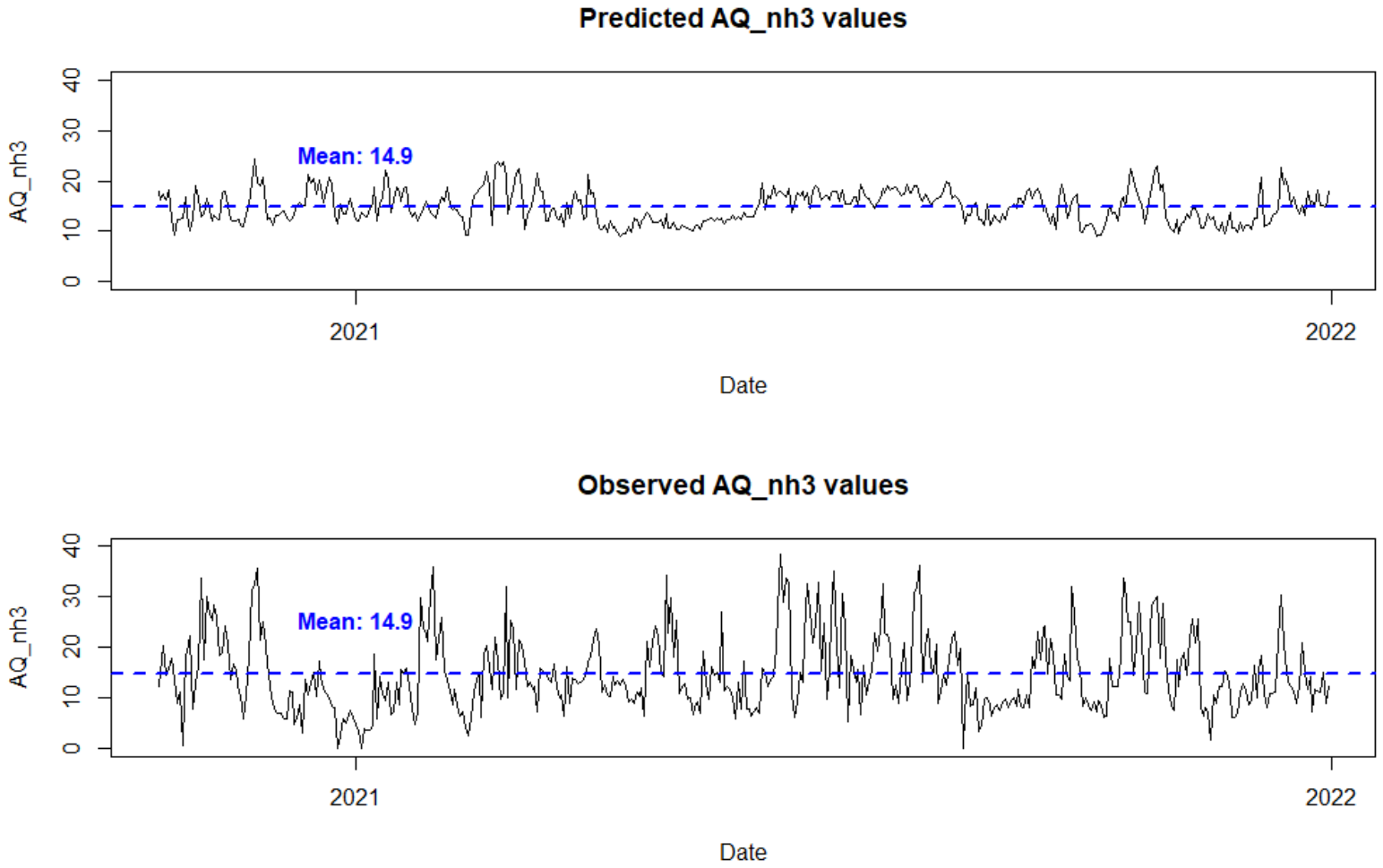


Figure: comparing predicted vs observed of AQ_nh3

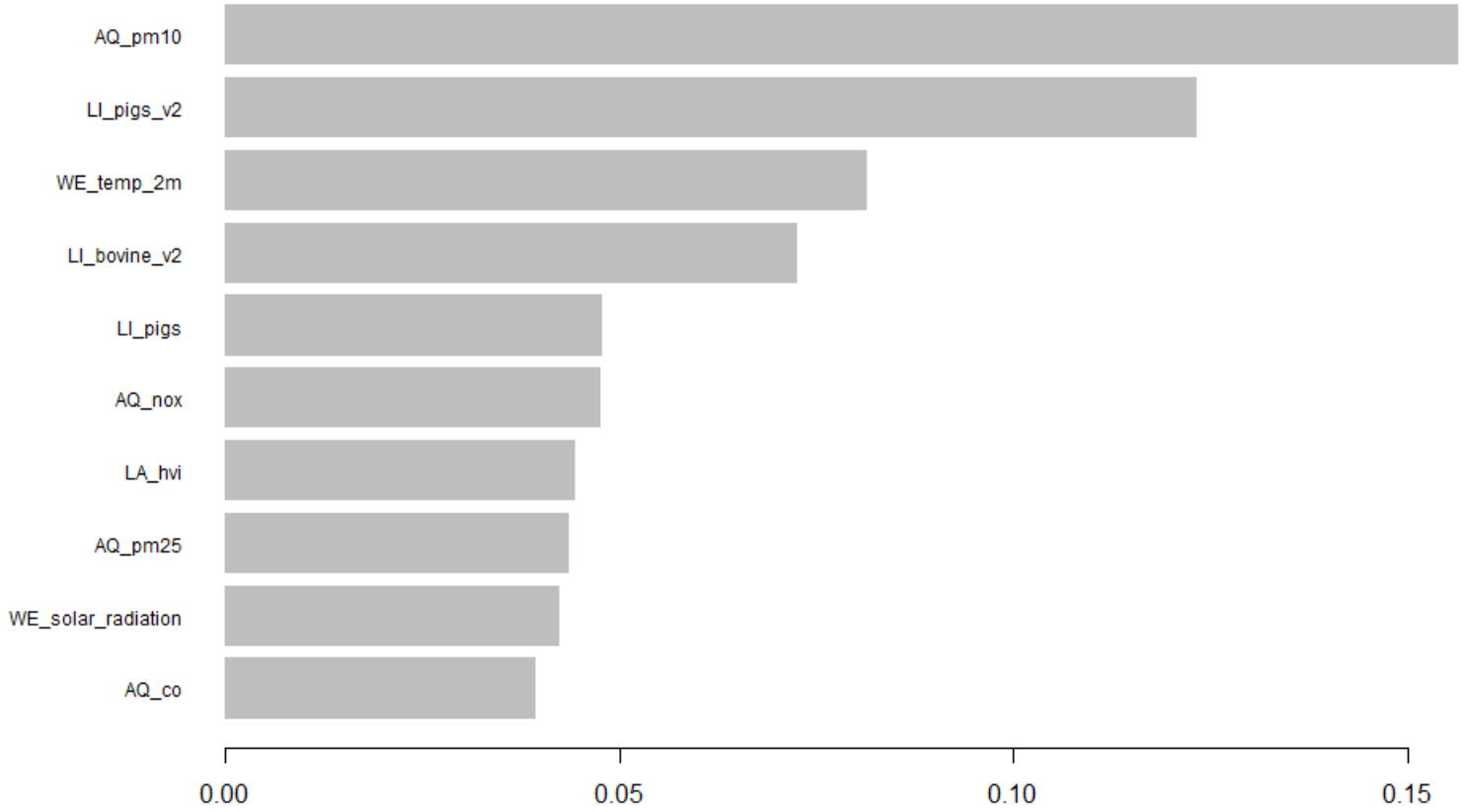


Figure: Gain of the 10 most important features used for splitting in the decision trees

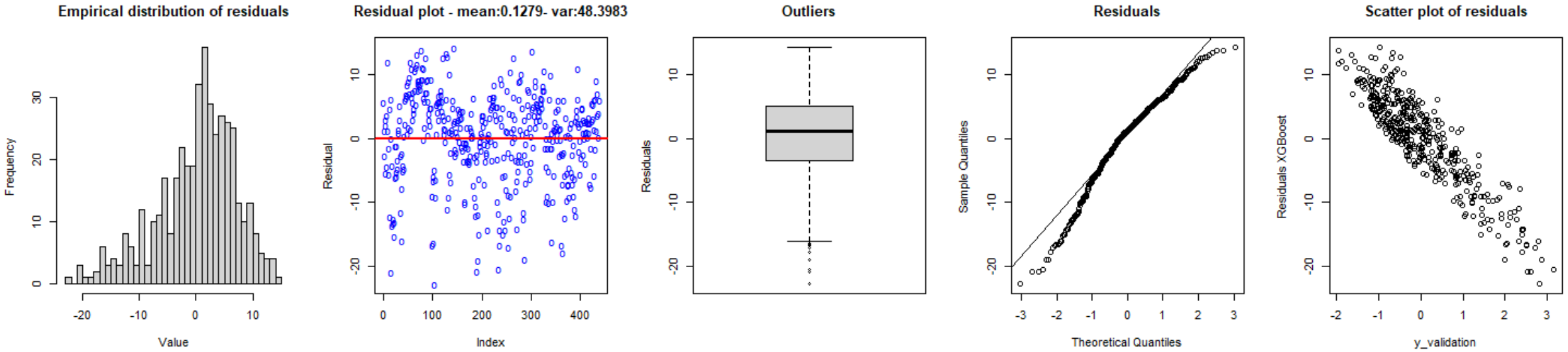


Figure: Residual behavior in validation

Prophet implementation

- Training 80% and validation 20%
- Add holidays component to the model as a list
- Add regressors to the model
- Automatic additive decomposition in the 4 components
- Customize plots offer by the library
- Residual analysis and testing

$$y(t) = g(t) + s(t) + h(t) + e(t)$$

Figure: where, y(t) refers to the forecast g(t) refers to the trend s(t) refers to the seasonality h(t) refers to the holidays for the forecast e(t) refers to the error term while forecasting **[8]**

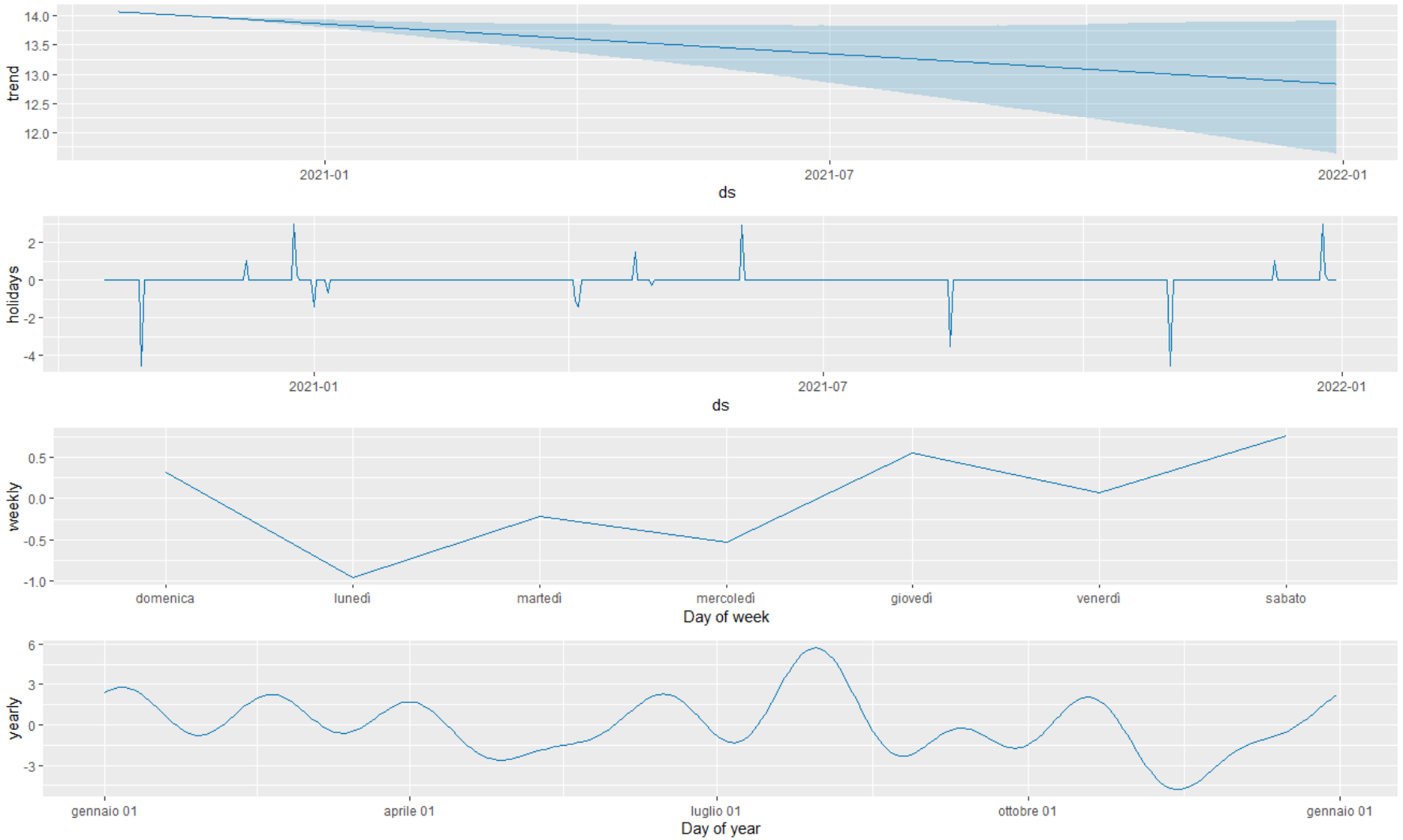


Figure: additive model decomposition over the validation set

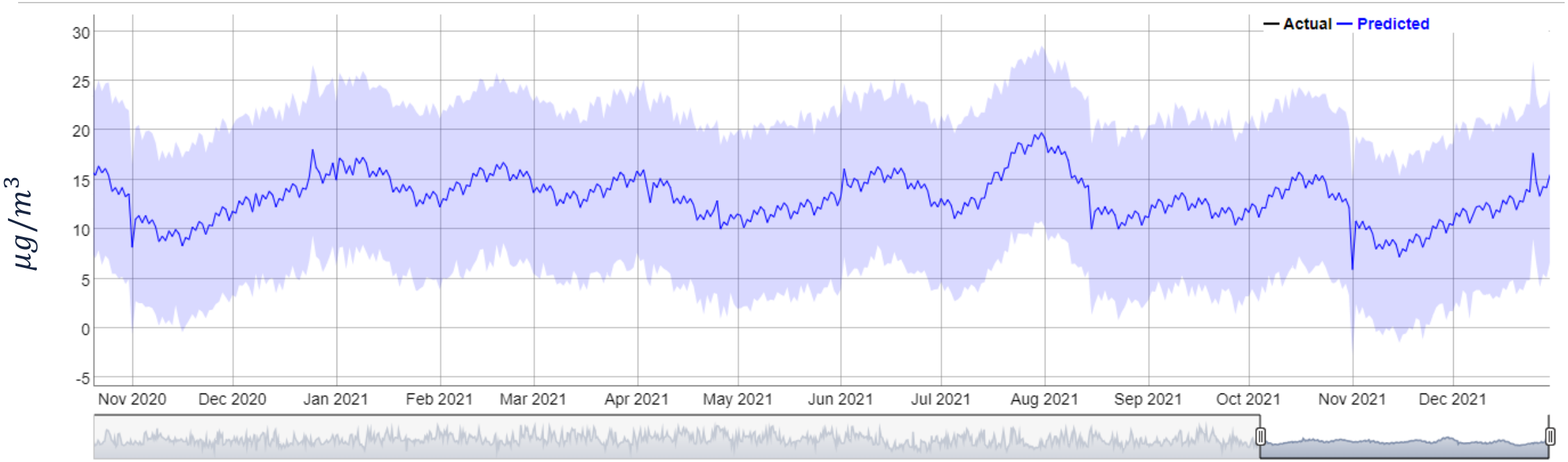


Figure: predicted AQ_nh3

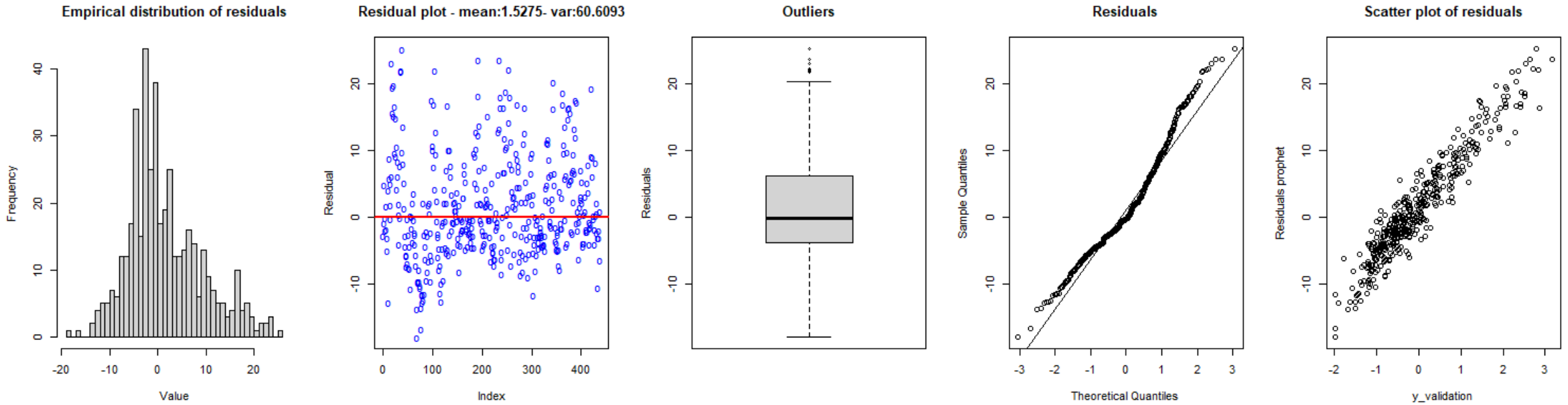


Figure: Residual behavior in validation

LSTM implementation

- Training 80% and validation 20%
- Minimizing a loss function in the training phase: **MSE**
- Optimizer used to update the weights of the parameters involved: **Adam optimizer**
  - Adaptive Learning Rate
  - Bias Correction
  - robustness and effectiveness in training
  - Computationally efficient
- `num_epochs`, the number of times the entire dataset will be passed forward and backward through the network during the training process
- Tuning on to `num_epochs` avoid overfitting
- Residual analysis and testing

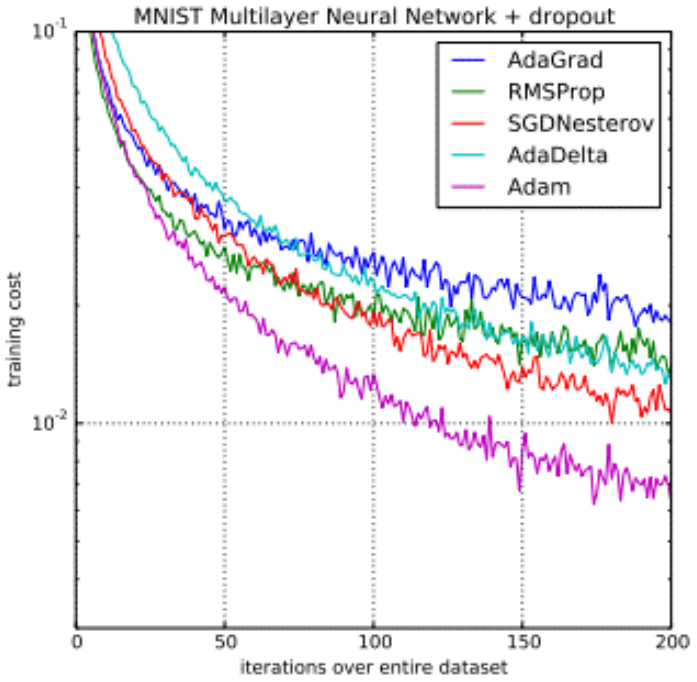


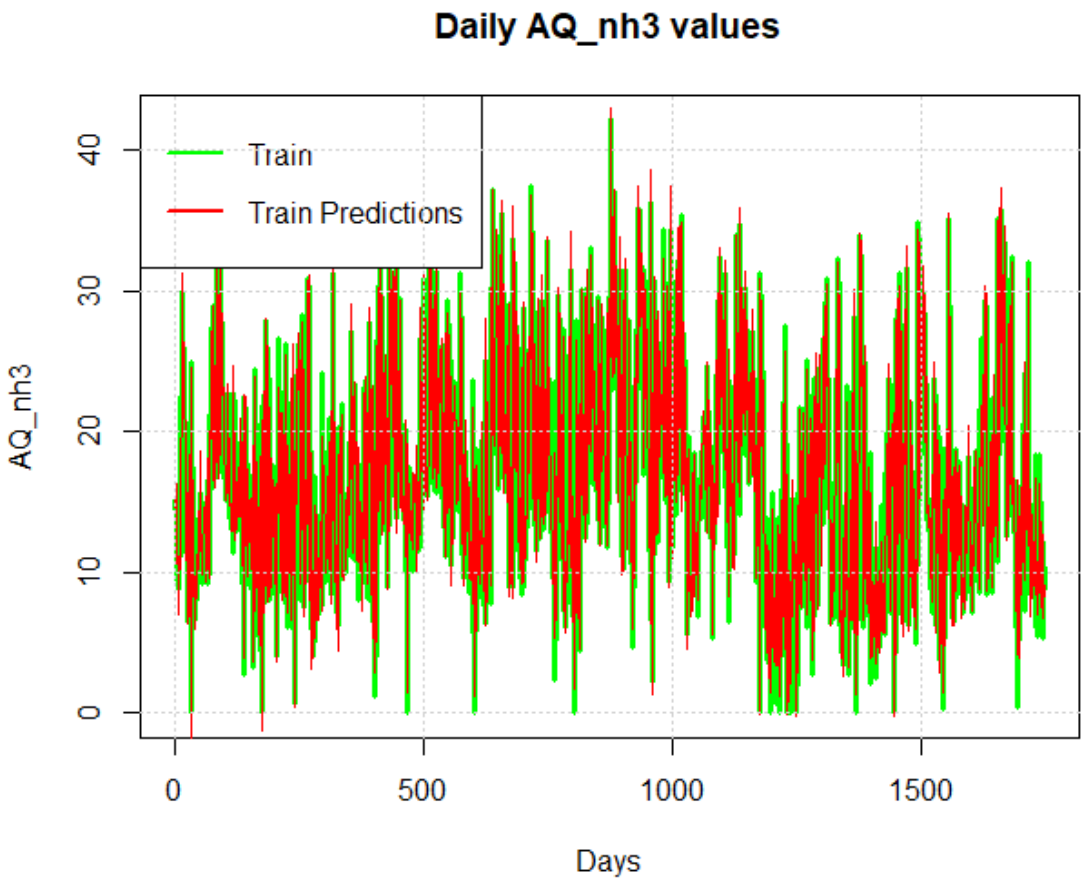
Figure: In "An overview of gradient descent optimization algorithms" Sebastian Ruder says: "Adam might be the best overall choice" **[9]**

Epochs	train $RMSE$	validation $RMSE$	Shapiro test	Breusch-Pagan Test
200	1.331	7.010	2.772e-05	0.253e-03
100	1.238	7.195	0.062	0.207e-03
50	2.223	7.173	4.409e-05	1.556e-05

Figure: tuning on `num_epochs` and results

Model	weekly	monthly	annual
200	0.025	0.543	0.299
100	0.058	0.355	0.065
50	2.116e-05	0.018	0.373

Figure: Ljung-Box tests executed to see any weekly, monthly or annual correlation between residuals



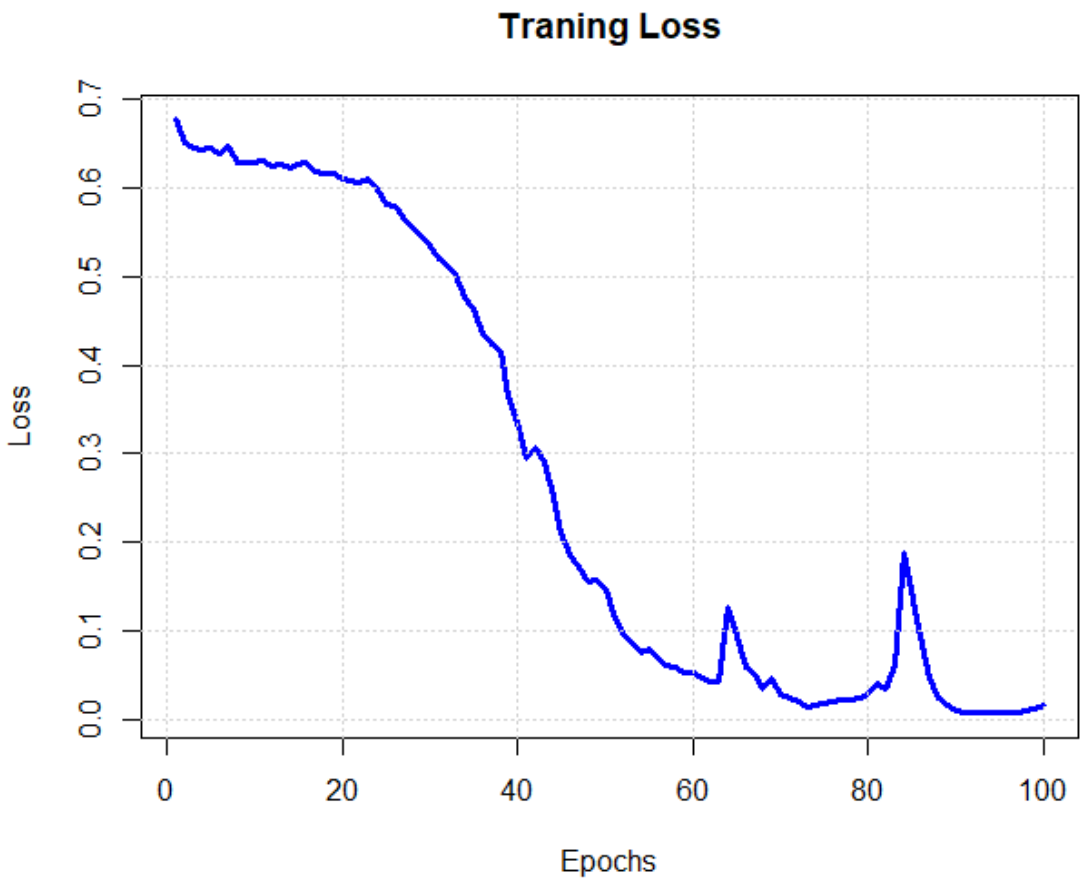


Figure: training LSTM network minimizing the loss function

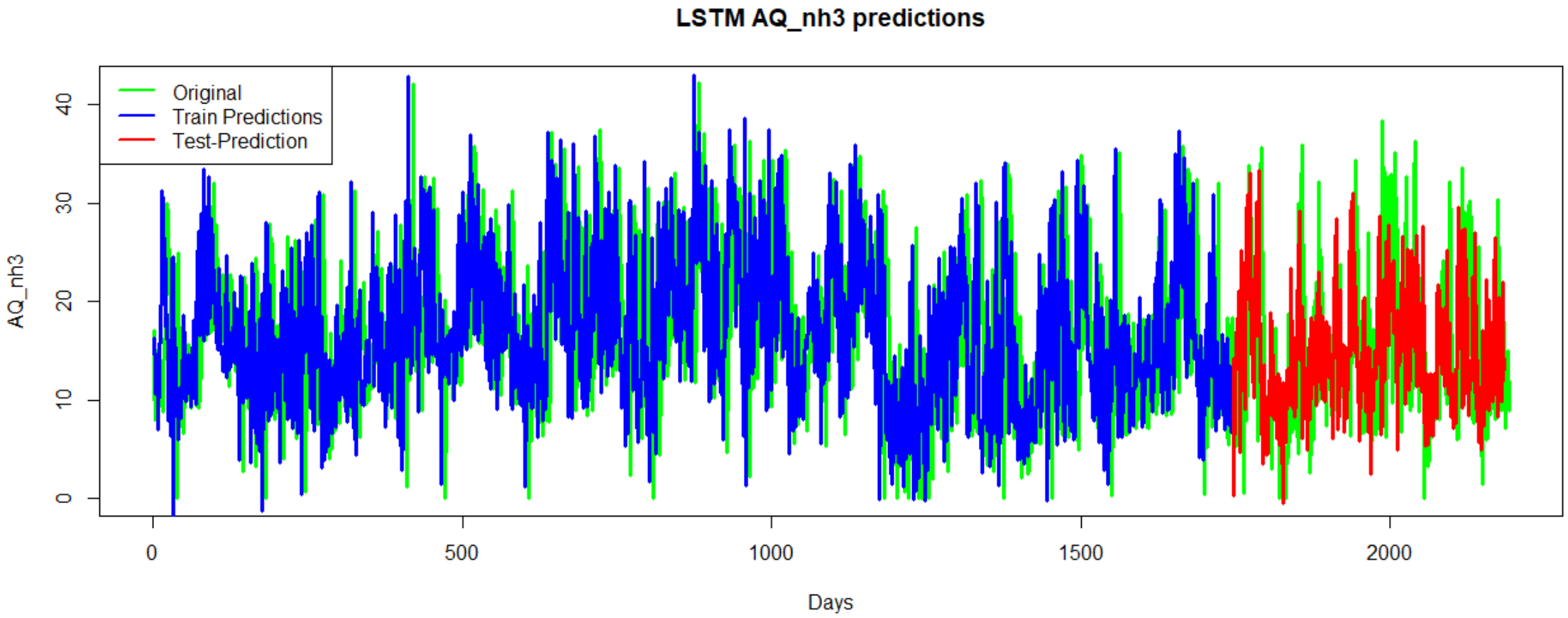


Figure: predicted AQ_nh3 by the LSTM model

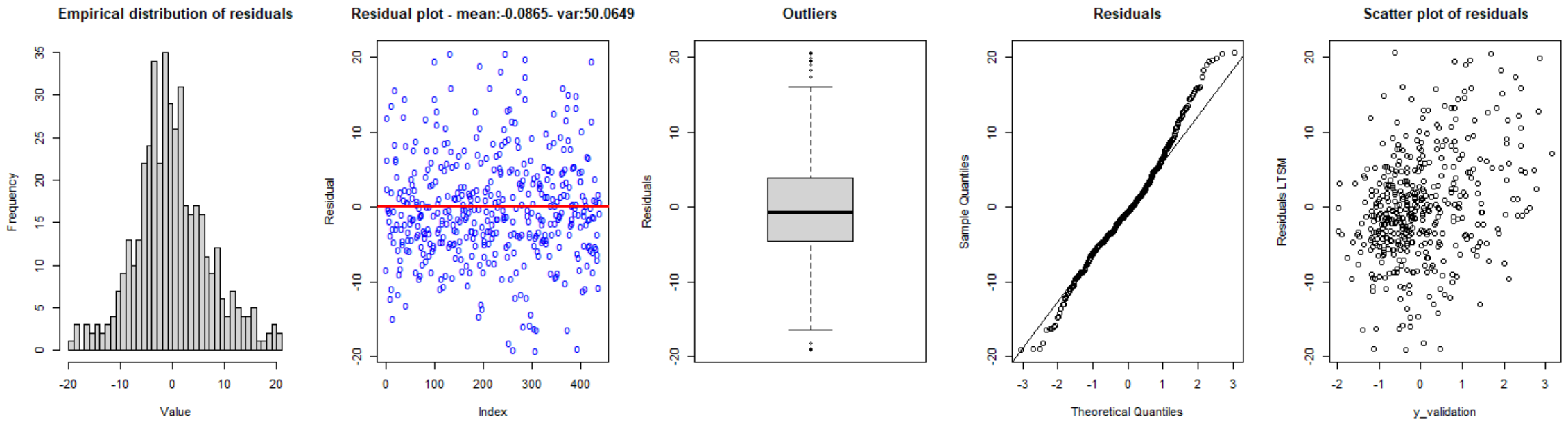


Figure: Residual behavior in validation

Machine learning conclusions

Model	train $RMSE$	validation $RMSE$	Shapiro test	Breusch-Pagan Test
XGBoost	4.281	6.934	1.227e-08	0.186
Prophet	6.772	7.924	4.721e-09	0.240
LSTM	1.238	7.195	0.062	0.207e-03

Figure: model statistics

Model	weekly	monthly	annual
XGBoost	2.200e-16	2.200e-16	2.200e-16
Prophet	2.200e-16	2.200e-16	2.200e-16
LSTM	0.058	0.355	0.065

Figure: Ljung-Box results

Packages

- library(bboot)
- library(ggplot2)
- library(bboot)
- library(forecast)
- library(imputeTS)
- library(stats)
- library(lmtest)
- library(tseries)
- library(xgboost)
- library(prophet)
- library(keras)

References

- **[1]** Gilbert P, Thornley P, Alexander S, Brammer J. Biomass gasification for ammonia production. International conference on polygeneration strategies; Sep 2009; Vienna. 2009
- **[2]** Fassò, Alessandro, et al. "Agrimonia: a dataset on livestock, meteorology and air quality in the Lombardy region, Italy." Scientific Data 10.1 (2023): 143.
- **[3]** A. Fassò, «AgrImOnIA: Open Access dataset correlating livestock and air quality in the Lombardy region, Italy». Zenodo, mag. 31, 2023. doi: 10.5281/zenodo.7956006.
- **[4]** Boufidi, E., Lavagnoli, S., & Fontaneto, F. (2020). A probabilistic uncertainty estimation method for turbulence parameters measured by hot-wire anemometry in short-duration wind tunnels. Journal of Engineering for Gas Turbines and Power, 142(3), 031007.
- **[5]** Guo, Rui, et al. "Degradation state recognition of piston pump based on ICEEMDAN and XGBoost.“ Applied Sciences 10.18 (2020): 6593
- **[6]** Prophet: forecasting at scale, By: Sean J. Taylor, Ben Letham, February 23, 2017

- **[7]** Loza, J. M. S. (2019). Shape sensing of deformable objects for robot manipulation (Doctoral dissertation, Université Clermont Auvergne [2017-2020]).
- **[8]** Time Series Analysis using Facebook Prophet in R Programming, geeksforgeeks.org, 22 Jul, 2020
- **[9]** Sebastian Ruder. An overview of gradient descent optimization algorithms, 2017.