Christian-Albrechts-Universität zu Kiel

Faculty of Mathematics and Natural Sciences
and
Faculty of Agricultural and Nutritional Sciences

Exposé

for Master thesis

Thesis title:

Correcting randomly shifting Compass Data by applying Machine Learning Methods in Time Series Analysis.

Supervised by:

Prof. Dr. Matthias Renz
Archaeoinformatics - Data Science
and
Prof. Dr. Martin Visbeck
Ozeanzirkulation und Klimadynamik

Name of the student: Razeeb Sarker
Address: Hansastr. 84, 24118 Kiel
Hauptfach: Environmental Management
email: stu218072@mail.uni-kiel.de
Immatriculation: 1139932

Table of Contents

List of Figures

1 Steps in Flow-chart

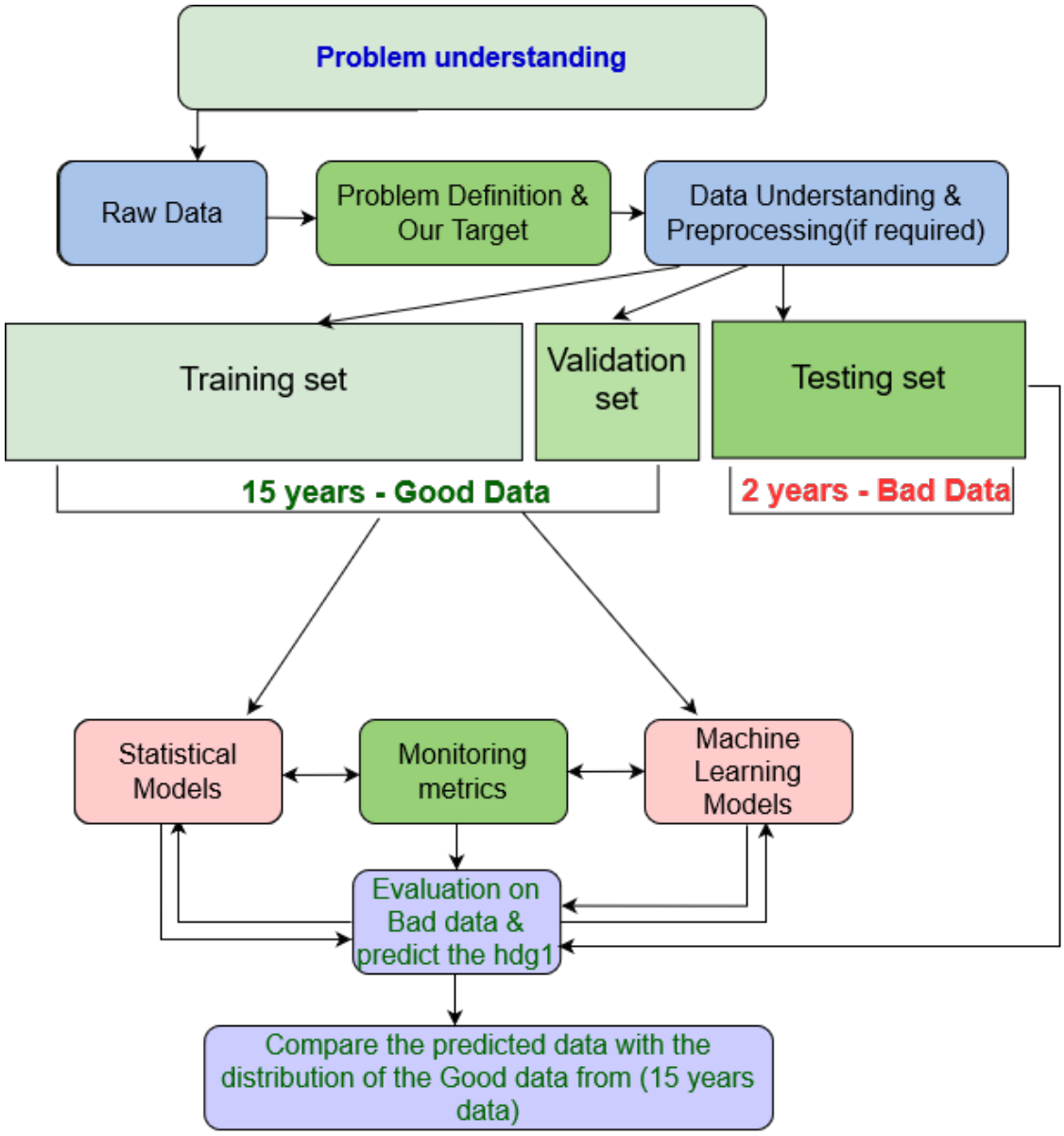

Figure 1: Steps in flow-chart

2 Problem Definition

As a part of the long-term ocean current surveillance project near New Cape Verde (approx. 23 degrees west longitude), two ADCPs (Acoustic Doppler Current Profiler) have been recording Equatorial Undercurrent (EUC) for the last 17 years (from Dr. Gerd Krahmann's problem-description.txt file). The devices collect the velocity and oscillation of underwater current, and to facilitate these measurements the devices also record other measures such as roll, pitch, device rotations at the same timestamp. Hence, the whole system's recorded data can be viewed as a multidimensional dataset Readings are collected from both ADCP sensors and stored in the device memories. Every two years the sensors are recovered from the underwater and the data is collected, and after necessary maintenance, the devices (or new devices) are again deployed on their respective position underwater. Doing so, in the last two years' data, we found that there are some irregularities and some sort of shocks and level shifts in the measurements (we term it as bad data). As the devices were in their fixed position for the whole time, and at the same time we do not have the ground truth about what was happening with the devices, but we want to get the measurements corrected for these two years.

The problem can visually be seen from Figure [2] and Figure [3] The green one is the good data, and the red one is the bad data (last 2 years' data). The bad data seems to come from a different distribution (mixture of two normal-distribution) than the normally distributed good data.

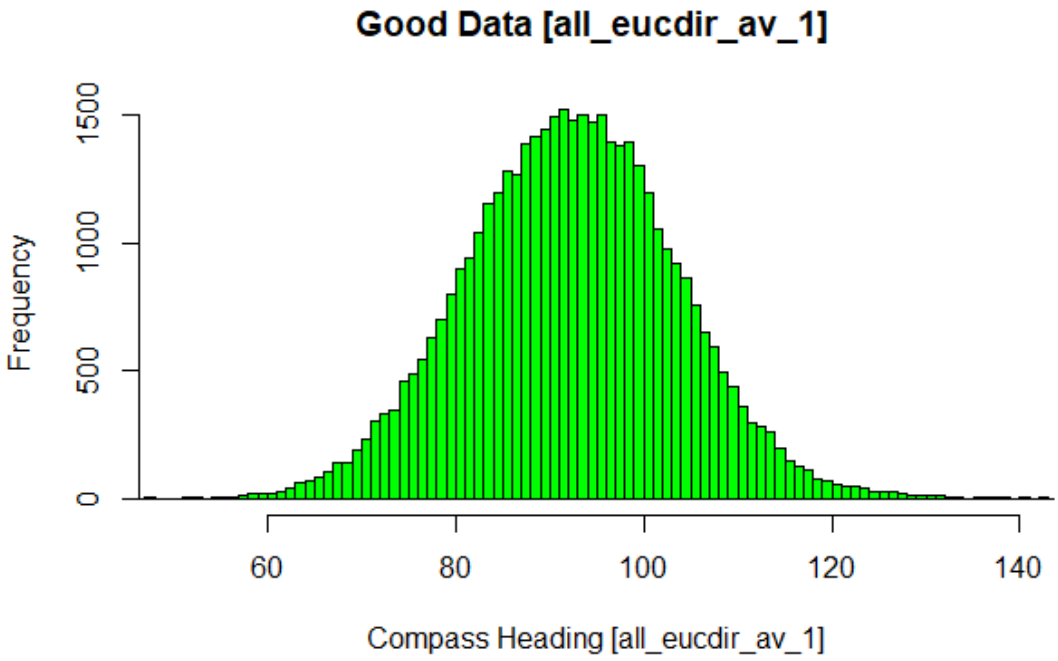

Figure 2: Histogram of Good Data (15 years data)

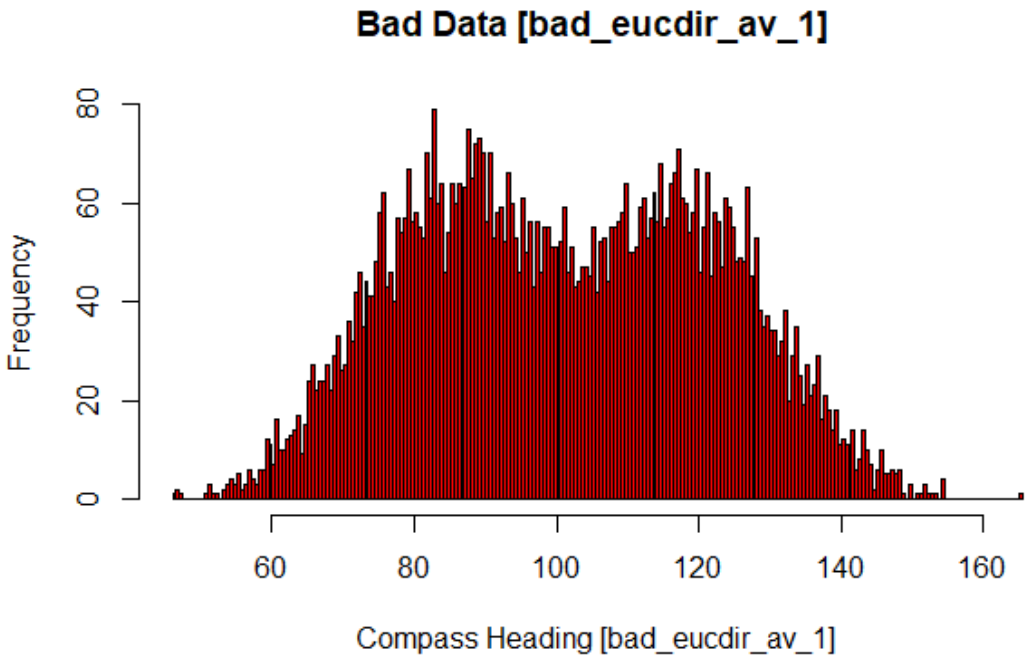

Figure 3: Histogram of Bad Data (last 2 years data)

3 Description of the variables:

Variable(s)	Description
all_eucdir_av_1	Equatorial Undercurrent (EUC) direction from the upward device
allhdg1	Compass heading of the upward looking device
allhdg2	Compass heading of the downward looking device
allpit1	Pitch from the upward looking device
allrol1	Roll from the upward looking device
allt2	Timestamp
allu1	East-west component of the upward looking device
allu1_rot	Rotation of the upward looking device (relative to the east-west u component)
allu2	East-west component of the downward looking device
allv1	**North-south** component of the upward looking device
allv1_rot	Rotation of the upward looking device (relative to the north-south v component)
allv2	**North-south** component of the downward looking device

Figure 4: Description table of the variables

Figure [4] shows the variable descriptions for the good data.

For the bad data, the description of the variables remains the same, the only difference is that these measurements were collected in the last two years from the problematic device.

4 Ideas for solution:

In this work, I will try to forecast the approximate values of a variable called "**eucdir_av_1**". As I will be dealing with multidimensional data, this problem can be turned into a supervised multivariate time series problem. The last 15 years' data that we already had rated as to be good data, will be our complete dataset for training (train-set & validation-set), and "**all_eucdir_av_1**" will be our training label. And the recent two years' data that have irregularities will be used as a test-set/or evaluation set. The main advantage in this approach is that we can test/evaluate our model(s) on completely new and unseen test-sample (bad dataset which is completely unknown to the trained model). In the end, I will be predicting the variable called "**bad_eucdir_av_1**" from a model trained on the good data from the last 15 years.

The train, validation, and test can be viewed as shown in Figure [5].

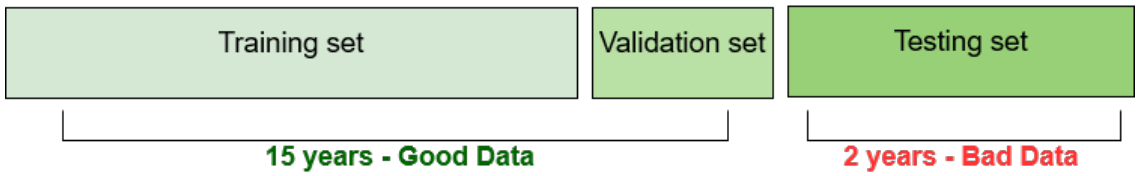

Figure 5: Data split for train, validation and test

5 Performance Measurement

Metrics are the key indicators of evaluating the performance of machine learning models. There are different choices of metrics for both classification and regression problems. For our task, regression problem, there are few popular metrics. Among them, MAE (Mean Absolute Error) and MAPE (Mean Absolute Percentage Error) are the most common.

Different statistical or machine learning models trained on the same data and evaluated on

the same metric are a clear indicator of the comparative performance of those models. The selection of appropriate evaluation metrics is also equally important with regard to the type of problem we are dealing with. Depending on the understanding of the data such as, the distribution, data patterns, final aim of the model, missing and extreme values of the data, the evaluation metrics should be selected. Sometimes it is customary also to use multiple metrics or a combination of matrices.

5.1 Mean Absolute Error (MAE):

The advantage of using MAE is that it can tolerate extreme values, MAE is a more natural measure of average error, is unambiguous, and is very easy to interpret [1].

$$\boldsymbol{MAE} = \frac{\mathbf{1}}{\boldsymbol{N}} \sum_{\boldsymbol{i=1}}^{\boldsymbol{N}} |\boldsymbol{y_i} - \boldsymbol{\hat{y}_i}|$$

Here,

$\boldsymbol{N}$ = number of training samples,
$\boldsymbol{y_i}$ = true label of the ith sample
$\boldsymbol{\hat{y}_i}$ = predicted value for ith training sample

5.2 Mean Absolute Percentage Error (MAPE):

MAPE is also an easily interpretable performance metric in machine learning, even well understood by non-technical persons as it shows the accuracy in percentage. There is one big disadvantage of using MAPE is that when the true value (y_i, the denominator) becomes zero or close to zero, it produces an infinite value [2] which may affect the model training and monitoring after deployment.

$$\boldsymbol{MAPE} = \frac{\mathbf{100}}{\boldsymbol{N}} \sum_{\boldsymbol{i=1}}^{\boldsymbol{N}} |\frac{\boldsymbol{y_i} - \boldsymbol{\hat{y}_i}}{\boldsymbol{y_i}}|$$

Here,

N = number of training samples,

y_i = true label of the ith sample

$\hat{y}_i$ = predicted value for ith training sample

5.3 Mean Squared Error (MSE):

MSE calculates the squared distance between the actual and the predicted values. As it squares the distances, it is more prone to outliers [3]. MSE is also used as the most common loss function (the objective function during the training and optimization of the ML model). The main difference is here when MSE is used as a loss function, it is differentiated to calculate the gradient during optimization whereas when used as an only metric it does not require to be differentiated.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

Here,

N = number of training samples,

y_i = true label of the ith sample,

$\hat{y}_i$ = predicted value for ith training sample

6 Literature review

Time series analysis is a technique to analyze time-dependent data points. This is different from other types of predictions and machine learning in a way that the sequence of the data points is equally important along with the values of corresponding discrete timestamps. In statistical approaches of time series analysis, autocorrelation is another important property of time series. Time series forecasting is used in various fields but not limited to finance, weather forecasting, economics, applied science, and signal processing, etc. Time series is also useful in environmental modeling when there is less ground truth or absence of relationships among the variables in a dynamical system. For our case, we are trying to find a relationship among the few physical oceanographic variables of ADCP. In other words,

we are trying to solve a physical oceanographic problem by turning it into a statistical or machine learning problem.

There are dozens of statistical and machine learning approaches to deal with multivariate time series. But it is hard to find a generalized model structure that fits all types of temporal data. This uncertainty in the model selection mainly comes from the nature of the data and pre-defined problems.

In the statistical approaches, vector autoregression, VAR [4], [5] and its extension such as VAR with exogenous variable influence, VARX [6], and Vector autoregressive moving-average, VARMA [7] and its VARMAX variant for exogenous effects are most popular.

In the neural networks approaches recurrent neural nets, and their extensions such as long-short time memories, LSTM [8], gated recurrent units, GRU [9]. Recently, decoder-encoder models such as time series generative adversarial networks (TimeGAN) have become popular. When a data point in time series is dependent on the previous as well as on the next values, in practice it is customary to use Bidirectional LSTM. In this case, during the training, the model gets information from both previous and from the next data or data points (based on the length of the sliding window).

6.1 Vector Autoregression (VAR):

Vector autoregression, VAR is the generalization of the autoregressive process of a univariate time series where a value at time t is calculated from the previous time(s) values as well as from the previous time (s) value of another univariate series. Say, we have two univariate time series processes y1_t, and y2_t, and we assume that they influence each other. We can write their VAR(1) process for only one lag period separately as follows,

$$\boldsymbol{y_{1,t}} = \boldsymbol{c_1} + \boldsymbol{a_{1,1}}\boldsymbol{y_{1,t-1}} + \boldsymbol{a_{1,2}}\boldsymbol{y_{2,t-1}} + \boldsymbol{e_{1,t}}$$

$$\boldsymbol{y_{2,t}} = \boldsymbol{c_2} + \boldsymbol{a_{2,1}}\boldsymbol{y_{1,t-1}} + \boldsymbol{a_{2,2}}\boldsymbol{y_{2,t-1}} + \boldsymbol{e_{2,t}}$$

$\boldsymbol{c_1}, \boldsymbol{c_2}$, are the intercept for the each series,

$\boldsymbol{a_{1,1}}, \boldsymbol{a_{1,2}}, \boldsymbol{a_{2,1}}, \boldsymbol{a_{2,2}}$, are the coefficient for respective series and lag(s),

$e_{1,t}, e_{2,t}$ are the error terms for the each series (error should have a zero mean and uncorrelated across the any previous time)

This two equation can be rewritten in matrix notation as follows-

$$\begin{bmatrix} y_{1,t} \\ y_{2,t} \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} + \begin{bmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{bmatrix} \begin{bmatrix} y_{1,t-1} \\ y_{2,t-1} \end{bmatrix} + \begin{bmatrix} e_{1,t} \\ e_{2,t} \end{bmatrix}$$

6.2 VARX (VAR with exogenous variable):

$$y_t = \sum_{\ell=1}^{p} \Phi_\ell y_{t-\ell} + \sum_{j=1}^{s} B_j x_{t-j} + \epsilon_t,$$

the first part of the equation is the VAR part, the second part of the equation is the exogenous variable, and the third part is the error (theoretically, a white noise) [10].

6.3 Cross Validation:

Before the data goes into the machine learning model, normally we split the whole data set into two parts, one is the training set and the other is the validation set (the test set is different for what we make the final prediction). But, there is a problem. When we split data, for example, 80% and 20%, there might be a situation where the model is doing well in the test set, but when we provide the real world data/or completely new samples to evaluate the model (for our case the bad data) the model might not perform well. To overcome this problem cross validation is a common technique. Cross validation brings stability in the model performance [11], and the model comparatively better generalised when sees a new example in deployment.

There are different techniques of cross validation and k-fold cross validation is the most commonly used one. Here, at first the value of k, the number of splits in the dataset is

Figure 6: K-Fold Cross Validation

decided. Then the model is trained on k different train and test sets. At the same time, the accuracy of each model is tracked. In Figure [6] (source: [12]) a simple visualization of k-fold cross validation can be seen.

But, k-fold cross validation has a small disadvantage when used for time series forecasting. k-fold cross validation does not maintain the temporal sequence of the data, but we want to keep our time dependencies in the time series forecasting. To overcome this problem there is a technique called walk-forward validation. The idea is very simple and intuitive, at first we train the model with less data (as per the window size we decide) and then we retrain the model with the new predictions we get over the time. It can be clearly seen from Figure [7] and Figure [8].

We can keep the window size fixed or let it grow with the availability of a new sample. Both are possible, and depends on the model accuracy and training time [13].

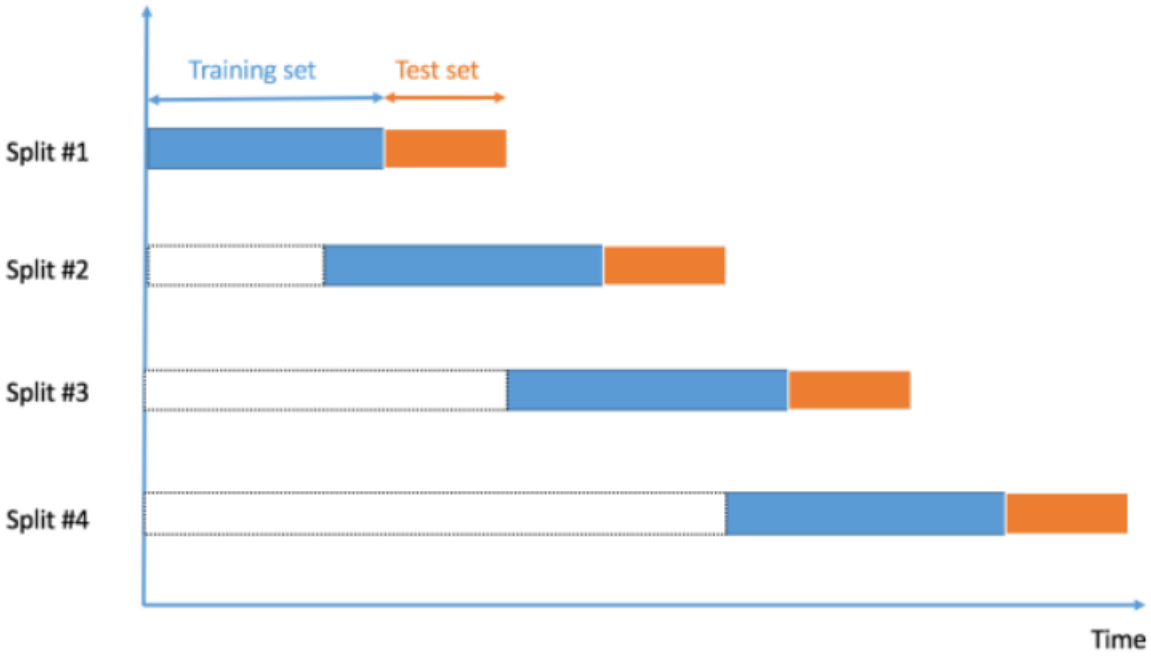

Figure 7: Walk-Forward validation with sliding window

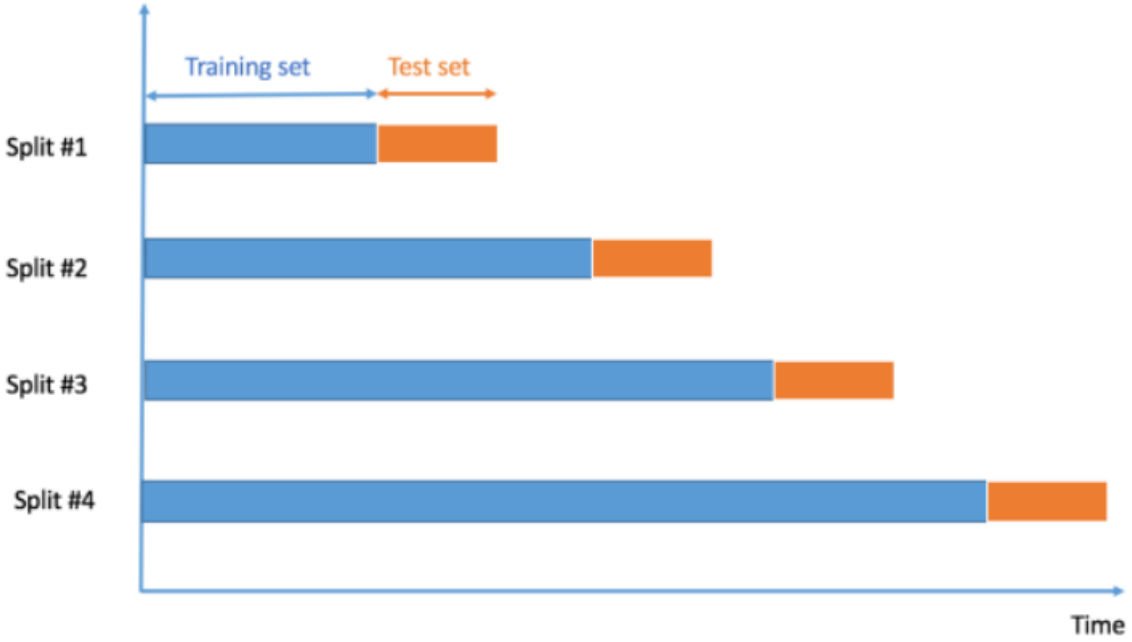

Figure 8: Walk-Forward validation with growing window

7 Structure of the Thesis

- Introduction
- Theoretical background
  - Basics of time series
  - Statistical approaches:

    In the statistical approaches, we try to fit the data we have into some predefined rules. Statistical learning requires less time and resources compared to the machine learning approach for forecasting. Based on the task and given data, sometimes statistical models may out perform a machine learning forecasting model.
  - Machine learning approaches: Here, we try to find out the underlying non-linear relationship between the given features and the labels. It is opposite to statistical learning, where we provide a predefined relationship.
- Model selection and performance measurement
- Results
- Discussion
- Conclusion
- Bibliography

8 Bibiliography

[1] C. J. Willmott and K. Matsuura, "Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance," *Climate research*, vol. 30, no. 1, pp. 79–82, 2005.

[2] S. Kim and H. Kim, "A new metric of absolute percentage error for intermittent demand forecasts," *International Journal of Forecasting*, vol. 32, no. 3, pp. 669–679, 2016.

[3] R. F. Gunst and R. L. Mason, "Biased estimation in regression: An evaluation using mean squared error," *Journal of the American Statistical Association*, vol. 72, no. 359, pp. 616–628, 1977.

[4] H. Lütkepohl, *New introduction to multiple time series analysis*. Springer Science & Business Media, 2005, pp. 24–26.

[5] R. J. Hyndman and G. Athanasopoulos, "Forecasting: Principles and practice," pp. 333–335, 2017.

[6] I. Wilms, S. Basu, J. Bien, and D. S. Matteson, "Interpretable vector autoregressions with exogenous time series," *arXiv preprint arXiv:1711.03623*, 2017.

[7] W. Scherrer and M. Deistler, "Vector autoregressive moving average models," 2019.

[8] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[9] d2l.ai, *D2l.ai*. 2021.Available: https://d2l.ai/chapter_recurrent-modern/gru.html

[10] I. Wilms, S. Basu, J. Bien, and D. S. Matteson, "Interpretable vector AutoRegressions with exogenous time series," *arXiv.org*. 2021.Available: https://arxiv.org/abs/1711.03623

[11] M. Stone, "Cross-validatory choice and assessment of statistical predictions," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 36, no. 2, pp. 111–133, 1974, doi: 10.1111/j.2517-6161.1974.tb00994.x.

[12] Datavedas, *Datavedas.com*. 2021.Available: https://www.datavedas.com/k-fold-cross-validation/

[13] R-blogger, *R-bloggers*. 2021.Available: https://www.r-bloggers.com/2020/03/time-series-cross-validation-using-crossval/

9 Appendices:

9.1 Appendix A: Citation Style Laguage

- Here, I used ‘`ieee-transactions-on-software-engineering.csl`’ csl style in the in-text citation, and in the bibliography. More information about the csl languages and avilable csl styles can be `found at this link`.