University College London
Faculty of Engineering Sciences

CEGEG076 Spatio-Temporal Data Mining

Spatio-Temporal Analysis on: Forecasting Tourist Data in Austria

Qiqing Huang, 17014982, ANN

Supratim Banik, 16118558, SVM

Florian Hübscher, 17135943, STARIMA

Claudio Navacchi, 18003316, RFR

Department of Civil, Environmental and Geomatic Engineering

4th May, 2018

Contents

1 Introduction

The aim of this report is to apply the methods and techniques presented in the lecture *"Spatio-Temporal Data Mining"* on a spatio-temporal dataset. A dataset description is provided in subsection 1.1. The subsequent section 2 provides an analysis of the dataset focusing on temporal (subsection 2.1), spatial (subsection 2.2) and spatio-temporal patterns (subsection 2.3).

The structure of the dataset restricts the applicability of the methods learned on spatio-temporal regression and prediction being the main focus of this work (subsection 1.2). The four methods chosen are: *Random Forest Regression (RFR)* (subsection 3.1), *Spatio-Temporal Auto-Regressive Integrated Moving Average (STARIMA)* (subsection 3.2), *Artificial Neural Networks (ANNs)* (subsection 3.3) and *Support Vector Machine (SVM)*(subsection 3.4). Liaw 2015 offers a documentation of random forests in *R* following the papers/manuals Breiman 2001 and Breiman 2002. Use case examples of *RFR* can be found in Liaw, Wiener, et al. 2002 and Segal 2004. *STARIMA* is proposed and documented in Pfeifer and Deutsch 1981. Recent applications and extensions are described in Islam-Khan, Landfeldt, and Damdhere 2012, Cheng et al. 2014 and Duan et al. 2016. Systematic introduction of ANNs can be found in Rojas's book (2013). The application of ANNs in forecasting has been well concluded by Zhang, Patuwo, and Hu (1998) and has been widely used in tourism area (Law and Au 1999, Burger et al. 2001, O. Claveria and S. Torra 2014). In case of the investigation using SVM/SVR, the research paper on application of support vector regression on tourism by M. Claveria and Torra 2016 and Witt and Witt 1992 provided a rich source for reference. While the tutorial on support vector machine by Scholkopf and Smola 2002 helped in understanding the SVR methodology.

Finally, all methods will be compared regarding meaningful statistics used for time-series analysis and an outlook for improvements will be given (section 4).

1.1 Dataset Description

Tourist data in Austria including two spatio-temporal variables, namely tourist arrivals and nights spent (stays), was chosen to be the dataset of interest [1]. It is available monthly-wise from 11/1973 until 11/2017. The spatial resolution is defined by the nine provinces of Austria. The variables are further subdivided by the country of origin. An exemplary depiction of the first three rows is shown in table 42.

Table 1: Data excerpt of the Austrian tourist dataset (top three rows).

Date	Province Code	Country of Origin Code	Arrivals	Stays
197311	W96-1	01	1792	14288
197311	W96-1	02	3845	18901
197311	W96-1	53	2	23

Figure 1 shows all tourist stays and arrivals aggregated by year. A spatial view is given by fig. 2.

1.2 Experimental Setup

It has been decided to predict the stays, since it is a more valuable parameter for tourism than arrivals. Each regression model should be trained and validated from 11/1973 to 12/2010 and from 01/2011 to 11/2017, respectively. All countries of origin were aggregated province-wise over time to simplify the dataset. Additionally, prediction success should be evaluated by using statistics such as RMSE and R^2 and n-step ahead instead of one-step ahead prediction.

[1]download available under https://www.data.gv.at/katalog/en/dataset/128e3b47-1e44-3791-bee2-3b06b2cd58f4

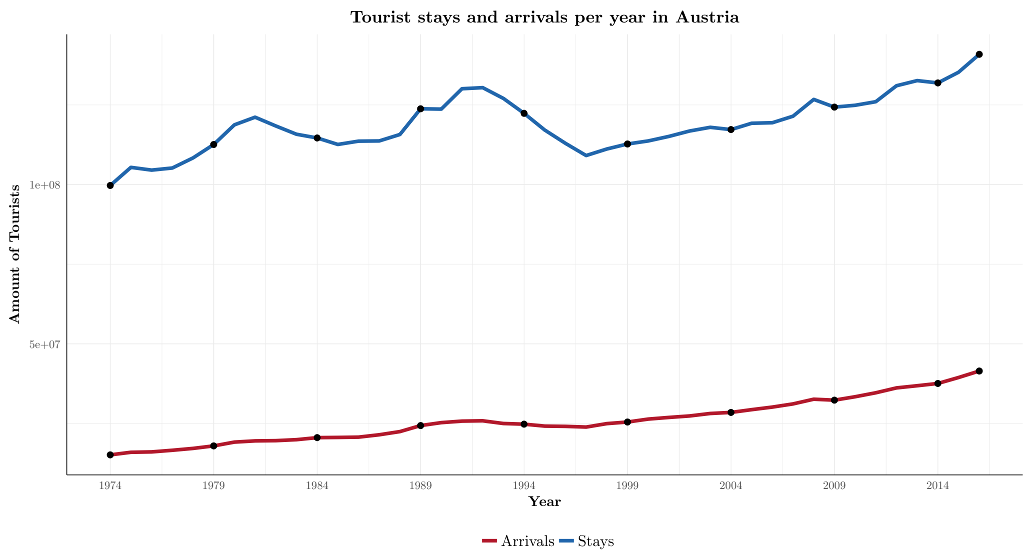


Figure 1: Aggregated tourist data depicting stays over night and arrivals. A clear trend is visible for both, whereas stays are characterised by larger fluctuations over time (weather conditions, advertisment, sport events, ...). A trend could be also caused by more accommodations taking part (or must take part) in the national collection of tourist data.

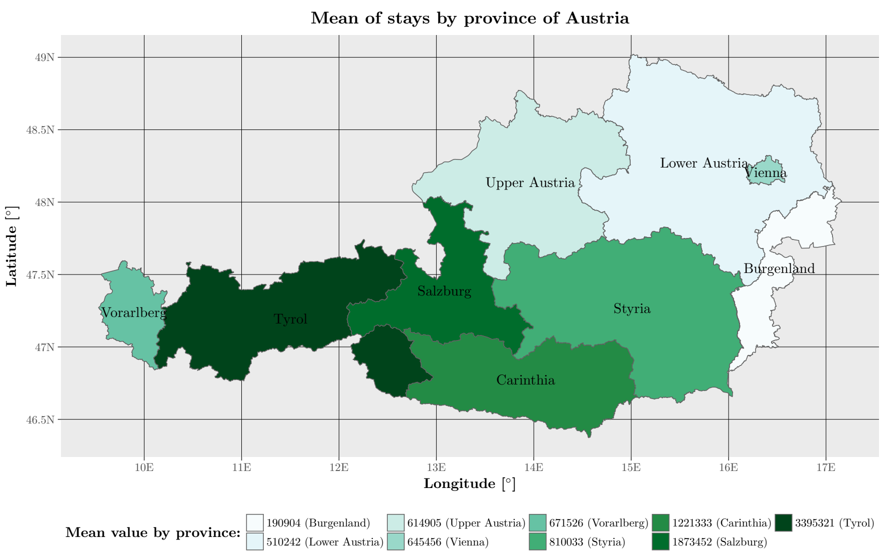


Figure 2: Austrian map showing the mean value of tourist stays per province from 1973-2017. The mountainous regions in the western part of Austria offer a broad spectrum of leisure activities, which is more attractive for longer stays, followed by Vienna being popular for its culture and history (short-time visits). All other provinces seem to be of minor importance for tourists.

2 Exploratory Spatio-Temporal Data Analysis

2.1 Temporal Patterns

The heatmap in figure 3 provides an overview of overnight tourist stays across Austria. It can be seen that Tyrol, Carinthia and Salzburg have the highest values and some provinces have two peaks while others have one. Carinthia is the most significant province having a descending trend.

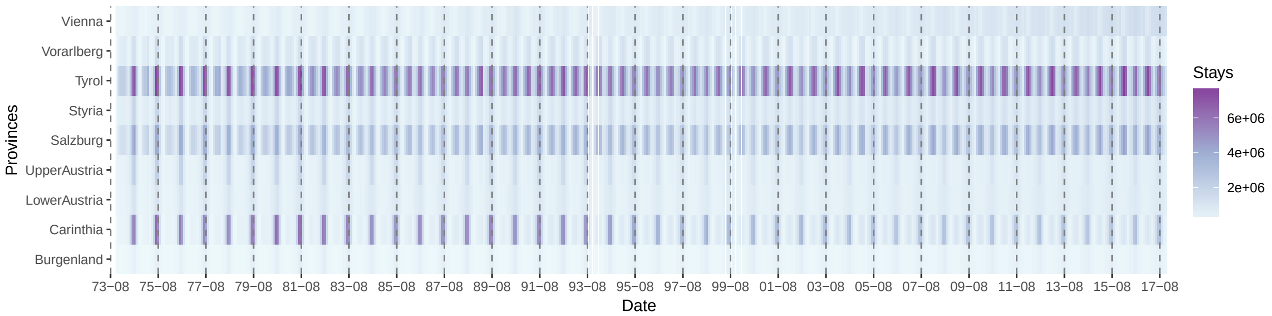


Figure 3: Heatmap of overnight stays time series.

The time series of overnight tourist stays is decomposed into a trend, seasonal and random part over time, which allows for a better understanding of stays behaviour for different provinces (see figure 4). The basic equation of the classical additive decomposition method is as follows:

$$y_t = T_t + S_t + E_t$$

where trend data T is computed using moving averages and seasonal data S is based on average detrended values for certain periods. The time series of Vienna shows a strongly rising trend starting approximately in 2010.

The autocorrelation function (ACF) shows for all 9 counties a predominant seasonality of order 12 (see figure 5). Using monthly data this corresponds to an annual cycle. For Vienna all values are positive but slightly decaying, whereas the others alternate between negative and positive values. This suggest that the time series of Vienna is not stationary yet. Some counties show additional peaks at lag 6, 18, 30, ... which suggest that their summer and winter tourism is significant. The corresponding partial autocorrelation functions (PACF) show consequently a large peak at lag 12 (see figure 6), but usually also larger values for lag 1, 11 and 13 suggesting that the neighbouring months (and their equivalent shifted by 12 months) have an influence on the number of stays for any specific month.

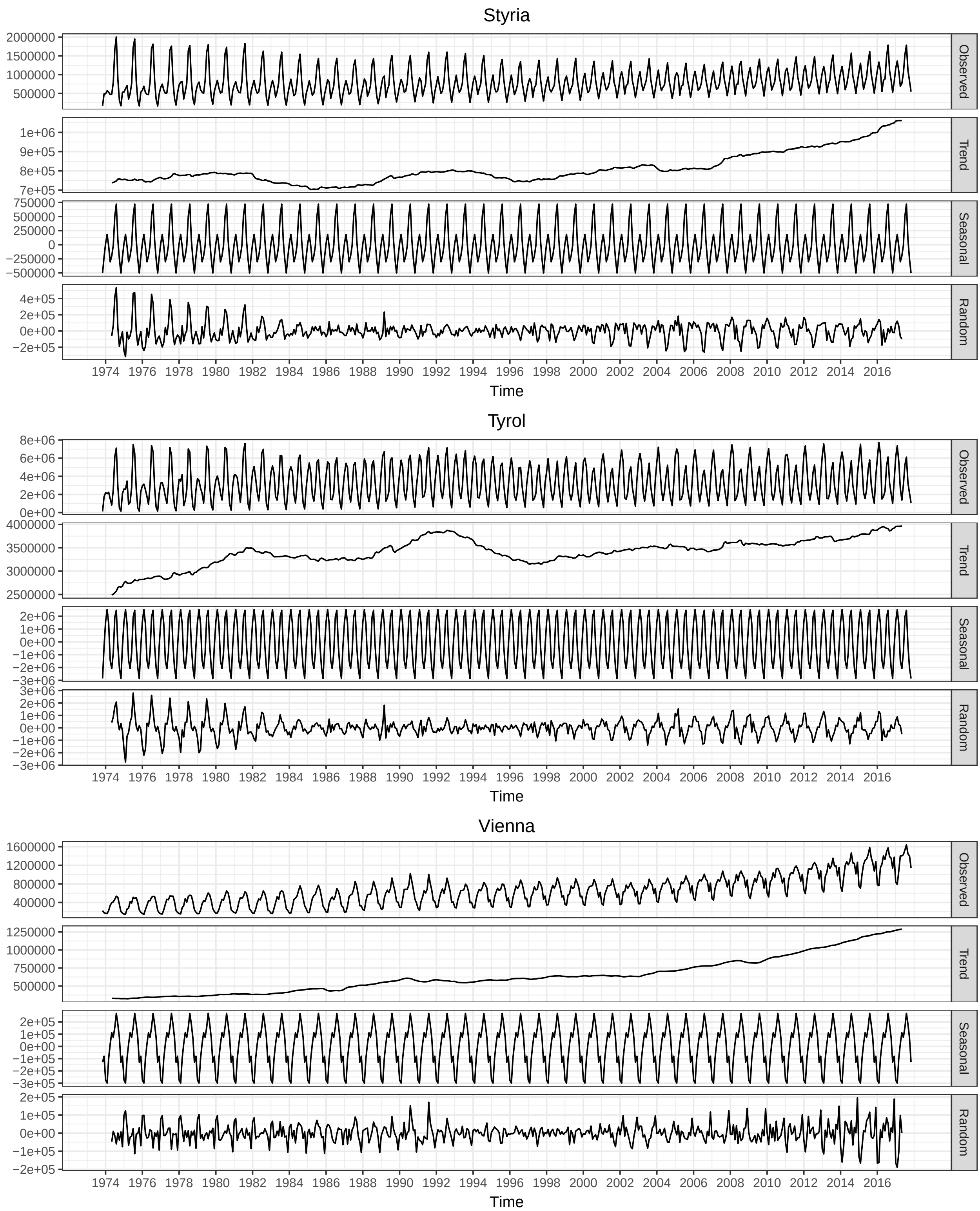


Figure 4: Decomposition of additive tourist stays time series.

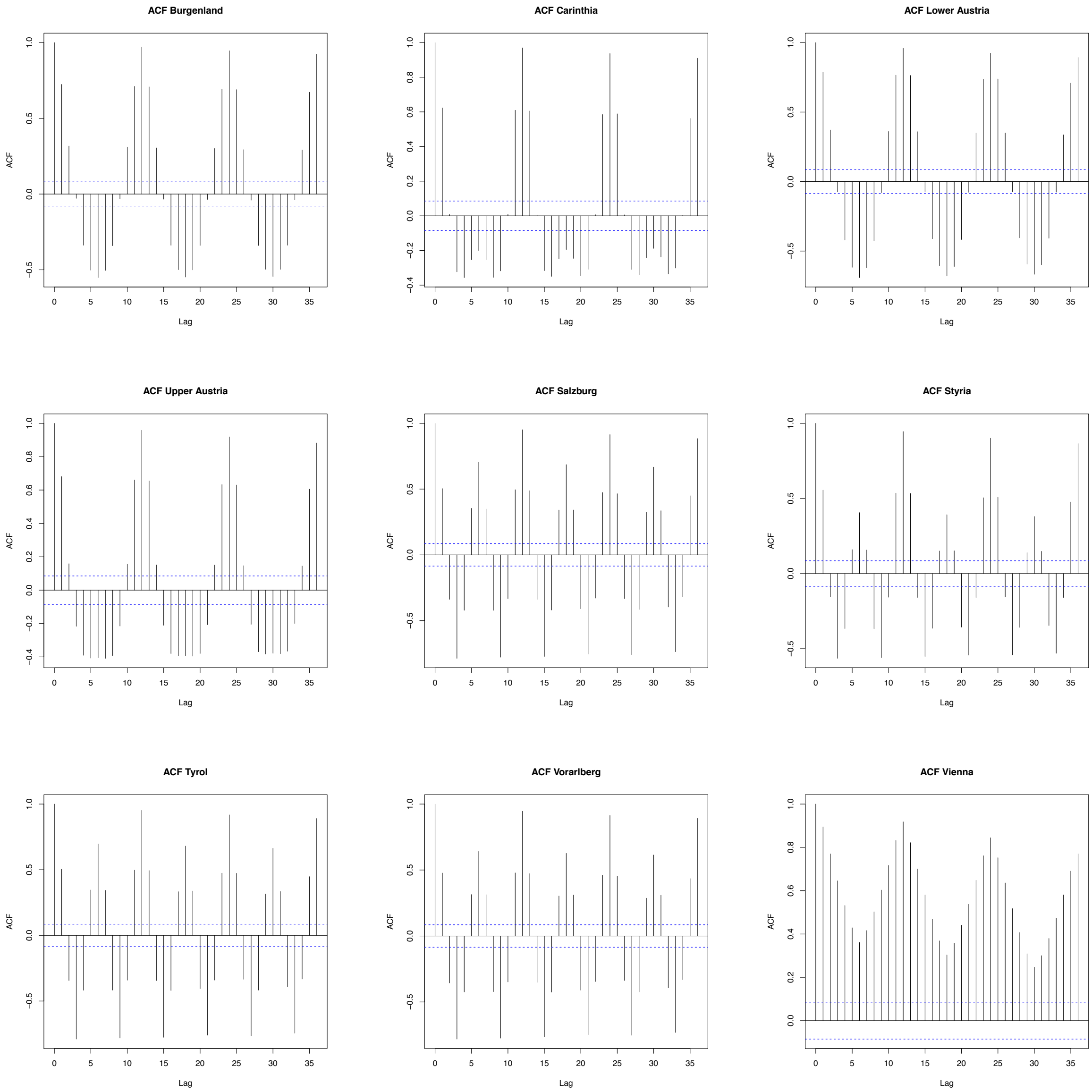


Figure 5: Autocorrelation functions of raw time series for all nine provinces (the blue dashed lines show the 95% confidence intervals).

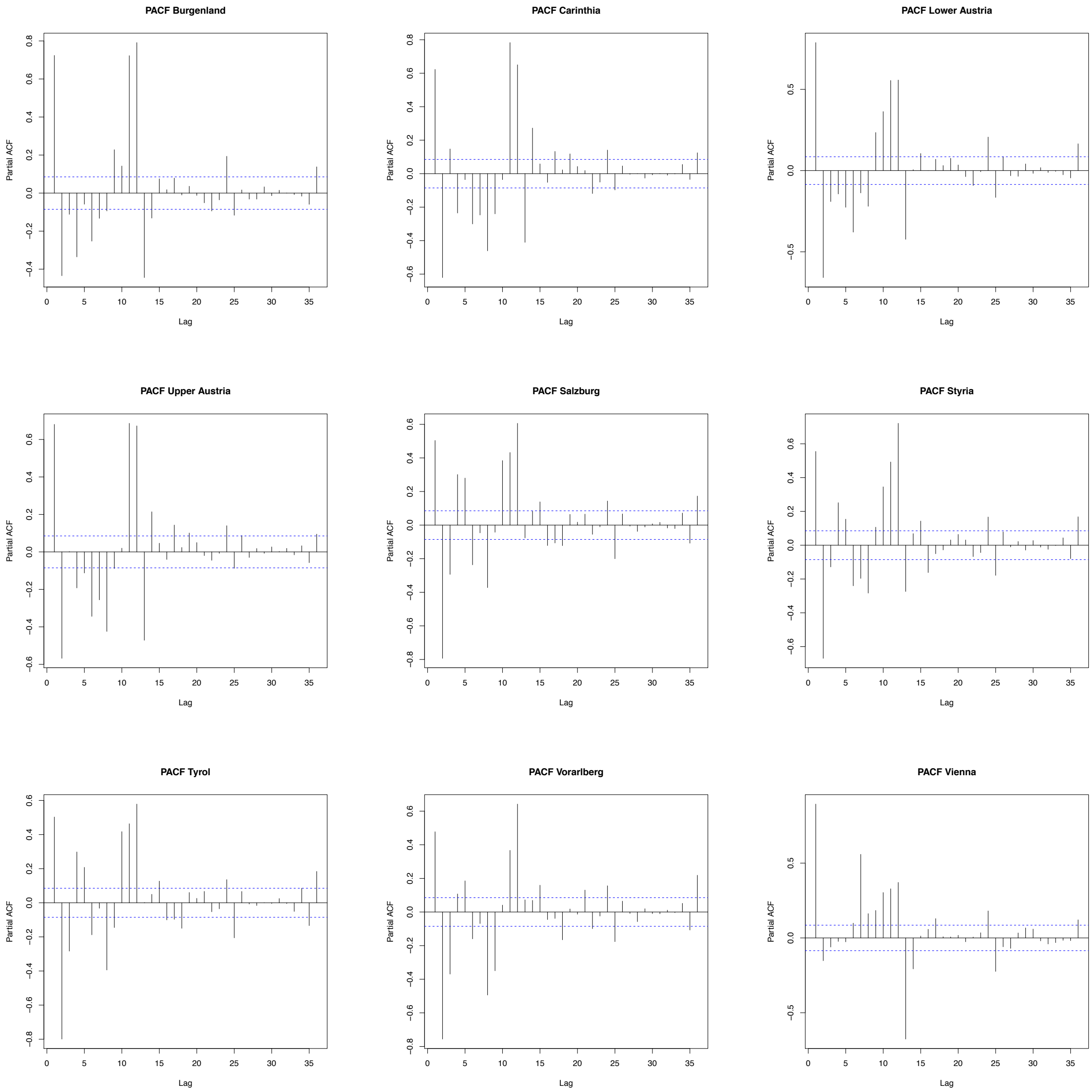


Figure 6: Partial autocorrelation functions of raw time series for all nine provinces (the blue dashed lines show the 95% confidence intervals)

2.2 Spatial Patterns

2.2.1 Distribution of Stays

Figure 7 shows the histograms of the stays distribution. A marked asymmetry can be seen in Burgenland, Carinthia, Lower Austria and Upper Austria. Two peaks are present in Salzburg, Tyrol and Vorarlberg.

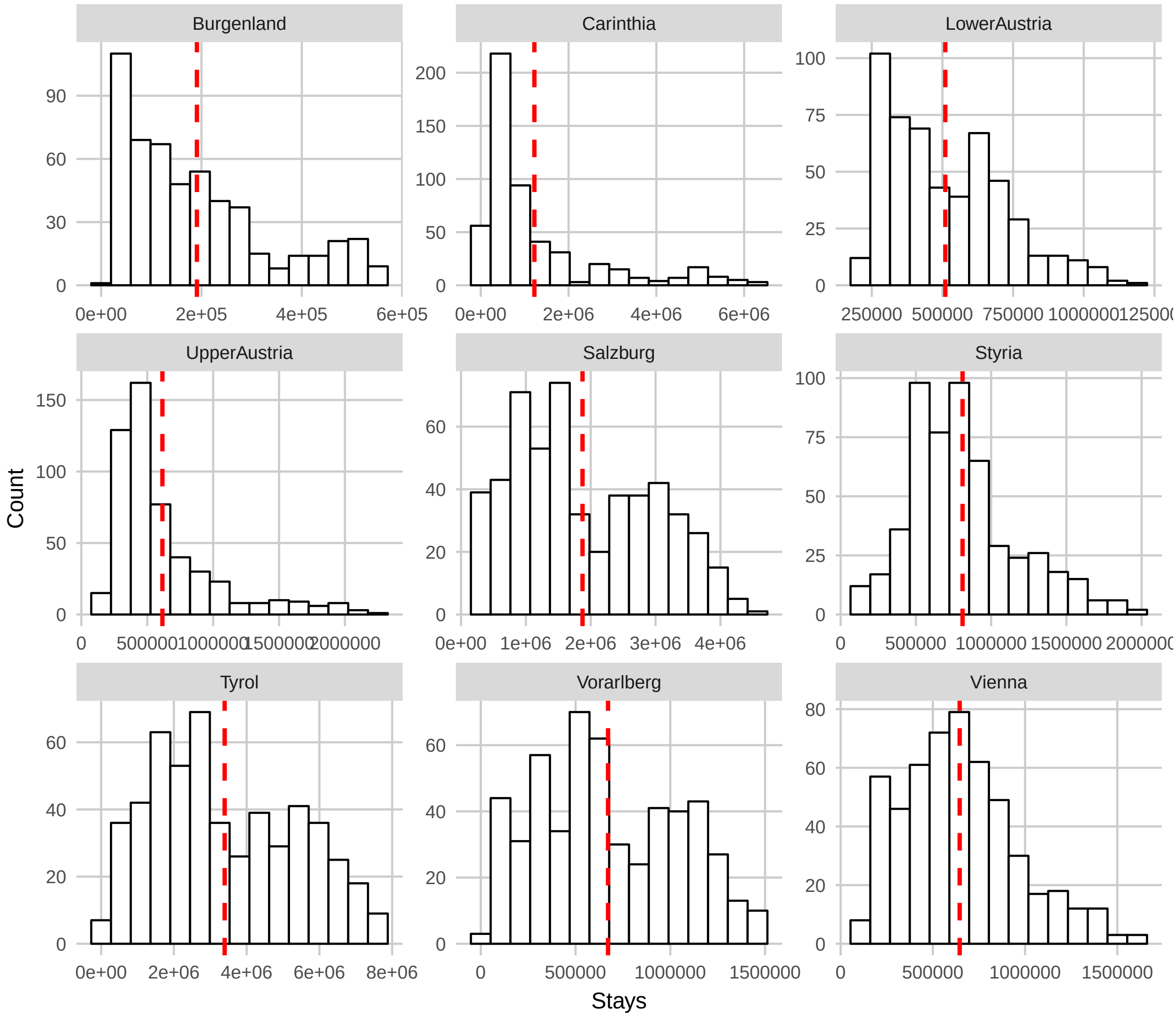


Figure 7: Histograms of stays. The red vertical line represents the mean of stays.

2.2.2 Moran's I

Moran's I is a significant indicator of presence of spatial autocorrelation in the data. This index is estimated for arrivals and stays and is as follows in table 2.

Table 2: Comparison of Global Moran's I between monthly tourist arrivals and overnight stays in Austria from 1974-2017 .

Variable	**Moran's I**	**Expectation**	**Variance**	**p-Value**	**Alternative Hypothesis**
Stays	-0.04715124	-0.12500000	0.03713860	0.3431	greater
Arrivals	-0.07697893	-0.12500000	0.04322940	0.4087	greater

The p-values in both cases of Moran's I are not significant. Hence, there is not much spatial autocorrelation and the null hypothesis that the processes are happening due to random chance cannot be rejected. Both Moran's I values are negative, which indicates that there is dispersion rather than clustering. To investigate further, the local Moran's I is calculated for any evidence of spatial autocorrelation at local level (see table 3).

Table 3: Local Moran's I for overnight stays, 1974-2017.

Province	$\mathbf{I}_i$	$\mathbf{E.I}_i$	$\mathbf{Var.I}_i$	$\mathbf{Z.I}_i$	**Pr(z > 0)**
Burgenland	-0.1186	-0.125	0.3514	0.0109	0.4957
Carinthia	0.1175	-0.1250	0.1923	0.5531	0.2901
Lower Austria	-0.3162	-0.1250	0.1302	-0.5298	0.7019
Upper Austria	0.0554	-0.1250	0.2195	0.3850	0.3501
Salzburg	-0.1909	-0.1250	0.1322	-0.1814	0.5720
Styria	-0.6492	-0.1250	0.1236	-1.4911	0.9320
Tyrol	0.1208	-0.1250	0.2060	0.5416	0.2940
Vorarlberg	0.2596	-0.1250	0.5703	0.5093	0.3053
Vienna	0.2972	-0.1250	0.5703	0.5591	0.2880

The local Moran's I shows a slight increase in spatial autocorrelation at local level, but still none of the p-values are significant. Mapping the local Moran's I results (figure 8) clearly shows presence of spatial clustering and dispersion. The north-eastern part of Austria consisting of Burgenland, Lower Austria, Upper Austria, Styria and Salzburg are forming a cluster of relatively low values, while the south-western part including Carinthia, Tyrol, and Vorarlberg is forming another cluster of high values. The only exception is Vienna, which has much higher tourist stays compared to its neighbours.

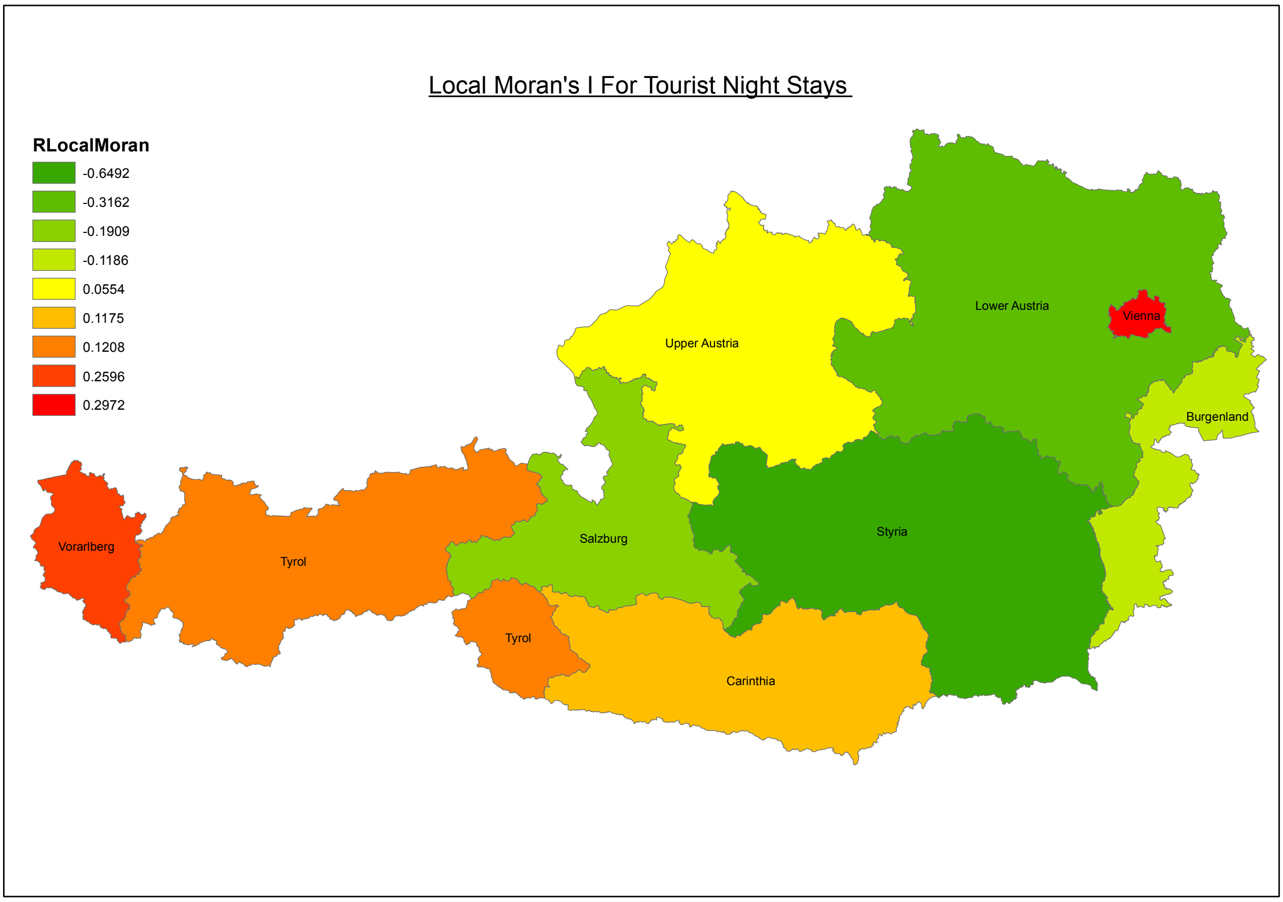


Figure 8: Local Moran's I for overnight stays, 1974-2017

One issue causing the low estimation of Moran's I could be the insufficient number of data entities (only nine provinces) used for estimating the spatial autocorrelation. One way to find out the presence of spatial autocorrelation and its impact on modelling prediction would be to use a space-time version of one of the machine learning methods by applying various spatial weight matrices and see if there is any significant improvement.

2.3 Spatio-Temporal Patterns

Similarly to the ACF and PACF their spatio-temporal (ST) equivalents reveal information about spatio-temporal patterns in the data (see figure 9). As spatial weights the shared border length of a counties neighbours is used (see table 4), although the neighbourhood adjacency first order leads to almost the same result. As only the data from Austria is available, the borders to neigbouring countries are ignored. The STACF at spatial lag 1 shows again a cyclic pattern which seems to be a mixture of all single ACF, thus showing the presence of autocorrelation once again. In the STPACF surprisingly the lags 4, 5 and 9 are significant whereas lag 12 is slightly smaller than the 95 % confidence interval.

Table 4: Spatial weight matrix weighted by shared border length of touching provinces. All values are normalised by the sum of each row.

	Burgenland	Carinthia	Lower Austria	Upper Austria	Salzburg	Styria	Tyrol	Vorarlberg	Vienna
Burgenland	0	0	0.6076	0	0	0.3924	0	0	0
Carinthia	0	0	0	0	0.3073	0.4643	0.2284	0	0
Lower Austria	0.2715	0	0	0.2855	0	0.2705	0	0	0.1726
Upper Austria	0	0	0.3919	0	0.3319	0.2761	0	0	0
Salzburg	0	0.2089	0	0.3138	0	0.1899	0.2873	0	0
Styria	0.1692	0.2347	0.2609	0.1940	0.1412	0	0	0	0
Tyrol	0	0.2783	0	0	0.5147	0	0	0.2070	0
Vorarlberg	0	0	0	0	0	0	1	0	0
Vienna	0	0	1	0	0	0	0	0	0

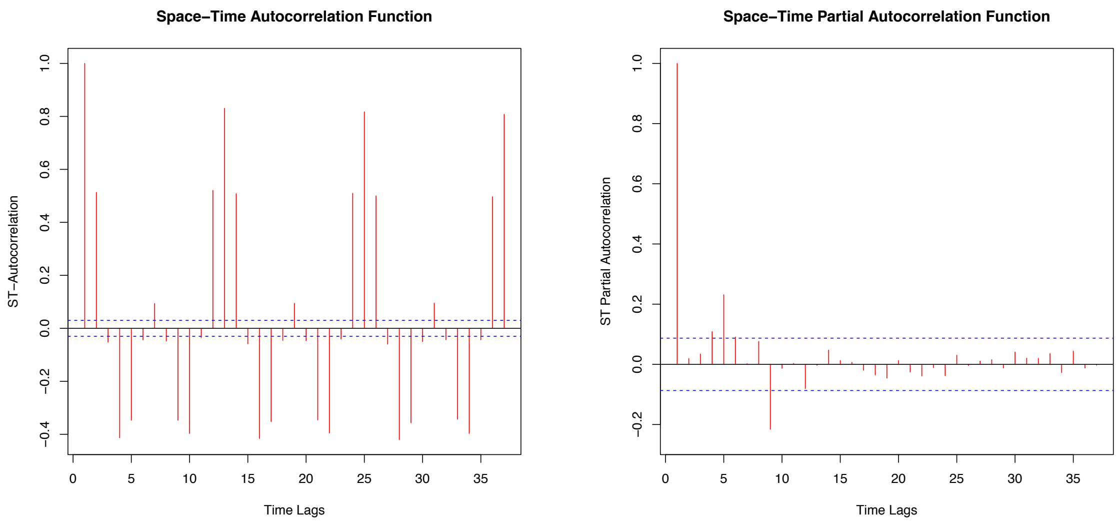


Figure 9: Spatio-temporal autocorrelation function and spatio-temporal partial autocorrelation function of raw time series with neighbourhood first order as spatial weights (the blue dashed lines show the 95% confidence intervals)

Another interesting relation between space and time is illustrated in figure 10. Every province shows the increase or decrease of tourist stays based on a linear model. A large positive slope for Tyrol and Vienna implies that throughout the years (from 1973-2017) more and more tourists have visited those provinces or tend to stay longer (assuming the amount of arrivals stays the same). Care has to be taken when using those provinces for time series models, where the data needs to be corrected for a trend. If the relation between stays and months is not linear as assumed for the trend reduction, large residuals will affect a proper model training. This implies, that those provinces might be the most difficult to predict.

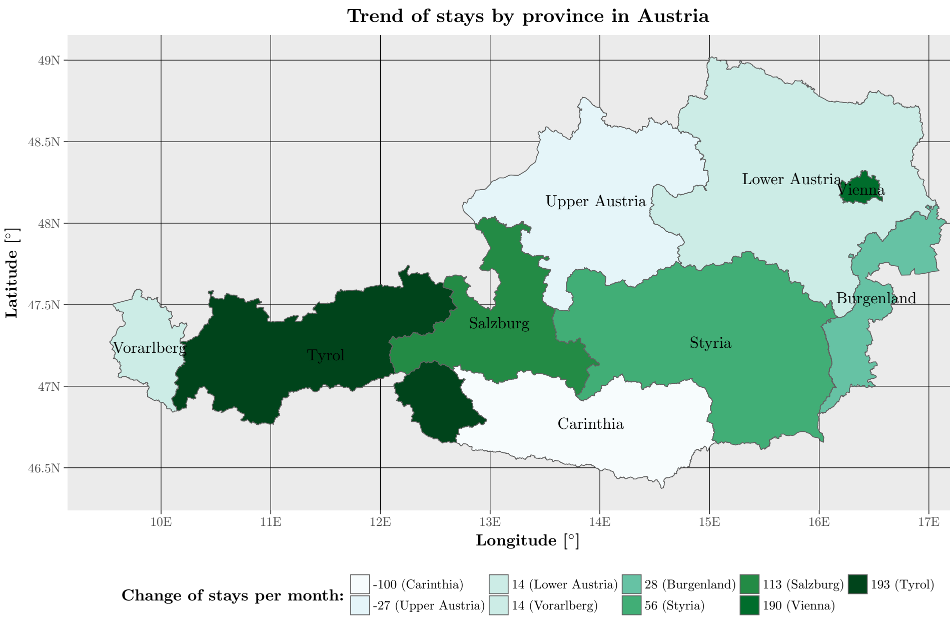


Figure 10: Austrian map shows slope value of a line (trend) representing the stays time series for each province from 1973-2017. This means that the depicted values indicate an increase/decrease of tourists per month.

3 Methods

Having a focus on regression, all four methods are described and applied on the tourist dataset in the subsequent sections. Each method, *RFR* (subsection 3.1), *STARIMA* (subsection 3.2), *ANN* (subsection 3.3) and *SVM* (subsection 3.4), is introduced by a literature review, requires different data pre-processing steps and is applied on the modified data. Finally, all results are shown, which are used for a method comparison in the discussion (see section 4).

3.1 Random Forest Regression (RFR)

The random forest algorithm was designed by Breiman (Breiman 2001), which unites his original idea of bootstrap aggreggation (*"bagging"*) and an additional randomness in feature selection (*"feature bagging"*).

3.1.1 Methodology

The first step in *RFR* involves a random selection with replacement of the input data for each predictor/feature (*"bagging"*). This ensures that each tree is constructed by using a different bootstrap sample opposing the problem of overfitting and reducing the generalisation error. On average, about one third of the data is excluded from training from each tree (*"out-of bag" (OOB)* data). `ntree`[2] is responsible for the size of the forest.

A single tree is built by using a slightly modified *Classification and Regression Trees (CART)* algorithm (*Decision Tree Learning* 2018). *CART* creates a binary decision tree structure leading to smaller data subsets per node level. At each node an attribute test is applied on data subsets for each feature, partitioning the source data into two subgroups. *RFR* uses *"variance reduction"* to define the "best split" and measures the impurity/difference between the parent and child nodes. All values of each feature are used for testing the impurity of the resulting split. *Residual Sum of Squares (RSS)* is taken to estimate the variance of the elements belonging to one node and the largest difference defines the best split. *Variance reduction* aims to minimise the variance along different node levels, thus ensuring that there are more similar elements in the same regions of the tree. Dominant predictors tend to be selected more often in *CART*, which leads to a higher correlation. Therefore, Breiman suggested to randomly select a subset of predictors to be taken into account for splitting a node (`mtry`). Moreover, trees are grown to full size (if e.g. `nodesize` isn't set).

RFR uses the mean of all single tree predictions as a final prediction (*Random Forest* 2018). Furthermore, it offers unbiased error metrics and other useful outputs, for instance the MSE and Pseudo-R^2 based on *OOB* data and variable importance (Liaw 2015). The latter one is computed by averaging *variance reduction* for each predictor.

[2] words written in this style refer to parameters of the *randomForest* package in *R*. For further details see Liaw 2015.

3.1.2 Experimental Setup

Various seasonal cycles are an essential feature of the data (see 2.1) being meaningful as explanatory variables. Therefore, stays data was lagged by all 12 months per province using the functions `t_embed` and `st_embed` [3]. Additionally, months (e.g. 1, 2, ..., 12) as themselves were also used as input data (`DateMod`).

Three different experiments were realised to find the optimal *RFR* model for prediction:

1. province-wise training/prediction (`t_embed`).
2. province-wise training/prediction including lagged data of first-order neighbours (`st_embed`).
3. training/prediction using stacked province-wise lags.

To assess the best method from above, some of the default parameters (`ntree` = 1000, `mtry` = $p/3$[4], `nodesize` = 5) were tuned after selecting the best model.

3.1.3 Analysis

To get an impression of how the different setups compete against each other, RMSE and R^2 values are shown in table 5, 6 and 7. In table 5, Lag 12 is clearly the most dominant variable, followed by lag 9, lag 11, lag 1, lag 6 and lag 3. Table 6, additionally including lagged data from neighbouring provinces, reveals that lag 12, lag 9, lag 6 and lag 3 are the leading predictors. For some provinces (e.g. Vienna) stays from neighbouring countries seem to be a better predictor than the province itself. The error metrics underline a worse model performance when including implicit spatial relations. Vienna has by far the worst R^2, which is even decreasing for the spatial *RFR*. The RMSE increases remarkably for nearly every province.

Using a simpler approach of stacking the lagged data, but ignoring any spatiality and province-wise distinction, turns out to be the best performing *RFR* model. In table 7, Vienna is characterised by the most significant improvement and all RMSE's are nearly lowered by half compared to the previous models.

[3] see practical 5 of STDM

[4] p equals to the total number of predictors

Table 5: Top 5 ranked important predictors and error metrics for a province-wise *RFR* model. Green represents the best province, red the worst.

Province	RMSE	R^2	1	2	3	4	5
Burgenland	21197	0.9769	Lag 12	Lag 1	Lag 11	DateMod	Lag 6
Carinthia	79623	0.9912	Lag 9	Lag 11	Lag 1	Lag 12	Lag 8
Lower Austria	35635	0.9703	Lag 12	Lag 6	Lag 11	Lag 1	Lag 7
Upper Austria	41249	0.9819	Lag 12	Lag 11	Lag 8	Lag 1	Lag 7
Salzburg	226560	0.9768	Lag 12	Lag 9	Lag 3	Lag 6	DateMod
Styria	125486	0.9208	Lag 12	Lag 9	Lag 3	Lag 11	Lag 1
Tyrol	337781	0.9818	Lag 12	Lag 9	Lag 3	Lag 6	DateMod
Vienna	231508	0.7848	Lag 12	Lag 1	Lag 10	Lag 11	Lag 2
Vorarlberg	71982	0.9695	Lag 12	Lag 9	Lag 3	Lag 6	DateMod

Table 6: Top 5 ranked important predictors and error metrics for a province-wise spatial *RFR* model. Lags are related to provinces in brackets (either first-order neighbours or the province itself). Green represents the best province, red the worst.

Province	RMSE	R^2	1	2	3	4	5
Burgenland	22348.0	0.9798	Lag 12 (L. Aus.)	Lag 11 (L. Aus.)	Lag 9 (Styria)	Lag 1 (L. Aus.)	Lag 9 (L. Aus.)
Carinthia	159818.0	0.9770	Lag 12 (Salzburg)	Lag 12 (Carinthia)	Lag 12 (Tyrol)	Lag 9 (Styria)	Lag 9 (Tyrol)
Lower Austria	35391.0	0.9764	Lag 12 (U. Aus.)	Lag 12 (L. Aus.)	Lag 6 (U. Aus.)	Lag 11 (L. Aus.)	Lag 1 (U. Aus.)
Upper Austria	31813.0	0.9864	Lag 12 (Salzburg)	Lag 12 (L. Aus.)	Lag 12 (U. Aus.)	Lag 9 (Styria)	Lag 11 (L. Aus.)
Salzburg	245599	0.9720	Lag 12 (Styria)	Lag 9 (Styria)	Lag 3 (Styria)	Lag 3 (Tyrol)	Lag 6 (Styria)
Styria	127284.0	0.9476	Lag 12 (L. Aus.)	Lag 12 (Carinthia)	Lag 12 (Salzburg)	Lag 12 (Styria)	Lag 12 (U. Aus.)
Tyrol	366814.0	0.9778	Lag 12 (Vorarlb.)	Lag 9 (Vorarlb.)	Lag 3 (Tirol)	Lag 3 (Vorarlb.)	Lag 6 (Vorarlb.)
Vienna	387520.0	0.7269	Lag 6 (L. Aus.)	Lag 5 (L. Aus.)	Lag 12 (L. Aus.)	Lag 10 (L. Aus.)	Lag 2 (L. Aus.)
Vorarlberg	90221.0	0.9590	Lag 12 (Vorarlb.)	Lag 3 (Tirol)	Lag 9 (Vorarlb.)	Lag 3 (Vorarlb.)	Lag 6 (Vorarlb.)

Table 7: Error metrics for a *RFR* model trained on stacked province-wise lagged data. Green represents the best province, red the worst.

	Burgen-land	Carinthia	Lower Austria	Upper Austria	Salz-burg	Styria	Tyrol	Vienna	Vorarl-berg
RMSE	15913	73530	19889	29487	173263	71818	317590	91999	69337
R^2	0.9809	0.9926	0.9810	0.9846	0.9786	0.9637	0.9812	0.9369	0.9681

To be able to detect the most important explanatory variables for the finally used method, variable importance was analysed (figure 11). `ntree` was set to 1000, since a lot of trees are necessary to get a stable estimate of variable importance (Liaw, Wiener, et al. 2002).

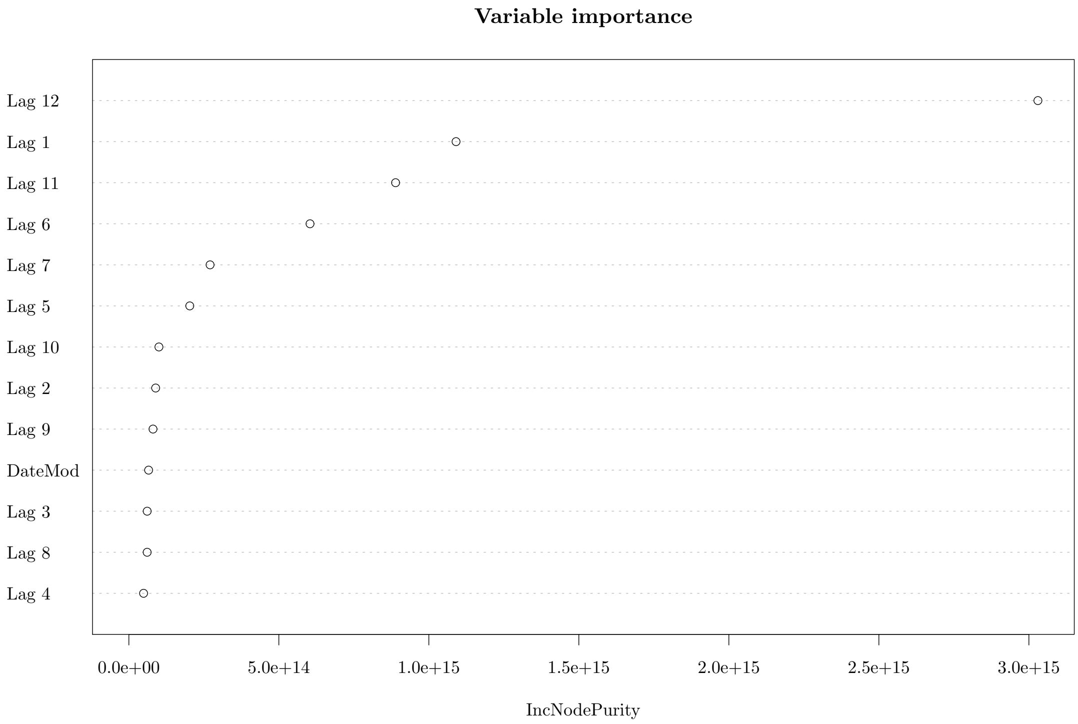


Figure 11: Variable importance of the third *RFR* method including lagged stays data and *DateMod*.

According to figure 11, it was decided to take lag 12, lag 11, lag 6 and lag 1 as predictors, since those variables seem to explain stays well enough (replicated conclusion of subsection 2.1). Moreover, less predictors yield a lower runtime.

The next step is dedicated to improve the performance by tuning the mentioned model parameters of the third *RFR* model. Firstly, 5-fold cross-validation was applied to determine the optimal `ntree`. A reliable error estimate can only be achieved by a higher `ntree` (more averaged predictions). Thus, cross validation is therefore thought to be more suitable, when investigating different values for `ntree`. The behaviour of runtime and prediction error depending on `ntree` can be seen in figure 12. The prediction error has its lowest value around `ntree` = 800 and the relation between runtime and `ntree` is approximately linear.

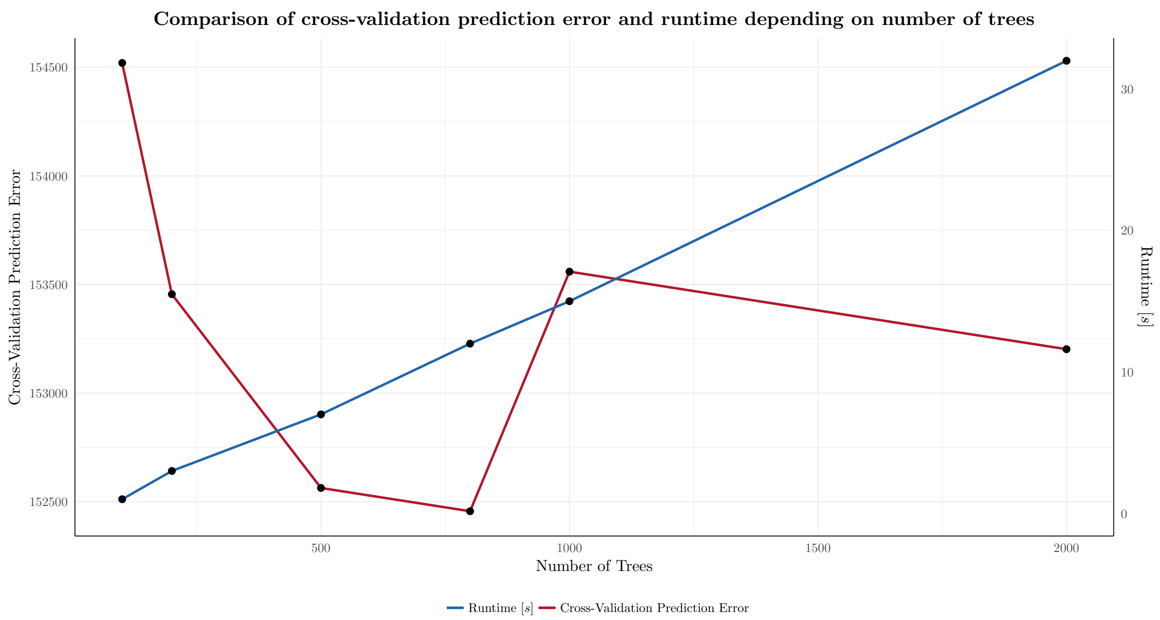


Figure 12: Runtime and cross-validation error comparison in relation to the number of trees `ntree`.

Secondly, figure 13 and 14 demonstrate that `mtry` = 2 and `nodesize` = 2 are the best choices having the highest pseudo-R^2, which shows the importance of *"feature bagging"* for the former one.

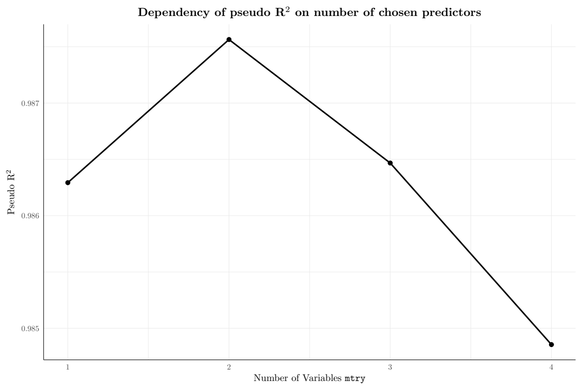


Figure 13: Performance dependency on number of predictors randomly selected (`mtry`).

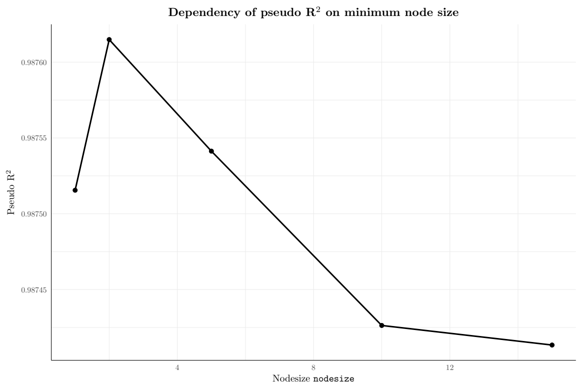


Figure 14: Performance dependency on minimum size of terminal nodes (`nodesize`).

3.1.4 Results

Using the optimised parameters and predictors, the error metrics in table 8 could be achieved.

Table 8: Final error metrics resulting from a *RFR* prediction comprising tuned model parameters. Green represents the best province, red the worst.

Province	Bias	RMSE	NRMSE	Std. Ratio	Pearson Correlation	R^2	Spearman Correlation
Burgenland	707	18130	0.0727	0.9967	0.9875	0.9752	0.9741
Carinthia	6286	79950	0.076	1.0182	0.9956	0.9913	0.9895
Lower Austria	-855	23383	0.041	1.0157	0.9867	0.9737	0.9813
Upper Austria	3726	31862	0.0526	1.0318	0.9896	0.9794	0.9703
Salzburg	30070	204958	0.0951	1.0596	0.9855	0.9713	0.9767
Styria	628	67876	0.0697	1.0569	0.9772	0.955	0.9768
Tyrol	54958	318733	0.0853	1.0703	0.9892	0.9786	0.9864
Vienna	2569	63687	0.0567	1.0083	0.9691	0.9392	0.9657
Vorarlberg	-2657	73886	0.104	1.0512	0.9809	0.9621	0.9743

In total numbers, Tirol has the highest RMSE followed by Salzburg. Bias and RMSE are clearly correlated with the amount of stays (see figure 2). To allow for a better comparison, NRMSE was calculated by dividing RMSE by the mean value. This leads to the best NRMSE for Lower Austria and worst for Vorarlberg. The standard deviation ratio is closest to one for Burgenland and the largest difference in scaling can be found in Tirol. Burgenland has the most promising overall statistics. The predictions for Carinthia, Lower Austria and Upper Austria also agree well with the reference stays data. All provinces have a high correlation, which can be directly seen in R^2 (very good temporal agreement). If the pearson and spearman correlation coefficient are equal, then both datasets are linearly correlated, which is approximately true for all regions.

A visual comparison of one-step ahead predicted and reference stays data can be seen in figure 15. It depicts an excellent agreement for provinces having a dominating annual peak (Burgenland, Carinthia, Lower Austria and Upper Austria) and larger differences for regions with winter and summer tourist offers, where peaks and troughs change more rapidly. The remarkable trend of Vienna is more or less modelled well.

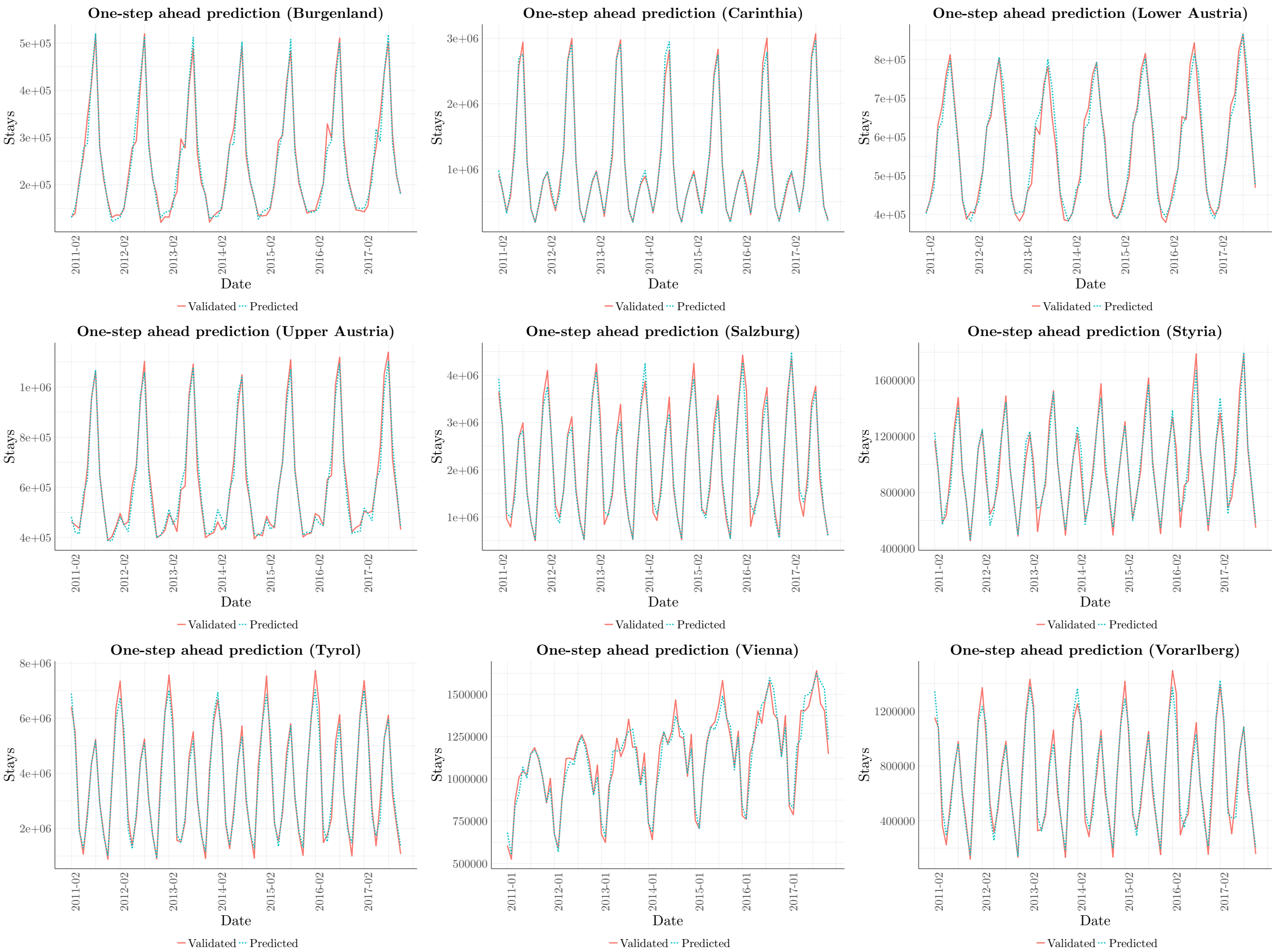
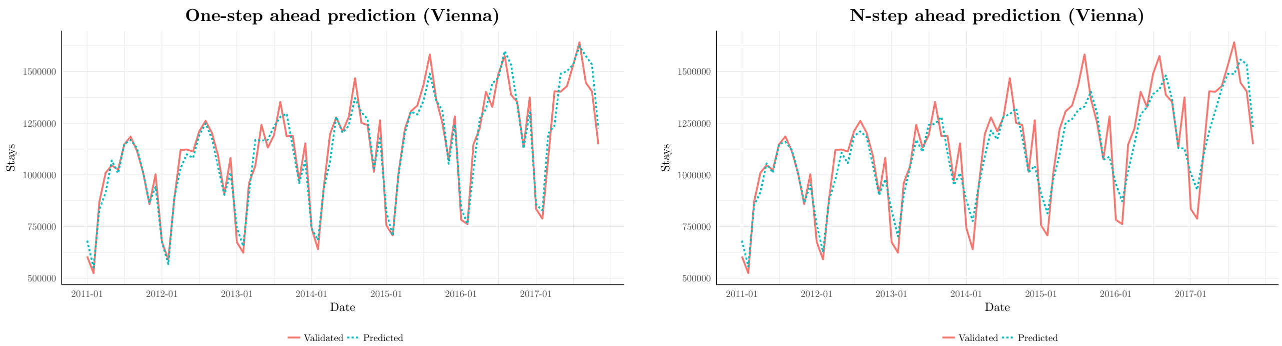


Figure 15: Comparison of predicted and validated time series from 01/2011 to 11/2017 (Note the different scaling of the y-axis.).

In forecasting it is more important to predict over multiple steps rather than one. Therefore, multistep-prediction was implemented for *RFR*. A comparison of one-step and n-step ahead prediction for Vienna can been seen in figure 16. The n-step ahead predicted curve experiences a slight shift of the main peak and a smoother time series with no inter-annual variability. RMSE and R^2 of the other provinces are shown in table 9. When using n-step ahead prediction, the RMSE increases (amplitude depending on seasonal complexity) and R^2 decreases.

Table 9: Error metrics resulting when using n-step ahead prediction with the final *RFR* model. Green represents the best province, red the worst.

	Burgenland	Carinthia	Lower Austria	Upper Austria	Salzburg	Styria	Tyrol	Vorarlberg
RMSE	19522	87145	33597	36349	309324	110711	431814	77906
R^2	0.9774	0.9896	0.9483	0.9799	0.9556	0.8787	0.9652	0.9614



Bias	RMSE	NRMSE	Std. Ratio	Pearson Correlation	R^2	Spearman Correlation
2569	63687	0.0567	1.0083	0.9691	0.9392	0.9657

Bias	RMSE	NRMSE	Std. Ratio	Pearson Correlation	R^2	Spearman Correlation
20004	99090	0.0881	1.2047	0.9315	0.8677	0.9392

Figure 16: Comparison of one-step (left) and n-step (right) ahead prediction for Vienna.

3.1.5 Summary and Discussion

RFR based on stacked province-wise lagged data has clearly shown to be applicable and leading to satisfying results. Only little success could be achieved by using province-wise training. This is reasonable, since the *RFR* model only knows values, which have occurred in the past. If there is any trend (values not known/trained), *RFR* is not able to handle it properly. Furthermore, it was found out that *RFR* for provinces comprising a two-seasonal tourist stays pattern performs worst. The complex and rapid changes of stays are difficult to be modelled by *RFR*, since the mean value of many single predictions smooths the predicted curve (section 3.1.1). This could be a reasonable explanation for larger deviations in Tirol, Salzburg, Styria and Vorarlberg.

Having a closer look at table 7 and 8 reveals that better statistics can be obtained by using a province-wise *RFR* model for two-seasonal provinces, whereas for Vienna results improve with the chosen configuration. Including lagged stays data from first-order neighbours did not improve forecasting and made it even worse. A reason could be that neighbouring provinces do not have a clear overlap of the seasonal onset and there are too many predictors (necessary to adjust `mtry`).

To sum up, *RFR* is a very simple method to use and observed changes by adjusting the used parameters were really small. Predictions with *RFR* could be further improved by including other predictors such as economy data of province-wise income resulting from tourism.

3.2 (ST)ARIMA

A short decomposition of the acronym *(ST)ARIMA* helps to understand what it includes. The simplest form are the autoregressive (AR) and moving average (MA) model. Combined together, they form the ARMA model which was first described by Gurland 1954, but got only popular after the publication of Box and Jenkins 1970. The integration part (I) was added to create *ARIMA* for handling non-stationary data, i.e. trends or seasonal data. The spatio-temporal ARIMA (*STARIMA*) model proposed by Pfeifer and Deutsch 1981 is an extension of the *ARIMA* model incorporating spatial autocorrelation. Both, *ARIMA* and *STARIMA* explicitly model a time series' behaviour based on its past values. There is no possiblity to include other explanatory variables. While the ARMA and *ARIMA* model are well established and used for analysing time series throughout all different disciplines, the *STARIMA* model seems to be far less used. Islam-Khan, Landfeldt, and Damdhere 2012; Cheng et al. 2014; Duan et al. 2016 use the latter for traffic flow predictions.

3.2.1 Methodology

The $ARIMA(p, d, q)$ model combines the following three parts:

- AR(p): Auto-regressive term of order p. Describes a linear regression of the current value against the p previous values.
- I(d): Differentiation of order d (number of differences). Used for achieving stationarity of the time series.
- MA(q): Moving average term of order q. Describes a linear regression of the current value against the forecast errors of the previous values.

ARIMA can be extended by a seasonal model as proposed by Pfeifer and Deutsch 1981 accommodating e.g. daily, weekly or annual cycles. The seasonal model is basically a second *ARIMA* model with a different lag used for the differentiation. The model is then extended to $ARIMA(p, d, q)(P, D, Q)_S$, where P, D and Q are the equivalents to the lower-case letters and S indicates the lag used for the seasonal model. The parameters are usually estimated using either a brute-force grid search or the Box-Jenkins method (Box and Jenkins 1970). The latter uses the ACF to estimate p and PACF to estimate q. Furthermore, the ACF is used for the decision whether a series is stationary or not and the resulting choice of d. The seasonal model is estimated similarly using the by lag S differentiated data. The *STARIMA* model, the spatio-temporal extension of the *ARIMA*, includes a spatial weight matrix as input describing to which extent a particular instance should be modelled based on other instances. Often simply neighbourhood first order or a distance related weighting is used.

Unfortunately (ST)ARMA requires (weak) stationary data and spatio-temporal data is often non-stationary, i.e. contains trends and seasonality. Sometimes the (weak) stationarity, i.e. a non-varying mean and auto-covariance over time, can be achieved by a differentiation in the I part, thus the use of *(ST)ARIMA*. If not, further pre-processing (e.g. applying the log-function) is necessary.

3.2.2 Experimental setup

For the following analysis, R's built-in *ARIMA* function and the *STARIMA* software package for R by Cheng and Wang[5] are used. The following differences between the two implementations were noticed and accommodated in an appropriate way:

- Only the *ARIMA* function allows to apply a seasonal model.
- *ARIMA* offers a multi-step ahead prediction whereas the *STARIMA* package only offers a one-step ahead prediction

[5] V2.0, January 2012, provided within the course CEGE076 Spatio-Temporal Data Mining

Thus, a recursive multi-step ahead prediction for *STARIMA* similar to the one used by Liu and Zio 2017 was implemented by the author, as multi-step ahead prediction seems more appropriate for our experiment. To handle the seasonality discovered in section 2.1, for *ARIMA* simply a seasonal model with $S = 12$ is used while for *STARIMA* the data was differentiated manually by lag 12 in pre-procesing. Compared to fig. 5 the ACF of the differentiated time series (fig. 17) show that the seasonality is largely eliminated and now only Vienna show some non-stationary behaviour, which can possibly later be eliminated using the I part of the *(ST)ARIMA* model. Additionally, in pre-processing the data was scaled before fitting the models for better numerical performance. Four different models *ST1*, *STn*, *STns* and *ARn* were trained and tested (see table 10). They differ in the base model used, the number of steps predicted ahead and the range of data used for training of the models. The test data was the same for all models.

Table 10: Summary of the models' characteristics

Model ID	**Model name**	**Base model**	**Prediction mode**	**Training data**
ST1	STARIMA 1-step ahead	STARIMA	single-step	11/1973 - 12/2010
STn	STARIMA n-step ahead	STARIMA	multi-step (recursive)	11/1973 - 12/2010
STns	STARIMA n-step ahead short	STARIMA	multi-step (recursive)	01/2000 - 12/2010
ARn	ARIMA n-step ahead	ARIMA	multi-step	11/1973 - 12/2010

For *ARIMA* and *STARIMA* the number of parameters to be estimated is six and three, respectively. A grid search with the range [0,1,2,3] for all parameters is used to find for each province the optimal combination leading to the lowest root-mean square error (RMSE) using the test data (see tables 11, 12, 13 and 14). The grid search is applicable, as the paramters of a *(ST)ARIMA* model are usually kept low (Brockwell and Davis 2002, chap. 6.5). Regardless of whether the models differ in the base model, prediction mode or even the training data, the optimal parameters are different for almost all provinces. As the upper limit 3 appears several times further sensitivity analysis was executed and showed that larger parameters increase the quality of the models only marginally. The *ARn* shows only a seasonal model for Vienna which does not agree with the interpretation of fig. 17.

Table 11: Model *ST1*: Optimal parameters

Province	**p**	**d**	**q**
Burgenland	2	0	1
Carinthia	2	0	2
Lower Austria	2	1	0
Upper Austria	3	0	3
Salzburg	1	0	3
Styria	3	0	3
Tyrol	1	0	2
Vorarlberg	1	0	1
Vienna	2	1	3

Table 12: Model *STn*: Optimal parameters

Province	**p**	**d**	**q**
Burgenland	2	1	0
Carinthia	2	0	0
Lower Austria	3	1	2
Upper Austria	2	1	0
Salzburg	1	0	1
Styria	2	1	3
Tyrol	3	0	2
Vorarlberg	3	1	3
Vienna	2	1	2

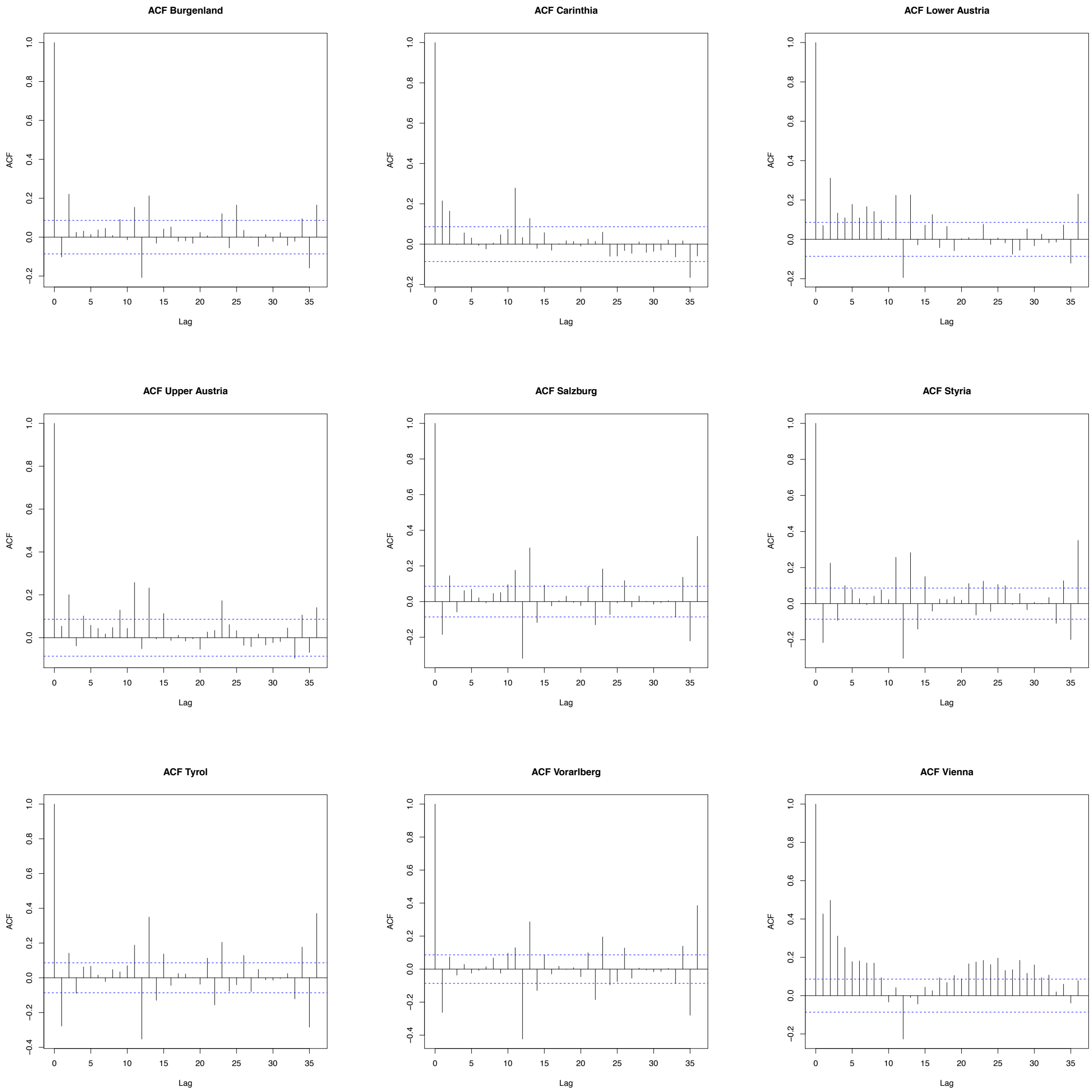


Figure 17: Autocorrelation functions of the time series differentiated by lag 12 (the blue dashed lines show the 95% confidence intervals)

Table 13: Model *STns*: Optimal parameters

Province	p	d	q
Burgenland	2	1	0
Carinthia	2	0	2
Lower Austria	3	1	3
Upper Austria	3	1	1
Salzburg	3	1	1
Styria	1	1	2
Tyrol	3	1	1
Vorarlberg	2	1	3
Vienna	2	1	0

Table 14: Model *ARn*: Optimal parameters; the seasonal model is of order $S = 12$

Province	p	d	q	P	D	Q
Burgenland	3	0	2	2	0	1
Carinthia	3	0	0	3	1	2
Lower Austria	3	2	1	0	1	1
Upper Austria	3	2	2	0	1	1
Salzburg	1	2	3	3	1	3
Styria	2	2	3	0	1	3
Tyrol	0	1	3	0	1	0
Vorarlberg	0	2	0	3	1	3
Vienna	0	0	0	2	1	2

3.2.3 Results

As performance measures the RMSE, the normalised RMSE (NRMSE) and the R^2 were used. While (N)RMSE measures the errors between the prediction and the observed data, R^2 is a measure for the similarity of the shape. As tables 15, 16, 17 and 18 show, all four models performed reasonably well. The best performing model for each province and each measure are listed in table 19. Although there is no dominant model, model *ARn* appears most often. The only experiment that never performs best is the *STn*. The predicted values of the most often best performing *ARn* model show generally a good agreement with the observed data (fig. 18).

Table 15: Model *ST1*: Performance measures

Province	RMSE	NRMSE	R^2
Burgenland	16,226	0.0654	0.9806
Carinthia	81,993	0.0781	0.9906
Lower Austria	19,184	0.0338	0.9825
Upper Austria	32,772	0.0543	0.9789
Salzburg	201,168	0.0927	0.9721
Styria	57,091	0.0585	0.9727
Tyrol	381,760	0.1013	0.9653
Vorarlberg	86,642	0.1211	0.9481
Vienna	52,210	0.0464	0.9588

Table 16: Model *STn*: Performance measures

Province	RMSE	NRMSE	R^2
Burgenland	16,598	0.0669	0.9823
Carinthia	93,968	0.0895	0.9881
Lower Austria	18,275	0.0322	0.9847
Upper Austria	31,678	0.0525	0.9848
Salzburg	300,198	0.1383	0.9620
Styria	67,985	0.0697	0.9657
Tyrol	387,000	0.1027	0.9743
Vorarlberg	78,188	0.1093	0.9599
Vienna	72,748	0.0647	0.9369

Table 17: Model *STns*: Performance measures

Province	RMSE	NRMSE	R^2
Burgenland	16,627	0.0670	0.9828
Carinthia	94,894	0.0904	0.9878
Lower Austria	18,193	0.0320	0.9849
Upper Austria	29,099	0.0482	0.9869
Salzburg	296,607	0.1366	0.9630
Styria	67,608	0.0693	0.9662
Tyrol	367,167	0.0974	0.9742
Vorarlberg	75,247	0.1052	0.9632
Vienna	73,869	0.0657	0.9393

Table 18: Model *ARn*: Performance measures

Province	RMSE	NRMSE	R^2
Burgenland	14,539	0.0586	0.9840
Carinthia	86,123	0.0821	0.9901
Lower Austria	18,025	0.0317	0.9850
Upper Austria	28,269	0.0468	0.9852
Salzburg	213,680	0.0984	0.9747
Styria	68,917	0.0707	0.9599
Tyrol	326,693	0.0867	0.9767
Vorarlberg	62,134	0.0868	0.9744
Vienna	59,897	0.0533	0.9545

Table 19: Best performing models according to (N)RMSE and R^2, respectively

Province	**(N)RMSE**	$\mathbf{R^2}$
Burgenland	ARn	ARn
Carinthia	ST1	ST1
Lower Austria	ARn	ARn
Upper Austria	ARn	STns
Salzburg	ST1	ARn
Styria	ST1	ST1
Tyrol	ARn	ARn
Vorarlberg	ARn	ARn
Vienna	ST1	ST1

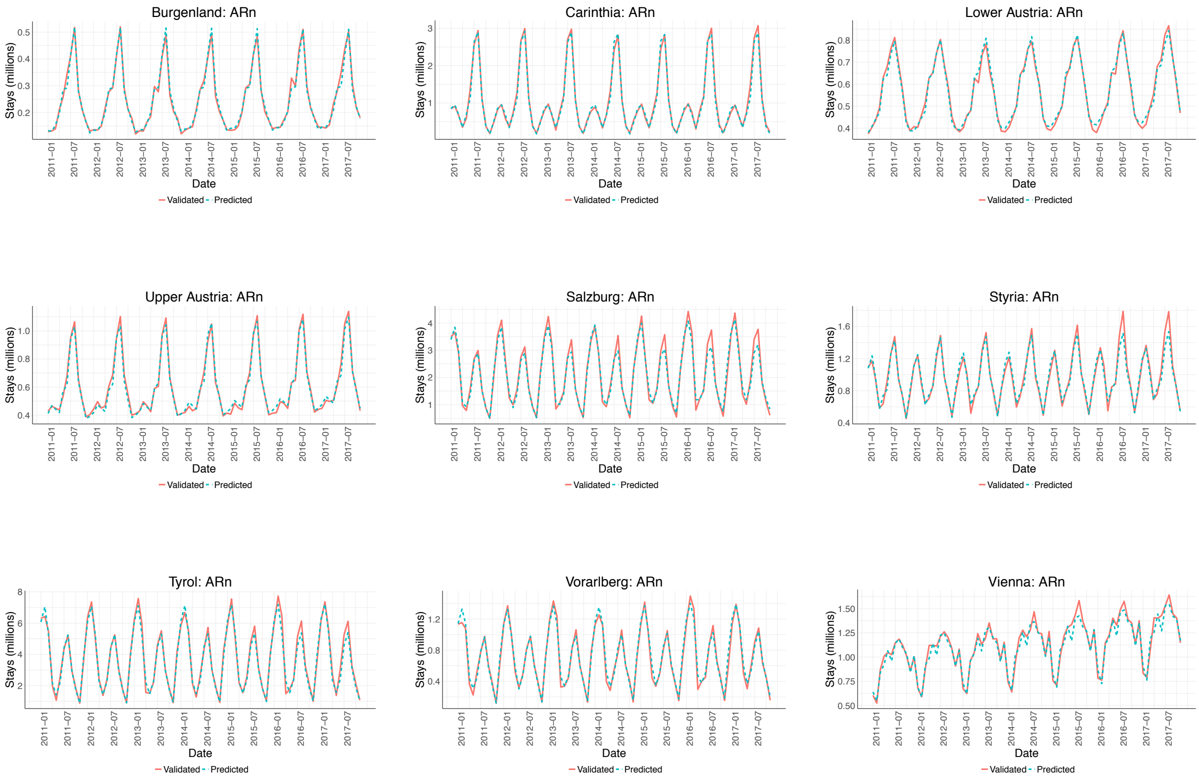


Figure 18: Results of the most often best performing *ARn* model

The differences of the models can be exemplarily illustrated using the Vienna (fig. 19). The one-step ahead prediction *ST1* follows the shape of the validated data quite close, especially also in the years 2016 and 2017, resulting in a low (N)RMSE and a high R^2 value. *STn* and *STns* look quite similar with minor differences in the prediction for summer 2017. *ARn* can model more accurately the high/low peaks than *STn* and *STns*, i.e. seems to have less smoothing, resulting in a lower RMSE.

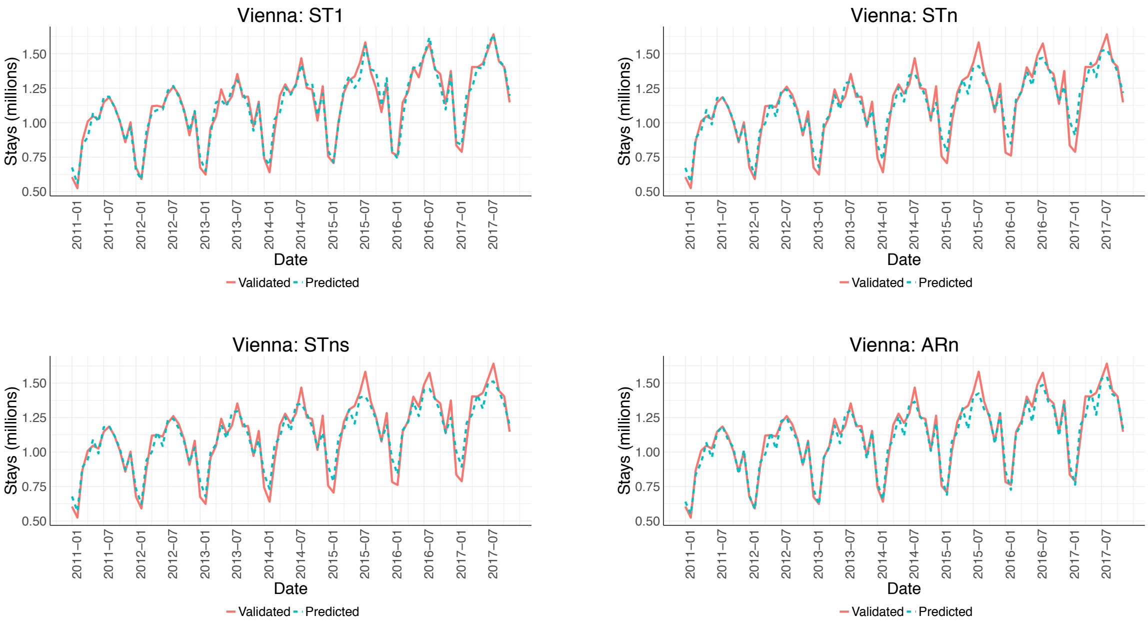


Figure 19: Comparison of the four methods at the example of Vienna

3.2.4 Discussion

The largest NRMSE values can be found in provinces with two peaks a year which do not evolve similarly over time (e.g. Tyrol, where the winter tourism got more important than the summer tourism in the 1990s). The lowest R^2 values has clearly Vienna which has the most complex shape of the time series and especially a different pattern every spring (possibly because the Easter holiday is not every year in the same month).

A critical comparison of the best-performing models *ST1* and *ARn* reveals that the one-step ahead prediction *ST1* has the clear advantage of relying only on observed data and not on predicted data. This advantage compared to the multi-step ahead predictions gets more important the longer the prediction period gets. It can thus be explained why it performs so well. But it should be kept in mind, that *ST1* can only predict one month ahead. The good performance of the non-spatial *ARn* model is possibly due to the lack of high spatial correlation and the fact that it includes a proper seasonal model.

Thus, it seems promisingly to implement a seasonal model for *STARIMA* like the one implemented for *ARIMA*. For keeping the models as simple as possible, a significance test for higher values of the parameters p, q, P and Q should be implemented, instead of just limiting the range in the grid search. As *STns* performs sometimes better than *STn*, some investigations should be made on the optimal range selection of the training data. More old data does obviously not necessarily improve the prediction.

3.3 Artificial Neural Networks

3.3.1 Methodology

ANNs are very useful in nonlinear time series forecasting. A basic feed-forward neural network (see figure 20) usually contains three layers with connected neurons and transmits information in one direction: from the input layer to the output layer. It learns from training data, update its parameters (weights and bias) and then predict unknown data.

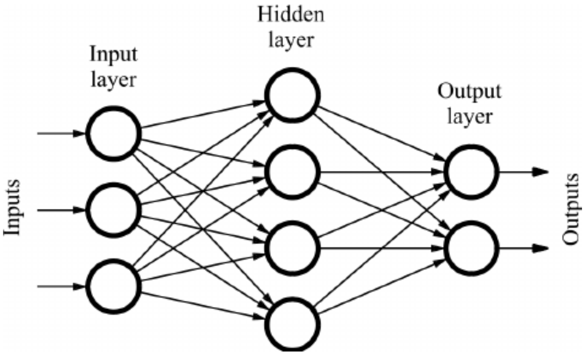


Figure 20: A feed forward neural network

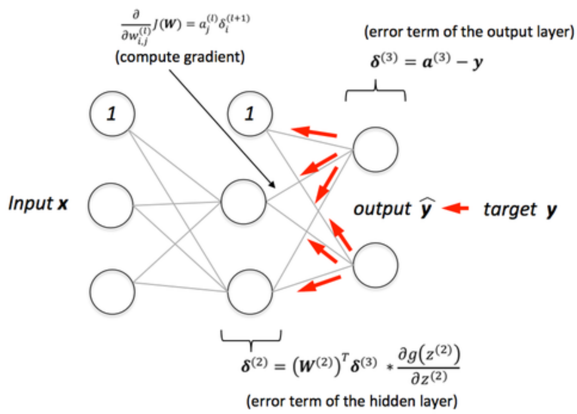


Figure 21: BP algorithm (Raschka 2018)

Backpropagation learning algorithm (BP) aims at minimizing the error. Instead of just propagating the information forward, it back-propagates the error, calculate the gradient of the error function based on chain rule and finally update the weights (see figure 21). For regular back propagation, a small learning rate is normally used, which leads to the slow training process.

Resilient Propagation algorithm (Rprop) is an iterative process, which was developed in 1993 in an attempt to improve upon the BP algorithm. Unlike BP, Rprop does not use the magnitude of the gradient; instead, it uses only the sign of the gradient and performs a local adaptation of the weight-updates (Riedmiller and Braun 1993). Therefore, it is more efficient and flexible.

3.3.2 Experimental Setup

In this section, multilayer perceptron neural networks (MLP) are applied to predict the monthly tourist overnight stays in Austria from Jan 2011 to Nov 2017 by testing both traditional BP algorithm and Rprop+ algorithm.

1. Data Preprocessing
   Data are scaled using Min-Max scaling method to make sure the minimum and maximum values of each time series map to the boundary [0,1]. Several ways of data processing have been considered. First, it's about whether to apply log-return or not. As it is shown in figure 7, several provinces have positiveskewness, however ANNs do not care much about distribution of data. Therefore, we use the raw data.
   Second, it is to decided whether to difference time series data or not. Previous experiments on predicting tourism data showed that ANN models trained by undifferenced data have better forecasting performance (Taieb 2014). Therefore, in this case we do not difference it.
   Besides, Makridakis et al. (1982) found that forecast results from ANN models trained by deseasonlized data are significantly more accurate than those trained by non-deseasonalised data. Although in this

paper non-deseasonalised data are used, deseasonalising can be a good way to improve the accuracy of model prediction.

2. Neural Network Construction

   - The Network Architecture
     Several factors need to be taken into consideration while designing a proper neural network. First, for the input layer, inputs are from three aspects: time delays data, monthly dummies and neighbourhoods' time delays data. Therefore, for different models, the specific selections of input variables have slightly differences and the number of nodes ranges from at least 3 to more than `12(lags)+11(monthly dummies)+n*12(lags)` (where n is the number of first-order neighbours).
     Second, for hidden layer, one layer is required in this case as previous findings suggest one is enough for time series forecasting (Dong, Fataliyev, and Wang 2013). The number of hidden nodes is decided through experiments, and the final optimal size of hidden layer varies for each province.
     Third, for output layer, the number of neurons is one, i.e. the output is the predicted next month overnight stays.
   - The Error and Activation Function
     The error function is sum of squared error (`SSE`) which is used to compare predicted values and outcome values and help generate new weights with learning rate during the training procedure. Activation function for the hidden layer is a sigmoid transfer function (`Logistic`) which is suggested by Klimasauskas who found it is better than the hyperbolic tangent function when learning the average behaviour. Finally, linear function is the activation function for output layer.
   - The Stopping Criteria
     Late stopping is one way to stop training iteration when a certain error condition is achieved. For traditional BP method, we set `maxit` parameter to 1000 through nnet package; for Rprop+ method, we set `threshold` to 0.01 and `maxstep` to 1e5 through nerualnet package as stopping conditions.
   - Performance Measures
     For each model, it will be trained several times and the optimal outcome model will be selected. Corresponding control parameters for training times are `repeats` and `rep` for the two methods, respectively. In this case, we use RMSE as the performance measure.

3.3.3 Model Selection and Results

1. Different Algorithms and Parameter Combinations
   For traditional back-propagation, grid search of both `size` and `decay` is needed; Here, the number of hidden nodes are ranged in the set {5,7,9} and weight decay possible values are from the following options {0.1,0.5,0.7,1}. The selected parameters are made based on a broader range of grid search, of which results are not presented.
   The best tune varies for different provinces (see table 20), for example, as for model of Corinthian, the optimal hidden nodes and weight decay are 5 and 1 respectively (see figure 22). The overall performance of BP models is good, except for Vienna whose model can only explain 78.9% of the response variability. Therefore, Rprop algorithm is introduced to see if it can improve the model. For this method, learning rate is not required to be specified. Thus, the only task is to find a proper number of hidden nodes. Generally, the increase of hidden neurons will help improve the accuracy of model. However, the problem is more computing time and danger of over fitting.
   Here, we set the maximum hidden nodes number to 12 which equals to the basic inputs 12 time lags and the minimum number is 6. In figure 23, each model is the optimal one from 35 repetitions and it can be seen that there is no much difference among models with different numbers of hidden nodes for most provinces and it does not follow the common sense that the more nodes the better fit either. Besides, Styria and Vienna are the probably the top two models which are hard to fit because their results are

more discrete than others. Table 21 shows the results of RMSE and R Square of the models and Vienna's R Square improves to 94.5%. From figure 24, it can be seen clearly that in the BP model Vienna has a bad fit with the problem of not catching up the growing trend while the Rprop model performs very well. It suggests that Rprop may be more able to update weights through learning past errors than traditional BP. Hence, we adopt Rprop algorithm for the following analysis.

	Size	Decay	RMSE	R Square
Burgenland	5	0.1	19948	0.9702
Carinthia	5	1	92357	0.9882
LowerAustria	5	0.3	25756	0.9761
UpperAustria	5	1	39680	0.9772
Salzburg	5	1	242359	0.9649
Styria	9	1	59874	0.9676
Tyrol	9	1	396216	0.9629
Vorarlberg	9	0.2	87849	0.9489
Vienna	9	0.3	243918	0.7892

Table 20: The Results of Best Tune for Transitional BP

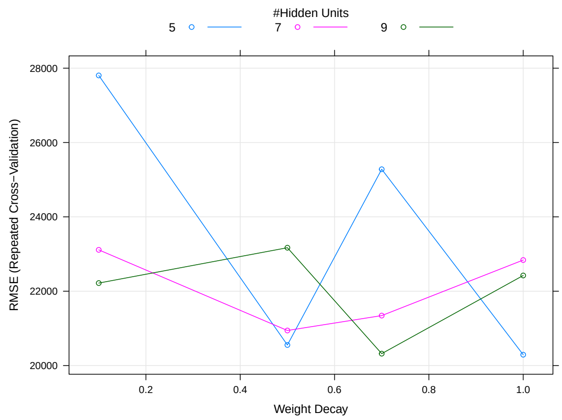


Figure 22: Example of Grid Search for BP

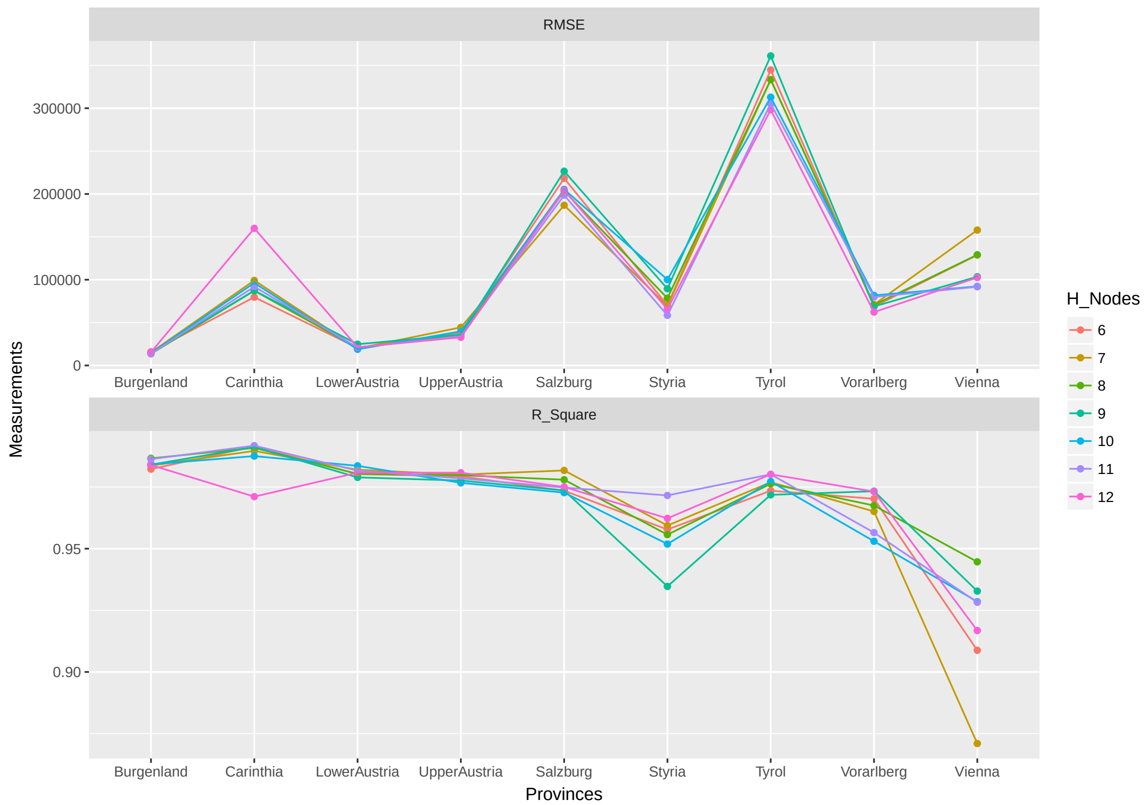


Figure 23: Different Hidden Nodes Comparison for Rprop+

	Burgenland	Carinthia	L.Austria	U.Austria	Salzburg	Styria	Tyrol	Vorarlberg	Vienna
H.Nodes	8	6	10	12	7	11	12	12	11
R^2	0.9866	0.9918	0.9836	0.9808	0.9817	0.9716	0.9802	0.9731	0.9447
RMSE	13593	79649	18865	32830	186846	58431	298194	62224	91772

Table 21: The Results of Best Hidden Nodes for Rprop+

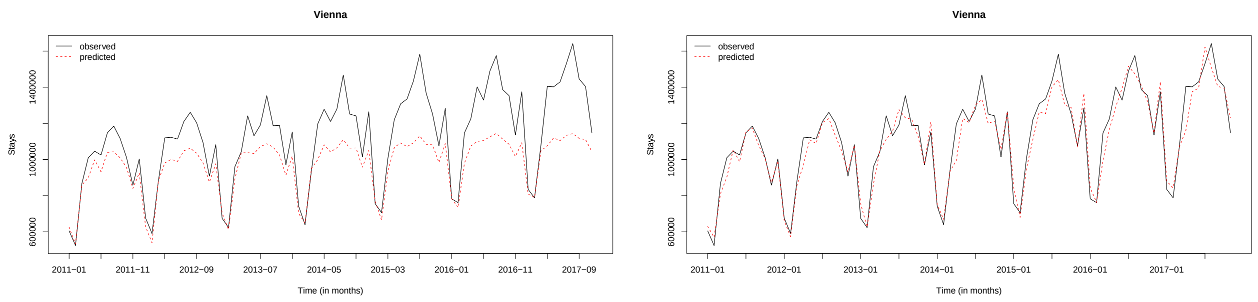


Figure 24: Observed data vs Predicted data of Vienna (Left: Traditional BP; Right: Rprop+)

2. Different Training Data Lengths
   Different training data lengths are tested in this section to see its effect on the accuracy of models, namely 5-year length, 10-year length, 20-year length and total 36 years. Among all the results (see table 22), the full range data length model performs the best. However, it takes more time than shorter length ones. Therefore, if accuracy and time consuming are both considered, mid length of training data would be a better option.

	Burgenland	Carinthia	L.Austria	U.Austria	Salzburg	Styria	Tyrol	Vorarlberg	Vienna
RMSE									
2005-2010	18095	92919	25507	34177	236439	69594	345634	81304	116732
2000-2010	13580	79112	20003	31319	174364	52095	337064	75572	70609
1990-2010	13078	79774	20968	33579	217653	63125	308482	84324	69924
1973-2010	13593	79649	18865	32830	186846	58431	298194	62224	91772
R Square									
2005-2010	0.9795	0.9882	0.9814	0.9781	0.9683	0.9609	0.9744	0.9612	0.9152
2000-2010	0.9866	0.9917	0.9811	0.9793	0.9801	0.9727	0.9751	0.9604	0.9471
1990-2010	0.9877	0.9911	0.9815	0.9781	0.9737	0.9663	0.9781	0.9598	0.9526
1973-2010	0.9866	0.9918	0.9836	0.9808	0.9817	0.9716	0.9802	0.9731	0.9447

Table 22: The Results for Models with Different Training Length

3. Different Ways of Incorporating Spatial-Temporal Information
   In this part, the effect of different length of lagged terms, monthly dummies and spatial adjacency matrix on ANN models are tested. First, we compare the test results of models with 4 different lengths lags and whether including monthly variables or not. It can be seen from table 23 that in most cases as the length of lagged term increases, the RMSE of the test set reduces, except for Vienna, for which the best predictor combination is one lag term (`Lag.1`) with monthly variables. One more finding is that by adding monthly dummies the average running time of models gets shorter. Therefore, based on the

results, models in following sections will still use Lag.1-Lag.12 as inputs but adding monthly dummies for models of Vorarlberg.

	Without Monthly Dummies				With Monthly Dummies			
	Lag.1	Lag.1 - 3	Lag.1 - 6	Lag.1 - 12	Lag.1	Lag.1 - 3	Lag.1 - 6	Lag.1 - 12
Burgenland	58,017	34,360	27,601	16,150	25,499	24,086	22,329	17,472
Carinthia	1,328,152	302,405	269,274	90,509	203,132	172,948	170,179	86,730
L.Austria	91,319	60,552	47,387	23,526	48,878	47,433	44,097	24,621
U.Austria	164,841	96,655	72,302	32,486	81,688	51,534	52,333	38,162
Salzburg	1,024,971	438,376	301,256	226,987	246,001	238,687	248,339	251,169
Styria	286,442	270,685	180,891	84,232	133,905	100,652	94,616	96,439
Tyrol	1,694,054	738,074	528,695	339,196	471,150	426,453	428,527	331,747
Vorarlberg	330,423	176,804	98,707	93,522	125,734	141,446	198,356	73,532
Vienna	243,420	228,745	185,026	181,462	86,786	159,606	152,804	154,390

Table 23: Test RMSE Results Comparison

The following table 24 shows the results of STANN models by adding first-order neighbours' time lags as inputs. The result does not show much improvement of accuracy compared to previous models probably due to the weak spatial autocorrelation based on spatial adjacency matrix.

	Burgenland	Carinthia	L.Austria	U.Austria	Salzburg	Styria	Tyrol	Vorarlberg	Vienna
R^2	0.9795	0.9748	0.9600	0.9744	0.9711	0.9359	0.9650	0.9669	0.9164
RMSE	16,610	138,396	31,072	39,717	202,805	91,163	413,935	73,773	87,763

Table 24: Results for Spatio-Temporal ANN Models.

4. Different Forecasting Horizons
One-step ahead prediction (see figure 25) and multi-step ahead prediction methods are adopted for forecasting. There are two different forecasting horizons: the long one is about seven years while the short one is only for one year. Usually, short term prediction can reach a higher accuracy (see table 25). However, for Burgenland and Vorarlberg long term one-step ahead prediction performs even better than short term prediction, which may indicate the ANN models are well trained. For short term prediction, the two methods do not have obvious differences but for long-term forecasting, multi-step ahead models produce a low performance in some provinces like Upper Austria and Vienna (see figure 26). It is possibly because ANNs is sensitive to the accumulation of errors for the long term forecasting horizon.

	Burgenland	Carinthia	L.Austria	U.Austria	Salzburg	Styria	Tyrol	Vorarlberg	Vienna
1-step ahead									
2010-2017	0.9876	0.9917	0.9831	0.9821	0.9806	0.9682	0.9792	0.9759	0.9545
2010-2011	0.9861	0.9947	0.9984	0.9928	0.9877	0.9749	0.9866	0.9759	0.9683
Multi-step ahead									
2010-2017	0.9618	0.9738	0.9736	0.9580	0.9681	0.9108	0.9750	0.9633	0.7352
2010-2011	0.9721	0.9936	0.9975	0.9939	0.9846	0.9777	0.9865	0.9637	0.9695

Table 25: R Square Results of One- and Multi-Step Ahead Prediction Models with Different Forecasting Horizons. Green Means Higher R Square Value.

3.3.4 Summary and Discussion

In conclusion, a number of MLP networks were obtained on basis of combining the following factors: different training range (4), the number of input nodes (8); with or without spatial information (1), the number of hidden nodes (6), and forecasting horizons (2). Generally, all models perform well but still different neural models with different combinations of input variables perform differently. In this article, lagged terms are chosen following a common approach in the literature (e.g. t-1, t-2, t-3). However, in the future, models can be improved through identifying a set of key important inputs by introducing feature selection techniques. To sum up, there is no formal method to build ANN models neither no standard criteria of the optimal number of hidden nodes. Therefore it is very necessary to do several experiments and repeat a certain times to get a proper result.

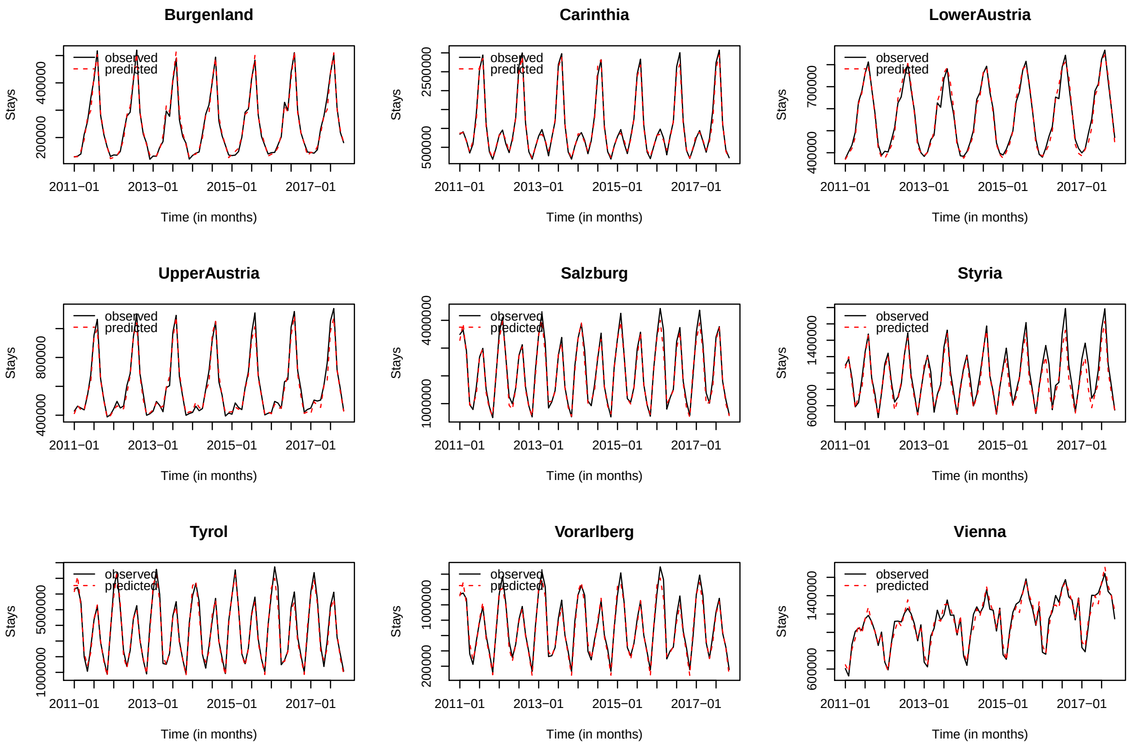


Figure 25: One-Step Ahead Prediction

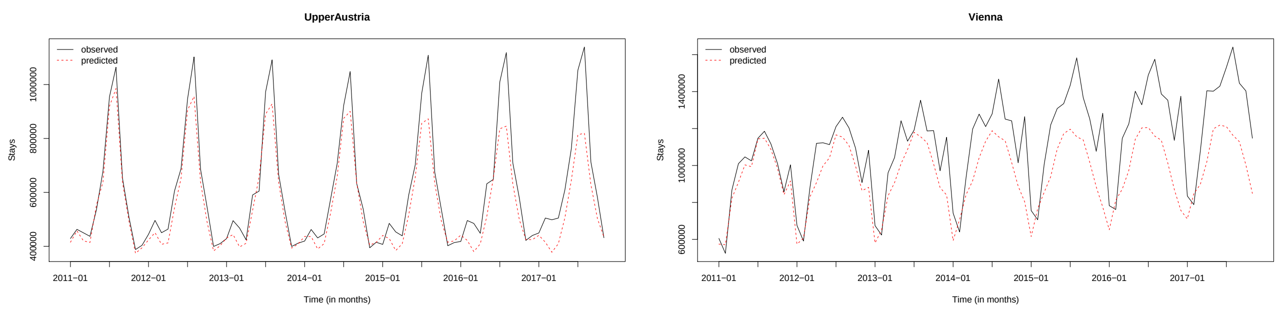


Figure 26: N-Step Ahead Prediction

3.4 Support Vector Machine

3.4.1 Methodology

Support Vector Machine (SVM) is a powerful machine learning method usually related to classification. However this could also be effectively used for regression. Here the regression mode also known as Support Vector Regression or SVR is used to predict the total night stays of tourist for each provinces in Austria using historical data. SVR was first introduced by Drucker and Vapnik 1997 as a version of SVM for performing non linear regression of high dimensional data. SVM looks at the extreme of the data points to classify objects. While support vector classification uses hyperplanes to split the data into separate classes the Support vector regression attempts to bind the data using extreme values. The extreme values or vectors also known as support vectors are used to define a regression function where the predicted values falls within a "tube" of width twice epsilon for related x values (Drucker and Vapnik 1997). Applying the concept of generalization performance (Boser and Vapnik 1992) an approximation of the regression function is estimated using a subset rather than the full dataset, which improves its performance on untrained data and also makes the method memory efficient. SVR controls the generalization performance by following the principle of structural risk minimization (V. Vapnik 1995) instead of empirical risk minimization (M. Claveria and Torra 2016). Various types of kernels are used for the estimation of the function. Some of the popular kernels are Linear kernel, Polynomial kernel and Gaussian RBF kernel. The kernels functions by mapping the data into higher dimensional space(Scholkopf and Smola 2002). Subsequently choosing the kernel is an important task and multiple runs are carried out using unique combination of kernels and specific parameters by grid search to finalize the model. Once chosen the parameters also has to be fine tuned to avoid over fitting to a certain subset, by k-fold cross validation of different subsets of the training data.

3.4.2 Experimental set up

There are nine provinces in Austria and the historical monthly data of tourist Night stays in the provinces from 1973 November to 2017 November is used for modelling. The auto regressive SVR model would be suitable in this case since there is insufficient knowledge of predictors which attract tourist to these places. Although according to Witt and Witt 1992 time series models perform better than regression models in case of short term seasonal prediction. Exploratory analysis of the tourist data set has revealed that there is indeed strong seasonal variation in the data for each of the provinces. In terms of spatial component of the data no significant spatial autocorrelation was found both for arrival and night stays, whereas slightly higher local Moran's I estimates did show presence of local spatial autocorrelation to some degree which varied over time from 1974 to 2017, but is not significant as well.
A total of 529 months of data points ranging from 1973, November to 2017, November is split for training and testing set at 2010 December. Thus 84% of monthly tourist night stays data from November, 1973 to December, 2010, is used for training and the rest for testing.
Besides, the SVR model requires different set of input parameters that can be varied for obtaining the best fit. First the general parameter is cost represented as C, which is a constant to control the amount of error to be allowed in the solution. The cost penalizes error hence larger the cost, less error is permitted. Second important parameter is epsilon which is the width of the tube. Choosing epsilon is significant since if epsilon is set very low then there is possibility of including the noise in the dataset, whereas if its very high then it would not be able to capture the variability of the dataset(M. Claveria and Torra 2016). These parameters are followed by the input of various kernels and their respective parameters.

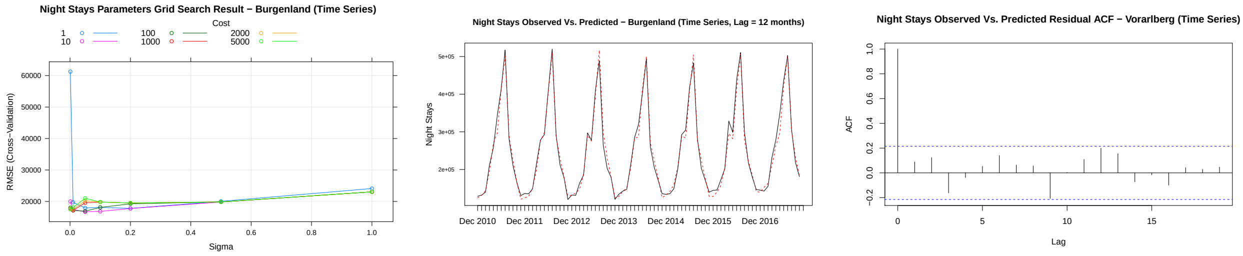
3.4.3 Results - Time Series SVR Model

In case of absence of spatial autocorrelation as revealed by the indices only the time series model would fit the data well.Therefore first a SVR time series regression is attempted using various time lags of month for creating the X and y testing and training components. This is initiated with Burgenland province, with lag 6 and 12 and the results are as follows.

Province	Kernel	RMSE Train	RMSE Test	R.Sq. Train	R.Sq. Test	Lag (m)
Burgenland	Radial Basis	25472.08	170770.2	0.9679858	0.9578215	6
Burgenland	Radial Basis	17361.51	14351.83	0.9864797	0.9851352	12

Table 26: Comparison Using Different Lags in Support Vector Regression (SVR)

The result shows that lag 12 is performing better than lag six, which is finalized in the main model followed by tuning the parameters with a grid search table. The selected model for Burgenland with the finalized parameters is used to predict Night Stays for the period of 2011 to 2017 and plotted against the observed data for the same duration.



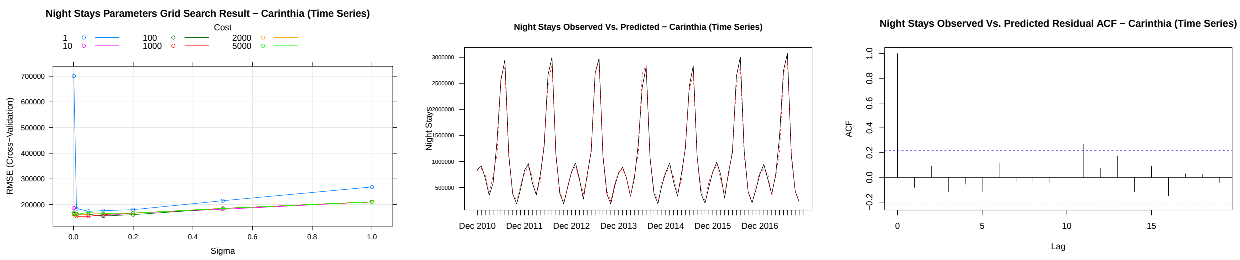
(a) Training errors for parameters (b) Prediction Vs. Observation Fit (c) Residual Autocorrelation

Figure 27: Burgenland Time Series Modelling

Province	Kernel	RMSE Train	RMSE Test	R.Sq. Train	R.Sq. Test	Lag (m)
Burgenland	Radial Basis	17361.51	160926.6	0.9864797	0.9848345	12

Table 27: Burgenland Final Model Result

The plot shows a tight fit against the observed night stays which is also validated by the residual ACF plot, which shows hardly any presence of significant autocorrelation in the residual. Subsequently the process is repeated for all the provinces using lags of six and 12 months while varying the kernels, followed by fine tuning their respective parameters with grid search. The results obtained are shown in the following.



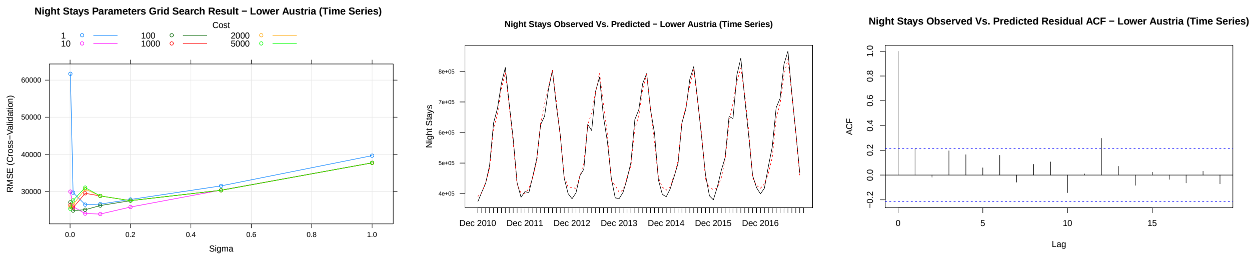
(a) Training errors for parameters (b) Prediction Vs. Observation Fit (c) Residual Autocorrelation

Figure 28: Carinthia Time Series Modelling using SVR

Province	Kernel	RMSE Train	RMSE Test	R.Sq. Train	R.Sq. Test	Lag (m)

Carinthia	Radial Basis	156698.50	84903.57	0.989495	0.9905645	12

Table 28: Carinthia Final Model Result

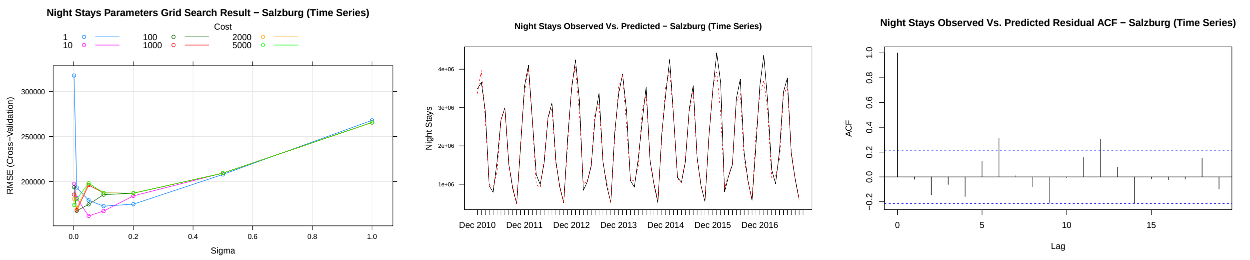


(a) Training errors for parameters (b) Prediction Vs. Observation Fit (c) Residual Autocorrelation

Figure 29: Lower Austria Time Series Modelling using SVR

Province	**Kernel**	**RMSE Train**	**RMSE Test**	**R.Sq. Train**	**R.Sq. Test**	**Lag (m)**
Lower Austria	Radial Basis	23651.09	19964.21	0.9873452	0.9824075	12

Table 29: Lower Austria Final Model Result

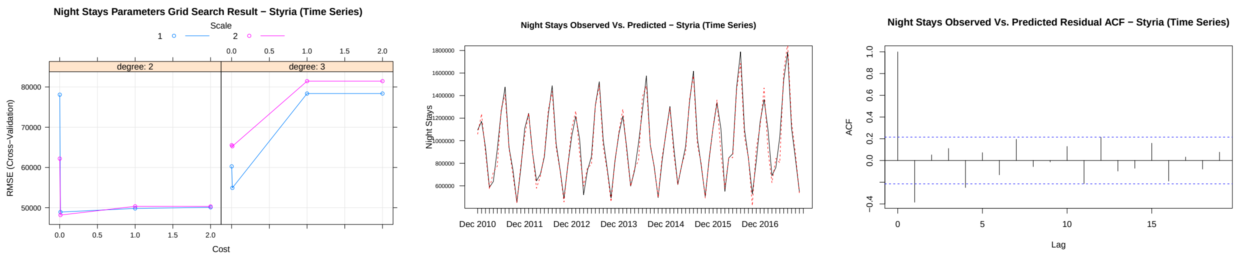


(a) Training errors for parameters (b) Prediction Vs. Observation Fit (c) Residual Autocorrelation

Figure 30: Salzburg Time Series Modelling using SVR

Province	**Kernel**	**RMSE Train**	**RMSE Test**	**R.Sq. Train**	**R.Sq. Test**	**Lag (m)**
Lower Austria	Radial Basis	160862.1	194640.2	0.9770268	0.9767925	12

Table 30: Salzburg Final Time Series Model Result

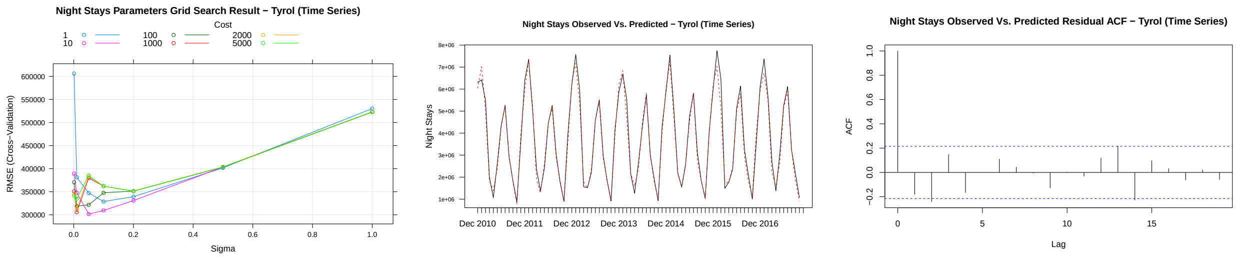


(a) Training errors for parameters (b) Prediction Vs. Observation Fit (c) Residual Autocorrelation

Figure 31: Styria Time Series Modelling using SVR

Province	Kernel	RMSE Train	RMSE Test	R.Sq. Train	R.Sq. Test	Lag (m)
Styria	Polynomial	49295.76	68445.17	0.9808474	0.9563754	12

Table 31: Styria Final Time Series SVR Model Result

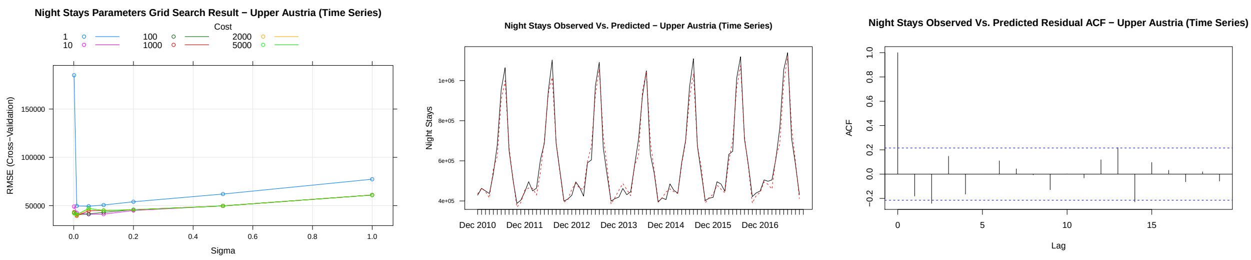


(a) Training errors for parameters (b) Prediction Vs. Observation Fit (c) Residual Autocorrelation

Figure 32: Tyrol Time Series Modelling using SVR

Province	Kernel	RMSE Train	RMSE Test	R.Sq. Train	R.Sq. Test	Lag (m)
Tyrol	Radial Basis	308242.8	306728.4	0.9750711	0.9786044	12

Table 32: Tyrol Final Time Series SVR Model Result

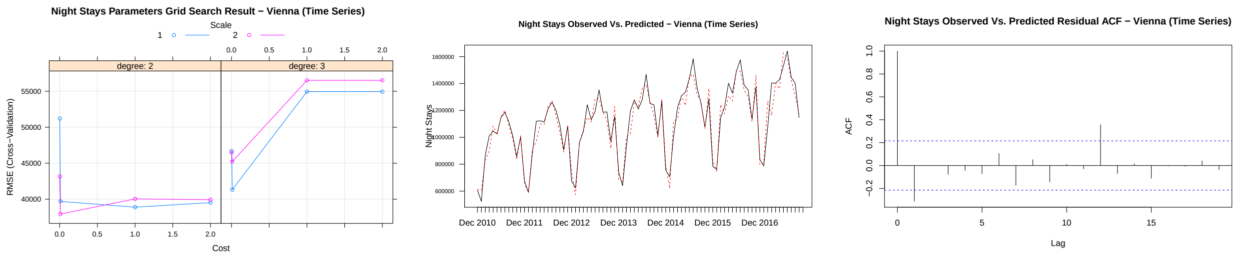


(a) Training errors for parameters (b) Prediction Vs. Observation Fit (c) Residual Autocorrelation

Figure 33: Upper Austria Time Series Modelling using SVR

Province	Kernel	RMSE Train	RMSE Test	R.Sq. Train	R.Sq. Test	Lag (m)
Upper Austria	Radial Basis	40096.51	33886.15	0.9904772	0.9788871	12

Table 33: upper Austria Final Time Series SVR Model Result

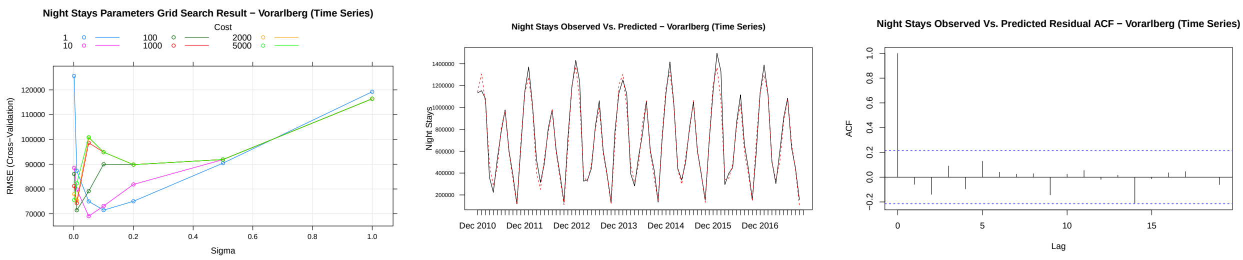


(a) Training errors for parameters (b) Prediction Vs. Observation Fit (c) Residual Autocorrelation

Figure 34: Vienna Time Series Modelling using SVR

Province	Kernel	RMSE Train	RMSE Test	R.Sq. Train	R.Sq. Test	Lag (m)
Vienna	Polynomial	39872.73	68355.33	0.9703751	0.932065	12

Table 34: Vienna Final Time Series SVR Model Result



(a) Training errors for parameters (b) Prediction Vs. Observation Fit (c) Residual Autocorrelation

Figure 35: Vorarlberg Time Series Modelling using SVR

Province	Kernel	RMSE Train	RMSE Test	R.Sq. Train	R.Sq. Test	Lag (m)
Vorarlberg	Radial Basis	70599.05	65505.49	0.9645311	0.9714037	12

Table 35: Vorarlberg Final Time Series SVR Model Result

Finally all the results are compiled in the table below. The result shows a high test R-square for all the provinces using the lag of 12 month rather than six. In most of the cases the RBF Kernel returns the best fit except in case of Vienna and Styria where the Polynomial kernel with degrees 2 out performed the RBF kernels. This is interesting since for the case of Vienna there seems to be multiple annual peaks of night stays compared to other provinces.

Province	Kernel	RMSE Train	RMSE Test	R.Sq. Train	R.Sq. Test	Lag (m)
Burgenland	Radial Basis	25472.08	170770.2	0.9679858	0.9578215	6
Burgenland	Radial Basis	17361.51	14351.83	0.9864797	0.9851352	12
Carinthia	Radial Basis	160763	151872.4	0.9750809	0.9882232	6
Carinthia	Radial Basis	156698.50	84903.57	0.989495	0.9905645	12
Lower Austria	Radial Basis	40460.03	33884.31	0.965073	0.9534268	6
Lower Austria	Radial Basis	23651.09	19964.21	0.9873452	0.9824075	12
Salzburg	Radial Basis	212233.70	318559	0.9601843	0.9359245	6
Salzburg	Radial Basis	160862.1	194640.2	0.9770268	0.9767925	12
Styria	Radial Basis	45190.7	90598.99	0.9834137	0.9470502	12
Styria	Polynomial	49295.76	68445.17	0.9808474	0.9563754	12
Tyrol	Radial Basis	416286.5	451811.5	0.9553336	0.9579362	6
Tyrol	Radial Basis	308242.8	306728.4	0.9750711	0.9786044	12
Upper Austria	Radial Basis	67516.86	69164.82	0.9749186	0.9076577	6
Upper Austria	Radial Basis	40096.51	33886.15	0.9904772	0.9788871	12
Vienna	Radial Basis	34655.17	376822.9	0.9779828	0.2135794	12
Vienna	Polynomial	39872.73	68355.33	0.9703751	0.932065	12
Vorarlberg	Radial Basis	88921.27	100286.6	0.9444394	0.9536337	6
Vorarlberg	Radial Basis	70599.05	65505.49	0.9645311	0.9714037	12

Table 36: Time Series Modelling Using Different Lags in Support Vector Regression (SVR)

3.4.4 Results - Space-Time SVR Model

The time series model discussed in the previous section fitted the observed data considerably well as was found from the RMSE and r-square values. Although no significant spatial autocorrelation was indicated in the data, which may be due to estimation using too few data points, a Space-Time Modelling of the data is attempted to verify if there would be any improvements in result. For implementing the space-time algorithm first the spatial weight matrix is estimated. Two types of weight matrix is tested to find the suitable weight matrix. One is the spatial adjacency weight matrix which is calculated on the basis of provinces sharing boundary with each other. Subsequently a SVR model is fitted using this weight matrix for Burgenland. The result shows that the RMSE values for the test increases for the test, though the model fits well on the training with high R-Squares for the training model.

Weight Matrix	RMSE Train	RMSE Test	R-Sq Train	R-Sq Test	Lag
W – Adjacency	8491.55	21580.54	0.9287	0.7237	2
W – Adjacency	5728.35	20252.70	0.9673	0.7730	3
W – Adjacency	3511.3470	17840.2300	0.9877	0.8192	4

Table 37: Space-Time Series SVR Model Result Using Adjacency Weight Matrix

Now the test is repeated using a second spatial weight matrix which is created using the length of the shared border of each neighboring provinces and row normalized. The parameters are retained from the previous model for comparison.

Weight Matrix	RMSE Train	RMSE Test	R-Sq Train	R-Sq Test	Lag
W – boundary	5481.2710	13408.2200	0.9699	0.8499	2

W – boundary	3547.4340	11452.3700	0.9874	0.9144	3
W – boundary	3028.29	11849.80	0.9908	0.9196	4

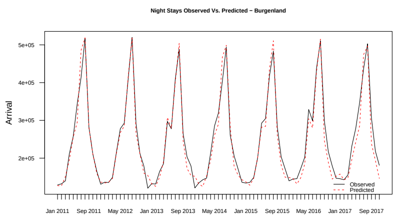
Table 38: Space-Time Series SVR Model Result Using Length of Boundary Shared Weight Matrix

The result shows that there is slight improvement in both the training as well as the test model. The residual autocorelation plot of both the type of weight matrices plot also shows there has been improvement with lesser ACF present in the residual for the length share matrix. Henceforth the shared length boundary spatial weight matrix is used to embed the data. Subsequently the data set is split into test and train, fitted to a space-time SVR model while fine tuning the model by varying the different general parameters along with the kernels of radial basis and polynomial function and the related kernel specific parameter. The result is measured using the RMSE values and the R square values and is summarized as follows:

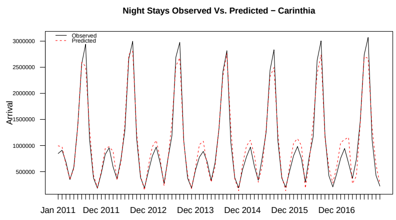
Provinces	**Train RMSE**	**Test RMSE**	**Train R Square**	**Test R Square**	**Kernel**
Burgenland	26246.46	34420.62	0.9688	0.9248	Polynomial
Carinthia	88727.83	168847.00	0.9964	0.9644	Radial Basis
Lower Austria	10982.09	26954.60	0.9974	0.9657	Radial Basis
Salzburg	107766.90	281138.70	0.9896	0.9465	Radial Basis
Styria	41728.64	73149.92	0.9864	0.9558	Radial Basis
Tyrol	267445.10	381466.00	0.9815	0.9645	Radial Basis
Upper Austria	58961.09	78319.51	0.9819	0.9359	Radial Basis
Vienna	32918.21	98301.73	0.9797	0.8970	Polynomial
Vorarlberg	41003.13	77126.05	0.9880	0.9592	Radial Basis

Table 39: Space-Time Series Final SVR Model Result Using Length of Boundary Shared Weight Matrix

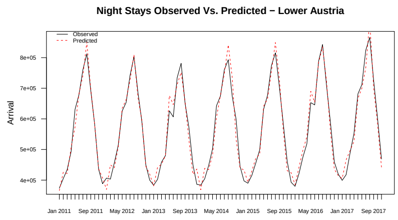
Among the various kernels the polynomial and Gaussian RBF kernels returned the best fit. The polynomial kernels performed better for data with irregular pattern of multiple peaks and troughs where RBF kernels were unable to capture the deviation especially for Burgenland and Vienna. In the rest of the instances RBF outperformed polynomial kernels in terms of both fit and training time. The training time for the polynomial kernels model were significantly higher than a SVR model using a Radial basis kernel.



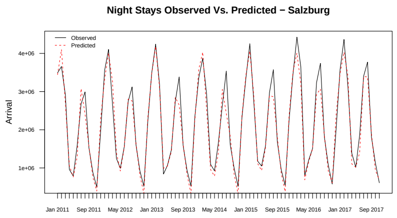
(a) Burgenland



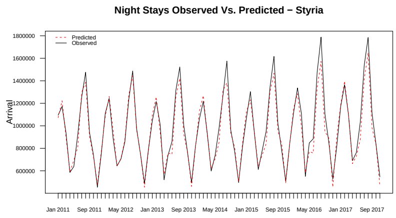
(b) Carinthia



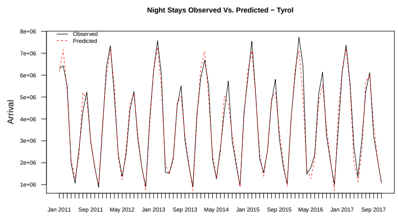
(c) LowerAustria



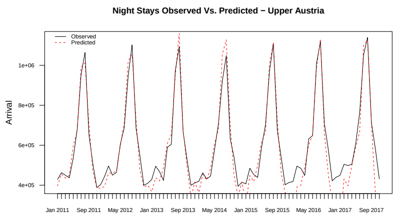
(d) Salzburg



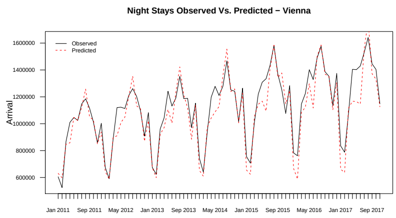
(e) Styria



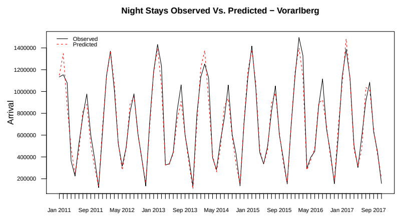
(f) Tyrol



(g) Upper Austria



(h) Vienna



(i) Vorarlberg

Figure 36: Observed vs Predicted Night Stays values for the SVR Space-Time model

3.4.5 Discussion

Comparison of the SVR model output from the time series versus the space-time series models clearly indicate that the time series model performs significantly better than the space-time model. Hence it could be concluded that there is not much spatial component in the data, as was indicated by Moran's I.

Provinces	Tm Test RMSE	Sp-Tm Test RMSE	Tm Test R-Sq	Sp-Tm Test R-Sq
Burgenland	14352	34421	0.9851	0.9248
Carinthia	84904	168847	0.9906	0.9644
Lower Austria	19964	26955	0.9824	0.9657
Salzburg	194640	281139	0.9768	0.9465
Styria	68445	73150	0.9564	0.9558
Tyrol	306728	381466	0.9786	0.9645
Upper Austria	33886	78320	0.9789	0.9359
Vienna	68355	98302	0.9321	0.8970
Vorarlberg	65505	77126	0.9714	0.9592

Table 40: Time Series Vs. Space-Time Series SVR Model Comparison of Night Stays

Another interesting output of the SVR model is the number of support vectors used by the model to train the data. The table below shows that the maximum number of support vectors used was 47 percent of the train data for the provinces of Tyrol and Vienna, while the least was for the province of Carinthia which was only around 19 percent of the data, and fitting the data considerably well with 0.989 r square. This is one of the strength of Support Vector Regression that it only requires a fraction of the data to fit a model reasonably well which prevents the model from over fitting.

Provinces	No. of Support Vectors	Percentage of Data	Kernel	Training Time(sec)
Burgenland	120	27.65%	Polynomial	123.54
Carinthia	81	18.66%	Radial Basis	2.92
Lower Austria	93	21.43%	Radial Basis	1.82
Salzburg	181	41.71%	Radial Basis	0.49
Styria	175	40.32%	Polynomial	1.14
Tyrol	204	47.00%	Radial Basis	1.12
Upper Austria	89	20.51%	Radial Basis	1.42
Vienna	204	47.00%	Polynomial	207.95
Vorarlberg	193	44.47%	Radial Basis	0.67

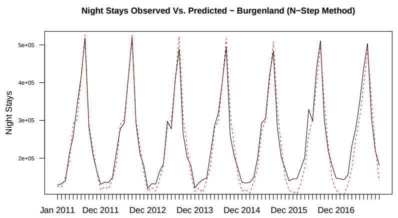
Table 41: No. of Support Vectors and Kernel Training Time for SVR Models

Finally the n-step method is used to predict tourist night stays from 2011, January to 2017, November using only 12 months of data of the year 2010 for each of the provinces. The predictions are then compared with actual data from during the same period

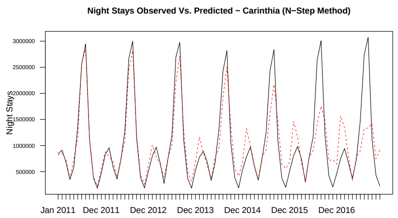
Provinces	RMSE	R-Square
Burgenland	26856	0.959427
Carinthia	417777	0.773698
Lower Austria	75217	0.827141
Salzburg	219775	0.974860
Styria	138864	0.895087
Tyrol	410525	0.971745
Upper Austria	92469	0.943894
Vienna	220018	0.581188
Vorarlberg	113202	0.938909

Table 42: Comparison of SVR N-Step Predicted Result to Observed Night Stays, 2011 – 2017

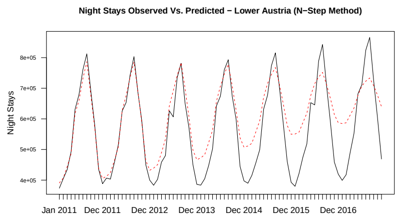
The best prediction is obtained for Salzburg with a r-square of 0.9748, whereas the worst is estimated for Vienna with a r-square of 0.5811.



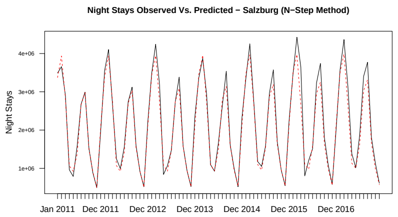
(a) Burgenland



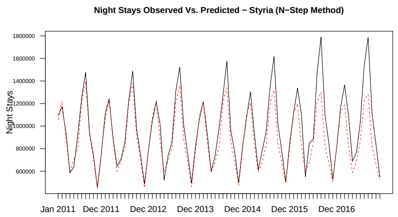
(b) Carinthia



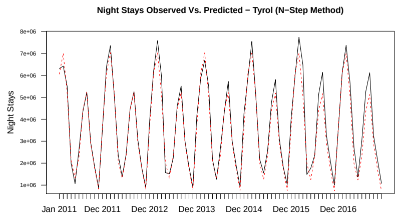
(c) LowerAustria



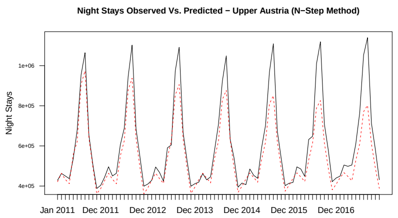
(d) Salzburg



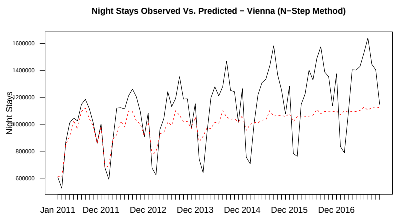
(e) Styria



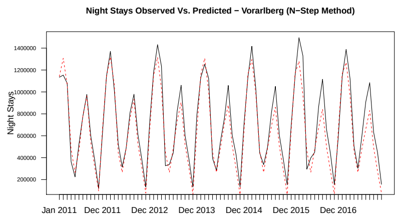
(f) Tyrol



(g) Upper Austria



(h) Vienna



(i) Vorarlberg

Figure 37: Observed vs Predicted Night Stays values Estimated by N-Step SVR model

Summarising the main findings from the experiment with the Support Vector Regression model it could be concluded that it is a very efficient method in terms of requiring the least number of data points to fit the data which does well to predict unseen data. However finding the optimum parameters implementing an elaborate grid search is critical for a good performance of the model since they are quite sensitive to the final outcome.

4 Discussion and Outlook

In this section, the results of the four different methods are compared, the experiment critically reviewed, recommendations for further research given and a prediction for tourist stays in Austria's provinces over the next five years displayed.

For the performance comparison (table 43 and 44) the best performing model of each method is used. This leads to a mixture of one- and multi-step ahead predicting models, but as seen in section 3.2 the multi-step ahead predictions do not necessarily perform worse. It can be seen easily that *RFR* and *SVM* perform almost always worse than the other two methods. *ANN* and *STARIMA* perform sometimes best, sometimes worst. *RFR* seems to be the worst performing model followed by *SVM* in terms of RMSE as well as R^2. Finally, *ANN* appears most often as best/next best performing method (especially in table 44). However, this finding raises the question why a non-spatial model performs best.

Table 43: RMSE of the best performing models of each method. Green represents the best model, red the worst.

Province	RFR	(ST)ARIMA	ANN	SVM
Burgenland	18,130	14,539	13,593	14,351
Carinthia	79,950	81,993	79,649	84,903
Lower Austria	23,383	18,025	18,865	19,964
Upper Austria	31,862	28,269	32,830	33,886
Salzburg	204,958	201,168	186,846	194,640
Styria	67,876	57,091	58,431	68,445
Tyrol	318,733	326,693	298,194	306,728
Vorarlberg	73,886	62,134	62,224	65,505
Vienna	63,687	52,210	87,913	68,355

Table 44: R^2 of the best performing models of each method. Green represents the best model, red the worst.

Province	RFR	(ST)ARIMA	ANN	SVM
Burgenland	0.9752	0.9840	0.9866	0.9851
Carinthia	0.9913	0.9906	0.9918	0.9905
Lower Austria	0.9737	0.9850	0.9836	0.9824
Upper Austria	0.9794	0.9869	0.9808	0.9788
Salzburg	0.9713	0.9747	0.9817	0.9767
Styria	0.9550	0.9727	0.9716	0.9563
Tyrol	0.9786	0.9767	0.9802	0.9786
Vorarlberg	0.9621	0.9744	0.9731	0.9714
Vienna	0.9392	0.9588	0.9605	0.9320

As a non-spatial method often performs best, it seems that in our experiment the spatial correlation is not large enough, the number of provinces too small or the spatial weight matrices used inappropriate for getting best predictions. Surprisingly, *ARIMA* designed for an explicit modelling of time series, especially seasonal ones, is not more accurate than the machine-learning technique *ANN*. However, one has to note, that (ST)ARIMA already uses n-step ahead prediction in the previous tables. To enable a more reasonable comparison, table 45 and 46 offer error metrics of the machine-learning methods regarding n-step ahead prediction. *RFR* surpasses *SVM* and even *ANN* when using n-step ahead prediction. Unfortunately, the tables also show that *RFR* has major issues in two-seasonal stays prediction. In respect of overall performance with n-step ahead prediction (see tables 43 and 44), (ST)ARIMA turns out to be the best model.

Table 45: RMSE of n-step ahead prediction of *RFR*, *ANN* and *SVM*. Green represents the best model, red the worst.

Province	RFR	ANN	SVM
Burgenland	19,522	27,247	26,856
Carinthia	87,145	182,007	417,777
Lower Austria	33,597	30,238	75,217
Upper Austria	36,349	56,583	92,469
Salzburg	309,324	260,445	219,775
Styria	110,711	102,530	138,864
Tyrol	431,814	384,418	410,525
Vorarlberg	77,906	81,598	113,202
Vienna	99,090	146,985	220,018

Table 46: R^2 of n-step ahead prediction of *RFR*, *ANN* and *SVM*. Green represents the best model, red the worst.

Province	RFR	ANN	SVM
Burgenland	0.9774	0.9618	0.9594
Carinthia	0.9896	0.9738	0.7737
Lower Austria	0.9483	0.9736	0.8271
Upper Austria	0.9799	0.9580	0.9439
Salzburg	0.9556	0.9681	0.9749
Styria	0.8787	0.9108	0.8951
Tyrol	0.9652	0.9750	0.9718
Vorarlberg	0.9614	0.9633	0.9389
Vienna	0.8677	0.7352	0.5812

The methods presented in section 3 should not only be compared in terms of accuracy but also in ease of use, interpretability and running time. Once the models are trained all methods can predict very fast (see table 47).

Table 47: Pros and cons of each method.

	RFR	(ST)ARIMA	ANN	SVM
Pros	– only 2-3 parameters – no pre-processing necessary – fast training (seconds) – variable importance plot gives hints for interpretation – good stability of n-step ahead predictions	– overall best predictions – explicit method, good interpretability of parameters – multi-step ahead prediction available by default for ARIMA – very fast training (sub-second)	– only 1-2 parameters – easy to handle – fast training (seconds) – model architecture is simple and clear with three layers and several nodes. – responses can be multivariate vectors	– fast training when using RBF (seconds) – performs well with high dimensional data – returns stable output and gives the same result for repeat runs – requires only a fraction of the dataset as support vectors to train
Cons	– no direct interpretability of parameters (‘black box’, randomness) – parameters have a small influence (contrarily to the lagged data)	– many (7) parameters (p,d,q,P,D,Q,S) to choose – pre-processing may be required – cannot include explanatory variables	– sensitive to data preprocessing, input variables and initial assigned weights. – hard to decide optimal hidden nodes number. – no interpretability of model (‘black box’)	– many (5-6) parameters depending on kernels – parameters are difficult to choose (interdependence, highly sensitive) – slow training when using polynomial (minutes)

In terms of the different provinces, the ones that show an almost constant pattern over the whole time period (e.g. Lower Austria, Upper Austria) can be predicted most accurately. Less accurate modelled can be provinces with two peaks a year like Carinthia, Tyrol and Vorarlberg. Most likely because the two peaks show an uncorrelated behaviour. Difficult to predict is also the highly variable shape of the annual pattern for Vienna. The shape is potentially caused by the number of weekends that are in a month (4 or 5) and the fact that the date of Easter (and all connected holidays) can vary up to roughly a month.

Furthermore, it should be considered that the machine-learning techniques could potentially predict more precise when additional, explanatory data would be included. It shall be referred to Lim 1997 for further details about explanatory variables of tourism. Unfortunately, in this work no additional data like flight movements or economic data could have been included, as it was not available, not documented[6], incomplete[7], in an insufficient temporal or spatial resolution or only covering a few epochs. Ideally the data of neighbouring provinces in foreign countries should be included in the forecasting as well, as all provinces but Vienna share a part of their border with foreign countries. Moreover, the choice of the weight matrices used (neighbourhood and border length weighted neighbourhood) seems to be unhelpful. Potentially number of streets (e.g. Openstreetmap data) or vehicles crossing the shared borders or the gradient of hotel or tourist attraction density as weights could improve the predictions. In a real case application, it should be first of all critically assessed which spatial

[6] e.g. https://www.data.gv.at/katalog/dataset/04f1d909-5e01-3387-aa0d-5057affe753d
[7] e.g. https://www.offenerhaushalt.at/

resolution is necessary. The given spatial resolution of provinces may not be the best choice.

In the end, one should never lose sight of the goal of a forecast. In our case it is to predict tourist stays in the future. Typically, the forecast should not only cover the next epoch. Thus, a multi-step ahead prediction is indispensable. We conclude with the prediction over the next five years calculated with the overall best performing *ARIMA* multi-step ahead predicting model *ARn* (see figure 38). Nevertheless, remember that these predictions are solely based on past data and do not show any uncertainty.

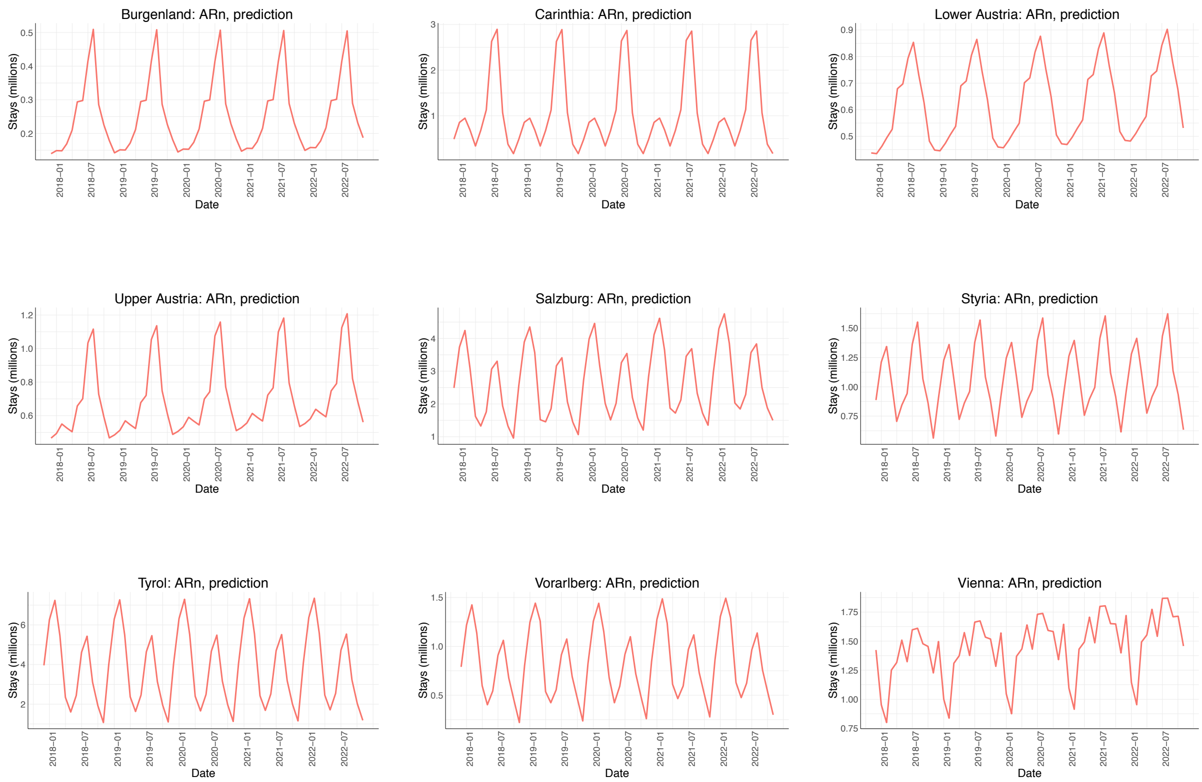


Figure 38: Prediction of tourist stays (in millions) in all Austria's provinces for the next 5 years (2017-12 till 2022-11)

References

Random Forest Regression

Breiman, Leo (2002). *Manual On Setting Up, Using, And Understanding Random Forests.* 3.1.

— (2001). "Random forests". In: *Machine learning* 45.1, pp. 5–32.

Decision Tree Learning (2018). https://en.wikipedia.org/wiki/Decision_tree_learning. Accessed: 2018-04-25.

Liaw, Andy (Oct. 2015). *Package 'randomForest'.*

Liaw, Andy, Matthew Wiener, et al. (2002). "Classification and regression by randomForest". In: *R news* 2.3, pp. 18–22.

Random Forest (2018). https://en.wikipedia.org/wiki/Random_forest. Accessed: 2018-04-25.

Segal, Mark R (2004). *Machine learning benchmarks and random forest regression.*

(ST)ARIMA

Box, George E. P and Gwilym M Jenkins (1970). *Time Series Analysis Forecasting and Control.* Wisconsin Univ Madison Dept Of Statistics.

Brockwell, P.J. and R.A. Davis (2002). *Introduction to Time Series and Forecasting.* Springer. ISBN: 0-387-95351-5.

Cheng, Tao et al. (Jan. 2014). "A Dynamic Spatial Weight Matrix and Localized Space–Time Autoregressive Integrated Moving Average for Network Modeling". In: *Geographical Analysis* 46.1, pp. 75–97. ISSN: 0016-7363.

Duan, Peibo et al. (Nov. 2016). "STARIMA-based traffic prediction with time-varying lags". eng. In: IEEE, pp. 1610–1615. ISBN: 978-1-5090-1889-5.

Gurland, John (1954). *Hypothesis Testing in Time Series Analysis.*

Liu, Jie and Enrico Zio (2017). "SVM hyperparameters tuning for recursive multi-step-ahead prediction". In: *Neural Computing and Applications* 28.12, pp. 3749–3763.

Pfeifer, Phillip E and Stuart Jay Deutsch (1981). "Seasonal Space-Time ARIMA Modeling". In: *Geographical analysis* 13.2, pp. 117–133.

Artificial Neural Networks

Burger, CJSC et al. (2001). "A practitioners guide to time-series methods for tourism demand forecasting—a case study of Durban, South Africa". In: *Tourism management* 22.4, pp. 403–409.

Claveria, Oscar and Salvador Torra (2014). "Forecasting tourism demand to Catalonia: Neural networks vs. time series models". In: *Economic Modelling* 36, pp. 220–228.

Dong, Guanqun, Kamaladdin Fataliyev, and Lipo Wang (2013). “One-step and multi-step ahead stock prediction using backpropagation neural networks”. In: *Information, Communications and Signal Processing (ICICS) 2013 9th International Conference on*. IEEE, pp. 1–5.

Law, Rob and Norman Au (1999). “A neural network model to forecast Japanese demand for travel to Hong Kong”. In: *Tourism Management* 20.1, pp. 89–97.

Makridakis, Spyros et al. (1982). “The accuracy of extrapolation (time series) methods: Results of a forecasting competition”. In: *Journal of forecasting* 1.2, pp. 111–153.

Palmer, Alfonso, Juan Jose Montano, and Albert Sesé (2006). “Designing an artificial neural network for forecasting tourism time series”. In: *Tourism Management* 27.5, pp. 781–790.

Raschka, Sebastian (2018). *Machine Learning FAQ*. URL: `https://sebastianraschka.com/faq/docs/visual-backpropagation.html` (visited on 04/15/2018).

Riedmiller, Martin and Heinrich Braun (1993). “A direct adaptive method for faster backpropagation learning: The RPROP algorithm”. In: *Neural Networks, 1993., IEEE International Conference on*. IEEE, pp. 586–591.

Rojas, Raúl (2013). *Neural networks: a systematic introduction*. Springer Science & Business Media.

Taieb, Souhaib Ben (2014). “Machine learning strategies for multi-step-ahead time series forecasting”. PhD thesis. Universit Libre de Bruxelles, Belgium.

Zhang, Guoqiang, B Eddy Patuwo, and Michael Y Hu (1998). “Forecasting with artificial neural networks:: The state of the art”. In: *International journal of forecasting* 14.1, pp. 35–62.

Support Vector Machine

Boser, Guyon and Vapnik (1992). “A training algorithm for optimal margin classifiers.” In: *COLT ’92 Proceedings of the fifth annual workshop on Computational learning theory*, pp. 144–152.

Claveria, Monte and Torra (2016). “Modelling tourism demand to Spain with machine learning techniques. The impact of forecast horizon on model selection.” In: *Revista de Economia Aplicada* 24, pp. 109–132.

Drucker Burges, Kaufman and Vapnik (1997). “Support vector regression machines.” In: *Advances in neural information processing systems* 9, pp. 155–161.

Scholkopf and Smola (2002). “Support Vector Machines and Kernel Algorithms.” In: *Statistics and Computing* 2e, p. 7.

Vapnik, V. (1995). “Support vector machine.” In: *Machine learning 20* 20, pp. 273–297.

Witt and Witt (1992). *Modeling and forecasting demand in tourism.* Academic Press. ISBN: 0-127-60740-4.

Miscellaneous

Lim, Christine (1997). “Review of international tourism demand models”. In: *Annals of tourism research* 24.4, pp. 835–849.