Project in Time Series Analysis

Arvid Gramer, Nils Romanus

"**Life can only be understood backwards,
but it must be lived forwards.**"
– Søren Kierkegaard (1844)

FMSN45
Lunds Tekniska Högskola, Lunds Universitet
December 2022

1 Introduction

Prediction of temperature is useful in many applications ranging from climate modeling to planning a holiday. This report examines different methods for temperature prediction using the Swedish town of Svedala as a case study. Section 4 covers methods that use exclusively historical values for predicting future values. Section 5 expands upon this and examines if using prognosis on nearby locations from SMHI as exogenous signals improves temperature prediction. Naturally temperature exhibits dynamics that vary by time and it is therefore reasonable to expect that the statistics of temperature data change over time. Therefore, section 6 examines time recursive methods for temperature prediction. Finally, in section 7, we test Meta's automatic forecasting API *Prophet*. Each constructed model is bench-marked against the naive predictor outlined in section 3

2 Working with Data

It can be quite difficult to immediately gain insight from data that has been gathered from real-world instruments. When trying to model the intrinsic behavior in the data it is therefore valuable to do some exploratory analysis followed by cleaning of the data.

2.1 Data Cleaning

As is often necessary, the data first needs to be cleaned. We begin with handling the missing values in the data set. We then proceed to explore the potential existence of outliers. Finally, we alter the exogenous input data to something closer to the real use case.

2.1.1 Missing value imputation

A visual inspection of a subset of the temperature data from Svedala shows that the data contains missing values. As one may note from the top plot in figure 1 the temperature is set to zero every day at 23:00. It is highly unlikely that the temperature is identically zero every day and one may therefore interpret these as missing values. In order to model the temperature data these missing values need to be imputed. This can be done by using linear interpolation of the data points at 22:00 and 24:00, thus creating a data set that is easier to make sense of. The linear interpolation results in the cleaner subset of data displayed in the middle plot of figure 1. The subset of temperature data displayed in the middle of figure 1 is more interpretable, however, one may still note that there are inconsistencies in the data. It is for example highly unrealistic that the temperature would remain identically zero between Aug 10 22:00 and Aug 11 14:00. By noting this, and similar inconsistencies, one can safely assume that all missing values are set to identically zero. We resolve this by viewing all zeros as missing values and imputing these through linear interpolation.

This results in the cleaned data set displayed in the bottom plot in figure 1. One could argue that viewing all zeros as missing values is an overly crude instrument since temperature actually may be identically zero. This argument is valid, however, the damage that can be caused by interpolating these values is minimal. If the temperature really is identically zero at some points it is likely that the temperature the hour before and after is close to zero. Hence the linear interpolation will result in a value that is close to the actual value of identically zero. Furthermore, since the granularity of the data is 0.1°C not many values are erroneously changed using this approach.

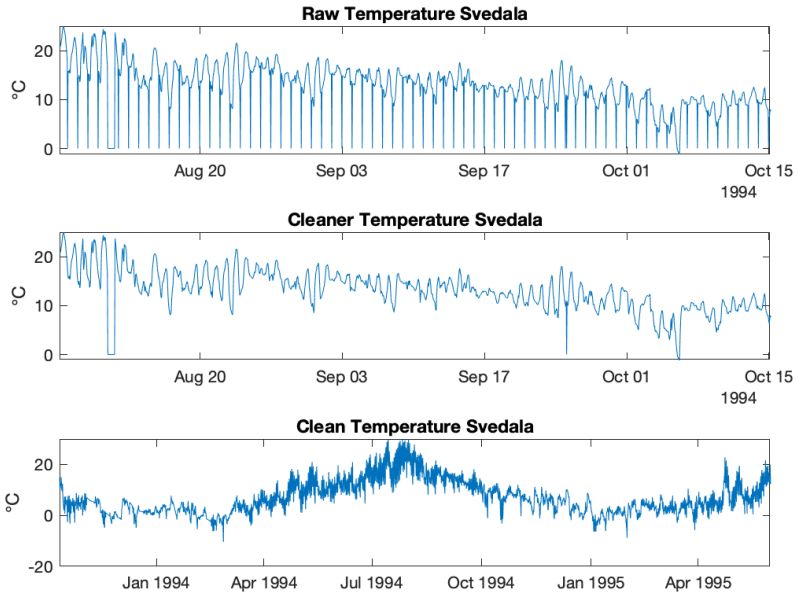

Figure 1: Top: Raw temperature data in Svedala between Aug 6 and oct 15. Middle: The result after linear interpolation of zeroes at 23:00, still showing some signs of missing values.
Bottom: Clean temperature data after having imputed all zeros ass missing values.

2.1.2 Handling outliers

The cleaned data set displayed in figure 1 shows no immediate signs of outliers. In order to further investigate the existence of outliers one may compare the autocorrelation function (ACF) estimate with the α-trimmed autocorrelation function (TACF). The TACF is an ACF estimate using a trimmed data set where the α % smallest and largest values are excluded from the estimate. The

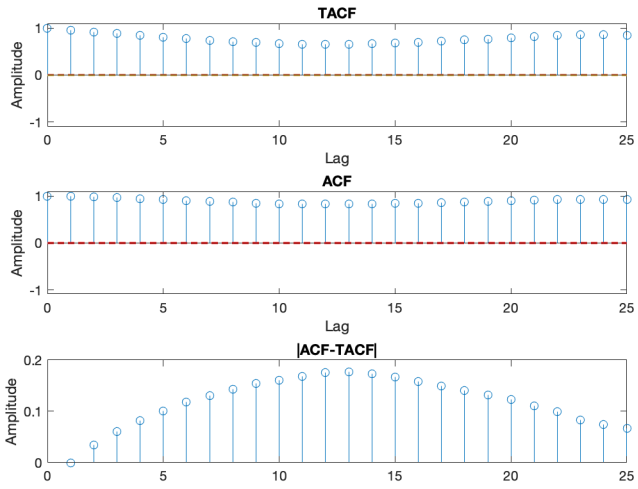

Figure 2: ACF, TACF, and absolute difference between these using a truncation parameter of $\alpha = 0.05$. Since they are similar, with just small differences for time lags around twelve hours, we do not employ any counter measures.

ACF having a similar appearance to the TACF suggests that outliers have an insignificant impact on the overall structure of the data. The ACF and TACF, using $\alpha = 0.05$, displayed in figure 2, shows that the impact of outliers is minimal.

2.1.3 Modifying exogenous signals

One part of the project is to use exogenous input to see if this enhances the predictive power of the model. The data sets that are given for this are series of predictions from SMHI, made three hours in advance. There are predictions for every three hours, at 10:00, 13:00 o'clock and so forth. Between these times the values are linearly interpolated. In order to make this data set more realistic, and closer to the setting in which a predictive model would be deployed, we have conducted the following steps. We have removed the interpolated data points, backshifted all data points three hours, and filled the values between the two predictions with the last prediction. Since the SMHI predictions are made three hours in advance, the prediction for 13:00 is available at 10:00, which is why we have shifted it three hours backward. However, the interpolated data points would then be based on information about the next prediction which would not be available. In order to remedy this we have removed them and forward-filled with the latest prediction. The resulting data is a time series where at every timestamp the value is equal to what would be available if the model was deployed as shown in figure 3.

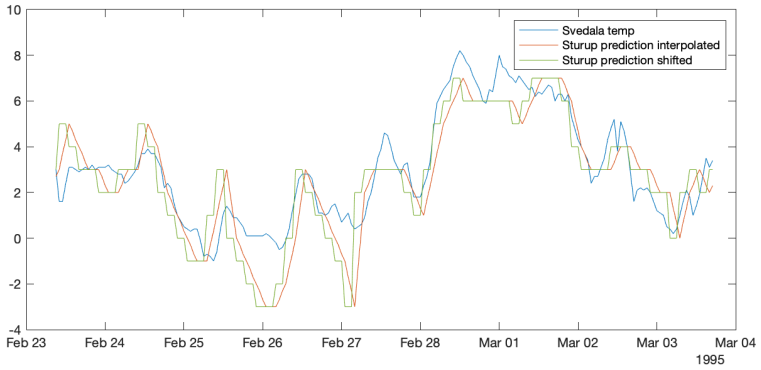

Figure 3: Shifting the input signal. This is made to make the right data available to the model at the right time.

2.2 Splitting

In order to properly design, validate and test a possible model it is beneficial to slice the data into corresponding subsets. This is done by picking 01-Sep-1994, 00:00:00 as the starting timestamp and the following ten weeks as modeling data. The two subsequent weeks are selected as validation data. Finally, 20 weeks after the last timestamp of the validation set, we use two weeks as test data. The modeling data is used to estimate the parameters of a model candidate. The validation data is used to evaluate a model that has been estimated using the modeling data. The test data is left untouched until a final evaluation of all models outlined in section 8.2 of the report. The splitting scheme can be visualized as shown in figure 4 which also includes the SMHI predicted data split by the same methodology.

2.3 Transformations

In order to make a time series more stationary it can sometimes be beneficial to apply a power transform to the data. A power transform may reduce heteroscedasticity, where the variance of the data changes over time. A way to measure the need of power transformation one may examine the Box-Cox normality plot. This is a metric that describes if, and in that case which kind of, power transformation could be useful. We perform this test on our training data and the resulting plot is displayed in figure 5. The λ that maximizes the BC-value is $\lambda = 1.3535$ which in theory suggests that the series should be element-wise raised to the power of 1.3535, however in accordance with section 4.4 in the course book it is often sufficient to use the standard transforms for $\lambda = ... - 0.5, 0, 0.5, 1, 2...$ which implies that we not apply a transform. No transformation might also be wiser since the BC-test is conducted on training data, and we are not sure about the characteristics of the future data used for

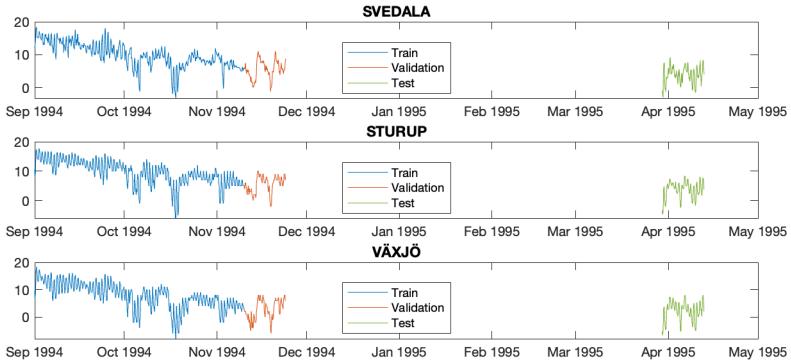

Figure 4: Modeling, validation and test data splits. The modeling data is used to find a good model and estimate parameters and the validation data is used to validate that the model also performs on unseen data. The test data is used when the final model is ready, to give an estimate of how the model would perform in real life.

validation and testing. A third argument for refraining from a transform, which connects to the previous, is that we work according to the motto of *KISS*; Keep It Simple, Stupid.

2.4 Trends

After investigating the need for transforms we study if there are any trends in the data. Since it is temperature over a long period of time, some kind of trend is expected. We begin by fitting a deterministic linear trend to our training which has an intercept of 14.47°C and a slope of -0.0055°C/day. This trend is shown in figure 6. One could have removed this from the data to make it zero-mean and detrended. However, our training data is during fall and the use case will be during all seasons, so this deterministic trend will probably be the opposite of how for example spring data would behave. Therefore we instead choose a stochastic approach and use a differentiator $\nabla = (1 - z^{-1})$ to remove any trend and also make the data zero mean.

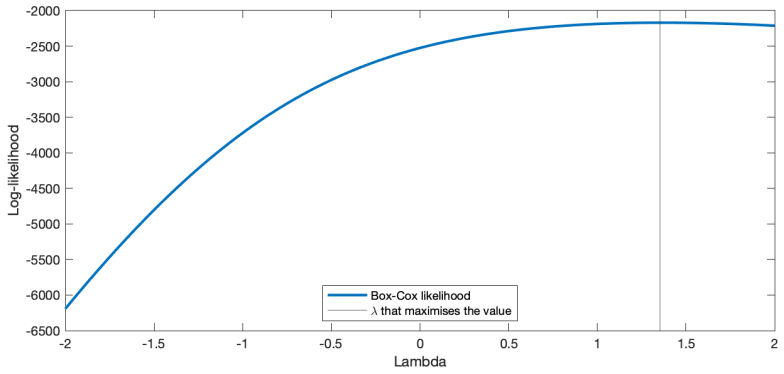

Figure 5: Box-Cox normality plot of the Svedala temperature data. The value that maximises this log-likelihood is suggested as a power transform exponent. $\lambda \approx 1$ indicated that no transform is needed.

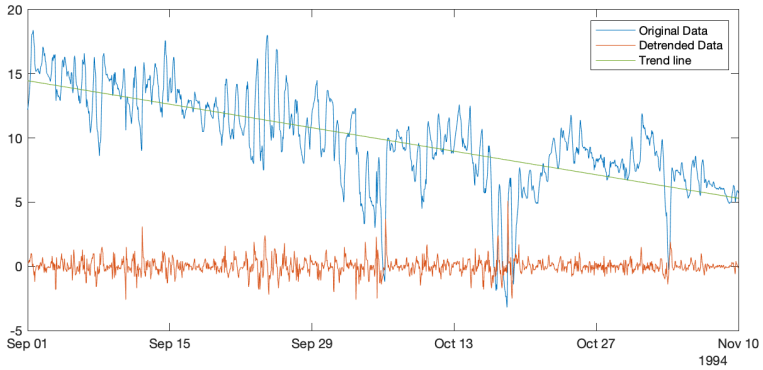

Figure 6: The unaltered data, with trend specified, together with the once differentiated data.

Data Set	$\sigma^2_{t+1\|t}$	$\sigma^2_{t+9\|t}$	σ^2_y
Modeling	1.5267	5.3761	13.5078
Validation	3.1007	11.9268	7.5529

Table 2: Naive Model prediction residual- and data variance.

3 Naively modeling temperature

A predictive model is not useful if it does not outperform a naive method for estimating future temperature. To benchmark possible model candidates one may compare them against the naive predictor. The naive predictor should be intuitive, yet provide some predictive power. In order to construct the naive predictor one can imagine how a human would go about predicting future temperature. It is reasonable to assume that the current temperature would be a good indicator of the temperature the following hour. It is also reasonable to assume that yesterday's temperature, during the same hour of the day, would have some predictive power, especially when extending the prediction horizon to, for example, nine hours. The naive model can be constructed using these two signals, without any parameter estimation, by simply adding yesterday's temperature with the current temperature and weighting them as equally important. The naive model hence becomes a seasonal auto-regressive (AR) model with fixed coefficients outlined in table 1.

	a_0	a_1	a_{24}
A(z)	1	-0.5	-0.5

	c_0
C(z)	1

Table 1: The Naive Predictor

Using the naive model to do a one- and nine step prediction yields the results displayed in figure 7. The performance on each corresponding data set is displayed in table 2, where $\sigma^2_{t+k|t}$ denotes the residual variance of a k-step prediction. The results displayed in table 2 indicate that the naive model has some 1-step predictive power, however the model seems to have a hard time doing a 9-step prediction. The non-whiteness of the prediction residuals displayed in table 3, suggests that the model fails to accurately capture the dynamics of the Svedala temperature data. Table 3 displays the results of the Monti whiteness test where $\epsilon^2_{t+k|t}$ denotes the k-step prediction residual.

4 Modeling with ARMA

In order to estimate future temperature values one may try to model the Svedala temperature data as a stationary stochastic process. As shown in section 2.4 the

Figure 7: The naive one- and nine-step predictions of temperature in Svedala, on modeling and validation data sets.

data contains a trend and is thus obviously not stationary. It is therefore suitable to detrend the data before trying to model it, as was discussed in section 2.4. We subsequently try to model the detrended data as an auto-regressive moving average (ARMA) process. To determine suitable model orders we examine the ACF and partial autocorrelation function (PACF) of the data shown in figure 8. The plots show ringing behavior with a strong periodicity of around 24. In order to remove the periodicity we de-season the detrended data by applying the $\nabla_s := (1 - z^{-s})$ operator to the data with a season of $s = 24$. The ACF and PACF of the detrended- de-seasoned data, displayed in figure 9, now look more akin to one of an ARMA process, however, there are still signs of periodicity at multiples of lags close to 24. In order to capture this dynamic we choose to incorporate a flexible de-seasoning component in the model that tries to capture the 24-hour day-night cycle in a less crude way. This is done by having the seasonal behavior be estimated in the same step as the remaining AR- and MA coefficients, i.e including terms such as a_{24} in the A-polynomial. Adhering to the *KISS*-rule, and noting the behavior at small lags of the ACF and PACF, we start off with an ARMA(1,1) model and examine the whiteness of the model residual. As this yields a non-white residual, according to the Monti test, we

Data Set	$\epsilon_{t+1\|t}$	$\epsilon_{t+9\|t}$
Modeling	$1545.52 \not< 36.42$	$2173.70 \not< 36.42$
Validation	$380.13 \not< 36.42$	$454.72 \not< 36.42$

Table 3: Results of Monti whiteness tests of prediction residuals using the Naive predictor. Red indicates non-white and green indicates a white prediction residual

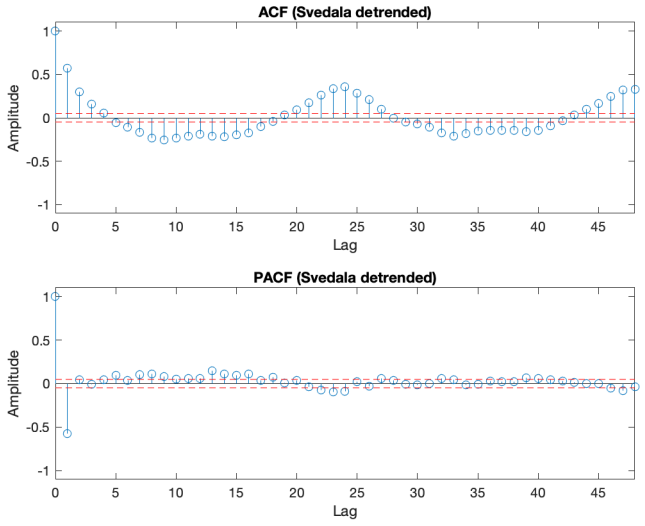

Figure 8: The autocorrelation and partial autocorrelation of the differentiated Svedala data. There are clear seasonal dependencies around time lag 24 hours.

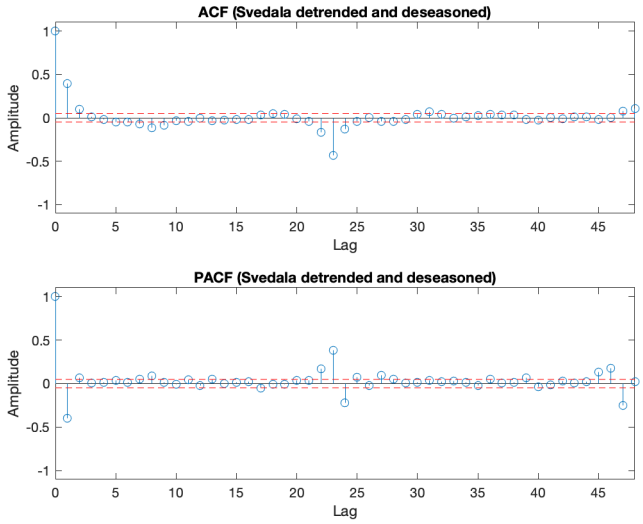

Figure 9: The autocorrelation and partial autocorrelation of the detrended and de-seasoned Svedala data. Some dependencies around 24 hours are still left.

add one AR coefficient yielding an ARMA(2,1) model. To capture the dynamics of the seasonality we also include the two coefficients a_{23} and a_{25} in the A-polynomial and one coefficient c_{24} in the C-polynomial. This yields a modeling residual that is white and has the ACF and PACF that are displayed in figure 10. The estimated coefficients of the model are displayed in table 4, this model will now be referred to as model A.

	a_1	a_2	a_{23}	a_{25}
$A(z)$	$-1.451(\pm 0.02)$	$0.5246(\pm 0.02)$	$-0.1295(\pm 0.01)$	$0.103(\pm 0.01)$

	c_1	c_{24}
$C(z)$	$-0.9661(\pm 0.01)$	$-0.02175(\pm 0.01)$

Table 4: Coefficients of model A: $\nabla_1 A(z) y_t = C(z) e_t$

In order to incorporate the detrending component we multiply the A-polynomial with the differentiating operator defined in section 2.4. Using model A to do a one- and nine-step prediction yields the predictions displayed in figure 11. The performance, on the modeling- and validation data sets, is displayed in table 5.

The results in table 5 indicate that model A outperforms the naive predictor. One may also note that the performance on the validation set is worse than on the modeling data. This is unsurprising as the parameters have been estimated using only data from the modeling set. The non-whiteness of the prediction

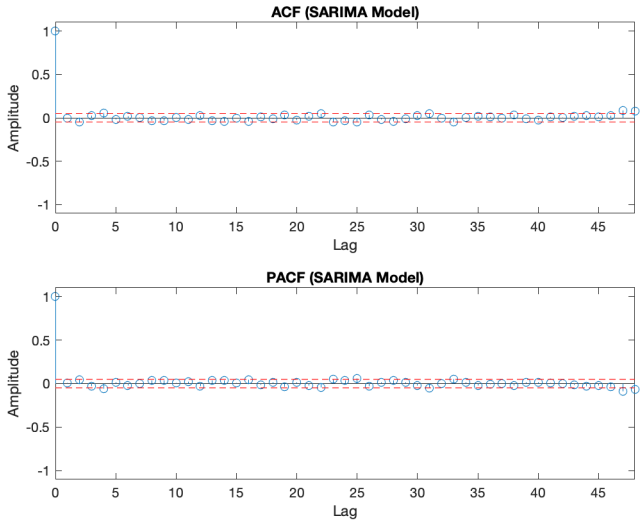

Figure 10: The autocorrelation and partial autocorrelation of the residuals from model A. Clearly, there is not much structure left in the residuals.

Figure 11: Model A one- and nine-step predictions of temperature in Svedala, on modeling and validation data sets.

Data Set	$\sigma^2_{t+1\|t}$	$\sigma^2_{t+9\|t}$	σ^2_y
Modeling	0.20722	3.6958	13.5078
Validation	0.12046	4.0546	7.5529

Table 5: Model A prediction residual- and data variance.

Data Set	$\epsilon_{t+1\|t}$	$\epsilon_{t+9\|t}$
Modeling	$40.08 \not< 36.42$	$2272.73 \not< 36.42$
Validation	$88.58 \not< 36.42$	$409.66 \not< 36.42$

Table 6: Results of Monti whiteness tests of prediction residuals using model A. Red indicates a failed whiteness test and green a successful whiteness test.

residuals displayed in table 6, suggests that the model fails to accurately capture the full dynamics of the Svedala temperature data. However, comparing table 6 with the whiteness of prediction residuals from the naive model, in table 3, one may note that the Monti test statistics are a lot closer to the whiteness limit. This suggests that model A captures more of the dynamics compared to the naive predictor, especially for the one-step prediction. One may also note that the Monti test statistic for the 9-step prediction, unfortunately, is still very far from the whiteness threshold.

5 Modeling temperature using external input

We try to enhance our predictions by utilising an external signal. We do this by fitting a Box-Jenkins model to the temperature, and use the nomenclature from the course book. The data we have at hand is SMHI:s predictions for the temperature in Sturup and Växjö, altered as described in section 2.1.3. We begin by choosing which data to use as input. To do this, we compare the cross-correlation between the two data sets and our objective: the temperature in Svedala. We see that in plot 12 the characteristics are similar, but the correlation with Sturup is stronger. This is what one could expect since Sturup is closer to Svedala geographically. Proceeding with the Sturup data as exogenous input, we begin the task of finding adequate model orders. This starts with trying to model the input as a process driven by white noise. We begin by differentiating the data to remove the trend and make it zero mean. We then examine the ACF and PACF to find suitable ARMA orders. Our goal is to find a model that describes the characteristics of the input signal as a process driven by white noise. Since there is a strong correlation for every three hours (naturally, since the values are constant for three consecutive hours) we find that the optimal (high order) model is one that covers several three-hour lags (3, 6, 9, 12, 15, 24). We trim away insignificant MA-components and are left with a whitening model for the exogenous input $\nabla_1 A_3(z)x_t = C_3(z)w_t$ where A_3 contains parameters $a_3, a_6, a_9, a_{12}, a_{15}, a_{24}$ and C_3 contains c_{12}, c_{24}. Although

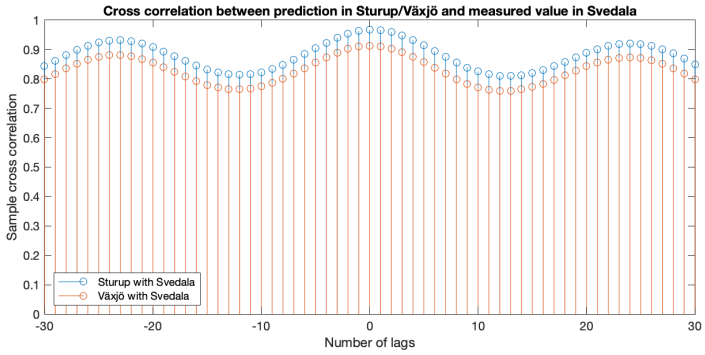

Figure 12: A comparison of the cross correlations between temperatures in SMHI:s prediction for Sturup and Växjö with the measured temperature in Svedala. Sturup has higher correlations for all time lags.

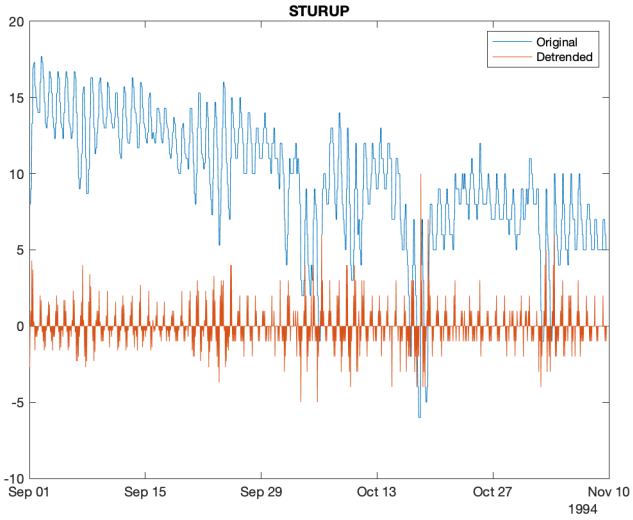

Figure 13: The temperature data for Sturup during the training period and the once differentiated data.

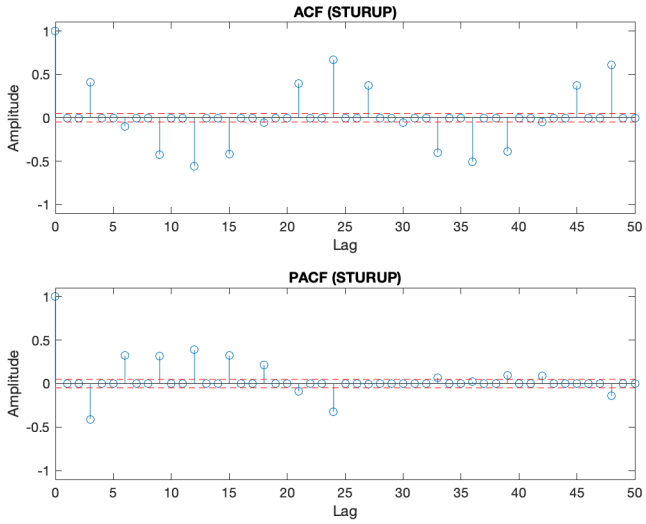

Figure 14: Auto Correlation Function and Partial Auto Correlation Function for the Sturup data. Clear dependencies for every three hours.

this polynomial is a tad large, it really covers all dependencies in the data which shows to be helpful in the next step. We filter the temperature in Svedala as well as the Sturup data through the inverse of this ARIMA-model and study the cross correlation between the residuals of these. This cross correlation is used to draw conclusions on what orders the $B-$ and A_2-polynomials should be of. Using the identification process outlined on page 133 in the course book, we draw the conclusion that the delay of the external signal should be $d = 0$ (expected from the zero lag cross correlation between the two data sets), the order of the $B-$polynomial should be $s = 0$ and the order of A_2-polynomial should be $r = 2$. Using these orders, we estimate a preliminary Box-Jenkins model without C_1 and A_1-polynomials (namely an Output Error-model if we allow ourselves to use control theory jargon) and study the residuals in terms of auto- and partial auto-correlation as well as the cross-correlation with the exogenous data. The cross correlation between input and error tells us if there is more information to extract from the exogenous input, which there clearly is. We therefore add a three hour time lag coefficient to A_2. In addition to this we use the ACF and PACF to estimate the orders for A_1 and C_1. After some testing of different model orders we find white residuals for a Box Jenkins-model $y_t = B(z)x_t/A_2(z) + C_1(z)e_t/A_1(z)$ with parameters found in table 7. The ACF and PACF for these residuals are found in figure 18, together with the cross correlation between the exogenous input and the residuals. Despite the last two parameters of the C_1-polynomial being statistically insignificant they proved to be vital for the nine step prediction on training and validation data.

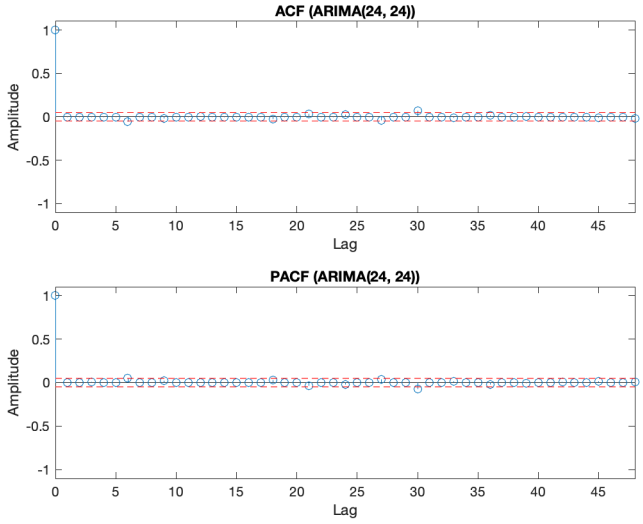

Figure 15: The residuals from modeling the exogenous input using an ARIMA model. It seems to capture most of the structure in the data.

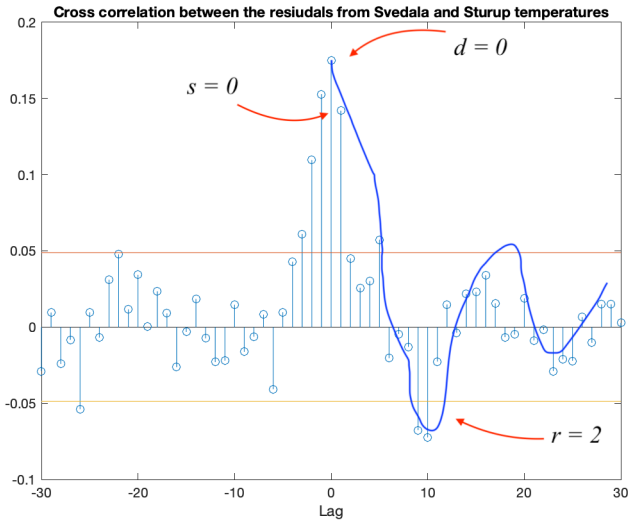

Figure 16: Cross correlation between residuals from Svedala and Sturup temperatures through the pre-whitening filter. From this we make a guesstimate of a preliminary BJ-model.

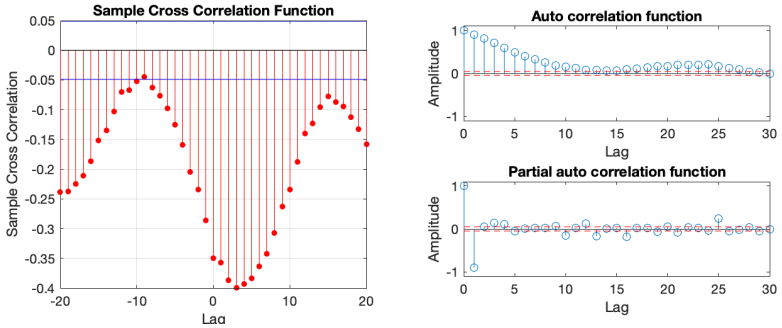

Figure 17: Left: the cross-correlation between the Sturup exogenous data and the residuals from our preliminary Box Jenkins-model. Right: the ACF and the PACF for the residuals from our preliminary Box Jenkins-model. We use these to further develop the model.

	a_1	a_2	a_3
$A_1(z)$	$-1.98(\pm 0.14)$	$1.307(\pm 0.18)$	$-0.2964(\pm 0.06)$
	a_{23}	a_{24}	a_{25}
	$-0.4455(\pm 0.05)$	$0.6131(\pm 0.05)$	$-0.1977(\pm 0.08)$
	c_1		
$C_1(z)$	$-0.5834(\pm 0.14)$		
	c_{23}	c_{24}	c_{25}
	$-0.3712(\pm 0.06)$	$0.063(\pm 0.08)$	$0.02277(\pm 0.04)$
	b_0		
$B(z)$	$0.1075(\pm 0.02)$		
	f_1	f_3	
$F(z)$	$0.9575(\pm 0.09)$	$0.09843(\pm 0.07)$	

Table 7: Model B: $y_t = B(z)x_t/A_2(z) + C_1(z)e_t/A_1(z)$ parameters with confidence bands

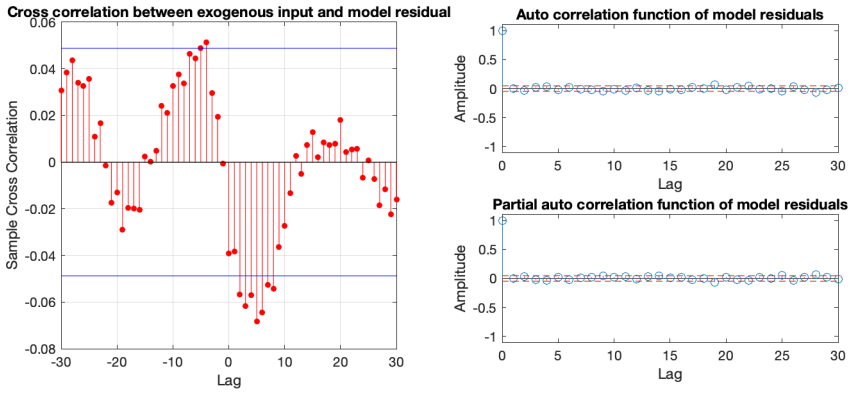

Figure 18: Left: The cross correlation between the Sturup data and our model residuals. The slight dependency around time lag five could be eliminated with a b_5-coefficient, but this showed to worsen the predictions. Right: ACF and PACF of BJ-model residuals. Quite white.

This could be explained by the extra 25 hour lag coefficient helps the model generalise better to when the seasonality changes. After all, the objective is to make a model with good predictive power. The same argument goes for the slight remaining dependency at time lag 5 in figure 18. We tried, and succeeded, with removing this dependency with a b_5-coefficient but this worsened the nine step prediction on both modelling and validation data.

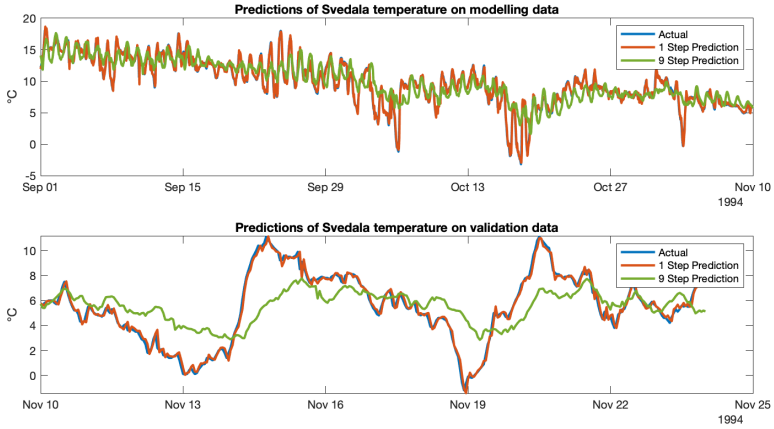

Figure 19: Model B one- and nine-step predictions of temperature in Svedala, on modeling and validation data sets.

Data Set	$\sigma^2_{t+1\|t}$	$\sigma^2_{t+9\|t}$	σ^2_y
Modeling	0.19962	3.5773	13.5078
Validation	0.10887	4.7321	7.5529

Table 8: BJ-model prediction residuals on modelling and validation data.

In order to make predictions with the Matlab `filter`-function, we need to translate the BJ-polynomials to ARMAX-form. We do this by identifying coefficients and dividing, ergo:

$$\begin{cases} y_t = \frac{B(z)}{A_2(z)}x_t + \frac{C_1(z)}{A_1(z)}e_t \\ K_A(z)y_t = K_B(z)x_t + K_C(z)e_t \end{cases} \implies \begin{cases} K_A = A_1A_2 \\ K_B = BA_1 \\ K_C = C_1A_2 \end{cases} \tag{1}$$

We make a one- and a nine step prediction on the training and validation data, both for the target temperature in Svedala and for the exogenous input data, the temperature predictions in Sturup. A plot of these predictions, together with the true temperature, are found in figure 19 and figure 20, respectively. The variance of the residuals on one and nine step predictions are presented in table 8. The quality of the prediction of the exogenous input is measured as the residual variance. These variances are presented in table 9. The variances of prediction residuals on the modeling set are smaller compared to the ones of model A, indicating the addition of the external signal provides predictive power during the modeling period. However, the performance of model B, on the validation set, is worse compared to model A, indicating that the Box-Jenkins model does not generalize as well. This might stem from the addition of an external signal creating a model that is overly complex for the underlying data.

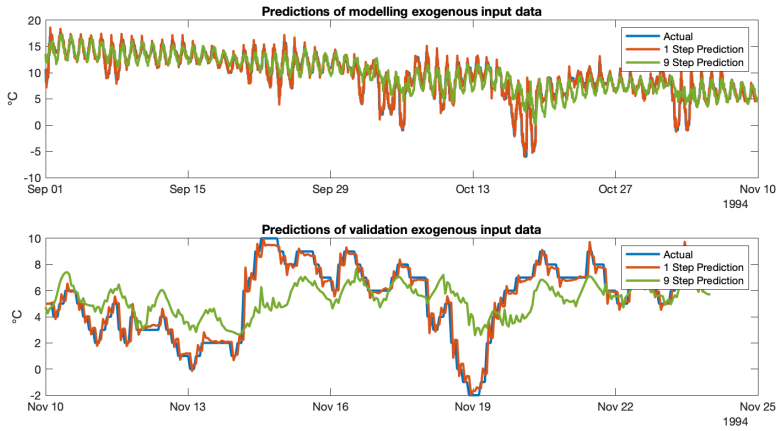

Figure 20: Model B one- and nine-step predictions of the exogenous SMHI prediction data for Sturup, on modeling and validation data sets.

Data Set	$\sigma^2_{t+1\|t}$	$\sigma^2_{t+9\|t}$	σ^2_y
Modeling	1.0654	4.658	13.5078
Validation	0.51458	5.4261	7.5529

Table 9: BJ-model prediction of exogenous input residuals on modelling and validation data.

I.e one could view model B as being over-fitted. Also, as one may note from table 9 the variance of the input prediction residual appears to be quite high, hinting at the quality of input prediction being poor. It might simply be the case that the input signal is difficult to predict and therefore only provides a small benefit in regard to the objective of predicting the Svedala temperature. On the other hand, the results of Monti whiteness tests on the prediction residuals, on both modeling- and validation data, displayed in table 10, are superior to those of model A. This is a surprising result as this would imply that model B captures more of the information present in the data. It could be the case that the simpler model A is a more accurate predictor, whilst having still having dependencies left in the prediction errors.

Data Set	$\epsilon_{t+1\|t}$	$\epsilon_{t+9\|t}$
Modeling	$34.23 < 36.42$	$2312.21 \not< 36.42$
Validation	$33.13 < 36.42$	$403.87 \not< 36.42$

Table 10: Results of Monti whiteness tests of prediction residuals using model B. Red indicates a failed whiteness test and green a successful whiteness test.

Data Set	$\sigma^2_{t+1\|t}$	$\sigma^2_{t+9\|t}$	σ^2_y
Modeling	0.19985	2.5626	13.5078
Validation	0.10701	2.1477	7.5529

Table 11: Model C prediction residual- and data variance.

Data Set	$\epsilon_{t+1\|t}$	$\epsilon_{t+9\|t}$
Modeling	$28.05 < 36.42$	$2125.63 \not< 36.42$
Validation	$26.70 < 36.42$	$387.93 \not< 36.42$

Table 12: Results of Monti whiteness tests of prediction residuals using model C. Red indicates a failed whiteness test and green a successful whiteness test.

6 Modeling using recursive estimation

In order to cope with the statistics of the temperature data changing over time one may construct a model that uses time-recursive parameter estimation. This can be done by using the parameters of model B, shown in table 7, as a baseline and constructing a Kalman filter in order for the parameters of the model to be re-estimated recursively instead of remaining fixed to the values estimated during the modeling period. As shown in equation (1) the BJ model B can be reformulated on the ARMAX form. Using this one may construct the state space representation shown in equation (2).

$$\begin{cases} \mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{e}_t \\ y_t = \mathbf{C}_{t|t-1}\mathbf{x}_{t|t-1} + w_t \\ C_{t|t-1} = \begin{bmatrix} -y_{t-1} & \dots & -y_{t-p} & e_{t-1} & \dots & e_{t-q} & u_{t-1} & \dots & u_{t-r} \end{bmatrix} \\ x_{t|t-1} = \begin{bmatrix} a_1 & \dots & a_p & c_1 & \dots & c_q & b_1 & \dots & b_r \end{bmatrix}^T \end{cases} \tag{2}$$

Where y_t denotes the temperature in Svedala, u_t the external signal, i.e the Sturup temperature predictions, a the coefficients of the K_A polynomial, c the coefficients of the K_C polynomial, b the coefficients of the K_B polynomial, $\mathbf{e}_t$ the process noise and w_t the measurement noise. After having formulated the state space representation of model B one may implement the Kalman filter update- and prediction scheme as defined in the book. This model will now be referred to as model C. By using model C one can make a one- and nine-step prediction on the modeling and validation data sets as shown in figure 21. The performance for each step size and data set is displayed in table 11. The result of Monti whiteness tests on all prediction residuals are displayed in table 12. Tables 11 and 12 show that the time recursive method of estimation outperforms the previous models. One may however note that the translation of the BJ model to an ARMAX form, through polynomial multiplication, results in a model with many coefficients. Figure 22 shows that many of these parameters tend towards zero during the parameter estimation process, despite being significant from the

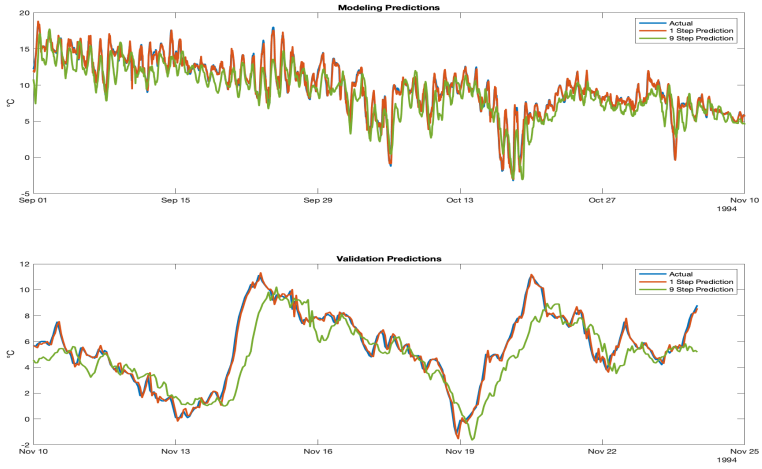

Figure 21: Svedala temperature, 1-step and 9-step prediction, using model C, on modeling- and validation sets.

Data Set	$\sigma^2_{t+1\|t}$	$\sigma^2_{t+9\|t}$	σ^2_y
Modeling	0.20791	3.1453	13.5078
Validation	0.10862	2.3771	7.5529

Table 13: Simplified model C prediction residual variance.

beginning in the BJ formulation outlined in table 7. One may also note that the initial state-space values are quite large, as a consequence of the reformulation from BJ to ARMAX. Figure 22 suggests that it might be beneficial to purge some of the unnecessary parameters as this would be in line with the *KISS* principle. Purging some of the parameters that appear close to zero in figure 22 results in a simpler model with far fewer parameters, however, at some cost of performance. The parameter estimates over time, for the simpler model, are displayed in figure 23 and the performance of the model in table 13. From the purged parameter estimates in figure 13 one may note that the coefficients of the B polynomial shrink during the estimation process. This hints at the external input not providing a significant amount of predictive power. Out of curiosity, one may therefore construct the Kalman filter for model A that uses no external inputs. This yields the parameter estimations displayed in figure 24 and performance displayed in table 14. The results in table 11, 13 and 14 show that the Kalman filter using the model B architecture, including all its parameters, yields the best results. Despite many of the coefficients being small, the external input appears to provide predictive power as the performance of the Kalman filter of model A is a lot worse. In all three versions of the recursive estimator, a smaller

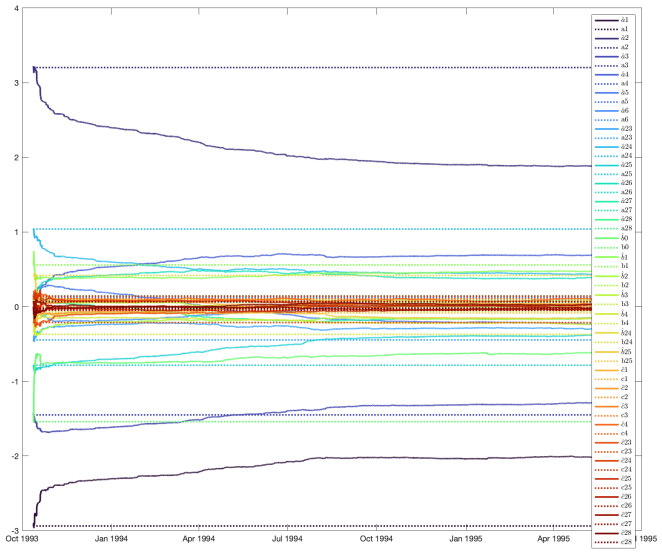

Figure 22: Kalman parameter estimation of model B. Original parameters are represented by a dotted line in the same color. Most parameters leave their inital values quite rapidly, but they are quite stable after some time.

Data Set	$\sigma^2_{t+1\|t}$	$\sigma^2_{t+9\|t}$	σ^2_y
Modeling	0.20846	4.0605	13.5078
Validation	0.11482	3.8719	7.5529

Table 14: Model A Kalman filter prediction residual- and data variance.

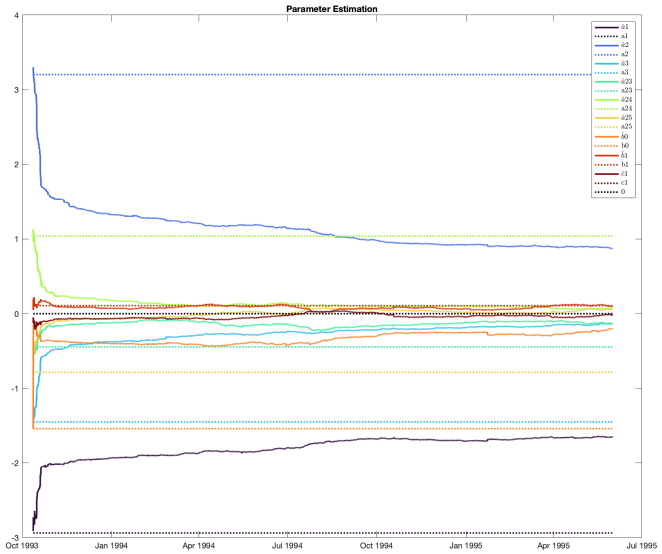

Figure 23: Kalman parameter estimation of the simplified version of model C. Original parameters are represented by a dotted line in the same color. Most parameters again leave their inital values quite rapidly.

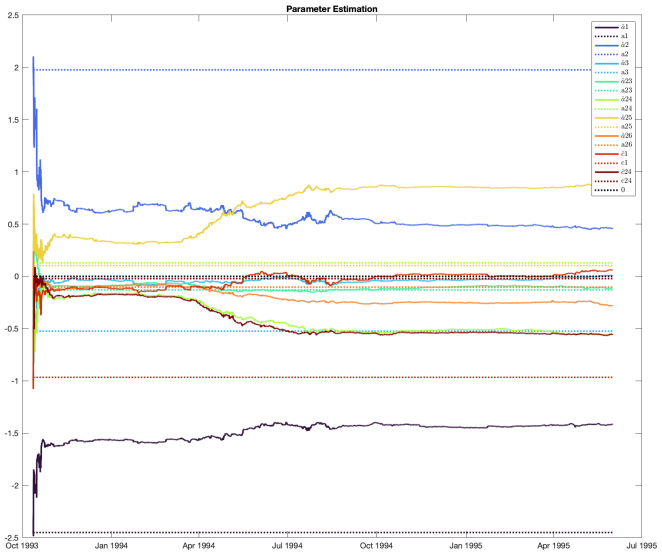

Figure 24: Kalman parameter estimation of model A. Original parameters are represented by a dotted line in the same color. Most parameters again leave their initial values quite rapidly, however these are slightly less stable than the previous models.

Data Set	$\sigma^2_{t+1\|t}$	$\sigma^2_{t+9\|t}$	σ^2_y
Modeling	6.2	6.5	13.5078
Validation	6.1	6.7	7.5529

Table 15: Model Prophet prediction residual- and data variance.

noise variance guess produced the best validation results. The results in tables 11, 13 and 14 were achieved using an equal noise variance of all states that was set to $\sigma_e^2 = 1 \times 10^{-6}$. The variance of the measurement noise proved to have an insignificant impact on the result and was set to the variance of previous 1-step prediction residuals, i.e $\sigma_w^2 \approx 0.22$. Overall a recursive parameter estimation scheme proves to be beneficial for predicting temperature.

7 Comparing with Prophet

Meta's forecasting API Prophet can provide a benchmark for the previously constructed methods. In order to provide a fair comparison we use a Prophet model that is completely out-of-the-box, i.e no parameter tuning, etc. The results of the Prophet forecaster, displayed in table 15 and visualized in figure 26, indicate that the automatic predictor does not fit the task of predicting the temperature in Svedala as it performs worse than the naive model. This might stem from the Prophet API being more well-suited for tasks that involve human affairs and not for weather events. An example of this is how the Prophet forecaster includes a component that uses the day-of-week as shown in figure 25. Temperature doesn't care about if it is Saturday or Monday, like a Facebook- or WhatsApp user would. As a consequence of the lacking performance, the Prophet model was discarded as a possible model candidate.

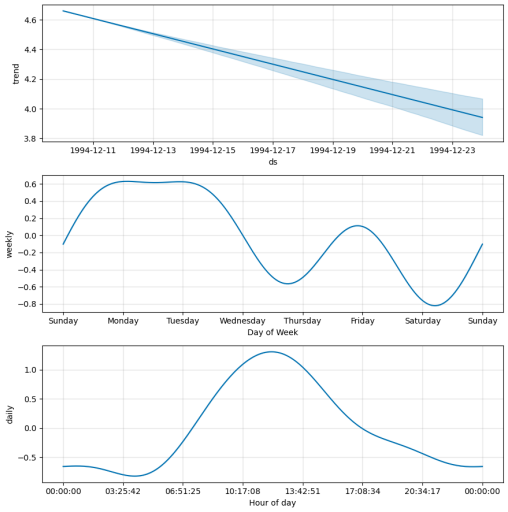

Figure 25: Components of the Prophet model. It has identified similar trend and daily variations of temperature as we have. However, it seems to have also estimated some kind of dependency between weekday and temperature.

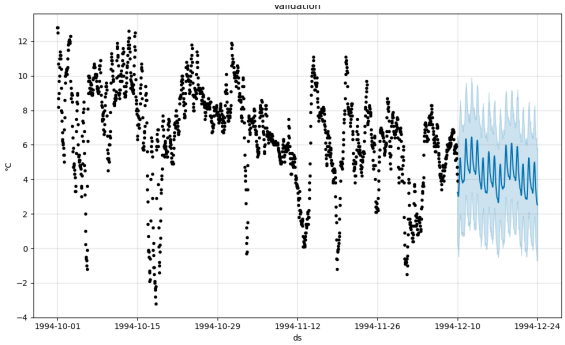

Figure 26: Prophet Validation Predictions

Model	$\sigma^2_{t+1\|t}$	$\sigma^2_{t+9\|t}$
Naive	3.1007	11.9268
A	0.12046	4.0546
B	0.10887	4.7321
C	0.10701	2.1477
Prophet	6.1	6.7
σ^2_y	7.5529	-

Table 16: Summary of Validation Results

Model	$\epsilon_{t+1\|t}$	$\epsilon_{t+9\|t}$
Naive	$380.13 \not< 36.42$	$454.72 \not< 36.42$
A	$88.58 \not< 36.42$	$409.66 \not< 36.42$
B	$33.13 < 36.42$	$403.87 \not< 36.42$
C	$26.70 < 36.42$	$387.93 \not< 36.42$

Table 17: Summary of results of Monti whiteness tests of prediction residuals on the validation set. Red indicates a failed whiteness test and green a successful whiteness test.

8 Results

We begin by giving a brief summary of the previously presented results on validation data. We then present the novel test results.

8.1 Summary of validation results

We compare the different models performance on validatin data in table 16. This suggest that the recursive method yields the best performance in terms of predictive power. This is strengthened by the fact that the one-step prediction residuals are white using model B and C as shown in table 17. As one may note from the ACFs of the prediction residuals in figure 27 there are still dependencies left in the one-step prediction residuals using model A and the naive model. Unfortunately, none of the nine-step validation prediction residuals are white.

8.2 Test Results

In order to perform an evaluation of all models we finally make predictions on the test data set. The one- and nine-step predictions on the test set, using all models, are displayed in figure 28. The prediction residual variances, in table 18, show a similar pattern to the validation performance in table 16. Once again, the recursive model outperforms all models using fixed parameter estimation. Unfortunately, none of the prediction residuals on the test set are white as indicated by the Monti test statistics, in table 19.

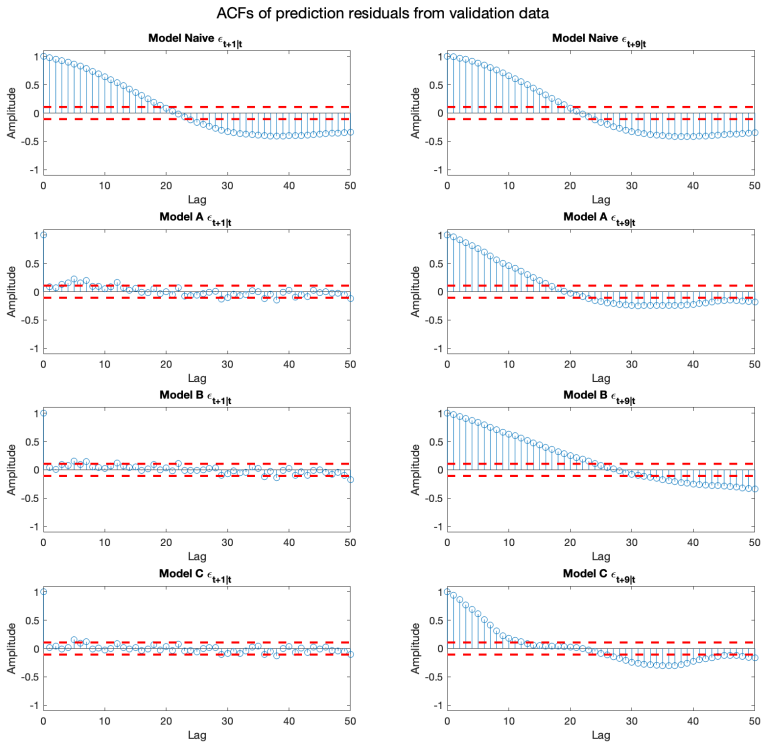

Figure 27: ACfs of one- and nine step prediction residuals on validation set using all models.

Model	$\sigma^2_{t+1\|t}$	$\sigma^2_{t+9\|t}$
Naive	1.8144	6.3152
A	0.23977	4.5599
B	0.23098	3.946
C	0.21926	3.0591
σ^2_y	5.8738	-

Table 18: Test Results

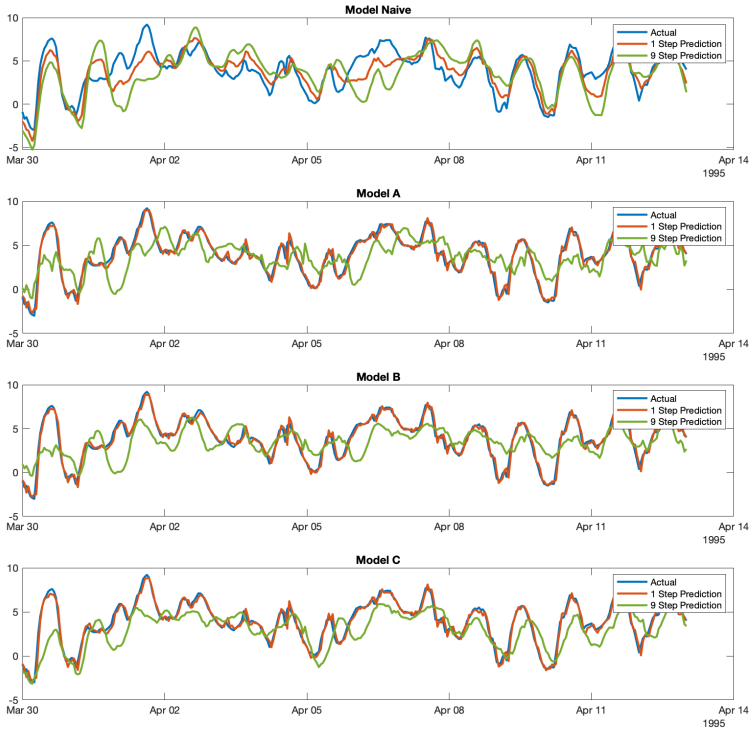

Figure 28: Test data 1- and 9-step predictions

Model	$\epsilon_{t+1\|t}$	$\epsilon_{t+9\|t}$
Naive	$341.41 < 36.42$	$469.75 < 36.42$
A	$41.89 < 36.42$	$496.08 < 36.42$
B	$39.93 < 36.42$	$490.03 < 36.42$
C	$41.24 < 36.42$	$497.02 < 36.42$

Table 19: Summary of results of Monti whiteness tests of prediction residuals on the test set. Red indicates a failed whiteness test and green a successful whiteness test.

9 Concluding Discussion

The validation- and test results in tables 16 and 18 show that a time recursive model is the best choice if one wants to predict the temperature in Svedala. As was mentioned in the introduction, this is an unsurprising result when taking the nature of the data in to account. Temperature naturally behaves differently depending on the season and length of the day-night cycle and it is therefore natural to assume that the statistics of such data change over time. A time recursive model captures this behavior and this is probably why model C has the largest predictive power. Another notable observation is that model B actually performs worse compared to model A on the validation data, indicating that the use of an external input decreases performance during the validation period. A possible explanation to this is found if we study the cross correlation between the temperatures in figure 12. Since the highest correlation is found for time lag zero, there is not much help using it as an external signal if it was not very successfully modelled. Studying table 9 leads to the conclusion that for us, this showed to be easier said than done. Perhaps this has to do with the complicated nature of the data, being forward filled with the previous value for two steps every three steps. Say that a change in temperature would often happened an hour before in Sturup than it did in Svedala. Then we would have an easier time . It is worth noting that it is the *prediction* of the Sturup temperature that has a zero lag correlation with Svedala, meaning that despite a three hour head start they still coincide. A plot of the correlation between unshifted Sturup temperature, which is the prediction for ten o'clock, presented at ten o'clock (but made at seven o'clock), and the Svedala measured temperature shows that the strongest correlation occur for a two hour time lag. From this it seems to be an easier task to do it the other way around, predicting Sturup using Svedala. Or maybe SMHI:s prediction of Sturup is just the naive prediction: the current temperature. At least, the model using exogenous input performs better on test data. A possible reason for this is that the input helps it generalise better to the seasonal change when we go from the autumn in training data to the spring in test data. From table 19 one may also note that none of the prediction residuals on the test set are white. This indicates that none of the models are able to capture the full dynamics of the data when making a prediction. One may however note that the Monti-test statistics for all estimated models are significantly closer to the whiteness threshold compared to the Naive model. At least, models A, B and C outperform the naive predictor, meaning that we have succeeded in extracting some information about the temperature, as well as (hopefully) this course.