

UNIVERSITY OF TARTU
FACULTY OF SCIENCE AND TECHNOLOGY
INSTITUTE OF MATHEMATICS AND STATISTICS

Adedokun Abdul-Baaki Dolapo
Modeling Complex Seasonalities
Actuarial and Financial Engineering
Master's Thesis (30 ECTS)

Supervisor: PhD Märt Möls

TARTU 2022

MODELING COMPLEX SEASONALITIES

Master thesis

Adedokun Abdul-Baaki Dolapo

Abstract

Time series data are sometimes affected by multiple cycles of different lengths. There can be a weekly cycle (better sales on Fridays), a monthly pattern (better sales at the beginning of the month as people have more cash after payday), and the effects of calendar seasonality (more tourists during summer, so better sales) might be present also. How to model multiple seasonality in one model? In this thesis, one could compare, for example, TBATS models (which allow multiple seasonalities) to alternative approaches

CERCS research specialisation: P160 Statistics, operations research, programming, financial and actuarial mathematics.

KeyWords: Time Series Forecasting, Complex Seasonalities, BATS, TBATS, Time series decomposition, Exponential Smoothing.

KEERUKATE SESCOONSETE MÕJUDE MODELEERIMINE

Magistritöö

Adedokun Abdul-Baaki Dolapo

Lühikokkuvõte

Aegridade andmeid mõjutavad mõnikord mitu erineva pikkusega tsüklit - võib olla nädalane tsükel (parem müük reedeti), kuu muster (Parem müük

kuu alguses, kuna inimestel on pärast seda rohkem raha Palgapäev) ja aastaaegade mõju (suvel rohkem turiste) võib samuti olemas. Kuidas arvestada mitut tsüklit ühes mudelis? Selles lõputöös saab võrrelda näiteks TBATS-i mudeleid (mis võimaldavad mitut hooajalisust) alternatiivsete lähenemisviisidega.

CERCS teaduseriala: P160 Statistika, operatsioonianalüüs, programmeerimine, finants- ja kindlustusmatemaatika.

Märksõnad: Aegridade prognoosimine, sessoonsed mudelid, BATS, TBATS, eksponentsiaalne silumine.

Contents

Introduction	4
1 Literature Review	5
2 Model Overview	7
2.1 Box-Cox transformation ARIMA errors Trend Seasonal components.	8
2.2 Trigonometric Seasonality, Box-Cox transformation ARIMA errors Trend Seasonal components.	10
2.3 Model Estimation and Selection	14
2.4 Facebook Prophet	18
2.5 ETS State Space Models	21
2.6 Evaluation Metrics	23
3 Experiment Framework and Forecasting Result	25
3.1 Datasets	25
3.2 Application of Models to Temperature Data	27
3.3 Application of Models to Electricity Price Data	33
Conclusions	38
References	39
Appendix 1.	42
Appendix 2.	45
Appendix 3.	51

Introduction

Seasonality is defined as the presence of variations that occur at regular intervals in a time series. Seasonality is the most significant component of time series, and it has become increasingly crucial as the frequency of time series produced in businesses has increased. Business time series data, for example, is collected and recorded in a considerably shorter time frame. The number of web page visitors, electricity consumption, and cash withdrawals from cash dispensers are all instances of business time series data that was previously collected quarterly or monthly but is now aggregated weekly, daily, or even hourly (Hyndman and Athanasopoulos [10]).

With the rise of high-frequency data and time series with closely spaced time intervals, a model capable of accurately modeling seasonal trends of time series is required. Forecasting of time series with complex seasonality and varying lengths of seasonality cycles is a difficult issue. As a result, in time series forecasting, detecting and accommodating the seasonality effect is essential.

This research will present the capabilities of automatic forecasting with TBATS models that account for seasonality using specific time series, e.g., with hourly data collected over several years, and thus generates a medium-term forecast with the specificity of a short-term forecast. It is ultimately one of the very few comprehensive models that address the entire forecasting process (from outliers' solution through decomposition and forecast to quantification accuracy). For this analysis of time series with complex seasonality, we used hourly data on the weather temperature in Estonia for six years, which made it possible to observe and use three seasonalities.

1 Literature Review

The choice of an appropriate model is one of the factors in determining correct forecasts for a given phenomenon. Given the number of various models available, our choice of the TBATS model may appear unusual. (Armstrong [1]), Box, Jenkins, and Ljung [2], Brockwell and Davis [3]).

The most widely used general forecasting techniques, such as Exponential Smoothing State Space Models (ETS, Hyndman et al. [9]), and Auto-Regressive Integrated Moving Average (ARIMA, Box, Jenkins, and Ljung [2]) introduced in 1970, are unable to cope with such a long seasonal cycle and thus have limited applicability in multiple seasonalities forecasting (Hyndman and Athanasopoulos [10]). As long as the ARIMA and exponential smoothing methods are not utilized for very complex time series, they are perfectly acceptable.

However, forecasts of more complicated time series require more advanced models that employ Bayesian procedures (Cottet and Smith [4]), Gaussian processes (M. Blum and Riedmiller [14]), and a variety of other methods (Zhou, Ang, and Poh [26]) depending on the data.

Multi-seasonality is one such feature. Most of these models are only suitable for time series with one or two seasonalities. The **TBATS**¹ model, which was introduced a few years ago, is the solution to this problem (De-Livera, Hyndman, and Synder [13])

There are four main approaches to dealing with seasonality in daily data in the literature. Seasonality can be treated in one of four ways:

1. Ignoring it and building a non-seasonal model.
2. By using seasonal lags.

¹TBATS: Trigonometric Seasonality, Box-Cox transformation ARIMA errors Trend Seasonal components.

3. By using seasonal indicator variables like Fourier terms or seasonal dummy variables.
4. By deseasonalizing the series before forecasting.

Miswan, Said, and Hussin [15] proposed an ARIMA model for forecasting Malaysian electricity load demand and compared it to a double exponential smoothing model. The ARIMA model was found to be more suitable than the other time series models. Nilgun and Betul [16] recently examined the TBATS and ARIMA models for predicting airline passenger volumes at airports. Although the TBATS model was complex, the results revealed that it performed well.

Power system price forecasting techniques are relatively recent procedures. Demand was already predicted in centralized marketplaces [6]. Competition has generated a new field of research, and various price forecasting algorithms are now in use.

Early publications by Skantze, Ilic, and Chapman [18] used jump-diffusion/mean reversion models to model electricity prices. T. Nakashima and Niimura [20] implemented fuzzy regression models that relate prices and demands to the Californian market. While many of the models discussed above deal with daily data, there's also been an increase increasing attention to modeling hourly electricity prices. Ramsay and Wang [17] employ neural networks to forecast prices in the England-Wales pool, as well as in California by Gao et al. [5] and the Victorian market by Szkuta, Sanabria, and Dillon [19].

2 Model Overview

Using a model to predict future values based on previously observed values is referred to as time series forecasting. Many scientists are familiar with time series forecasting but often struggle with particular data types. One of these data types is multi-seasonal data, whereby seasonal variations are frequent (e.g., hourly data may contain a daily pattern, weekly pattern, and an annual pattern).

Holt-Winters (HW) exponential smoothing algorithms are commonly used to forecast time series with single (additive or multiple) seasonal patterns, producing excellent results. On the other hand, these models are unable to handle complex seasonal time series with multiple seasonality, non-integer seasonality, and dual-calender effects. Taylor [21] added a second seasonal component to the single seasonal Holt-Winters model to handle time series with two seasonal patterns. This necessitates the estimation of many values for the initial seasonal components, especially when the seasonal patterns' frequencies are high, which may result in over-parameterization.

Simple seasonal trends, such as quarterly and monthly data, were evaluated. On the other hand, higher frequency time series frequently show more complex seasonal patterns. Daily data, for example, may have weekly and annual patterns. There are three types of seasonality in hourly data: a daily pattern, a weekly pattern, and a yearly seasonal pattern. Even weekly data might be difficult to estimate since it often follows a yearly pattern with an average seasonal period of $365.25/7 \approx 52.179$.

With high-frequency data recording, numerous seasonal patterns are becoming more frequent. The volume of calls at call centers, daily hospital admissions, requests for cash at ATMs, use of power and water, and access to computer websites are all instances of different seasonal patterns that might occur.

Furthermore, if the seasonal period is extended, the resulting model will likely have more parameters. When one seasonal period is a multiple of the other, this becomes

easier. According to the exponential smoothing model, the white noise process is not serially correlated. In practice, this assumption is not always correct because it might sometimes behave like an AR(1) process.

Hyndman et al. [9] presented ETS model improvements to accommodate a range of seasonal trends and tackle the correlated error problem. To prevent nonlinear difficulties, these authors limited the models to homoskedastic models, and when there is a certain type of non-linearity, the Box-Cox transformation (Box and Cox, 1964) is implemented.

2.1 Box-Cox transformation ARIMA errors Trend Seasonal components.

The **BATS**² model is a Box-Cox transformation that uses an exponential smoothing method and an ARIMA model to determine residuals. With minimal complexity and redundancy, the BATS model estimates seasonal time-series data. BATS is an acronym for Model Key Attributes for more complexity and redundancy: B: Box-Cox Transformation A: ARIMA Errors T: Trend S: Seasonal Components.

Basic Holt-Winters additive and multiplicative techniques are among the most commonly used seasonal models in the innovative state-space framework. Taylor [21] modifies the Holt-Winters linear form to include a second seasonal component. Box-Cox transformed observations with the parameter ω are represented by the notation $y_t^{(\omega)}$, where y_t represents the observation at time t . To include T seasonal patterns, we can expand exponential smoothing models as follows:

$$y_t = l_{t-1} + b_{t-1} + s_t^{(1)} + s_t^{(2)} + d_t \quad (1)$$

²BATS Box-Cox transformation ARIMA errors Trend Seasonal components.

$$y_t^{(\omega)} = \begin{cases} \frac{y_t^\omega - 1}{(\omega)}; & \omega \neq 0 \\ \log y_t; & \omega = 0 \end{cases} \quad (2)$$

$$y_t^{(\omega)} = l_{t-1} + \phi b_{t-1} + \sum_{i=1}^T s_{t-m_i}^{(i)} + d_t \quad (3)$$

$$l_t^{(w)} = l_{t-1} + \phi b_{t-1} + \alpha d_t \quad (4)$$

$$b_t^{(\omega)} = \phi b_{t-1} + \beta d_t \quad (5)$$

$$s_t^{(i)} = s_{t-m_i}^{(i)} + \gamma_i d_t \quad (6)$$

$$d_t = \sum_{i=1}^p \varphi_i d_{t-i} + \sum_{i=1}^q \theta_i \varepsilon_{t-i} + \varepsilon_t \quad (7)$$

where m_1, \dots, m_T denote the seasonal periods, l_t is the local level in period t , b_t is the trend with damping in period t , $s_t^{(i)}$ represents the i th seasonal component at time t , d_t denotes the residual $ARMA(p, q)$ process and ε_t is a Gaussian white noise process with zero mean and constant variance σ . The smoothing parameters are given by α , β and seasonal smoothing γ_i for $i = 1, \dots, T$.

Because the Box-cox transform can handle non-linear data and keep the variance more or less constant, the BATS model outperforms the simple space model. The auto-correlation problem can be solved using the residual ARMA model. They can get the point prediction, but they can also get an interval forecast. Explanatory variables cannot be added to the BATS model, which is one of its limitations. Multi-season periods must be nested, and the parameter for high-frequency seasonality will be quite large.

2.2 Trigonometric Seasonality, Box-Cox transformation ARIMA errors Trend Seasonal components.

De-Livera, Hyndman, and Synder [13] introduced an alternate method that fully automates the usage of fourier terms in conjunction with an exponential smoothing state space model and the Box-Cox transformation. There will be instances when it produces poor outcomes, like with any automated modeling framework, but it can be a beneficial method in specific circumstances.

A **TBATS**³ model differs from dynamic harmonic regression because seasonality can change slowly over time in the TBATS model, while harmonic regression conditions force seasonal patterns to repeat periodically without change. However, a disadvantage of TBATS models is that they can be slow to estimate, especially with long time series.

TBATS is a forecasting method for modeling time series data. The main purpose is to use exponential smoothing to forecast time series with complex seasonal patterns. TBATS is a time series model that helps you process data with multiple seasonal patterns, that is, data that changes over time.

TBATS is superior to BATS because it can handle complex and multiple seasonality time series data. The TBATS model is an extension of the BATS model, which is similar to the BATS model except that it does not include trigonometric regressors. TBATS is an acronym for key model function. T: Trigonometric seasonality B: Boxcox transformation A: ARIMA error T: Trend S: Seasonal component. The forecast package's *tbats()* command can be used to fit the TBATS model.

The TBATS $(\omega, \phi, p, q, m_1, k_1, \dots, m_T, k_T)$ model has its roots in the exponential smoothing methods and the parameters are as follows: ω is the Box-Cox parameter, ϕ is the damping parameter, p and q are the ARMA parameters, m_1, \dots, m_T is the seasonal periods, and k_i is the number of harmonics necessary for the i th seasonal

³TBATS: Trigonometric Seasonality, Box-Cox transformation ARIMA errors Trend Seasonal components.

component. The Box-Cox transformation $y_t^{(\omega)}$ with parameter ω can be expressed by the following equation:

Model:

$$y_t^{(\omega)} = \begin{cases} \frac{y_t^\omega - 1}{(\omega)}; & \omega \neq 0 \\ \log y_t; & \omega = 0 \end{cases} \quad (8)$$

$$y_t^{(\omega)} = l_{t-1} + \phi b_{t-1} + \sum_{i=1}^T s_{t-m_i}^{(i)} + d_t \quad (9)$$

$$l_t = l_{t-1} + \phi b_{t-1} + \alpha d_t \quad (10)$$

$$b_t^{(\omega)} = (1 - \phi)b + \phi b_{t-1} + \beta d_t \quad (11)$$

$$s_t^{(i)} = s_{t-m_i}^{(i)} + \gamma_i d_t \quad (12)$$

$$d_t = \sum_{i=1}^p \varphi_i d_{t-i} + \sum_{i=1}^q \theta_i \varepsilon_{t-i} + \varepsilon_t \quad (13)$$

where $y_t^{(\omega)}$ is the time series at moment t (Box-Cox transformed), $s_t^{(i)}$ is the i th seasonal component at time t , l_t is the local level in period t , b_t is the trend with damping in period t , d_t denotes an $ARMA(p, q)$ process for residuals, α and β is the smoothing parameters, ϕ is the trend damping, φ_i and θ_i are $ARMA(p, q)$ coefficients and ε is the Gaussian white noise process with zero mean and constant variance σ .

The trigonometric representation of seasonal components based on Fourier series (West and Harrison [25], Harvey [7]) has been introduced for greater flexibility:

Fourier Seasonal terms:

$$s_t^{(i)} = \sum_{j=1}^{(k_j)} s_{j,t}^{(i)} \quad (14)$$

$$s_{j,t}^{(i)} = s_{j,t-1}^{(i)} \cos \lambda_j^{(i)} + s_{j,t-i}^{*(i)} \sin \lambda_j^{(i)} + \gamma_1^{(i)} d_t \quad (15)$$

$$s_{j,t}^{*(i)} = s_{j,t-1}^{(i)} \sin \lambda_j^{(i)} + s_{j,t-i}^{*(i)} \cos \lambda_j^{(i)} + \gamma_2^{(i)} d_t \quad (16)$$

$$\lambda_j^{(i)} = \frac{2\pi j}{m_i} \quad (17)$$

Model parameters T represents the amount of seasonalities, m_i is the length of i th seasonal period, k_i is the amount of harmonics of i th seasonal period, $s_{j,t}^{*(i)}$ is the stochastic level of the i th seasonal component, and $s_{j,t}^{(i)}$ is the stochastic growth of the i th seasonal component needed to describe seasonal changes over time, and $\gamma_1^{(i)}, \gamma_2^{(i)}$ is the seasonal smoothing pair for each time period for $i = 1, \dots, T$.

Each seasonality is modeled with a trigonometric representation based on the fourier series. $2(k_1 + k_2 + \dots + k_T)$ initial values is estimated in the TBATS model. Another feature is the ability to model the seasonal effects of the non-integer lengths. For example, in the light of a series of daily observations, a leap year with a length of 365.25 can be modeled. TBATS can handle data with incorrect seasonal periods, non-overlapping periods, and high-frequency data. It can do multiple seasonalities without increasing too many parameters and the weaknesses of the TBATS model is that it cannot add explanatory variables.

The prediction accuracy of every forecasting technique should be evaluated. During prediction, accuracy measures check for error between obtained and actual values. The accuracy of forecast methods is usually measured using the Root Mean Squared Error (RMSE) and Median Absolute Error (MAE) metrics.

Seasonal trends can be found in many different time series. Some patterns, most

notably weekly series, have a non-integer period. For illustration, we will utilize temperature data from Tartu, Estonia. If we had examined data from other regions, we would have obtained similar results. For Tartu, Estonia, hourly temperature data were obtained from 1 January 2016 to 31 December 2021. The data is collected at the University of Tartu's Laboratory of Environmental Physics. The station is on the Institute of Physics building. This time series exhibit multiple seasonal patterns with a high frequency. Figure 1 shows a daily seasonal pattern with period 24 and a weekly seasonal pattern with period $24 * 7 = 168$ for Estonia daily temperatures. An annual seasonal trend is present in a longer version of this series. Daily hospital admissions, requests for cash at ATMs Machine, power and water usage, and access to computer websites are also examples of multiple seasonal patterns.

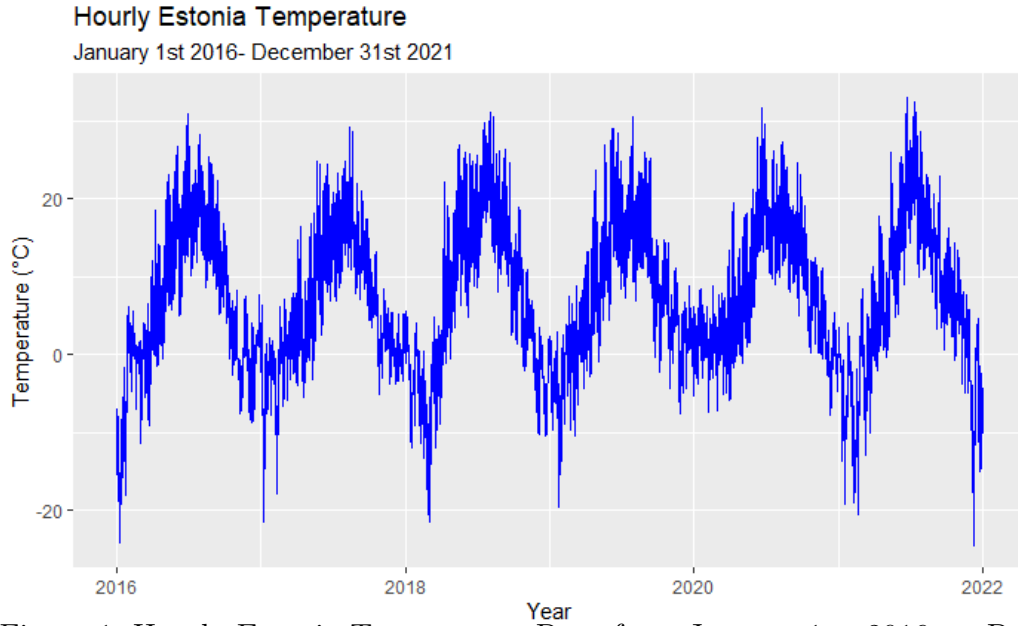


Figure 1: Hourly Estonia Temperature Data from January 1st, 2016, to December 31st, 2021

2.3 Model Estimation and Selection

In this section, estimation procedures for modeling complex seasonality with TBATS as demonstrated by De-Livera, Hyndman, and Synder [13], finding initial values for the procedures, and choosing the best model are discussed.

Maximum likelihood estimation

To compute the underlying model (9) likelihood function and obtain maximum likelihood estimates (MLEs). On the assumption that $\varepsilon_t \sim N(0, \sigma^2)$, the vector containing the Box-Cox parameters, and smoothing parameters and ARMA coefficients, is denoted as v as well as the initial states, x_0 , can be estimated from the observed data, $\mathbf{y} = (y_1, \dots, y_n)$ by maximizing the likelihood. This indicates that the transformed series density is $y_t^{(\omega)} \sim N(w'x_{t-1}, \sigma^2)$, implying that the data density is as follows:

$$p(y_t^{(\omega)} | x_0, v) = \prod_{t=1}^n p(\varepsilon_t) = \frac{1}{(2\pi\sigma^2)^{\frac{n}{2}}} \exp\left(\frac{-1}{2\sigma^2} \sum_{t=1}^n \varepsilon_t^2\right)$$

Therefore, using the Jacobian of the Box-Cox transformation, the density of the original series is

$$\begin{aligned} p(y_t^{(\omega)} | x_0, v, \sigma^2) &= p(y_t^{(\omega)} | x_0, v, \sigma^2) \left| \det \left(\frac{\partial y_t^{(\omega)}}{\partial y} \right) \right| = p(y_t^{(\omega)} | x_0, v, \sigma^2) \prod_{t=1}^n y_t^{\omega-1} \\ &= \frac{1}{(2\pi\sigma^2)^{\frac{n}{2}}} \exp\left(\frac{-1}{2\sigma^2} \sum_{t=1}^n \varepsilon_t^2\right) \prod_{t=1}^n y_t^{\omega-1}. \end{aligned}$$

By setting the partial derivative of \mathcal{L} with respect to σ^2 to zero, the maximum

likelihood estimate of the error variance is obtained.

$$\hat{\sigma}^2 = n^{-1} \sum_{t=1}^n \varepsilon_t^2. \quad (18)$$

As a result, the log likelihood is equal to

$$\mathcal{L}(x_0, v, \sigma^2) = \frac{-n}{2} \log(2\pi\sigma^2) - \frac{\sum_{t=1}^n \varepsilon_t^2}{2\sigma^2} + (\omega - 1) \sum_{t=1}^n \log y_t. \quad (19)$$

The result is obtained by substituting (18) into (19), multiplying by -2, and omitting constant terms.

$$\mathcal{L}^*(x_0, v) = n \log \left(\sum_{t=1}^n \varepsilon_t^2 \right) - 2(\omega - 1) \sum_{t=1}^n \log y_t. \quad (20)$$

To obtain maximum likelihood estimates, this quantity (20) can be minimized.

We'll need some initial x_0 and v estimates to begin the optimization. If the data requires a Box-Cox transformation, an initial value for the Box-Cox parameter ω must be approximated first. This could be accomplished by inspecting the data under various transformations or just starting with $\omega=0$. (We discovered that using this initial value resulted in better model selection and forecasting than using alternative initial values for ω .)

Let $m^* = \lfloor \max(m_1, m_2, \dots, m_T) \rfloor$. Then, for the first few seasons of transformed data, we construct a $2xm^*$ moving average. Denote this by $f_t(t = m^*/2 + 1, m^*/2 + 2, \dots)$ let $z_t = y_t^{(\omega)} - f_t$.

For TBATS models, the seasonal component is then approximated using

$$z_t \approx \sum_{t=1}^T \sum_{j=1}^{k_i} a_j^{(i)} \cos(\lambda_j^{(i)}, t) + b_j^{(i)} \sin(\lambda_j^{(i)}, t), \quad (21)$$

where $a_j^{(i)}$ and $b_j^{(i)}$ are estimated by regressing z_t against the trigonometric terms.

The i th seasonal component's initial seasonal states can then be set to $a_j^{(i)}$ and $b_j^{(i)}$. In order to derive $a_j^{(i)}$ and $b_j^{(i)}$ for BATS models, we repeat the same technique but substitute k_i with $m_i/2$ for even frequencies and $(m_i-1)/2$ for odd frequencies in equation (21). The initial seasonal state estimates are defined as $\hat{z}_t = \sum_{j=1}^{k_i} \left[\hat{a}_j^{(i)} \cos(\lambda_j^{(i)}, t) + \hat{b}_j^{(i)} \sin(\lambda_j^{(i)}, t) \right]$ using these values.

In the TBATS model, model selection is automated by minimizing the AIC.

$$AIC = \mathcal{L}^*(x_o, v) + 2K$$

where x_o is an estimate of the initial states vector and v is an estimate of the vector containing the Box-Cox transformation parameter, the dampening parameter, and the ARMA process coefficients. K is the total number of all the parameters in v plus the number of seed states in x_o .

Furthermore, rather than evaluating every possible combination, the number of harmonics is chosen using a method that involves first approximating the seasonally de-trended data with a regression, then gradually adding harmonics and testing their significance until a model with a minimal AIC emerges.

According to De-Livera, Hyndman, and Synder [13], a two-step procedure is used to determine the ARMA process's order (p, q) . First, a suitable model without an ARMA component is chosen, and then the Hyndman and Khandakar [8] algorithm is applied to the residuals of that model to estimate p and q . The selected first model is then fit again with the ARMA errors components included. However, the latter is only retained if the new augmented model has a lower AIC than the model without the ARMA component.

The TBATS model also has the advantage of decomposing a complex seasonal series into trend, seasonal, and irregular components. The Exponential smoothing state-space model with Box-Cox transformation, ARMA errors, Trend, and Sea-

sonal components(BATS) is another model variation that might be considered an extension of the ETS model to support more complex seasonal patterns.

De-Livera, Hyndman, and Synder [\[13\]](#) state that the BATS model has drawbacks, including the inability to manage non-integer seasonal periods and the potential for a large number of seed states in seasonal patterns with long seasonal periods.

2.4 Facebook Prophet

Facebook Prophet⁴ is a powerful time series forecasting package for Python and R. It takes care of the technicalities required to generate forecasts, such as model selection and feature engineering. It was created by Facebook's Core Data Science team [23] and released open-source in December 2016. It provides multiple seasonality modeling methods, ranging from the Auto-regressive Integrated Moving Average (ARIMA) models to exponential smoothing via Holt's Linear Exponential Smoothing, Holt-Winters. Growth $g(t)$, seasonality $s(t)$, holidays $h(t)$, and error e_t are the sum of three time functions plus an error term (Taylor and Letham [22]).

$$y(t) = g(t) + s(t) + h(t) + \varepsilon_t \quad (22)$$

The Growth Function:

The new concept introduced into Facebook Prophet is that the growth trend can exist at any point in the data or be changed at "change points," as Prophet refers to them. The growth function models the data's general trend. Change points are when the data shifts from one direction to another.

$$g(t) = (k + a(t)^T \delta)t + (m + a(t)\gamma) \quad (23)$$

k is the increase rate, δ is the rate arrangements, m is the offset parameter, and γ_j is set to $s_j \delta_j$ to provide a continuous function in equation (23). Change points can be detected automatically by Prophet by setting a sparse prior on δ . The Fourier series, on the other hand, is used to incorporate daily, weekly, and annual seasonalities. You can also customize the growth function and the amount of data taken into account in automatic change point detection by adjusting the power of the change points.

⁴<https://github.com/facebook/prophet>.

The Seasonality Function:

A Fourier Series as a function of time is the seasonality function. The sum of several successive sines and cosines is a simple way to think about it. Some coefficients are multiplied by each sine and cosine term. In the instance of Facebook Prophet, this sum can approximate practically any curve or the seasonality (cyclical trend) in our data. It appears as follows:

$$s(t) = \sum_{n=1}^N \left(a_n \cos \left(\frac{2\pi nt}{P} \right) + b_n \sin \left(\frac{2\pi nt}{P} \right) \right) \quad (24)$$

The Holiday/Event Function:

When a holiday or major event changes the forecast, the holiday function allows Facebook Prophet to adjust forecasting. It requires a list of dates (there are built-in dates of US holidays, or you can specify your dates). It can also explore a selection of days around dates (think of the period between Christmas and New Year's, holiday weekends, Thanksgiving link to Black Friday/Cyber Monday, and so on).

Facebook uses Prophet to forecast metrics, including page load times, server utilization, and future ad and product performance. It has also been used to forecast in various fields, like finance and climate research. It has been used to model the spread of infectious diseases, predict future stock price movements, forecasts the number of customers who will visit a website in the coming weeks and election results.

Prophet can be used with any time series data that has been stored in Hierarchical Data Format version 5 (HDF5) format. The data must have a date and time index, but other attributes can also index it. Multiple seasonalities are supported by the package, which means that the time series data can have multiple seasonal cycles, such as daily, weekly, quarterly, and yearly patterns in the same model by using a factor to identify each data point. It also supports multiple input variables in the same model and can handle time series with missing observations.

Prophet’s forecasting method uses machine learning to predict the future, which means that the model will adapt to new patterns. Prophet is trained with historical data and uses that to predict what might happen in the future. The algorithm has two steps: training and prediction. It is also easy to use and deploy, making it a good candidate for production forecasting. Prophet has some great documentation available on [Github](#), including installation instructions for [Python](#) and [R](#). Companies including Google, Airbnb, Lyft, New Relic, Stripe, Heroku, and Digital Ocean use Prophet.

The Prophet model has advantages and disadvantages over other forecasting methods: it is a state-of-the-art algorithm but requires many data to work well. Prophet’s main limitation is that it requires at least twice as much data to predict the number of time steps in our forecast. For instance, one needs 6 months of data to make a forecast for 3 months.

Prophet can handle non-stationary time series data and has several built-in models for ARIMA modeling (such as ARMA, ARIMAX, and ARIMA), exponential smoothing (such as Holt Winters), seasonal adjustment (such as X-12-ARIMA-SEATS and X-13-ARIMA-SEATS), and trend decomposition (such as HoltWinters Trend).

Prophet also has several features that include

1. The ability to handle multiple time series at once (both fitting models and calculating forecasts).
2. The ability to fit non-seasonal ARIMA models (and other types of models).
3. The ability to fit models with multiple seasonality (e.g., quarterly and monthly).
4. The ability to fit long-memory ARIMA (and other) models.
5. The ability to fit time series with a non-stationary mean, including the ability to handle heteroskedasticity or variance changes.

2.5 ETS State Space Models

The **ETS**⁵ model, which stands for Error-Trend-Seasonality, is a time series decomposition model (Hyndman and Athanasopoulos [10]). It divides the series into three parts: error, trend, and seasonality. When dealing with time-series data, it is a uni-variate forecasting model. Trend methods model, exponential smoothing, and ETS decomposition are some of the principles included in this model.

The ETS models are a class of time series models that have an underlying state space model that includes a level component, a trend component (T), a seasonal component (S), and an error term (E). The exponential smoothing techniques algorithms generate point forecasts. The ETS (Error Trend and Seasonality, or Exponential Smoothing) models can generate prediction (or forecast) intervals in addition to point forecasts. There are two models for each method: one with additive errors and the other with multiplicative errors. If the smoothing parameter values are the same, the model's point forecasts are similar. However, they will produce different prediction intervals.

Each method is labeled as **ETS**(\cdot, \cdot, \cdot) for (Error, Trend, Seasonal components). This label can also be thought of as Exponential Smoothing. Using the notation; Error: Additive ("A") or multiplicative ("M"), Trend: None ("N"), additive ("A"), multiplicative ("M"), or damped ("A_d" or "M_d"), Seasonality: None ("N"), additive ("A") or multiplicative ("M"), the possibilities for each component (or state) are: Error= {A, M}, Trend = {N, A, A_d} and Seasonal= {N, A, M}.

Simple Exponential Smoothing with Addictive Errors

Simple exponential smoothing is yet another name for the simplest of the ETS models. It is equivalent to the (A, N, N) model in ETS terms, which is a model with additive errors, no trend, and no seasonality. Holt's technique has the following state space formulation [10]:

⁵ETS: Error - Trend and Seasonality, or Exponential Smoothing

$$y_t = y_{t-i} + \varepsilon_t \quad (25)$$

$$l_t = l_{t-i} + \alpha \varepsilon_t \quad (26)$$

This state space formulation can be transformed into a new one, a forecast, and a smoothing equation (as with all ETS models):

$$\text{Forecast equation :} \quad \hat{y}_{t|t-1} = l_{t-1} \quad (27)$$

$$\text{Smoothing equation :} \quad l_t = \alpha y_{t-1} + (1 - \alpha)l_{t-1} \quad (28)$$

The "error correction" form is obtained by rearranging the smoothing equation for the level.

$$l_t = l_{t-1} + \alpha(y_{t-1} - l_{t-1}) \quad (29)$$

$$l_t = l_{t-1} + \alpha \varepsilon_t \quad (30)$$

$\hat{y}_{t|t-1}$ is the forecast/expectation. The forecast corresponds to the preceding level in the simple exponential smoothing model. The next level is calculated using the second equation (smoothing equation) as a weighted average of the previous level and the preceding observation. l_t is the level (or the smoothed value) of the series at time t and $\varepsilon_t = y_{t-1} - l_{t-1} = \hat{y}_t - \hat{y}_{t|t-1}$ denotes the residual at time t , $\varepsilon_t \sim NID(0, \sigma^2)$.

Hyndman and Athanasopoulos [10] derived the innovative state space model for each of the exponential smoothing methods in the same way. The equations for all of the models in the ETS framework are listed in the table below.

ADDITIVE ERROR MODELS

Trend	Seasonal		
	N	A	M
N	$y_t = \ell_{t-1} + \varepsilon_t$ $\ell_t = \ell_{t-1} + \alpha \varepsilon_t$	$y_t = \ell_{t-1} + s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + \alpha \varepsilon_t$ $s_t = s_{t-m} + \gamma \varepsilon_t$	$y_t = \ell_{t-1} s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + \alpha \varepsilon_t / s_{t-m}$ $s_t = s_{t-m} + \gamma \varepsilon_t / \ell_{t-1}$
A	$y_t = \ell_{t-1} + b_{t-1} + \varepsilon_t$ $\ell_t = \ell_{t-1} + b_{t-1} + \alpha \varepsilon_t$ $b_t = b_{t-1} + \beta \varepsilon_t$	$y_t = \ell_{t-1} + b_{t-1} + s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + b_{t-1} + \alpha \varepsilon_t$ $b_t = b_{t-1} + \beta \varepsilon_t$ $s_t = s_{t-m} + \gamma \varepsilon_t$	$y_t = (\ell_{t-1} + b_{t-1}) s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + b_{t-1} + \alpha \varepsilon_t / s_{t-m}$ $b_t = b_{t-1} + \beta \varepsilon_t / s_{t-m}$ $s_t = s_{t-m} + \gamma \varepsilon_t / (\ell_{t-1} + b_{t-1})$
A _d	$y_t = \ell_{t-1} + \phi b_{t-1} + \varepsilon_t$ $\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha \varepsilon_t$ $b_t = \phi b_{t-1} + \beta \varepsilon_t$	$y_t = \ell_{t-1} + \phi b_{t-1} + s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha \varepsilon_t$ $b_t = \phi b_{t-1} + \beta \varepsilon_t$ $s_t = s_{t-m} + \gamma \varepsilon_t$	$y_t = (\ell_{t-1} + \phi b_{t-1}) s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha \varepsilon_t / s_{t-m}$ $b_t = \phi b_{t-1} + \beta \varepsilon_t / s_{t-m}$ $s_t = s_{t-m} + \gamma \varepsilon_t / (\ell_{t-1} + \phi b_{t-1})$

MULTIPLICATIVE ERROR MODELS

Trend	Seasonal		
	N	A	M
N	$y_t = \ell_{t-1} (1 + \varepsilon_t)$ $\ell_t = \ell_{t-1} (1 + \alpha \varepsilon_t)$	$y_t = (\ell_{t-1} + s_{t-m}) (1 + \varepsilon_t)$ $\ell_t = \ell_{t-1} + \alpha (\ell_{t-1} + s_{t-m}) \varepsilon_t$ $s_t = s_{t-m} + \gamma (\ell_{t-1} + s_{t-m}) \varepsilon_t$	$y_t = \ell_{t-1} s_{t-m} (1 + \varepsilon_t)$ $\ell_t = \ell_{t-1} (1 + \alpha \varepsilon_t)$ $s_t = s_{t-m} (1 + \gamma \varepsilon_t)$
A	$y_t = (\ell_{t-1} + b_{t-1}) (1 + \varepsilon_t)$ $\ell_t = (\ell_{t-1} + b_{t-1}) (1 + \alpha \varepsilon_t)$ $b_t = b_{t-1} + \beta (\ell_{t-1} + b_{t-1}) \varepsilon_t$	$y_t = (\ell_{t-1} + b_{t-1} + s_{t-m}) (1 + \varepsilon_t)$ $\ell_t = \ell_{t-1} + b_{t-1} + \alpha (\ell_{t-1} + b_{t-1} + s_{t-m}) \varepsilon_t$ $b_t = b_{t-1} + \beta (\ell_{t-1} + b_{t-1} + s_{t-m}) \varepsilon_t$ $s_t = s_{t-m} + \gamma (\ell_{t-1} + b_{t-1} + s_{t-m}) \varepsilon_t$	$y_t = (\ell_{t-1} + b_{t-1}) s_{t-m} (1 + \varepsilon_t)$ $\ell_t = (\ell_{t-1} + b_{t-1}) (1 + \alpha \varepsilon_t)$ $b_t = b_{t-1} + \beta (\ell_{t-1} + b_{t-1}) \varepsilon_t$ $s_t = s_{t-m} (1 + \gamma \varepsilon_t)$
A _d	$y_t = (\ell_{t-1} + \phi b_{t-1}) (1 + \varepsilon_t)$ $\ell_t = (\ell_{t-1} + \phi b_{t-1}) (1 + \alpha \varepsilon_t)$ $b_t = \phi b_{t-1} + \beta (\ell_{t-1} + \phi b_{t-1}) \varepsilon_t$	$y_t = (\ell_{t-1} + \phi b_{t-1} + s_{t-m}) (1 + \varepsilon_t)$ $\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha (\ell_{t-1} + \phi b_{t-1} + s_{t-m}) \varepsilon_t$ $b_t = \phi b_{t-1} + \beta (\ell_{t-1} + \phi b_{t-1} + s_{t-m}) \varepsilon_t$ $s_t = s_{t-m} + \gamma (\ell_{t-1} + \phi b_{t-1} + s_{t-m}) \varepsilon_t$	$y_t = (\ell_{t-1} + \phi b_{t-1}) s_{t-m} (1 + \varepsilon_t)$ $\ell_t = (\ell_{t-1} + \phi b_{t-1}) (1 + \alpha \varepsilon_t)$ $b_t = \phi b_{t-1} + \beta (\ell_{t-1} + \phi b_{t-1}) \varepsilon_t$ $s_t = s_{t-m} (1 + \gamma \varepsilon_t)$

2.6 Evaluation Metrics

The models were compared using forecasting errors. When a model is tested on a N data samples, several metrics can be used to evaluate its performance (J.Hyndman and B.Koehler [12]), We evaluate the forecast accuracy of our models by means of the Root Mean Squared Error (RMSE), and the Median Absolute Error (MAE) defined as

$$\text{MAE} = \text{Median}(|e_t|)$$

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{t=1}^N e_t^2}$$

where:

e_t is the forecast error, $e_t = y_t - \hat{y}_t$,

y_t is the actual value,

\hat{y}_t is the forecast value.

Hyndman and Athanasopoulos [10] defined time series cross-validation as a statistical method of evaluating model performance using training and test sets. The method of cross-validating the time-series model that can be utilized is cross-validation on a rolling basis. Begin by training the model with a small subset of data, forecasting for later data points, and then evaluating the accuracy of the forecasted data points. The same forecasted data points are included in the next training dataset, and additional data points are forecasted.

3 Experiment Framework and Forecasting Result

3.1 Datasets

When choosing a dataset for study, there are various elements to consider, including availability, privacy, and data scale. Although there are few public datasets with enough data points for forecasting complex seasonal time series, we were able to locate the hourly weather [dataset](#), which has seasonality in its temperature degree pattern and no missing values. This dataset⁶, used in this study, covers weather temperature from January 2016 to the end of December 2021 as shown in figure 1. It contains a total of 52578 data points, and it has seasonal variations. It can be seen from Figure 2, the daily visualization of the weather temperature between June 1st 2017 to June 8th 2017 to reflect the weekly and daily seasonality in the data.

The data set is divided into an in-sample period comprising the first 439818 observations (1st January 2016 to 31st December 2021), which represent the initial training set, 13 iterations were carried out in total to train the model, and in each iteration, an additional 500 new observations are included in the new training set. An out-of-sample period composed of the 2160 observations will be used to assess the forecasting abilities of the different models.

⁶First Data Source : <https://meteo.physic.ut.ee/>

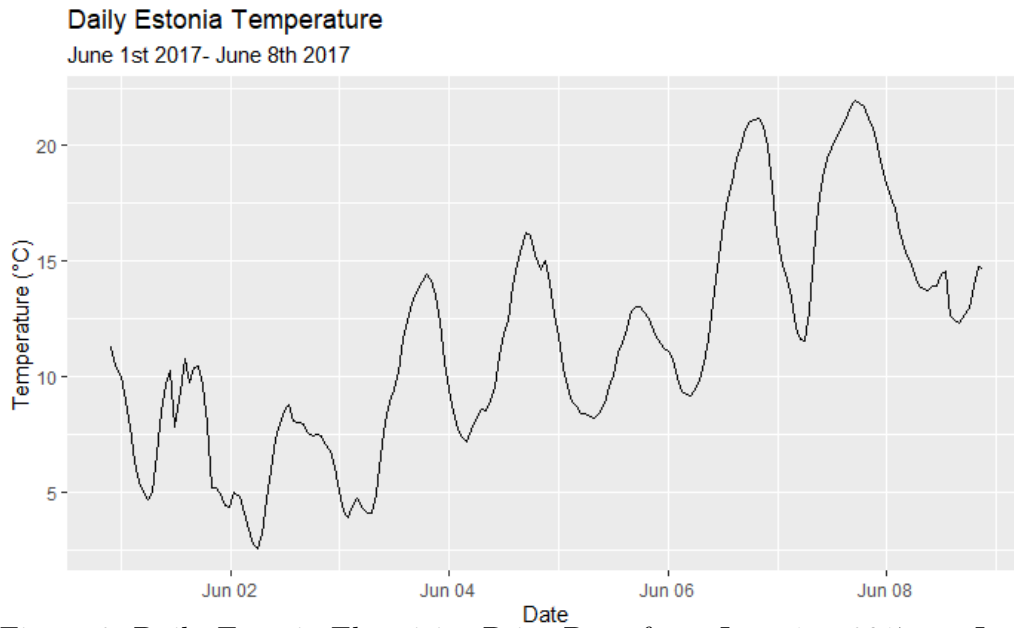


Figure 2: Daily Estonia Electricity Price Data from June 1st, 2017, to June 8th, 2017

The second [dataset](#) contains hourly electricity prices in four Baltic states: Latvia, Lithuania, Finland, and Estonia. Forecasting electricity prices can assist the government in developing power production strategy. In this scenario, we choose hourly electricity prices in Euro per megawatt (Euro/MWh) and they are plotted in Figure 3 for Estonia from January 1, 2016, to December 31, 2020. There are 43816 data points in this dataset⁷

There is a noticeable positive jump around the middle of the year 2020 and also at the beginning of December 2020, it can be assumed that this period corresponds to Christmas, and due to the holiday, there is a high demand creating a high electricity price. A similar but much smaller effect can be seen at the beginning of the year 2016, 2018 and also at the middle of the year 2019, which may be a similar effect caused by the high usage of electricity in the winter season. A rising trend beginning with October until February can also be seen, and this effect can be due to colder weather creating increased consumption due to heating requirements.

⁷Second Data Source : <https://dashboard.elering.ee/>

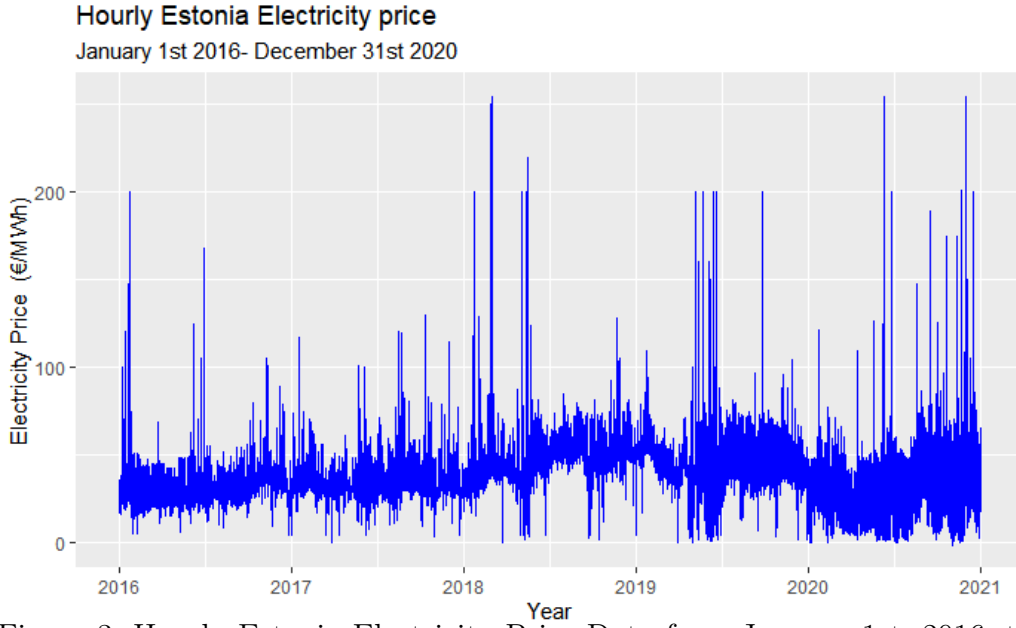


Figure 3: Hourly Estonia Electricity Price Data from January 1st, 2016, to December 31st, 2020

To choose an appropriate forecasting model, data must first be analyzed. After selecting a model, it can be implemented with training data and evaluated using test data. The raw data from Figure 1 and 3 were evaluated.

This is how the forecasting exercise is set up. The in-sample data is used to estimate the model of interest, and forecasts are computed for different horizon up to 2160 hours (3 months ahead forecast). For each model, two measures of forecast error are calculated: Root Mean Square Error (RMSE) and Median Absolute Error (MAE).

3.2 Application of Models to Temperature Data

Firstly, time series data with a single seasonality are analyzed using ETS models. The selected ETS model is **ETS (A, A_d , N)**, also known as the damped trend method with additive errors, which means additive error, damped additive trend, and no seasonality.

For hourly data, the model selection procedure led to the model: **TBATS (1,**

$\{0,0\}$, —, $\langle 24,4 \rangle$, $\langle 168,6 \rangle$, $\langle 8760,6 \rangle$). In this model, Box-Cox transformation equals 1 (no Box-Cox transformation), ARMA (0,0) errors are present, the no damping parameter, 4 Fourier pairs have a period of $m_1=24$ (daily), 6 Fourier pairs have a period of $m_2=168$ (weekly), and 6 Fourier pairs have a period of $m_3=8760$ (annual) which can be written as

$$\begin{aligned}
y_t &= l_{t-1} + b_{t-1} + s_{t-1}^{(1)} + s_{t-1}^{(2)} + s_{t-1}^{(3)} + \alpha d_t \\
b_t &= b_{t-1} + \beta d_t \\
s_t^{(1)} &= \sum_{j=1}^4 s_{j,t}^{(1)} \\
s_{j,t}^{(1)} &= s_{j,t-1} \cos\left(\frac{2\pi jt}{24}\right) + s_{j,t-1}^* \sin\left(\frac{2\pi jt}{24}\right) + \gamma_1^{(1)} d_t \\
s_{j,t}^{*(1)} &= -s_{j,t-1} \sin\left(\frac{2\pi jt}{24}\right) + s_{j,t-1}^* \cos\left(\frac{2\pi jt}{24}\right) + \gamma_2^{(1)} d_t \\
s_t^{(2)} &= \sum_{j=1}^6 s_{j,t}^{(2)} \\
s_{j,t}^{(2)} &= s_{j,t-1} \cos\left(\frac{2\pi jt}{168}\right) + s_{j,t-1}^* \sin\left(\frac{2\pi jt}{168}\right) + \gamma_1^{(2)} d_t \\
s_{j,t}^{*(2)} &= -s_{j,t-1} \sin\left(\frac{2\pi jt}{168}\right) + s_{j,t-1}^* \cos\left(\frac{2\pi jt}{168}\right) + \gamma_2^{(2)} d_t \\
s_t^{(3)} &= \sum_{j=1}^6 s_{j,t}^{(3)} \\
s_{j,t}^{(3)} &= s_{j,t-1} \cos\left(\frac{2\pi jt}{8760}\right) + s_{j,t-1}^* \sin\left(\frac{2\pi jt}{8760}\right) + \gamma_1^{(3)} d_t \\
s_{j,t}^{*(3)} &= -s_{j,t-1} \sin\left(\frac{2\pi jt}{8760}\right) + s_{j,t-1}^* \cos\left(\frac{2\pi jt}{8760}\right) + \gamma_2^{(3)} d_t
\end{aligned}$$

where d_t is an ARMA process (0,0), and $\alpha = 1.3034$, $\beta = 0$, $\gamma_1^{(1)} = 0.0166$, $\gamma_1^{(2)} = -5.9233$, $\gamma_1^{(3)} = 0.0157$, $\gamma_2^{(1)} = 0.0298$, $\gamma_2^{(2)} = -0.0004$, and $\gamma_2^{(3)} = -0.0027$ are smoothing parameters. The total number of initial seasonal values are $2(4+6+6)$

is equal to 32.

The decomposition result is shown in Figure 4 of the historical hourly electricity Price in Estonia using TBATS confirms the series is characterized by three seasonalities (daily, weekly, and yearly). Each panel represents the original data, the trend, the daily seasonality, the weekly seasonality, and the yearly seasonality, respectively. However, the frequency of the first seasonal component (season 1) about the length of the time series makes identification possible only after a graph for a shorter period (e.g., one week) has been visualized. Taken as an example, the first seasonal component for June 1st, 2017, to June 8th, 2017 (Figure 2) clearly shows a daily cycle.

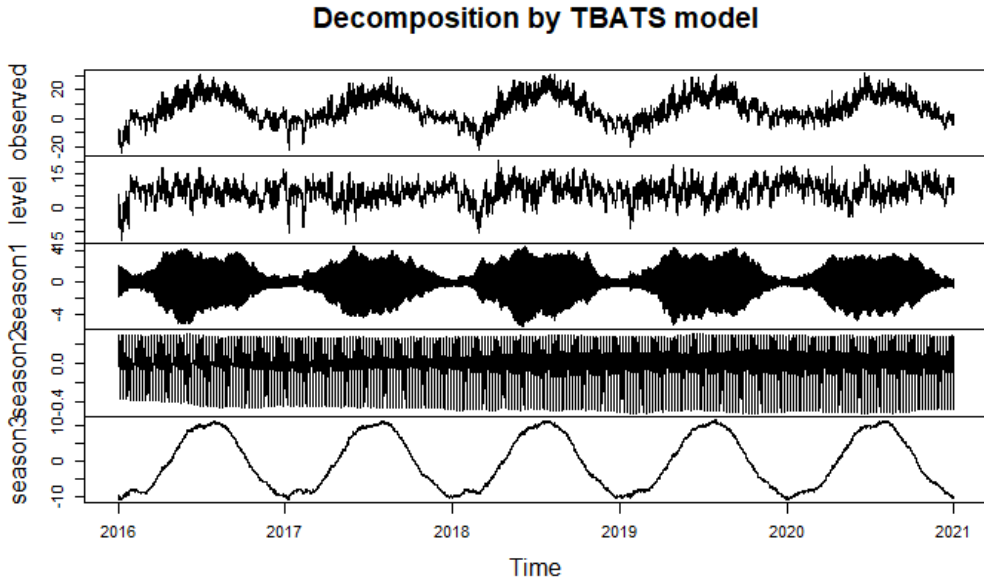


Figure 4: Decomposition of Historical Hourly Weather Temperature in Estonia using TBATS.

Subsequently, the Prophet model was initialized, with in-sample data of 4 years for training the model. The three figures from top to bottom are the growth trend, weekly trend, yearly trend and daily trend provided by Prophet additive models, respectively.

Figure 5 reveals an obvious peak in the annual trend of high temperature during

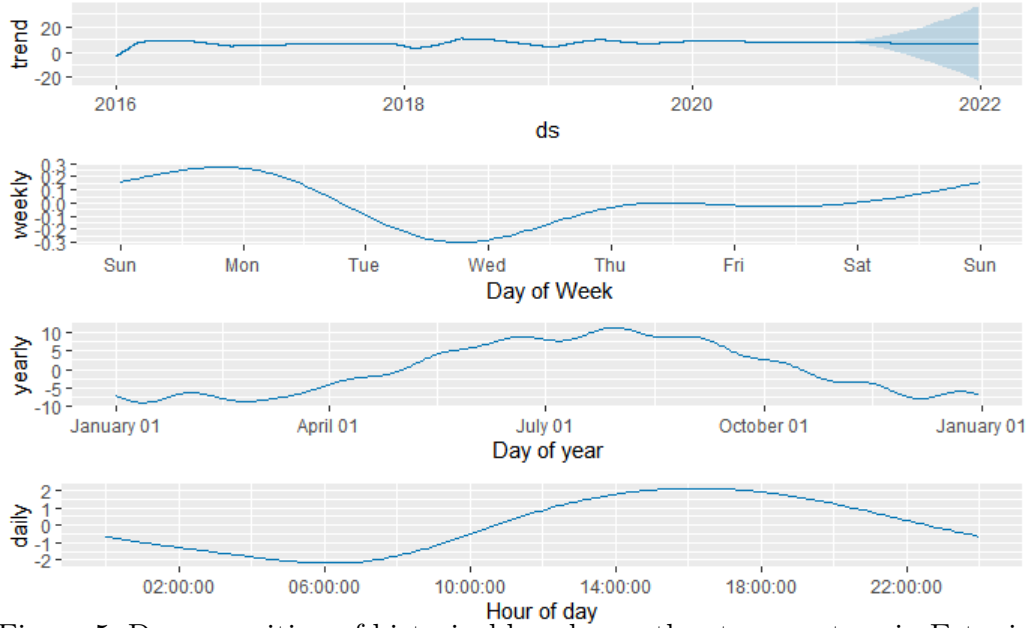


Figure 5: Decomposition of historical hourly weather temperature in Estonia using PROPHET.

summer from early June to late September, and the temperature was lowest from December to February, which is during the winter period of the year. In addition, according to the decomposition, the daily temperature is lowest during the night and increases through daylight. For this purpose, two error components, RMSE and MAE (explained in section 2.6) have been computed to check the out-of-sample performances for different forecasting horizons(hour): $h = 1, 6, 12, 48, 72, 720, 2160$. Figure 6 represents the comparison between RMSE and MAE error measures obtained by applying these selected models.

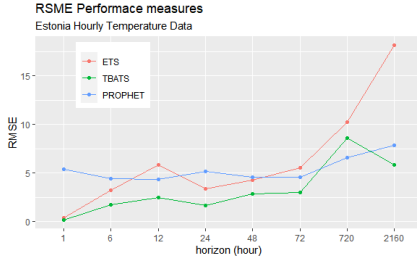
Table 1 and 2 show the model performance of predictions over different time horizons up to 2160 (3 months) ahead forecast using the RMSE and MAE evaluation metrics. From the results, it is evident that the TBATS is a better forecasting model that performs better than all other models in terms of low error, better fitting, and better forecasting than others while PROPHET also perform well on a long term forecasts (up to about 3 month ahead) than ETS which has higher error metrics.

Model	Forecasting horizon (hour)							
	1	6	12	24	48	72	720	2160
ETS (A, A_d, N)	0.39	3.22	5.83	3.38	4.25	5.53	10.23	18.12
TBATS	0.17	1.73	2.51	1.63	2.87	2.99	8.57	5.84
PROPHET	5.36	4.43	4.35	5.19	4.55	4.57	6.55	7.86

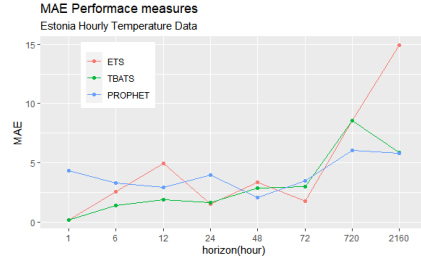
Table 1: **RMSE** Performance Matrix of Selected Models to Historical Estonia Temperature Data for Different Horizons

Model	Forecasting horizon (hour)							
	1	6	12	24	48	72	720	2160
ETS (A, A_d, N)	0.18	2.55	4.96	1.54	3.37	1.73	8.56	14.90
TBATS	0.17	1.37	1.90	1.63	2.87	3.00	8.57	5.85
PROPHET	4.31	3.30	2.91	3.94	2.06	3.45	6.05	5.78

Table 2: **MAE** Performance Matrix of Selected Models to Historical Estonia Temperature Data for Different Horizons



(a) RMSE across different forecast horizons comparing out-of-sample performances



(b) MAE across different forecast horizons comparing out-of-sample performances

Figure 6: Forecasting Accuracy for Temperature from 1 to 2160 hours (i.e. 3 months)

Figure 7 and 8 illustrates the 3 months ahead forecast comparison based on the out-of-sample data for our selected models. The TBATS model: **TBATS (1, {0,0}, —, <24,4>, <168,6>, <8760,6>)** retains the trend of the time series which indicates that the TBATS model is best-suited to the data. The forecasts retained all three seasonalities, making them more realistic for shorter and longer periods.

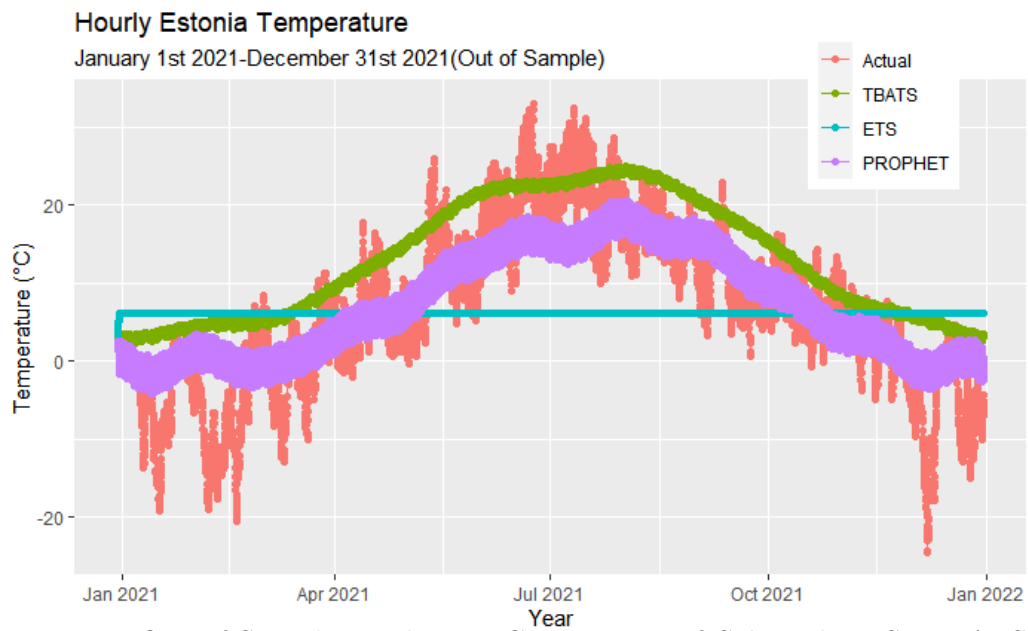


Figure 7: Out-of-Sample Prediction Comparison of Selected ETS, TBATS, and PROPHET Model.

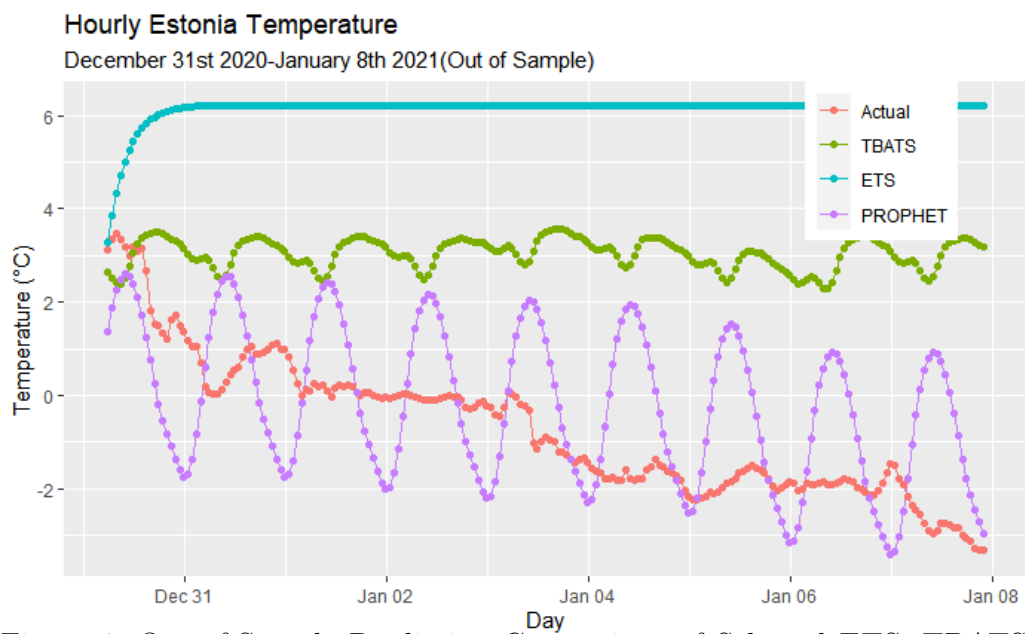


Figure 8: Out-of-Sample Prediction Comparison of Selected ETS, TBATS, and PROPHET Model.

3.3 Application of Models to Electricity Price Data

The data reflects the (spot) electricity price recorded at one-hour (frequency) intervals. A visual analysis of the data establishes strong, complex levels of seasonality. Figure 3 plots the electricity price data for five years from January 1, 2016, to December 31, 2020, and it is evident that the data exhibits the typical features of electricity prices and contains several periods of extreme volatility and price spikes.

In this study, to evaluate the performance of the proposed model, three individual models, such as ETS, TBATS, and PROPHET, were considered. The ETS and TBATS models were implemented and optimized using the Forecast package for R [11] while the prophet package for R [24] is used to model PROPHET.

Two subsets of the time series were generated: in-sample and out-of-sample (test-samples). The in-samples were used to build the model, whereas the test samples were used to assess the performance of the models. The in-sample contained 35040 (approximately four years) observations, and 8760 (a year) datapoint was used as out-of-samples. The models are evaluated based on one-step-ahead (1 hour) prediction up to 2160 (3 months). The selected models were used for performance analysis by making the aforementioned experimental setup.

The selected ETS model is **ETS(A, N, A)**, which means additive error, No trend, and Addictive seasonality. A standard model selection procedure was followed to obtain the **TBATS (1, {0,0}, 0.8, <24, 5>)** model and the estimated TBATS model was derived by applying a Box-Cox transformation of 1 representing no use of Box-Cox transformation, the order of ARMA error is (0, 0), a damping parameter of 0.8 (essentially doing nothing), and five Fourier pairs/harmonics with seasonal periods of 24. The decomposition of the electricity price time series is shown in Figure 9. The forth panels exhibit the daily seasonal components accommodated by the TBATS model which can be expressed as follows:.

$$\begin{aligned}
y_t &= l_{t-1} + b_{t-1} + s_{t-1} + \alpha d_t \\
b_t &= b_{t-1} + \beta d_t \\
s_t &= \sum_{j=1}^5 s_{j,t} \\
s_{j,t} &= s_{j,t-1} \cos\left(\frac{2\pi jt}{24}\right) + s_{j,t-1}^* \sin\left(\frac{2\pi jt}{24}\right) + \gamma_1 d_t \\
s_{j,t}^* &= -s_{j,t-1} \sin\left(\frac{2\pi jt}{24}\right) + s_{j,t-1}^* \cos\left(\frac{2\pi jt}{24}\right) + \gamma_2 d_t
\end{aligned}$$

where $\alpha = 0.009560$, $\beta = -0.2078$, $\gamma_1 = 0.0156$, $\gamma_2 = 0.0009$ are smoothing parameters.

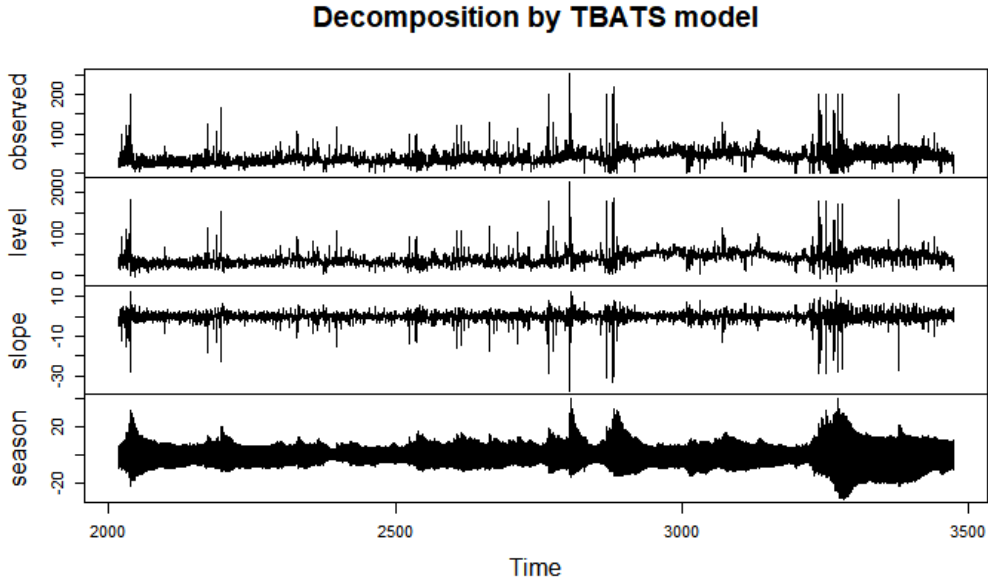


Figure 9: Decomposition of historical hourly electricity price in Estonia using TBATS.

Based on the current prediction framework, it is clear from the decomposition results of PROPHET in Figure 10 that the weekly decomposition found weekdays to be high electricity prices than weekends which shows a gradual decline.

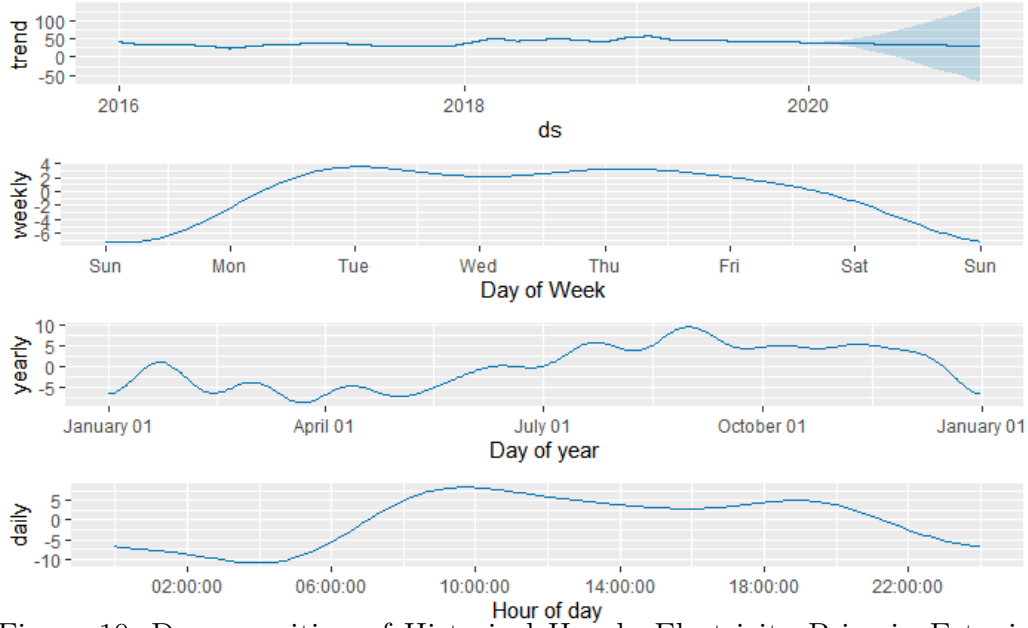


Figure 10: Decomposition of Historical Hourly Electricity Price in Estonia using PROPHET.

Table 3 and 4 show the forecasting evaluation metrics (RMSE and MAE values) of the selected time series forecasting models for different forecast time horizons as $h = 1$ (1 hours), 6 (6 hours), 12 (12 hours), 24 (1 day), 48 (2 days), 72 (3 days), 720 (1 month), 2160 (3 months) ahead is generated.

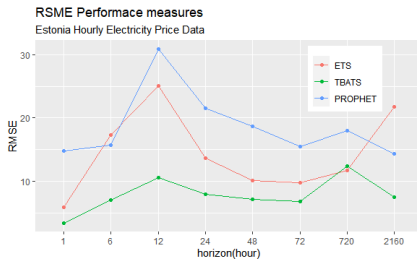
Model	Forecasting horizon (hour)							
	1	6	12	24	48	72	720	2160
ETS (A, N, A)	5.91	17.32	25.07	13.74	10.13	9.82	11.79	21.72
TBATS	3.46	7.08	10.64	8.02	7.20	6.82	12.43	7.53
PROPHET	14.781	15.76	30.81	21.51	18.67	15.46	18.02	14.41

Table 3: **RMSE** Performance Matrix of Selected Models to Estonia Electricity Price Data for Different Horizons

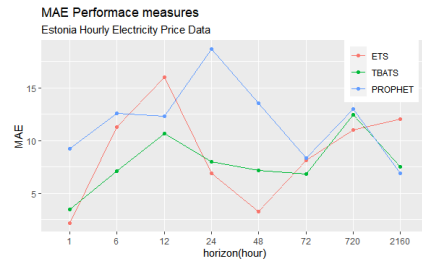
Figure 11 compares the out of sample forecasting accuracy of the selected ETS, TBATS, and PROPHET models. The TBATS model provides much better forecasting accuracy than the other models. Experiment results in Figure 12 and 13 show the comparison of our model forecast with actual time series data, while ETS

Model	Forecasting horizon (hour)							
	1	6	12	24	48	72	720	2160
ETS (A, N, A)	2.19	11.29	16.02	6.92	3.28	8.16	11.03	12.05
TBATS	3.46	7.08	10.64	8.02	7.20	6.82	12.43	7.53
PROPHET	9.26	12.58	12.29	18.65	13.52	8.37	13.01	6.89

Table 4: **MAE** Performance Matrix of Selected Models to Estonia Electricity Price Data for Different Horizons



(a) RMSE across different forecast horizons comparing out-of-sample performances



(b) MAE across different forecast horizons comparing out-of-sample performances

Figure 11: Forecasting Accuracy for Electricity Price from 1 to 2160 hours (i.e. 3 month).

and TBATS forecasts quite well compared with the original time series and can be considered satisfactory.

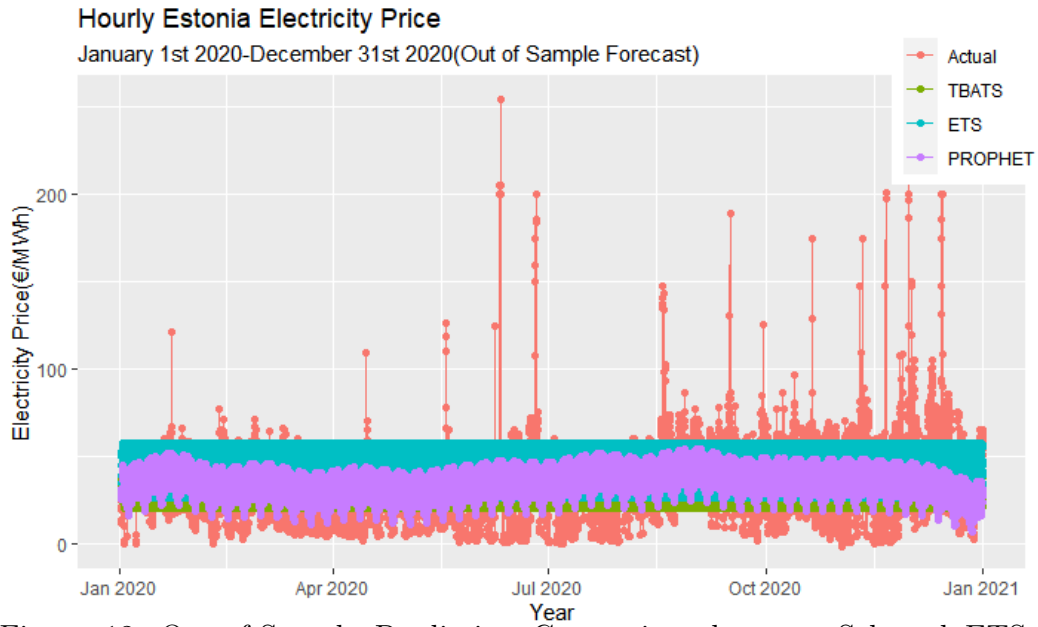


Figure 12: Out-of-Sample Prediction Comparison between Selected ETS, TBATS, and PROPHET Model.

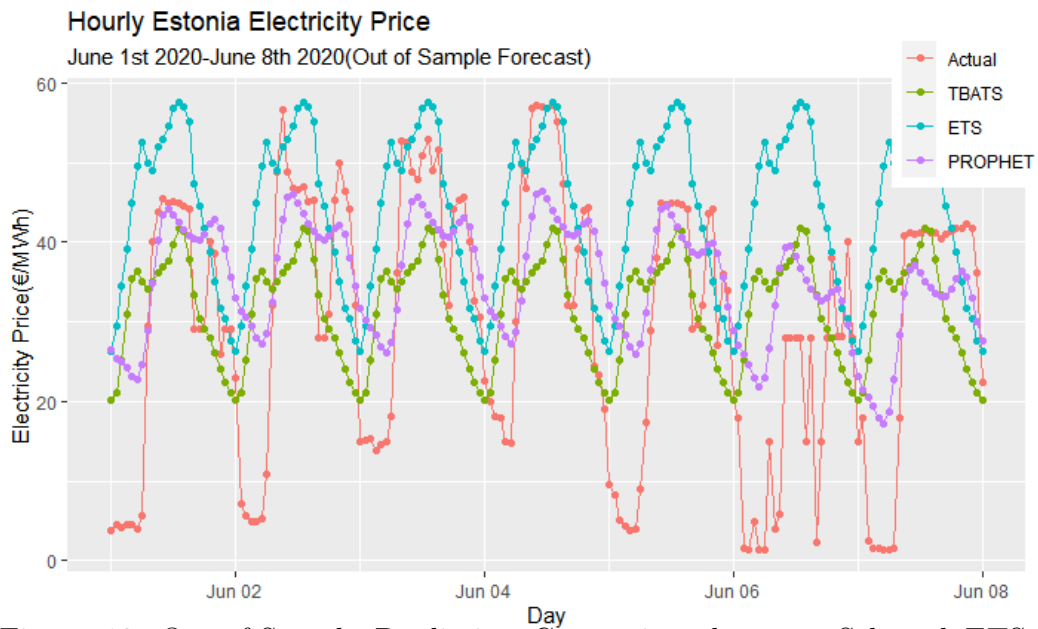


Figure 13: Out-of-Sample Prediction Comparison between Selected ETS, TBATS, and PROPHET Model.

Conclusions

In Today's era, the complex seasonality frequently appears in long time series such as daily, hourly, and weekly time series data. Due to the limitations of traditional forecasting methods to handle complex seasonality, new state-space modeling methods like TBATS have become more important.

Real-time prediction with high accuracy depends on the model used for training and testing automated forecasting procedure that integrates seasonality detection, estimation, model selection, and forecasting for complex seasonality time series.

The TBATS, ETS, and PROPHET models are compared in this work, and their ability to make short-term and long-term forecasts are investigated. Two error measures, RMSE and MAE, were studied at different forecasting horizons to compare these models' forecasting accuracy. The study analyses reveal complex seasonal variation, and forecast values were produced and compared for different forecasting horizons for up to 3-months using three selected models. By Examining the accuracy metric values in Table 1 - Table 4, we conclude that in most cases, the TBATS model outperforms the ETS and PROPHET, which has the least minimum RMSE, and MAE values.

We have investigated the model's performance using financial (electricity price) and non-financial (temperature) time series. The real-life data allowed us to demonstrate that the TBATS model outperforms ETS and PROPHET for long and short-term forecast horizons as it can capture complex seasonality.

Hence, we conclude that the TBATS forecasting model can be used as an easy-to-implement and automatic, strong baseline in complex time series forecasting space.

References

- [1] S. Armstrong. *Principles of Forecasting: A Handbook for Researchers and Practitioners*. Third. Kluwer Academic Publishing, 2001.
- [2] G. E. P. Box, G. M. Jenkins, and G. M Ljung. *Time Series Analysis: Forecasting and Control*. Wiley, 2015.
- [3] P. J. Brockwell and R. A. Davis. *Introduction to Time Series and Forecasting*. Springer, 1996.
- [4] R. Cottet and M. Smith. “Bayesian Modeling and Forecasting of Intra-day Electricity Load”. In: *Journal of Operational Research Society* 98 (2003), pp. 839–849.
- [5] F. Gao et al. “Forecasting power market clearing price and quantity using a neural network method”. In: *Power Engineering Summer Meet., Seattle, WA* (2000), pp. 2183–2188.
- [6] G. Gross and F. D. Galiana. “Short-term load forecasting”. In: *IEEE* 75 (1987), pp. . 1558–1573.
- [7] Adrew C. Harvey. *Forecasting, Structural Time Series Models and the Kalman Filter*. Cambridge University Press, 1989.
- [8] R. J. Hyndman and Y. Khandakar. “Automatic time series forecasting: The forecast package for R”. In: . *Journal of Statistical Software* 27 (2008). URL: <https://dx.doi.org/10.18637/jss.v027.i03>.
- [9] Rob J Hyndman et al. *Forecasting with Exponential Smoothing: the State Space Approach: The State Space Approach*. Springer, 2008.
- [10] Rob J. Hyndman and George Athanasopoulos. *Forecasting: Principles and Practice*. Third. OTexts, 2018. URL: <https://otexts.com/fpp3/>.

- [11] Rob J. Hyndman and Yeasmin Khandakar. *Automatic time series for forecasting: the forecast package for R. Monash University, Department of Econometrics and Business Statistics*. 6/07. 2007.
- [12] Rob J. Hyndman and Anne B. Koehler. *Another look at measures of forecast accuracy. International Journal of Forecasting*. 2006, pp. 679–688.
- [13] A. M. De-Livera, R. J. Hyndman, and R. D. Snyder. *Forecasting time series with complex seasonal patterns using exponential smoothing. Journal of the American Statistical Association*. 2011, pp. 1513–1527. URL: <https://www.tandfonline.com/doi/abs/10.1198/jasa.2011.tm09771>.
- [14] M. Blum and M. Riedmiller. “Electricity Demand Forecasting Using Gaussian Processes”. In: *AAAI Press* (2013), pp. 10–13.
- [15] N. H. Miswan, R. M. Said, and N. H. Hussin. “Comparative Performance of ARIMA and DES Models in Forecasting Electricity Load Demand in Malaysia”. In: *International Journal of Electrical Computer Sciences IJECS-IJENS* 16 (2016), pp. 6–9.
- [16] F. Nilgun and M. Betul. “Incorporating Explanatory Effects of Neighbour Airports in Forecasting Models for Airline Passenger Volumes”. In: *Proceedings of 5th the International Conference on Operations Research and Enterprise System* (2016), pp. 178–185.
- [17] B. Ramsay and A. J. Wang. “An electricity spot-price estimator with particular reference to weekends and public holidays,” in: *UPEC Manchester, U.K* (1997), pp. 371–374.

- [18] P. Skantze, M. Ilic, and J. Chapman. “Stochastic modeling of electric power prices in a multi-market environment”. In: *Power Engineering Winter Meet., Singapore* (2000), pp. 1109–1114.
- [19] B.R. Szkuta, L. A. Sanabria, and T. S. Dillon. “Electricity price short-term forecasting using artificial neural networks”. In: *IEEE Trans. Power Syst* 14 (1999), pp. 851–857.
- [20] M. Dhalival T. Nakashima and T. Niimura. “Electricity market data representation by fuzzy regresion models”. In: *presented at the Power Eng. Summer Meet., Seattle* (2000).
- [21] James W. Taylor. “Short-term electricity demand forecasting using double seasonal exponential smoothing”. In: *Journal of Operational Research Society* 54 (2003), pp. 799–805.
- [22] S. Taylor and B. Letham. “Forecasting at Scale”. In: *The American Statistician* 17 (2018), pp. 837–45. URL: [doi : 10 . 1080 / 00031305 . 2017.1380080](https://doi.org/10.1080/00031305.2017.1380080).
- [23] Facebook’s Core Data Science team. *Prophet: Automatic Forecasting Procedure*. Github. 2016. URL: [https : / / github . com / facebook / prophet](https://github.com/facebook/prophet).
- [24] Facebook’s Core Data Science team. *Prophet Forecasting At Scale*. URL: <https://facebook.github.io/prophet/>.
- [25] Mike West and Jeff Harrison. *Bayesian Forecasting and Dynamic Models*. Springer, 1997.
- [26] P. Zhou, B. W. Ang, and K. L. Poh. “A Trigonometric Grey Prediction Approach to Forecasting Electricity Demand”. In: *Energy, Issue* 31 (2006), pp. 2839–2847.

Appendix 1: R code for data exploration and visualisation

```
1 library(forecast);library(tidyverse);library(forecast);library(
  xts);library(ggplot2);library(prophet);library(lubridate);
  library(dplyr);library(scales);library(tidyr);library(pacman)
  ;library(fpp2);require(tseries);library(tidyr)
2
3 #####
4 #Data 1: Temperature Dataset#####
5 #####
6 #Read in Temperature dataset
7 data <- read.table(file = "thesisdata2.txt", sep = ",", header =
  TRUE)
8 #date manipulation
9 dat.xts <- xts(data$Temperature, as_datetime(data$Times))
10 ep<-endpoints(dat.xts, on="hours", k=1) #on= how your data is
  spaced ,k=how you want to club it
11 dat.xts.hourly<-period.apply(dat.xts, FUN=mean,INDEX=ep)
12 df <- fortify(dat.xts.hourly)
13 names(df) <- c("Date", "Temperature")
14 #convert temperature data into multiple seasonality time series
  data
15 x11 <- msts(dat.xts.hourly, seasonal.periods=c(24,24*7,365*24),
  start=2016)
16 autoplot(x11, color="blue") + xlab("Year") + ylab("Temperature (
  C)") + labs(title = "Hourly Estonia Temperature", subtitle =
  "January 1st 2016– December 31st 2021")
17 #Seasonal Subseries Plot
18 ggsubseriesplot(x11) + ggtitle("Seasonal Subseries plot: Hourly
  Estonia Temperature")+ xlab("Month") + ylab("Temperature")
19 #Test for Autocorrelation
```

```

20 checkresiduals(x11)
21 #####
22 #Dataset 2– Estonia Energy Consumotion###
23 #####
24 #Read in yearly dataset from 2016–2021
25 data2 <- read.table(file = "electricity-nps price_20220408 (1).
      csv", sep = ";", header = TRUE)
26 data3 <- read.table(file = "electricity-nps price_20220408 (2).
      csv", sep = ";", header = TRUE)
27 data4 <- read.table(file = "electricity-nps price_20220408 (3).
      csv", sep = ";", header = TRUE)
28 data5 <- read.table(file = "electricity-nps price_20220408 (4).
      csv", sep = ";", header = TRUE)
29 data6 <- read.table(file = "electricity-nps price_20220408 (5).
      csv", sep = ";", header = TRUE)
30 #bind all dataset
31 total <- rbind(data2, data3, data4, data5, data6) #used to create
      total dataframe
32 #create new dataframe of dates and prices of electricity
33 dfnew <- data.frame(total$Kuup ev..Eesti.aeg., total$NPS.Eesti)
34 names(dfnew) <- c("Date", "Prices")
35 dfnew$Prices = gsub(",", ".", dfnew$Prices) #change decimal point
      from , to .
36 dfnew$Prices = as.numeric(as.character(dfnew$Prices))
37 dfnew$Date <- strptime(dfnew$Date, format = "%d.%m.%Y %H:%M ")
38 format(dfnew$Date, format = "%Y-%m-%d %H:%M")
39 #multiple seasonality for energy price data
40 x3 <- msts(dfnew$Prices, seasonal.periods=c(24), start=2016)
41 #Some Exploratory Analysis
42 #Normal Plot
43 autoplot(x3, color="blue") + xlab("Year") + ylab("Electricity
      Price ( /MWh)") + labs(title = "Hourly Estonia Electricity
      price", subtitle = "January 1st 2016– December 31st 2020")

```

```

44 #Seasonal Plot
45 ggseasonplot(x3) + ggtitle("Seasonal plot: Hourly Estonia
    Electricity price")+ xlab("Month") + ylab("Electricity Price"
    )
46 #Seasonal Subseries Plot
47 ggsubseriesplot(x3) + ggtitle("Seasonal Subseries plot: Hourly
    Estonia Electricity price")+ xlab("Month") + ylab("
    Electricity price")
48 #Test for Autocorrelation
49 checkresiduals(x3)

```

Appendix 2: R code for ETS, TBATS and PROPHET

```
1 train1= subset(x11, end=24*365*5)
2 #Apply TBATS and forecast
3 temptbats111<- tbats(train1, use.box.cox = NULL, use.trend = NULL,
  use.damped.trend = NULL, seasonal.periods = NULL, use.arma.
  errors = TRUE)
4 fc44.8= forecast(temptbats111, h=24*365)
5 accuracy(fc44.8, x11)
6
7 x12 <- msts(dat.xts.hourly, seasonal.periods=c(24), start=2016)
8 train10= subset(x12, end=24*365*5)
9 #Apply ETS and forecast
10 temptbats222<- ets(train10)
11 fc550.8= forecast(temptbats222, h=24*365)
12 accuracy(fc550.8, x12)
13 #####
14 ##Apply prophet to temperature dataset##
15 #####
16 stats1=data.frame(y=(df$Temperature)
17                   ,ds=df$Date)
18 stats1=aggregate(stats1$y,by=list(stats1$ds),FUN=sum)
19 colnames(stats1)<- c("ds","y")
20 #Removing last 8760(1 year) rows from dataframe, storing last
  measurements in a separate vector.
21 series1=stats1[1:(nrow(stats1)-8760),]
22 actual1=tail(stats1,n=8760)
23 train1 = series1
24 test1 = actual1
25 future1.8 = make_future_dataframe(model1, periods = 24*365)
26 forecast1.8 = predict(model1, future1.8)
27 #####
28 #Apply TBATS on electricity price Data##
```

```

29 #####
30 train20= subset(x3, end=24*365*4)
31 #Apply TBATS and forecast
32 temptbats20<- tbats(train20, use.box.cox = NULL, use.trend = NULL,
    use.damped.trend = NULL, seasonal.periods = NULL, use.arma.
    errors = TRUE)
33 fc20.8= forecast(temptbats20, h=24*365)
34 accuracy(fc20.8,x3)
35 #####
36 #Apply ETS on electricity price Data##
37 #####
38 x4= ts(dfnew$Prices, frequency=24,start=2016)
39 train30= subset(x4, end=24*365*4)
40 #Apply ETS and forecast
41 ets01<- ets(train30)
42 fcets01.8=forecast(ets01, h=24*365)
43 accuracy(fcets01.8,x4)
44 #####
45 ##prophet with electricity price data
46 #####
47 stats=data.frame(y=(dfnew$Prices)
48                 ,ds=dfnew$Date)
49 stats=aggregate(stats$y,by=list(stats$ds),FUN=sum)
50 colnames(stats)<- c("ds","y")
51 # Removing last 8760(1 year) rows from dataframe, storing last
    measurements in a separate vector.
52 series10=stats[1:(nrow(stats)-8760),]
53 actual10=tail(stats,n=8760)
54 train10 = series10
55 test10 = actual10
56 model10 = prophet(train10, daily.seasonality=TRUE)
57 future10.8 = make_future_dataframe(model10, periods = 24*365)
58 forecast10.8 = predict(model10, future10.8)

```

```

59 #####
60 #EVEALUATION METRICS
61 #####
62 forecastM_tbats=matrix(NA, 13, 24*3*30)
63 forecastM_ets=matrix(NA, 13, 24*3*30)
64 forecastM_prophet=matrix(NA, 13, 24*3*30)
65
66 errors_tbats=matrix(NA, 13, 24*3*30)
67 errors_ets=matrix(NA, 13, 24*3*30)
68 errors_prophet=matrix(NA, 13, 24*3*30)
69
70 for (i in 1:13){
71   # i=1
72   print(i)
73   currentTime=24*365*4+(i-1)*500
74   train <- subset(x3, end=currentTime)
75   test  <- subset(x3, start=currentTime+1, end=currentTime+24*3*
76     30)
77   tbats_i <- tbats(train, use.box.cox = NULL, use.trend = NULL, use
78     .damped.trend = NULL, seasonal.periods = NULL, use.arma.
79     errors = TRUE)
80   # Up to 3 months forecast
81   forecast_tbats <- forecast(tbats_i, h=24*3*30)
82   forecastM_tbats[i,] <-forecast_tbats$mean
83   errors_tbats[i,] <- test-forecast_tbats$mean
84   print("ets")
85   ets_i <- ets(train)
86   forecast_ets <- forecast(ets_i, h=24*3*30)
87   forecastM_ets[i,] <- forecast_ets$mean
88   errors_ets[i,]= test-forecast_ets$mean
89   print("prophet")
90   train20.1 <- stats[, 1:currentTime,]
91   colnames(train20.1) = c("ds", "y")

```



```

89 prophet_i <- prophet(train20.1)
90 future <- make_future_dataframe(prophet_i, periods = 24*3*30,
    freq = 3600)
91 forecast_prophet <- predict(prophet_i, future)[(currentTime+1)
    :(currentTime+24*3*30),]$yhat
92 forecastM_prophet[i,] <- forecast_prophet
93 errors_prophet[i,] <- test - forecast_prophet
94 }
95 median(abs(errors_prophet[,3]))
96 median(abs(errors_ets[,2]))
97 median(abs(errors_tbats[,3]))
98
99 MAE_prophet=rep(NA, 24*30*3)
100 MAE_ets=rep(NA, 24*30*3)
101 MAE_TBATS=rep(NA, 24*30*3)
102
103 for (i in 1:(24*30*3)){
104   MAE_prophet[i]=median(abs(errors_prophet[,i]))
105   MAE_ets[i]=median(abs(errors_ets[,i]))
106   MAE_TBATS[i]=median(abs(errors_tbats[,i]))
107 }
108
109 RMSE_prophet=rep(NA, 24*30*3)
110 RMSE_ets=rep(NA, 24*30*3)
111 RMSE_TBATS=rep(NA, 24*30*3)
112
113 P=errors_prophet**2
114 E=errors_ets**2
115 TB=errors_tbats**2
116
117 for (i in 1:(24*30*3)){
118   RMSE_prophet[i]=sqrt(mean(P[,i]))
119   RMSE_ets[i]=sqrt(mean(E[,i]))

```

```

120 RMSE_TBATS[i]=sqrt(median(TB[,i]))
121 }
122
123 plot(RMSE_TBATS)
124 lines(RMSE_ets, col=2)
125 lines(RMSE_prophet, col=3)
126
127 data_long <- melt(rsmemetric, id = "h")
128 ggplot(data_long, aes(x = factor(h), y = value, colour = variable
129   , group = variable)) +
130   geom_line()+geom_point()+labs(x = "h") +labs(y = "RSME")+ labs(
131     title = "RSME Performace measures for 5 years ",subtitle =
132     "Estonia Hourly Temperature Data") +
133   theme(legend.title = element_blank())+theme(legend.position = c
134     (0.2, 0.8),legend.background = element_rect(fill = "white")
135   )
136
137 data_mae <- melt(maemetric, id = "h")
138 ggplot(data_mae, aes(x = factor(h), y = value, colour = variable,
139   group = variable)) +
140   geom_line()+geom_point()+labs(x = "h") +labs(y = "MAE")+ labs(
141     title = "MAE Performace measures for 5 years ",subtitle = "
142     Estonia Hourly Temperature Data") +
143   theme(legend.title = element_blank())+theme(legend.position = c
144     (0.2, 0.8),legend.background = element_rect(fill = "white")
145   )
146
147 data_rsme2 <- melt(rsmemetric2, id = "h")
148 ggplot(data_rsme2, aes(x = factor(h), y = value, colour =
149   variable, group = variable)) +
150   geom_line()+geom_point()+labs(x = "h") +labs(y = "RSME")+ labs(
151     title = "RSME Performace measures for 4 years ",subtitle =
152     "Estonia Hourly Electricity Price Data") +

```

```

140 theme(legend.title = element_blank()) + theme(legend.position = c
      (0.2, 0.8), legend.background = element_rect(fill = "white")
      )
141
142 data_mae2 <- melt(maemetric2, id = "h")
143 ggplot(data_mae, aes(x = factor(h), y = value, colour = variable,
      group = variable)) +
144   geom_line() + geom_point() + labs(x = "h") + labs(y = "MAE") + labs(
      title = "MAE Performace measures for 4 years ", subtitle = "
      Estonia Hourly Electricity Price Data") +
145   theme(legend.title = element_blank()) + theme(legend.position = c
      (0.2, 0.8), legend.background = element_rect(fill = "white")
      )

```

Appendix 3.

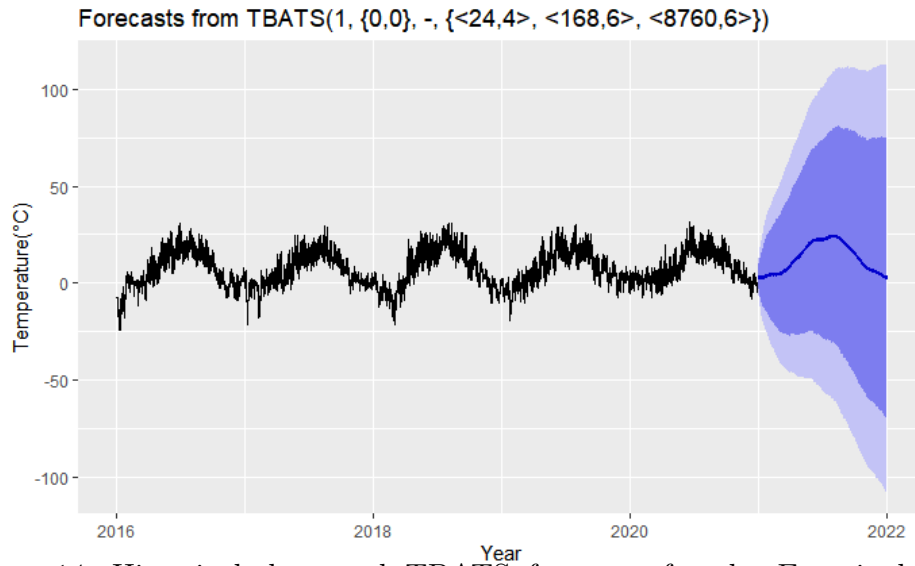


Figure 14: Historical data and TBATS forecasts for the Estonia hourly weather temperature (1 year ahead Forecast).

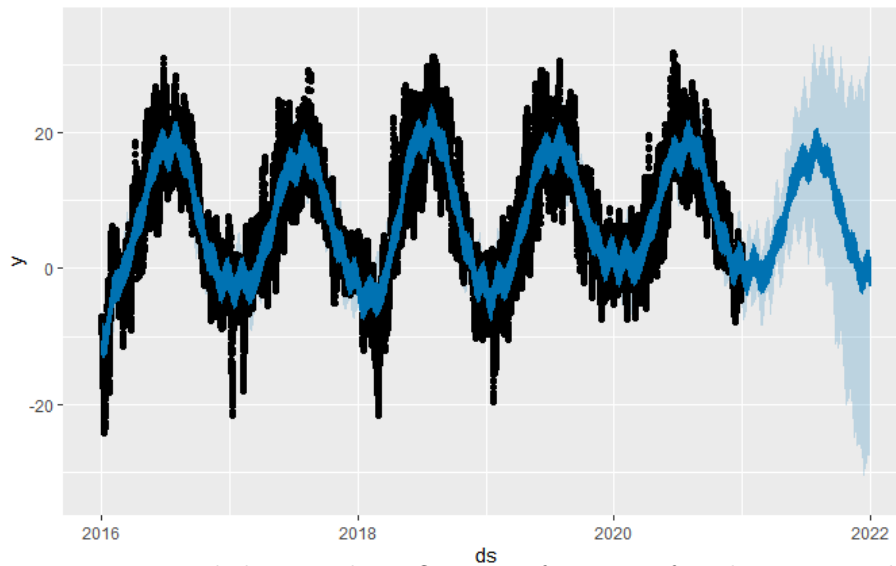


Figure 15: Historical data and PROPHET forecasts for the Estonia hourly weather temperature (1 year ahead Forecast).

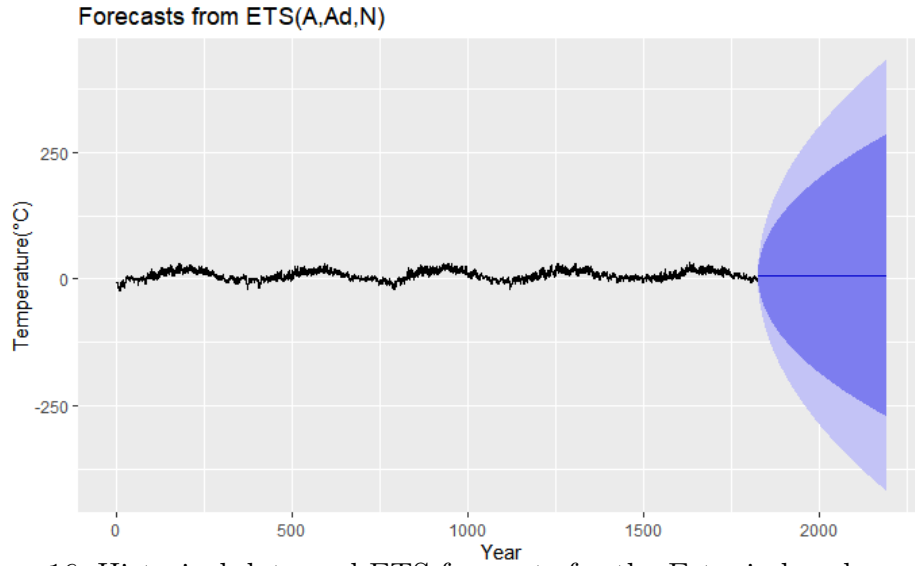


Figure 16: Historical data and ETS forecasts for the Estonia hourly weather temperature (1 year ahead Forecast).

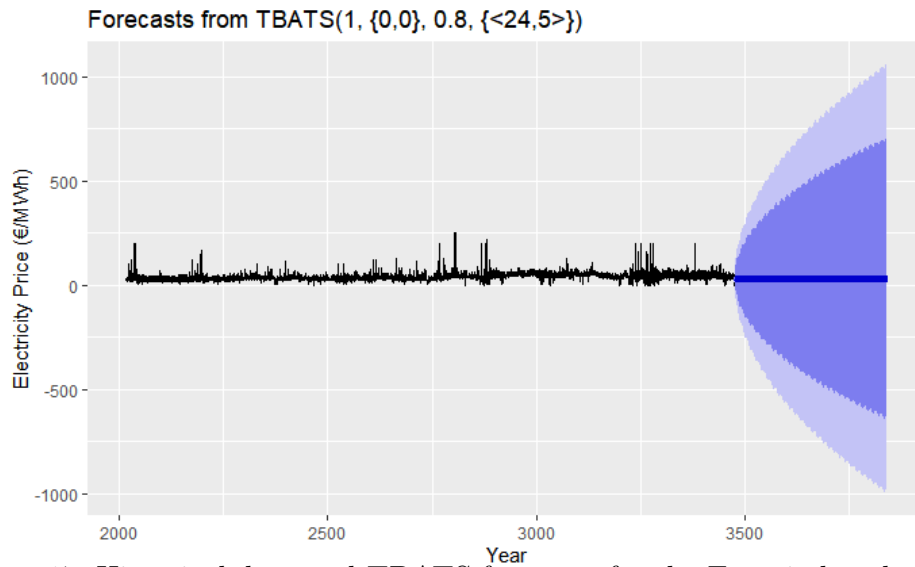


Figure 17: Historical data and TBATS forecasts for the Estonia hourly Electricity Price (1 year ahead Forecast).

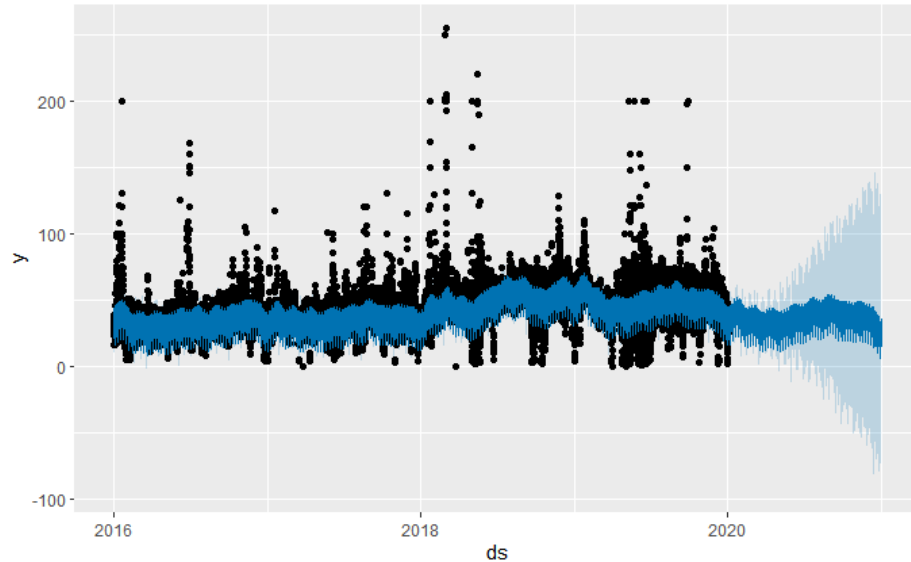


Figure 18: Historical data and PROPHET forecasts for the Estonia hourly Electricity Price (1 year ahead Forecast).

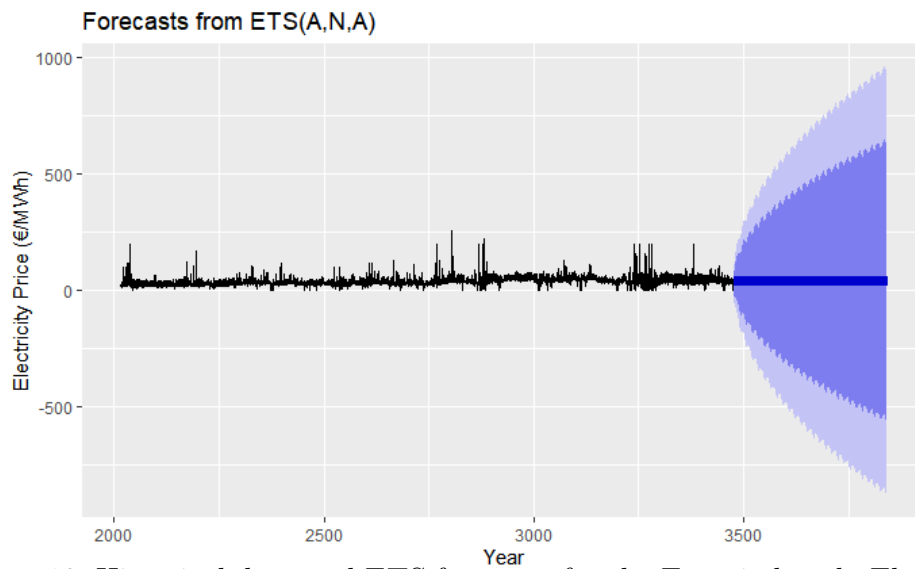


Figure 19: Historical data and ETS forecasts for the Estonia hourly Electricity Price (1 year ahead Forecast).

Non-exclusive licence to reproduce thesis and make thesis public

I, **Adedokun Abdul-Baaki Dolapo**,

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright, **Modeling complex seasonalities**, supervised by **Märt Möls**.
2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.
3. I am aware of the fact that the author retains the rights specified in p. 1 and 2.
4. I certify that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Adedokun Abdul-Baaki Dolapo

17/05/2022