Department of Mathematics
TUM School of Computation, Information and Technology
Technical University of Munich

Hierarchical Time Series Forecasting for Electricity Demand

Master's Thesis

Paraskevi Kiriakidou

Thesis for the attainment of the academic degree

Master of Science

at the TUM School of Computation, Information and Technology of the Technical University of Munich

Supervisor:

Prof. Dr. Felix Dietrich

Advisors:

Dr. Marisa Mohr (inovex GmbH)
Dr. Lea Petters (inovex GmbH)

Submitted:

Munich, 07. February 2025

I hereby declare that this thesis is my own work and that no other sources have been used except those clearly indicated and referenced.

Munich, 07. February 2025 Paraskevi Kiriakidou

Acknowledgements

First and foremost, I would like to express my sincere gratitude to Prof. Dr. Felix Dietrich for the opportunity to write my thesis at the Chair of Physics-Enhanced Machine Learning and for his invaluable guidance and support throughout this journey.

I am especially grateful to Dr. Marisa Mohr and Dr. Lea Petters for their immense mentorship, supervision, and encouragement. Their insights and expertise have been invaluable. I also extend my thanks to inovex GmbH for funding this thesis and providing essential computational resources.

My deepest gratitude goes to my friends and family for their unwavering support, without which I could not have completed my studies.

A special thank you to Julian Posch, Niklas Schulte-Geers, Barış Onarıcı, Dilvan Sabir, Gentrit Fazlija, and Awayah Shahid for keeping me motivated over the past months through countless coworking sessions and uplifting conversations. Katharina König and Julia Koch, thank you for always being there for me.

Finally, a heartfelt thanks goes to my partner, James Gray, for his unwavering support and encouragement. Thank you for always having my back and believing in me.

Abstract

Hierarchical Time Series frequently appear in real-world applications. In this thesis, we focus on the hierarchical time series given by the electricity demand in the United States. To our best knowledge, this is the first time that this dataset is used for hierarchical time series forecasting in its entirety. We make two main contributions. Our first main contribution consists of applying various forecasting and reconciliation techniques and evaluating which of them performs best on this dataset. Our second main contribution is using a novel clustering heuristic in order to obtain a new hierarchical structure on the time series data. Our clustering method focuses on capturing local patterns in the data as opposed to usual clustering techniques which focus on global similarity of time series. As a result, we obtain a better forecasting accuracy at the bottom level with respect to all metrics that we examine.

Contents

1 Introduction

Hierarchical Time Series occur naturally in real-world applications. Whenever there is data collected over time, they form a time series. A hierarchical time series appears whenever there are time series data collected at various hierarchical layers. Frequently, such a hierarchy on time series is given by geographic conditions, i.e. data is collected regionally and statewide or on some higher geographic structure.

The hierarchical time series that we work with in this thesis is given by electricity demand in the United States. The electricity demand is captured locally by so-called balancing authorities and aggregated to regional groupings and, on the total level, country-wide for the entirety of the United States. Hence, for our data, the hierarchy is naturally an attribute of the electricity grid. The motivation behind working with this dataset in the literature about hierarchical time series, frequently, the same datasets are used, focusing on certain application areas. In particular, to our best knowledge, this concrete dataset has not been used in this context in its entirety.

Whenever a hierarchy on time series data is present and we perform forecasting on more than one of the levels, the question of how to make the predictions on the different levels of the hierarchy consistent is raised. When a higher level of the hierarchy is the aggregation of several lower levels of the same hierarchy and we make forecasts on both of the levels, we would like to make sure that the forecasts adhere to the aggregation structure. This is done by reconciliation, which we explain later in more detail. Which method of reconciliation is best suited for the use case strongly depends both on the dataset and the forecasting method used.

This thesis aims at making progress on several topics. First, since this dataset has not been used in hierarchical time series forecasting before, we would like to compare several forecasting models together with their reconciliations and evaluate which of them performs better in terms of the forecasting accuracy on this particular dataset. Moreover, it is an interesting question whether we can improve on the naturally existing hierarchy. To be more detailed, when we do not use the naturally present hierarchy, but instead use a different hierarchy that we extract from the data, can we improve the forecasting accuracy? We answer this question in the affirmative. In more detail, we perform a clustering on the time series to obtain the new hierarchy. For clustering the time series data, we use a heuristic which we believe to better capture the short term changes, as opposed to clustering the time series once over the whole course of time.

Applying clustering to hierarchical time series is not an entirely new idea, in the following, we summarize some research that has been done in this area. We make no claim of this list being complete, at the opposite, due to spatial constraints, we limit ourselves to a selection of references that we found to be most important. A much more extensive overview over the research in the area of hierarchical time series forecasting may be found in [4].

In [33] clustering is used for hierarchical time series forecasting. The authors start by clustering the time series several times with a varying number of clusters, putting similar time series in the same cluster. As they perform clustering more than once, they arrive at a large number of candidate clusters. The choice of clusters for the new hierarchical structure is done automatically via the sparse penalty method. Evaluation of their approaches on real-life electricity and solar power datasets indicate an improvement in the accuracy of the prediction which uses the chosen clustering as the hierarchical structure.

Another approach that has been explored is combining graph structures with hierarchical time series forecasting. This idea was worked on in [13], where a Hierarchical Graph predictor is constructed. First, the authors show how to incorporate the hierarchical structure in a graph neural network. Then, they turn

towards a setting in which the hierarchical structure is learned from the data by the neural network via graph pooling. Finally, they show how reconciliation along the learned hierarchical structure is done. The experiments that they conduct show that their approach can at least achieve state-of-the-art accuracy.

A novel clustering algorithm for hierarchical time series is presented in [20]. It is a model-free method that performs clustering on each aggregated level. The algorithm is evaluated on several datasets and shows a better clustering performance. Furthermore, it is also used for forecasting and is shown to reduce the computation time especially in use cases where forecastings for many hierarchical time series are to be done.

In [44], several research questions concerning clustering in hierarchical time series are answered. It is shown that when given a hierarchical time series consisting only of two levels, introducing an intermediate level with the help of clustering can improve the forecasting performance. Secondly, the authors claim that, having performed a clustering to obtain a new hierarchical structure, the new structure of the hierarchy has a bigger influence on the forecasting performance than the similarity of time series inside the clusters. Finally, an approach of using an average of reconciled forecasts stemming from different hierarchical structures obtained by clustering is employed, resulting in improved forecasting accuracy compared to any single hierarchical structure in experiments.

The thesis is structured as follows. In chapter 2, we review the theoretical background of time series forecasting, focusing on the forecasting models that we use in our experiments, in particular Exponential Smoothing (ETS) and Autoregressive Integrated Moving Average (ARIMA). Subsequently, we turn our attention towards hierarchical time series, describing several approaches for forecasting and reconciliation. Afterwards, we give a brief introduction to the clustering algorithms needed for this thesis. Finally, an overview over electricity supply in the United States is given, in order to give context for the dataset which we use.

Chapter 3 contains the main part of this thesis, this being the experiments conducted. It begins with a closer introduction to the electricity data set which serves as our experimental data set. We also clarify how the dataset was prepared before the start of the experiments. After that, we explain in detailed fashion how the experiment was set up, including both the technical details of the experiments as well as how we evaluated the experiments. Moreover, we connect the theoretical background from chapter 2 with the experiments by outlining how we applied the models to our dataset. Thereafter, we direct our attention towards the clustering of the hierarchical time series. In order to motivate the conducted experiments, we first inspect a case theoretically, using methods of forecasting and reconciliation which we also use in our experiments. Finally, we present the results of our experiments.

We provide a summary of the experiments conducted and the results thereof in the last chapter of this thesis.

2 Background and State of the Art

In this chapter, we will introduce several key concepts and provide an overview of the state of the art for hierarchical time series forecasting. We will begin by exploring time series forecasting. Since it is a vast field, we will concisely overview the essential concepts needed for a foundational understanding. Afterwards, we will discuss hierarchical time series and introduce the concept of reconciliation. Finally, we will understand how electricity operates in the US and discuss which methods exist specifically for electricity demand forecasting.

2.1 Time Series Forecasting

A time series is a sequence of data points collected sequentially over time - usually at evenly spaced intervals. There are many examples of various domains where we utilize time series data. In finance, we track stock prices and exchange rates. In economics, we monitor indicators such as GDP, inflation, and unemployment rates, and we need to plan future demand and production rates in retail. Temperature and precipitation data are essential for tracking climate change. Similarly, healthcare relies on time series data to track patient vital signs, disease rates, and healthcare resource utilization. The most prominent example is during the COVID-19 pandemic when we relied on time series data to predict the spread of the virus.

Formally one can define a time series as a stochastic process.

Definition 2.1.1 (Stochastic Process [36]) *A stochastic process is a collection of random variables* $\{y_t\}$ *indexed by* $t = 1, ..., T$*. The observations of a stochastic process are referred to as realizations.*

A realization of a time series y_t is assumed to be composed by a signal μ_t and a noise term e_t such as

$$y_t = \mu_t + e_t.$$

What makes time series unique is that, unlike other data, the observations do not arise independently from a common population. Because of their temporal dependence, the value of an observation is influenced by previous observations [14]. Therefore, methods based on the assumption of independent data cannot be applied. The linear dependence between observations can be measured as in classical statistics, using the autocovariance function

$$\gamma(s,t) := \mathrm{cov}(y_s, y_t) = \mathrm{E}[(y_s - \mu_s)(y_t - \mu_t)],$$

for $\mu_t := \mathrm{E}(y_t)$. Forecasting is one of the most common tasks applied to time series data and is especially crucial for decision-making and planning. The objective is to model the stochastic mechanism of the observed series, given T data points, to predict future values $\hat{y}_{T+h|T}$, where h is the forecast horizon. After fitting a model to a set of observations $\{y_1, ..., y_T\}$, referred to as training data, and predicting values for subsequent observations $\{y_{T+1}, y_{T+2}\}$, referred to as test data, the forecast error

$$\epsilon_{T+h} = y_{T+h} - \hat{y}_{T+h|T} \tag{2.1}$$

can be calculated, which is the difference between the forecast and the actual observed value. When calculating the difference between the training data's fitted and observed values, we talk about residuals. Forecasting techniques are based on the assumption that historical patterns will continue in the future and can, therefore, be learned. This involves learning underlying structures such as seasonality and trends, as

well as accounting for irregularities or random noise. Key characteristics that are important for many models are stationarity, particularly weak stationarity (also known as second-order or covariance stationarity), and autocorrelation.

Stationarity is a concept that describes the regular behavior of a time series, meaning that the probabilistic behavior of the series does not change over time. Since this is a too strong requirement for most applications, and it is not easily possible to assess if a time series fulfills this, a weaker version exists that only requires the first two moments of the series to stay stable over time.

Definition 2.1.2 (Weak Stationarity) *A time series $\{y_t\}$ is considered weakly stationary if it satisfies*

- $\mu(t) = \mu(t+u)$ *for all* t, u,
- $\gamma(t+u, t) = \gamma(u, 0)$ *for all* t, u *and*
- $E(y_t^2) < \infty$.

We will use "stationary" to describe this version of stationarity and omit the "weakly" part. Stationarity is an essential concept because many time series forecasting models are based on the assumption that the data are stationary. Non-stationary data, which shows trends, changing variances, or seasonality, often must be transformed (e.g., through differencing) to achieve stationarity before applying these models.

There are many approaches that can be used to test if a time series is stationary. Augmented Dickey-Fuller (ADF) is a statistical test that determines as the null hypothesis the presence of a unit root (i.e., it is non-stationary), while the alternative hypothesis is that the series is stationary. The Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test offers a complementary approach by testing for stationarity as the null hypothesis. Using both the ADF and KPSS tests together allows for a more robust assessment of stationarity.

Two key tools for analyzing the temporal dependence in a time series are the autocorrelation function (ACF) and the partial autocorrelation function (PACF).

Definition 2.1.3 (ACF, PACF [11][36]) *The autocorrelation function (ACF) at lag k is defined as*

$$\rho_k := \frac{\gamma(t, t+k)}{\sqrt{\gamma(t,t)\gamma(t+k, t+k)}}$$

and measures the linear predictability of the series at time $t+k$, y_{t+k}, using only y_t. The partial autocorrelation function (PACF) is the quantity ϕ_{kk}, regarded as a function of the lag k, that satisfies together with ϕ_{kj} the set of equations

$$\rho_j = \phi_{k1}\rho_{j-1} + ... + \phi_{k(k-1)}\rho_{j-k+1} + \phi_{kk}\rho_{j-k}$$

for $j = 1, 2, ..., k$ and measures the correlation between y_t and y_{t-k} with the linear effect of $y_{t-1}, ..., y_{t-k+1}$ removed.

The ACF measures the correlation between observations in a time series that are separated by a certain number of time lags. It helps to identify the degree of correlation between current values and past values at different time lags. A significant autocorrelation at a given lag suggests a dependency between those time points. The partial autocorrelation function (PACF), on the other hand, measures the correlation between observations at a specific lag after removing the effects of shorter lags. It provides a more focused view of the direct relationship between observations at a certain lag, excluding the influence of intermediate values. By examining the patterns in the ACF and PACF plots, we can gain insights into the underlying structure of the time series, such as seasonality or the number of lags needed in a forecasting model.

Over the years, multiple models have been developed for time series forecasting, each with different assumptions about the data. In general, there are two approaches: the frequency domain approach and the time domain approach.

- The frequency domain approach focuses on decomposing a time series into its underlying cyclical or seasonal components by examining the data's frequency characteristics. Techniques such as fourier analysis are often used in this approach to identify dominant cycles or periodicity.
- The time domain approach involves modeling the dependencies between observations over time directly. This is the more commonly used approach in many forecasting applications and forms the basis for a range of well-known models, including autoregressive and exponential smoothing methods.

We will focus on three foundational models from the time domain approach: the persistence model, exponential smoothing (ETS), and seasonal autoregressive integrated moving average (SARIMA).

Persistence Model

The persistence model, also known as the naive model, is a very intuitive and one of the simplest forecasting techniques. It assumes that the most recent observation is the best predictor of future values.

Definition 2.1.4 (Persistence Model [27]) *In the persistence model all forecasts are set to the value of the last observation,* $\hat{y}_{T+h|T} = y_T$.

An extension of this approach is the seasonal persistence model, which is specifically suitable for time series with recurring seasonal patterns. Instead of forecasting based on the most recent value, this model predicts future values based on the observation from the same point in the previous seasonal cycle.

Definition 2.1.5 (Seasonal Persistence Model [27]) *In the seasonal persistence model all forecasts are set to the values of the last observed season,* $\hat{y}_{T+h|T} = y_{T+h-m(k+1)}$*, where* m *is the seasonal length, and* k *is the integer part of* $(h-1)/m$.

Despite their simplicity, the persistence models already work remarkably well for many use cases and can serve as good baselines for comparison. Their advantage also lies in their minimal computational requirements.

Exponential Smoothing (ETS)

The persistence model assumes that only the most recent observation provides information for the future and is thus the only important one. An alternate assumption could be to assume all past observations are important with equal weight. Both assumptions are very extreme. It would be more realistic to assume something in the middle, where more importance is given to recent data points, while not generally disregarding all of them. This is exactly the idea behind exponential smoothing models, which are a class of widely used forecasting methods that apply exponentially decreasing weights to past observations.

Definition 2.1.6 (Simple Exponential Smoothing) *In the simplest form of the exponential smoothing models, the one step ahead forecast is*

$$\hat{y}_{T+1|T} = \alpha y_T + \alpha(1-\alpha)y_{T-1} + \alpha(1-\alpha)^2 y_{T-2} + \cdots,$$

where $0 < \alpha < 1$ *is the smoothing constant.*

Setting $\alpha = 1$ leads to the persistence model and setting $\alpha = 0$ corresponds to the idea of all past observations being equally important. It can be easily verified that

$$\begin{aligned} \hat{y}_{t+h|t} &= \ell_t \\ \ell_t &= \alpha y_t + (1-\alpha)\ell_{t-1} \end{aligned}$$

corresponds to the simple exponential smoothing model. The smoothed value ℓ_t is called level in this representation. For the forecasts beyond the training data $t = T$ needs to be set and in this simple model the forecasts are flat, i.e. $\hat{y}_{T+h|T} = \hat{y}_{T+1|T} = \ell_T$. This simple exponential smoothing model has been extended multiple times leading to a class of different models. Holt (1957) added a component to model trend and updated the model to

$$\begin{aligned} \hat{y}_{t+h|t} &= \ell_t + hb_t \\ \ell_t &= \alpha y_t + (1-\alpha)(\ell_{t-1} + b_{t-1}) \\ b_t &= \beta(\ell_t - \ell_{t-1}) + (1-\beta)b_{t-1}, \end{aligned} \tag{2.2}$$

where $0 < \alpha < 1$ is the smoothing parameter for the level ℓ_t and $0 < \beta < 1$ is the smoothing parameter for the slope of the trend b_t [23]. The h-step ahead forecasts are no longer flat as a result of the constant trend. It has been observed that this method tends to over-forecast for longer forecast horizons. This is why Gardner and Mckenzie (1985) further extended the model to also include a parameter to dampen the trend, leading to

$$\begin{aligned} \hat{y}_{t+h|t} &= \ell_t + (\phi + \phi^2 + \cdots + \phi^h)b_t \\ \ell_t &= \alpha y_t + (1-\alpha)(\ell_{t-1} + \phi b_{t-1}) \\ b_t &= \beta(\ell_t - \ell_{t-1}) + (1-\beta)\phi b_{t-1}, \end{aligned}$$

where $0 < \phi < 1$ is the damping parameter [18]. In this version, the trend of the forecasts in the distant future becomes constant, while the forecasts in the near future still have a trend. Finally, Holt (1957) and Winters (1960) further extended the method to also incorporate a seasonality. In the additive version, the component form is

$$\begin{aligned} \hat{y}_{t+h|t} &= \ell_t + hb_t + s_{t+h-m(k+1)} \\ \ell_t &= \alpha(y_t - s_{t-m}) + (1-\alpha)(\ell_{t-1} + b_{t-1}) \\ b_t &= \beta(\ell_t - \ell_{t-1}) + (1-\beta)b_{t-1} \\ s_t &= \gamma(y_t - \ell_{t-1} - b_{t-1}) + (1-\gamma)s_{t-m}, \end{aligned}$$

where $0 < \gamma < 1$ is the smoothing parameter for the seasonal component s_t and k is the integer part of $(h-1)/m$ [23][42]. The level is modeled as a weighted average between the seasonally adjusted observation and the non-seasonal forecast for time t and the seasonal component is a weighted average between the current seasonal index and the seasonal index of the same season m time periods ago. The trend is the same as in Holt's linear method as described in Equation (2.2). This version is suitable for time series with constant seasonal variations. When the seasonal variations are changing proportionally to the level of the series, the multiplicative version

$$\begin{aligned} \hat{y}_{t+h|t} &= (\ell_t + hb_t)s_{t+h-m(k+1)} \\ \ell_t &= \alpha\frac{y_t}{s_{t-m}} + (1-\alpha)(\ell_{t-1} + b_{t-1}) \\ b_t &= \beta(\ell_t - \ell_{t-1}) + (1-\beta)b_{t-1} \\ s_t &= \gamma\frac{y_t}{\ell_{t-1} + b_{t-1}} + (1-\gamma)s_{t-m} \end{aligned}$$

is preferred. Even more variations are possible. By combining all different choices we have seen for the trend and seasonal component we can obtain 9 different exponential smoothing methods. We can choose the trend to be constant or dambed and the seasonal component to be additive or multiplicative, and we can choose to not include those components at all.

Seasonal Autoregressive Integrated Moving Average

In this chapter, we will give an overview over the ARIMA and SARIMA models that we will use for forecasting in our experiments. The material presented here can also be found in [12] and [27]. While exponential smoothing models describe trend and seasonality, ARIMA models describe the autocorrelations in the data. Both are widely used approaches and complement each other. To fully understand ARIMA models, we first need to introduce some concepts, starting with an autoregression model.

Definition 2.1.7 (Autoregressive Series) *A time series y_t that satisfies $y_t = c + \phi_1 y_{t-1} + ... + \phi_p y_{t-p} + \epsilon_t$, where ϵ_t is white noise, ϕ_u are parameter coefficients and c is a constant, is called an autoregressive series of order p, for short, AR(p).*

The next value observed in the time series is a linear combination of the most recent observations. This resembles classical regression, with the difference that it regresses the variable against itself, i.e. for prediction, the p last values of the time series itself are used. Instead of using past values of the forecast, a moving average model uses past forecast errors.

Definition 2.1.8 (Moving Average Series) *A time series y_t that satisfies $y_t = c + \epsilon_t + \theta_1 \epsilon_{t-1} + ... + \theta_q \epsilon_{t-q}$, where ϵ_t is white noise, θ_u are parameter coefficients and c is a constant, is called a moving average series of order q, MA(q).*

In particular, the forecast for a time series of type MA(q) uses a weighted average of the forecast errors in the last q time steps. Moving average models and stationary autoregressive models are related in the sense that a stationary model of type AR(p) can be rewritten as a MA(∞) model [27]. The next model combines both of the previous models.

Definition 2.1.9 (Autoregressive Moving Average Series - ARMA) *A time series y_t that satisfies $y_t = \phi_1 y_{t-1} + ... + \phi_p y_{t-p} + \epsilon_t + \theta_1 \epsilon_{t-1} + ... + \theta_q \epsilon_{t-q}$, is called an autoregressive moving average series of order (p, q), ARMA(p, q).*

Clearly, the first p terms are taken from the autoregressive model, while the last q terms belong to the moving average model. ARMA models cannot be applied on non-stationary data. As mentioned before, a non-stationary time series can become stationary by differencing. Integrating differencing into the ARMA model leads us to ARIMA models, which are applicable also on non-stationary time series. For that reason, we now describe the process of differencing in more detail. First, we introduce the backshift operator in order to simplify the notation.

Definition 2.1.10 (Backshift operator) *The backshift operator B applied to a time series y_t shifts the data back one time step, i.e. we have $By_t = y_{t-1}$*

Using this notation, we introduce differencing.

Definition 2.1.11 (Differencing) *Given a time series y_t, the differenced time series is given by the differences between consecutive time steps, i.e. $y'_t = (1 - B)y_t$. We may perform differencing more than once on a time series, the result being a higher order differenced time series. More precisely, for any $k \geq 1$, the k-th order differenced time series is obtained as $y_t^{(k)} = (1 - B)^k y_t$*

Alternatively, it can be be useful to apply seasonal differencing to remove seasonality from a time series.

Definition 2.1.12 (Seasonal differencing) *The seasonal difference of a time series y_t is the difference between the current time value and the value from the last season. If the season returns with a period m, then the seasonal difference is obtained by applying the operator $(1 - B^m)$ to y_t.*

In practice, it is rarely necessary to perform differencing more than twice, and might even cause information loss when overdifferencing.

The ARIMA (AutoRegressive Integrated Moving Average) model incorporates differencing, moving average and autoregressive models into one model. Afterwards, we will introduce a version of this, which in addition uses seasonal differencing, making it suitable also for seasonal time series.

Definition 2.1.13 (Autoregressive Integrated Moving Average - ARIMA) *The essential idea of this series is to first apply differencing and then the combination of autoregression and moving average that we have already seen in ARMA. In detail, for ARIMA, we have*

$$\left(1 - \sum_{i=0}^{p} \phi_i B^i\right)(1-B)^d y_t = c + \left(1 + \sum_{j=0}^{q} \theta_j B^j\right)\epsilon_t$$

We abbreviate this model by ARIMA(p, d, q), where p is the order of autoregression, q the order of the moving average model that we use, and d is the degree of differencing that is applied.

A further extension of ARIMA is a more sophisticated model, that introduces seasonal autoregressive and moving average components, making it suitable for seasonal data. It also introduces seasonal differencing, allowing for more flexibility in stationarizing the time series.

Definition 2.1.14 (Seasonal Autoregressive Integrated Moving Average - SARIMA) *For the SARIMA model, we have seven parameters, (p, d, q) for the non-seasonal part (as before), and (P, D, Q) and m for the seasonal part. The first six parameters have the same purpose as before, the additional parameter m for the seasonal part determines the seasonal differencing. In formulas, this means that*

$$\left(1 - \sum_{i=0}^{p} \phi_i B^i\right) \cdot \left(1 - \sum_{i=0}^{P} \Phi_i B^{im}\right) \cdot (1-B)^d \cdot (1-B^m)^D y_t = c + \left(1 + \sum_{j=0}^{q} \theta_j B^q\right) \cdot \left(1 + \sum_{j=0}^{q} \Theta_j B^{jm}\right)\epsilon_t$$

SARIMA captures both seasonal and non-seasonal patterns in the data through a combination of autoregression, differencing, and moving averages. From now on we will simply refer to both models as ARIMA, and state wether the seasonal variant is used.

2.2 Hierarchical Time Series

Only a few real-life situations can be fully captured by a single time series. In most cases, multiple time series are observed instead, following a hierarchical structure across various levels.

For instance, during the COVID-19 pandemic, infection rates were recorded for each city and then aggregated to higher levels, such as regions or countries, to monitor and better understand how the virus spread. Governments and health authorities had to rely on multi-level forecasts to make critical decisions, such as implementing lockdowns and distributing medical resources. The accuracy and consistency of forecasts across these levels became essential, as each level influenced policies that directly impacted public health and safety.

Such types of time series, which follow a hierarchical structure, are called hierarchical time series. The hierarchical structure does not necessarily emerge geographically. Instead, it can also be abstract and represent, e.g., product categories in sales. Forecasting hierarchical time series has gained a lot of attention in recent years, as hierarchies in time series are often occurring. The main challenge in forecasting hierarchical time series is that the forecasts at different levels will generally not align with the hierarchical structure of the time series. If a time series is the aggregate of two other time series, i.e $Z = X + Y$ the forecasts will almost never align $\hat{Z} \neq \hat{X} + \hat{y}$. We will start by formally defining hierarchical time series.

Definition 2.2.1 (Hierarchical Time Series) *A n-dimensional time series $y_1, ...y_T$ that adheres to some known linear constraints is called hierarchical time series (HTS). Let $y_t \in \mathbb{R}^n$ be the vector of observations of all time series in the hierarchy at time t and let $y_{i,t}$ be the value of the time series at node i and at time t. Let $b_t \in \mathbb{R}^{n_b}$, $n_b < n$ be the vector of the most disaggregated series at time t, called bottom-level series, and let a_t be the vector of the $n_a = n - n_b$ aggregated time series. It must hold, that $a_t = Ab_t$, where A is an $n_a \times n_b$ aggregation matrix. Furthermore, $y_{Tot,t}$ denotes the total of the time series at the top level at time t.*

There are two possible representations of a hierarchical time series: The structural representation and the zero constraints representation. In the structural representation the full set of time series $y_t = \begin{bmatrix} a_t \\ b_t \end{bmatrix}$ can be obtained by $y_t = Sb_t$, where $S = \begin{bmatrix} A \\ \mathbb{I}_{n_b} \end{bmatrix}$. The matrix S is called summing or structural matrix. This representation illustrates how the data at lower levels aggregates to higher levels. In the alternative zero constraints representation the aggregation structure is represented with a constraint matrix C such that $Cy_t = 0$. The matrix may not be full rank. Starting from the structural representation one can obtain the constraint matrix by setting $C = \begin{bmatrix} \mathbb{I}_{n_a} & -A \end{bmatrix}$. The series a_t do not necessarily have to be aggregates of b_t and can include any linear combination of the bottom-level series. Therefore, there is no requirement for S, A or C to only contain 0s or 1s and they can include any real values that specify linear constraints on the time series.

The goal when forecasting hierarchical time series is that the forecasts should fulfill the same linear constraints as the observed data. Forecasts with this property are called coherent.

Definition 2.2.2 (Coherence) *The linear subspace of $\mathbf{R}^n$ that contains all forecasts for which the linear constraints hold, is called the coherence subspace. It is spanned by the columns of S and denoted by $\mathcal{S}$. A set of h-step-ahead point forecasts $\hat{y}_{T+h|T} \in \mathbf{R}^n$ are coherent if $\hat{y}_{T+h|T} \in \mathcal{S}$.*

Following this, one could define hierarchical time series as an n-dimensional multivariate time series, such that all observed values $y_1, ...y_T$ and all future values $y_{T+1}, y_{T+2}, ...$ lie in the coherent subspace, i.e. $y_t \in \mathcal{S}$ for all t. According to this definition hierarchical time series, do not necessarily have to follow a hierarchy, and as stated before, do not have to adhere to aggregation. Alternatively, the series could be grouped, without strict level order, have temporal hierarchies, overlapping temporal hierarchies, or a mixture of the mentioned [32]. However, our focus will lie on strictly hierarchical time series, formed by aggregation.

It is possible to add a statistical discrepancy term to the aggregation constraints that captures possible inaccuracies from the process of recording the data. In that case, the full set of time series would be obtained by

$$y_t = Sb_t + \delta_t.$$

In the following we will disregard such discrepancy term.

The hierarchical time series problem describes the goal of finding forecasts that are coherent. In the best case those forecasts also lead to an increased accuracy.

2.2.1 Single-Level Approaches

The traditional and most intuitive approach for dealing with hierarchical time series is to select one level of the hierarchy, generate forecasts for that level, and use those forecasts to construct also forecasts for the other levels by linearly combining them [6].

Definition 2.2.3 (Bottom-Up) *In the bottom-up approach, forecasts for the bottom-level series $\hat{b}_{T+h|T} \in \mathbf{R}^{n_b}$ are first generated. Then a set of coherent forecasts are obtained by*

$$\tilde{y}_{T+h|T} = S\hat{b}_{T+h|T}.$$

The advantage of bottom-up is, that no information from the most disaggregated series is lost. At the same time, bottom-level series might be noisy and challenging to forecast [6]. In that case, the aggregated forecasts will have decreased accuracy in comparison to forecasting immediately at this level.

Definition 2.2.4 (Top-Down) *In the top-down approach, forecasts for the top-level series $\hat{y}_{Tot,T+h|T} \in \mathbf{R}$ are first generated. Then a set of coherent forecasts are obtained by*

$$\tilde{y}_{T+h|T} = Sp\hat{y}_{Tot,T+h|T},$$

where $p = (p_1, ..., p_{n_b})$ is a vector of proportions, that disaggregate the top-level forecasts for the bottom-level series, which are then aggregated up by the summing matrix S to obtain the remaining $n_a - 1$ forecasts. It holds that $p\hat{y}_{Tot,T+h|T} = \hat{b}_{T+h|T}$ [19].

Assuming that the total-aggregate series is the least challenging to forecast, this approach leverages the accurate forecasts from the top-level. Nonetheless, due to the loss of information, disaggregation might be challenging. The proportions for disaggregation can be chosen in multiple ways. Traditionally, the proportions of observed historical data have been utilized, which requires to forecast only the top-level time series, but which does not capture characteristics of the lower-level series. Instead, Athanasopoulos, Ahmed, and Hyndman (2009) proposed to disaggregate based on the proportions of forecasts outperforming the alternative disaggregation approaches [2]. Panagiotelis et al. (2021) showed that a remaining limitation of top-down methods is that they can introduce bias [32].

Finally, a combination of the two approaches can be taken.

Definition 2.2.5 (Middle-Out) *In the middle-up approach, one middle-level is selected and forecasts on that level are generated. For lower-level series top-down is applied, and for higher-level series bottom-up.*

Since single-level approaches mainly utilize information from one level of the hierarchy, the accuracy on the other levels strongly depends on the performance of the forecasts of the selected level. Furthermore, they do not take advantage of the fact that multiple time series are available, which could theoretically have the potential to increase also accuracy in addition to becoming coherent.

Until recently, those were the sole methods that existed for obtaining coherent hierarchical time series forecasts and the initial research's focus was on comparing those methods with each other [4]. There are plenty of research papers, comparing those methods for a variety of applications. The findings favor bottom-up approaches, but it also becomes apparent that it strongly depends on the use case and data. Dangerfield and Morris (1992) made an interesting observation in their study, where bottom-up forecasts were more accurate, especially when the two bottom-level series were highly correlated [15]. In contrast, Huddleston, Porter, and Brown (2015) argue that top-down methods are better suitable for noisy data on the lower levels. In conclusion, a modeler needs to take different methods into consideration and compare them.

2.2.2 Forecast Reconciliation

A rather new concept, is forecast reconciliation, which tackles the limitations of single-level approaches. The general idea is easy: Given independently generated forecasts for all time series in the hierarchy, how can they subsequently be adjusted to make them coherent?

Definition 2.2.6 (Forecast Reconciliation) *Let $\hat{y}_{T+h|T} \in \mathbb{R}^n$ be a vector of, so called, base forecasts for all time series in the hierarchy, stacked in the same order as y_t, and let $G \in \mathbb{R}^{n_b \times n}$ be a matrix that maps $\hat{y}_{T+h|T}$ to $\mathbb{R}^{n_b}$. Then*

$$\tilde{y}_{T+h|T} = SG\hat{y}_{T+h|T}$$

is called reconciled forecasts [39].

If SG corresponds to a mapping from $\mathbb{R}^{n_b}$ to $\mathcal{S}$, then the reconciled forecasts are coherent [32]. SG denotes here a two step approach, where we create first reconciled bottom-level forecasts, as a weighted combination of all base forecasts, and then obtain the aggregate series, through aggregation with the mapping matrix S. Obviously, the single-level approaches can also be formulated as reconciliation problems by setting $\mathbf{G} = \begin{pmatrix}\mathbf{0}_{(n_b \times n - n_b)} & \mathbf{I}_{n_b}\end{pmatrix}$ and $\mathbf{G} = \begin{pmatrix}p & \mathbf{0}_{(n_b \times n-1)}\end{pmatrix}$.

Let

$$\hat{e}_{t+h|t} = y_{t+h} - \hat{y}_{t+h|t},$$
$$\tilde{e}_{t+h|t} = y_{t+h} - \tilde{y}_{t+h|t}$$

denote the h-step-ahead in-sample base forecast errors, and the h-step-ahead reconciled forecast errors, for $t = 1, 2, \ldots, T - h$, respectively.

The following approach is based on the intuition that larger variance of the base forecast errors, will permit larger range of adjustments for forecast reconciliation [39].

Minimum Trace Reconciliation

Wickramasuriya, Athanasopoulos, and Hyndman (2019) formulated the reconciliation problem, as a minimization problem of

$$\mathrm{Var}(\tilde{e}_{t+h|t})$$

which is the h-step-ahead covariance matrix of the reconciled forecast errors [41]. Based on the assumption that the base forecasts are unbiased with the additional constraint to only obtain unbiased reconciled forecasts, a unique solution

$$G_h = \left(S' W_h^{-1} S\right)^{-1} S' W_h^{-1}$$

exists, where W_h is the positive definite covariance matrix of the h-step-ahead base forecast errors [39]. Despite having a unique solution, the challenge here lies in estimating W_h, especially for $h > 1$. A simple assumption is $W_h = k_h W_1$, $\forall h$, where $k_h > 0$, which yields a constant solution of G_h in regards of the forecast horizon h. Table 2.1 lists the most popular estimators for W_h, where $\widehat{W_1}$ denotes the unbiased covariance estimator based on the in-sample one-step-ahead base forecast errors (i.e., residuals), diag$(\cdot)$ constructs a diagonal matrix using a given vector and Diag$(\cdot)$ using the diagonal elements of the input matrix [39].

Table 2.1 Forecast reconciliation methods for different estimators of W, taken from [39].

Reconciliation method	$W_h \propto$
OLS [30]	I
WLSs [5]	diag$(S1)$
WLSv [29]	Diag$(\widehat{W_1})$
MinTs [41]	$\lambda\,\mathrm{Diag}(\widehat{\mathbf{W_1}}) + (1-\lambda)\widehat{W_1}$

All the above mentioned estimators, resulting to different reconciliation methods, are imposing an unbiasedness condition on the base forecasts and reconciled forecasts. Furthermore, *OLS* and *WLSs* do not utilize the data when estimating W_h. Instead *WLSs* uses structural information from the summing matrix.

Empirical Risk Minimization

Ben Taieb and Koo (2019) relaxed the unbiasedness assumption and formulated the reconciliation problem as a regularized empirical risk minimization problem by expanding the training window incrementally, one observation at a time. The objective is

$$\min_{G_h} \frac{1}{(T - T_1 - h + 1)n} \|Y_h^* - \tilde{Y}_h^* G_h^* S'\|_F^2 + \lambda \|\operatorname{vec}(G_h)\|_1,$$

where T_1 denotes the minimum number of observations used for model training, $\|\cdot\|_F$ is the Frobenius norm, $\|\cdot\|_1$ is the L_1 norm, $\operatorname{vec}(\cdot)$ denotes the vectorization of a matrix,

$$Y_h^* = [y_{T_1+h}, \ldots, y_T]', \quad \tilde{Y}_h^* = [\hat{y}_{T_1+h|T_1}, \ldots, \hat{y}_{T|T-h}]',$$

and $\lambda \geq 0$ is a regularization parameter. When $\lambda = 0$, the problem reduces to an empirical risk minimization (ERM) problem without regularization [9].

A variety of methods have been developed over the last couple of years, including cross-temporal methods, probablistic approaches and also machine learning models with regularization. Athanasopoulos et al. (2024) provides an extensive review over the existing methods [4]. In this study we will focus on the single-level approaches, on all four presented variants of Minimum Trace Reconciliation, as well as Empirical Risk Minimization (ERM).

2.3 Clustering

We will introduce now three concepts that we will need later when clustering the time series.

Principal Component Analysis

Principal Component Analysis (PCA) is a widely used method for dimensionality reduction. Bishop (2006) defines PCA in two ways, which are shown to be equivalent [10]:

1. "The orthogonal projection of the data onto a lower dimensional linear space, known as the principal subspace such that the variance of the projected data is maximized."
2. "The linear projection that minimizes the average projection cost, defined as the mean squared distance between the data points and their projections."

The direction of the linear projection that satisfies the above definitions can be shown to be the eigenvector of the data covariance matrix corresponding to the largest eigenvalue. This eigenvector is called the first principal component.

Additional principal components can be obtained by taking the eigenvectors corresponding to the next largest eigenvalues, though each additional principal component will explain less variance than the previous one. Thus, we can reduce the dimensionality of our data while minimizing the loss of information by

$$\tilde{\mathbf{X}} = \mathbf{X}\mathbf{U}.$$

Where $\mathbf{X}$ is the matrix of our original data as row vectors and $\mathbf{U}$ is the PCA matrix consisting of the first n principal components as column vectors.

K-Means clustering

Bishop (2006) defines clustering intuitively as a partitioning of the data into some number of clusters, where a cluster can be seen as "a group of data points whose inter-point distances are small compared to the distances to points outside the cluster" [10].

K-Means clustering is a well-known clustering method that formulates this problem as a minimization of the sum of squared differences between each data point x_n and a control point defining its assigned cluster, μ_k. The objective function is

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} ||x_n - \mu_k||^2$$

where $r_{nk} \in \{0, 1\}$ is an indicator variable such that $r_{nk} = 1$ if x_n is assigned to cluster k and $r_{nj} = 0$ for $j \neq k$. This measure we are minimizing is sometimes also called the Within-Cluster Sum of Squares (WCSS).

The problem of minimizing J jointly over r_{nk} and μ_k is known to be NP-Hard, so it is common to approximate a solution using the expectation maximization (EM) algorithm [1]. The EM algorithm alternates holding one of the variables constant and optimizing the other until some measure of convergence is achieved.

K-Medoids clustering

K-Means clustering has several weaknesses. Among these is that K-Means is specifically defined to use a (squared) Euclidean metric, which limits the type of data that can be clustered and can make the problem nonrobust to outliers.

K-Medoids is a generalization that allows any dissimilarity metric $\mathcal{V}$ to be used and has the objective function

$$\tilde{J} = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \mathcal{V}(x_n, \mu_k).$$

K-Medoids is also trained using the EM algorithm (Bishop, 2006).

2.4 U.S. Electricity Grid and Balancing Authorities

To provide a comprehensive understanding of the dataset and its structure, we begin with an overview of the electricity grid in the United States, emphasizing the role of Balancing Authorities.

The electricity grid in the United States is a complex high-voltage transmission network that moves electricity quickly over large areas and connects 145 million customers. Since electricity cannot be stored in large amounts, it must be produced in real-time at power plants. The generated electricity is transported through substations, transformers, transmission lines, and distribution lines [38].

The U.S. electric grid consists of

- over 7,300 power plants,
- nearly 160,000 miles of high-voltage power lines, and

- millions of miles of low-voltage power lines and distribution transformers [37].

At the highest level, the lower 48 states are covered by three major interconnections [38]. They connect local grids and thereby ensure the reliability of the power system by providing multiple routes for power to flow and allowing generators to supply electricity to many load centers. The interconnections themselves exchange power only limited. Furthermore, each interconnection consists of multiple balancing authorities (BAs) of various sizes.

1. The Eastern Interconnection covers the east of the Rocky Mountains and a portion of North Texas and consists of 31 balancing authorities in the United States and 5 in Canada.

2. The Western Interconnection covers the west of the Rocky Mountains and consists of 34 balancing authorities in the United States, two in Canada, and one in Mexico.

3. The Electric Reliability Council of Texas (ERCOT) consists of only one balancing authority and covers most of Texas [37].

Balancing authorities manage the actual operation of the electric system to ensure reliable power delivery to customers, including homes, businesses, and industries across America. Balancing authorities operate under the mandatory reliability standards established by the North American Electric Reliability Corporation (NERC) and approved by the U.S. Federal Energy Regulatory Commission (FERC) [24]. Each balancing authority ensures that electricity generation matches consumption in real-time, that frequencies are within limits, and that costs are minimized by optimizing the use of different generating units. Balancing authorities can also exchange electricity with each other. There are more than 60 balancing authorities in the U.S., each overseeing a specific geographic area, which can sometimes have overlapping borders [22]. These areas, also referred to as Control Areas, are monitored by the balancing authorities. Typically, these regions encompass the retail service territories of multiple electric power utilities. Furthermore, each control area is interconnected with its neighbors, with the purpose of faciliated emergency support, coordinated operations, and power trading [16].

Electricity flows freely within an interconnection, following the laws of physics rather than fixed transmission paths. Transfers between balancing authorities are managed by adjusting generation output and not by switching physical lines. However, electricity does not flow freely between interconnections. Special conversion ties connect them, preventing disturbances from spreading [34].

North American electric systems operate on 60 Hz alternating current, with frequency as the primary indicator of the balance between electricity supply and demand. Deviations from 60 Hz signal imbalances—higher frequency means excess supply, while lower frequency indicates insufficient supply. Significant deviations can cause system failures, equipment damage, or blackouts. An imbalanced system can lead to a grid collapse and extended outages and therefore, it is crucial to ensure the balance of electricity demand and supply. Balancing authorities may initiate rolling blackouts in severe cases to prevent total system failure [34].

A balancing authority can be a utility, a Power Market Administrator (PMA), a Regional Transmission Organization (RTO), or an Independent System Operator (ISO) [22]. Independent System Operators (ISOs) and Regional Transmission Organizations (RTOs) are groups of utilities that formed regional entities. Following FERC's initiative to promote independent grid operation, they were established as neutral market facilitators. They do not own transmission or generation infrastructure, nor do they regulate retail electricity prices. The key difference between ISOs and RTOs is that RTOs meet additional FERC requirements related to regional transmission planning and oversight, while ISOs primarily focus on grid operation and market administration. In regions without an ISO or RTO, particularly in the Southeast and the West, utilities fulfill these functions within their own service territories. Currently, three ISOs and four RTOs operate within the U.S. [24].

To ensure reliable power at the lowest cost within their service area, balancing authorities dispatch generation by first using the least expensive generator. The dispatch process consists of two phases: the day-ahead commitment dispatch and real-time system dispatch [22].

- **Day-ahead commitment:** The balancing authorities schedule generation for the following 24-hour period by forecasting electricity demand and deciding which generation units should commit to be activated for each hour. They consider the minimum running times and flexibility of the generating units, as well as factors such as weather, historical energy consumption data, and economic activity [24].
- **Real-Time System Dispatch:** The actual load will differ from the forecasted demand. Therefore, the balancing authorities must adjust the dispatch from the day-ahead commitment in real time and minimize the overall costs. Demand, generation, and interchange must continually be balanced. The real-time system dispatch is typically controlled automatically [24].

3 Hierarchical Time Series Forecasting for Electricity Demand

This chapter is dedicated to conducting experiments in order to answer our two main research questions. Firstly, we would like to assess the performance of various hierarchical reconciliation techniques within the context of U.S. electricity systems. Secondly, we would like to perform a clustering, obtaining a new hierarchical structure on our time series, and compare the forecasting results with the original hierarchical structure.

We begin by presenting the components and features of our dataset, and the preprocessing steps we performed before starting the experiments. After that, we describe how the experiments were set up on a technical level, and we also explain how the experiments were evaluated. We then describe the experiment to tackle our first research question, using ETS and ARIMA with various hierarchical reconciliation techniques to perform forecasting. For our second research question, we first give a mathematical introduction to outline the research idea and then start our experiment. We describe the clustering procedure that we employed and illustrate the resulting hierarchical structure. Finally, we conclude the chapter by displaying the results of our experiments, evaluating the forecasting accuracies of the different models for the first experiment, and showing that our new hierarchical structure improves the bottom level forecasts in the second experiment.

3.1 Dataset

3.1.1 Description

As described in Section 2.4, the U.S. electricity grid is managed by a network of balancing authorities (BAs), each responsible for maintaining the balance between electricity supply, demand, and interchange within their areas. The geographical structure of these authorities — from national and regional levels to individual balancing authorities — provides a natural framework for hierarchical time series forecasting.

The U.S. Energy Information Administration (EIA) is a federal agency within the U.S. Department of Energy that collects and provides independent statistics and analysis on energy production, consumption, and trends to inform policymakers, businesses, and the public. The EIA 930 dataset is one of their publicly available resources offering high-frequency data on electricity operations across the U.S. It is collected directly from the electricity balancing authorities that operate the grid. EIA provides an online dashboard for a basic dataset overview, an API, and downloadable CSVs. Various other datasets and general information material about energy in the U.S. are also available [34].

Form EIA-930 data attempts to represent a physical picture of net generation, demand (or load), and interchange on the U.S. electric grid.

- **Net generation** is a metered value of the output of generating units within one balancing authority.
- **Interchange** is the physical flow of electricity over the transmission lines (called tie lines) between the electric systems of neighboring balancing authorities.
- **Demand** is derived by subtracting actual interchange from net generation.

Electricity that is used to store energy (e.g., via pumped storage or batteries) is excluded from the reported actual demand and instead included as a negative contribution to net generation.

The electricity data is captured at multiple levels of aggregation — from individual balancing authorities to broader regional groupings formed geographically. The total aggregate gives the total electricity consumption of all the lower 48 states of the U.S. Table 3.2 shows all balancing authorities with their corresponding regional grouping.

Table 3.1 Generation-only balancing authorities sorted by region

BA Code	BA Name	Time Zone	Region Code	Region Name
YAD	Alcoa Power Generating, Inc. - Yadkin Division	Eastern	CAR	Carolinas
AVRN	Avangrid Renewables, LLC	Pacific	NW	Northwest
GRID	Gridforce Energy Management, LLC	Pacific	NW	Northwest
GWA	NaturEner Power Watch, LLC	Mountain	NW	Northwest
WWA	NaturEner Wind Watch, LLC	Mountain	NW	Northwest
SEPA	Southeastern Power Administration	Central	SE	Southeast
DEAA	Arlington Valley, LLC	Arizona	SW	Southwest
HGMA	New Harquahala Generating Company, LLC	Arizona	SW	Southwest

The data is submitted hourly - no later than within 60 minutes of the end of the operating hour - and daily by 7:00 a.m. eastern prevailing time. The hourly data consists of the actual demand in megawatts by each hour's ending time rounded to the nearest integer and their corresponding date-time stamps using the coordinated universal time (UTC). Unless stated otherwise, we will use UTC for any date and time information in this thesis. Once a day, additional data is reported, such as net generation by energy source, day-ahead demand forecast, and metered tie-line flow [17].

Commercial arrangements such as pseudo ties and dynamic scheduling can alter the numbers for demand, net generation, and energy flows compared to their physical measurements. Reported demand might include power used outside a balancing authority's boundaries or exclude some power dispatched within the system by other units. Similarly, net generation may count power from generators outside the balancing authority or leave out power from units controlled by other systems. Energy transfers may also be reported between balancing authorities that are not directly interconnected. Metered values of balancing authorities are normally adjusted to incorporate these commercial arrangements. Official reports, such as those for EIA-930, use only the physical measurements leading potentially to significant differences in calculated demand. Demand forecasts are based on regular operating practices, which include these arrangements, and can make comparisons between forecasted and physically reported demand less meaningful for some balancing authorities [34]. We thus did not compare the reported demand forecasts with our experiment results and only utilized the actual demand.

Furthermore, the quality of the data might be affected by reporting issues of the balancing authorities due to real-time disruptions leading to anomalous or missing data. EIA publishes the raw hourly data as received but performs basic imputations on the daily and aggregated data sets and warns users to be cautious when interpreting the data. Therefore, we will directly use the raw hourly balancing authority data, impute the missing data on our own, and subsequently aggregate the data to obtain the regional and national level series. In order to further assess the quality of the data set, we performed a plausibility check on all known constraints, such as the fact that the reported values for demand, generation, and interchange are expected to balance hourly.

Some balancing authorities' responsibilities lie only in generation, meaning they consist of a power plant or group of power plants and do not directly serve retail customers. Therefore, they do not report demand or demand forecasts; they only report net generation and interchange. Table 3.1 lists all generation-only balancing authorities. Furthermore, eight balancing authorities report demand additionally disaggregated

by subregions, which can be load zones, weather zones, or local balancing authorities within their territory. All of them lie in separate geographical regions. Occasionally, the electric system of a balancing authority is incorporated into another system, leading to their retirement. We did not consider all balancing authorities in the experiments and excluded

- retired balancing authorities,
- generation-only balancing authorities since they do not report demand and
- subregional data since this sub-level occurs maximally only once per region.

The U.S. Energy Atlas is another service the EIA provides, which offers a geospatial data catalog along with an interface for web map applications. Among the available datasets is a shapefile illustrating the electric power balancing authorities across the lower 48 states. This data is based on the Control Areas information sourced from the Homeland Infrastructure Foundation Level Database (HIFLD) as of April 2022. It is important to note that balancing authorities do not have precisely defined geographical boundaries. Consequently, the shapefiles may overlap between different authorities and may also contain gaps in coverage. While these geographic representations give a general sense of each balancing authority's area of influence, they should not be interpreted as rigid borders. For our analysis, we utilized this dataset primarily to gain a general overview of the locations and sizes of the control areas, as well as for visualization purposes [8].

Table 3.2 Active balancing authorities sorted by region

BA Code	BA Name	Time Zone	Region Code	Region Name	Subregion
BANC	Balancing Authority of Northern California	Pacific	CAL	California	No
CISO	California Independent System Operator	Pacific	CAL	California	Yes
IID	Imperial Irrigation District	Pacific	CAL	California	No
LDWP	Los Angeles Department of Water and Power	Pacific	CAL	California	No
TIDC	Turlock Irrigation District	Pacific	CAL	California	No
CPLE	Duke Energy Progress East	Eastern	CAR	Carolinas	No
CPLW	Duke Energy Progress West	Eastern	CAR	Carolinas	No
DUK	Duke Energy Carolinas	Eastern	CAR	Carolinas	No
SC	South Carolina Public Service Authority	Eastern	CAR	Carolinas	No
SCEG	Dominion Energy South Carolina, Inc.	Eastern	CAR	Carolinas	No
SPA	Southwestern Power Administration	Central	CENT	Central	No
SWPP	Southwest Power Pool	Central	CENT	Central	Yes
FMPP	Florida Municipal Power Pool	Eastern	FLA	Florida	No
FPC	Duke Energy Florida, Inc.	Eastern	FLA	Florida	No
FPL	Florida Power & Light Co.	Eastern	FLA	Florida	No
GVL	Gainesville Regional Utilities	Eastern	FLA	Florida	No
HST	City of Homestead	Eastern	FLA	Florida	No
JEA	JEA	Eastern	FLA	Florida	No
SEC	Seminole Electric Cooperative	Eastern	FLA	Florida	No
TAL	City of Tallahassee	Eastern	FLA	Florida	No
TEC	Tampa Electric Company	Eastern	FLA	Florida	No
PJM	PJM Interconnection, LLC	Eastern	MIDA	Mid-Atlantic	Yes

Continued on next page...

BA Code	BA Name	Time Zone	Region Code	Region Name	Subregion
AECI	Associated Electric Cooperative, Inc.	Central	MIDW	Midwest	No
LGEE	Louisville Gas and Electric Company and Kentucky Utilities Company	Eastern	MIDW	Midwest	No
MISO	Midcontinent Independent System Operator, Inc.	Eastern	MIDW	Midwest	Yes
ISNE	ISO New England	Eastern	NE	New England	Yes
AVA	Avista Corporation	Pacific	NW	Northwest	No
BPAT	Bonneville Power Administration	Pacific	NW	Northwest	No
CHPD	Public Utility District No. 1 of Chelan County	Pacific	NW	Northwest	No
DOPD	PUD No. 1 of Douglas County	Pacific	NW	Northwest	No
GCPD	Public Utility District No. 2 of Grant County, Washington	Pacific	NW	Northwest	No
IPCO	Idaho Power Company	Pacific	NW	Northwest	No
NEVP	Nevada Power Company	Pacific	NW	Northwest	No
NWMT	NorthWestern Corporation	Mountain	NW	Northwest	No
PACE	PacifiCorp East	Mountain	NW	Northwest	No
PACW	PacifiCorp West	Pacific	NW	Northwest	No
PGE	Portland General Electric Company	Pacific	NW	Northwest	No
PSCO	Public Service Company of Colorado	Mountain	NW	Northwest	No
PSEI	Puget Sound Energy, Inc.	Pacific	NW	Northwest	No
SCL	Seattle City Light	Pacific	NW	Northwest	No
TPWR	City of Tacoma, Department of Public Utilities, Light Division	Pacific	NW	Northwest	No
WACM	Western Area Power Administration - Rocky Mountain Region	Arizona	NW	Northwest	No
WAUW	Western Area Power Administration - Upper Great Plains West	Mountain	NW	Northwest	No
NYIS	New York Independent System Operator	Eastern	NY	New York	Yes
SOCO	Southern Company Services, Inc. - Trans	Central	SE	Southeast	No
AZPS	Arizona Public Service Company	Arizona	SW	Southwest	No
EPE	El Paso Electric Company	Arizona	SW	Southwest	No
PNM	Public Service Company of New Mexico	Arizona	SW	Southwest	Yes
SRP	Salt River Project Agricultural Improvement and Power District	Arizona	SW	Southwest	No
TEPC	Tucson Electric Power	Arizona	SW	Southwest	No
WALC	Western Area Power Administration - Desert Southwest Region	Arizona	SW	Southwest	No
TVA	Tennessee Valley Authority	Central	TEN	Tennessee	No
ERCO	Electric Reliability Council of Texas, Inc.	Central	TEX	Texas	Yes

3.1.2 Data Preparation

The downloadable CSV files used in this study are provided on a six-month basis, with data availability starting from July 2015. To avoid potential anomalies associated with the initial COVID-19 period, we focused on data from 2021-01-01 onwards. At the time of data retrieval, June 2024 was not yet completed; therefore, the dataset spans up to 2024-05-31.

First, we removed balancing authorities that were either generation-only or had been retired, resulting in a refined dataset with 53 active balancing authorities. We then thoroughly reviewed the dataset for missing values. Given the nature of time series data, the total number of missing values is less important, but rather their distribution within each time series. In particular, consecutive missing values pose a more significant challenge, as longer gaps are harder to impute accurately without introducing bias into the analysis.

Table 3.3 Number of missing demand values per BA before truncation. Relative specifies the percentage of missing demand values in relation to the total number of rows per BA. Sequences corresponds to the number of blocks of consecutive missing values.

BA	Absolute	Relative	Sequences
SW/TEPC	792	2.692%	2
CENT/SPA	144	0.482%	1
NW/PSCO	120	0.401%	1
FLA/SEC	55	0.184%	3
FLA/JEA	48	0.161%	1
CAL/LDWP	25	0.084%	1
CAR/SCEG	25	0.084%	1
FLA/TEC	24	0.080%	1
FLA/FPL	24	0.080%	1
MIDW/LGEE	20	0.067%	4
MIDA/PJM	1	0.003%	1

Missing demand values were identified for 11 balancing authorities, with missing data accounting for 0.080% of the entire dataset, as can be seen in Table 3.3. A notable balancing authority was SW/TEPC, which exhibited a long gap of 744 consecutive missing values from 2021-02-18 08:00:00 to 2021-03-21 07:00:0 — corresponding to 2.489% of its data. Imputing such a big gap could significantly affect the experimental outcomes.

Table 3.4 Number of missing demand values per BA after truncation. Relative specifies the percentage of missing demand values in relation to the total number of rows per BA. Sequences corresponds to the number of blocks of consecutive missing values.

BA	Absolute	Relative	Sequences
CENT/SPA	144	0.519%	1
NW/PSCO	120	0.433%	1
FLA/SEC	50	0.180%	2
FLA/JEA	48	0.173%	1
SW/TEPC	48	0.173%	1
CAL/LDWP	25	0.090%	1
CAR/SCEG	25	0.090%	1
FLA/FPL	24	0.087%	1
FLA/TEC	24	0.087%	1
MIDA/PJM	1	0.004%	1

To constrain the impact of the imputation method, we excluded this period from the analysis, restricting the dataset to the time frame from 2021-04-01 01:00:00 to 2024-05-31 00:00:00 and can see in Table 3.4 how this changed the distribution of missing values. After truncating the data, the proportion of missing data slightly increased to 0.173% with missing data for 10 of the 53 balancing authorities. However, the longest sequence of consecutive missing values was reduced to six days, corresponding to 0.519% of the affected balancing authority's data — a considerable improvement from the prior 2.489%. This distribution of miss-

ing data is more balanced, and the shorter gaps minimize the risk of imputation bias in our subsequent experiments and analysis.

We used a weighted average of two automatically fitted Exponential Smoothing (ETS) models with additive seasonality for each consecutive gap to handle the missing data. One model was trained on the data preceding the gap, while the second was fitted to the data following the gap. For the former, the post-gap series was reversed, and the model was fitted backward, starting from the end of the series. The gap was imputed using a weighted average of the predictions of both models, leveraging information from both directions of the time series. More precisely, the interpolation is calculated as

$$\text{imputation} = (n - i) \cdot \text{forward} + i \cdot \text{backward} \qquad \text{for } i = 1, ..., n,$$

where n represents the length of the gap. This approach ensures that both the preceding and succeeding data points contribute to the imputation process. Figure 3.1 illustrates two examples of the resulting interpolations.

Finally, we constructed a coherent hierarchical dataset by aggregating the balancing authorities as bottom-level series. These were aggregated according to their geographical relationships to obtain region-level series. For the total-level series, we aggregated all balancing authorities. It is important to note that these aggregated values do not precisely match the resources available on the EIA dashboard due to two primary reasons:

- We excluded certain balancing authorities from our dataset.
- EIA applies its own imputation procedures when generating aggregate values.

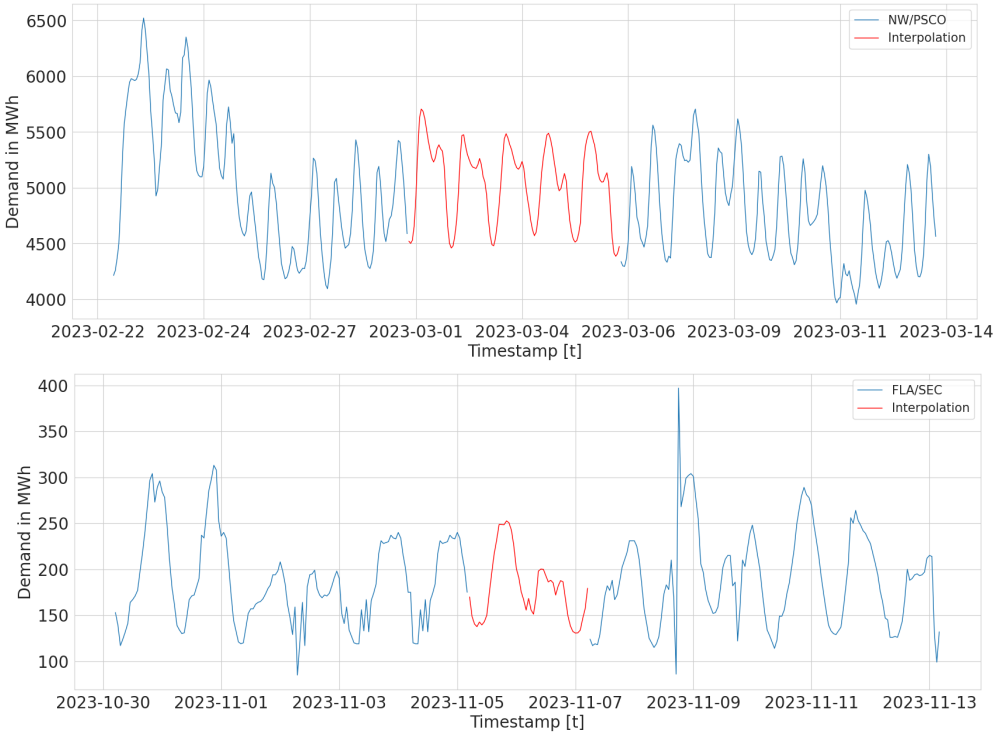

Figure 3.1 Interpolation of the missing demand values of NW/PSCO and FLA/SEC.

3.2 Experimentation Setup

3.2.1 Evaluation Setup

Splitting data is a fundamental step in model evaluation. The primary goal is to ensure that a model is both reliable and generalizable - meaning that the model can perform well on unseen data. If we used the same data for learning and evaluating our model, the model could overfit by learning the data by heart, performing perfectly on the same data set but failing to generalize to new data. Since we aim to predict the unseen future in forecasting, evaluating the model's ability to generalize to data it has not encountered during training is crucial.

When splitting data for model evaluation, there are typically two approaches: the simple train-test split and the more comprehensive train-validation-test split. The train-test split divides the data into a training set used to fit the model and a test set used to evaluate its performance on completely unseen data. It is crucial to separate the test set early on and abstain from performing any modeling or data analysis on it. However, if the process of fitting a model on the train set and evaluating its performance on the test set is repeaed multiple times to increase the model's performance (e.g., by model tuning), this approach can lead to biased performance estimates. In such cases, adding an intermediate validation set is important, as it allows for model selection and hyperparameter tuning without impacting the final performance evaluation.

In this study, we initially divided the data into three sets, namely

- **training set**, with data from 2021-04-01 01:00:00 until 2022-05-31 00:00:00,
- **validation set**, with data from 2022-05-31 01:00:00 until 2023-05-31 00:00:00, and
- **test set**, with data from 2023-05-31 01:00:00 until 2024-05-31 00:00:00.

This was done to maintain the option of having a holdout set for unbiased evaluation if needed. However, since no hyperparameter tuning or model optimization was performed, the test set was finally not required. The parameters of the models were determined analytically and not by iteratively refitting the models to enhance the results. The goal was not to find the best-performing model but to apply different methods on our dataset and compare the performance of their forecasts. This approach eliminates the risk of modeler-induced bias, as we made no subjective adjustments to the model's configurations. Therefore, we did not use the test set and will, from now on, refer to the above mentioned validation set as the test set.

The performance evaluation can highly depend on the way the data is split. The model might predict one specific day of the year very well while failing to do the same for another. To assess the robustness of predictive models, it is thus important to perform cross-validation by testing the model's performance across different subsets of data. In classical cross-validation, data are randomly shuffled and split into k-folds, with each fold serving as a test set while the remaining k-1 folds are used for training, resulting in k-train-test sets. An important assumption for this process is that the data is independent and identically distributed, which evidently does not hold for time series where temporal dependencies exist between observations. Randomly shuffling time series data before splitting it would cause future data to leak into the training set and be used to predict past events. As Hewamalage, Ackermann, and Bergmeir (2023) stated, this would essentially transform the forecasting problem into an imputation problem [21]. Time-dependent cross-validation methods preserve the chronological structure of the data. These methods train models on a subset of the data up to a specific time point and test them on subsequent unseen data [3]. This approach better reflects realistic forecasting scenarios, where models are deployed to predict future events based on past observations.

In this study, we performed time-dependent cross-validation for 365 iterations, corresponding to each day of the year, and used a rolling window approach. We fitted the model on the training set and evaluated its 24-step ahead predictions and its reconciliations against the first 24 values of the test set. Subsequently, we rolled the training window forward by removing the first 24 values from the test set and appending

them to the training set. We repeated this process 365 times, covering the complete validation set. For the train window to remain fixed in size, we also removed the first 24 values of the training set at each step. This experimental design allows a robust assessment of the model's ability to capture seasonal variations, trends, and potential shifts in the data over time. We chose a forecasting window of 24 steps ahead to imitate the operational practices of balancing authorities.

When working with multiple time series that are not entirely independent of each other, data leakage in model training can occur not only within each series but also between different series. If one series contains future information for another series, it could be included in the training set [21]. Although, this does not directly affect the base forecasts, as they are produced independently, we must be cautious during the reconciliation step. Table 3.2 shows that the balancing authorities have different time zones, even within the same regional groupings, as they are spread across the U.S. This means that effects, like sunrise and sunset, are not aligned between all series when using UTC timestamps. An intuitive idea would be to align the series based on their local time zones to leverage common occurring effects. However, this would cause data leakage from the future of some series to the past of others. Thus, we did not realign the series and utilize their local time timestamp converted to UTC. Moreover, this resembles a more realistic situation as balancing authorities must complete their day-ahead planning, based merely on the available information up to this point.

What makes hierarchical time series forecasting special is that it consists of differently scaled time series across multiple levels of aggregation, with each level serving a different forecasting objective. This poses a further challenge, when evaluating hierarchical forecasts, as we need to consider accuracy measures across different levels and scales.

Athanasopoulos and Kourentzes (2023) recommend

- using a variety of metrics due to the multi-objective nature of hierarchical time series,
- applying metrics on different levels of the hierarchy since solely looking at the overall average is not representative,
- assessing metrics over multiple forecast windows (as we did through cross-validation), and
- comparing forecast performance to an appropriate baseline [3].

Due to the game-theoretic perspective of hierarchical time series, there will likely be no model that performs the best across all metrics and levels.

Furthermore, forecasts at different levels may not similarly influence decision-making. For a single balancing authority's operations, accuracy at the lowest level is most significant. However, considering that balancing authorities can exchange electricity with neighboring balancing authorities, the accuracy on a regional level will represent how well balancing authorities can operate together. Finally, the total level can be a measure of general grid stability.

Given these challenges, we utilize a set of well-established accuracy metrics, including Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Root Mean Squared Scaled Error (RMSSE), Mean Absolute Scaled Error (MASE), and Directional Symmetry (DS). Given forecast errors ϵ_t, as defined in Equation (2.1), MAE and RMSE are defined as

$$MAE = \frac{1}{n}\sum_{t=1}^{n}(|\epsilon_t|) \quad \text{and}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(\epsilon_t^2)}.$$

Similarly, MASE and RMSSE are defined as

$$MASE = \frac{1}{n}\sum_{t=1}^{n}(|q_t|) \quad \text{and}$$
$$RMSSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(q_t^2)},$$

where

$$q_t = \frac{\epsilon_t}{\frac{1}{T-m}\sum_{j=m+1}^{T}\left|y_j - y_{j-m}\right|} \quad \text{and}$$
$$q_t^2 = \frac{\epsilon_t^2}{\frac{1}{T-m}\sum_{j=m+1}^{T}(y_j - y_{j-m})^2}$$

are the scaled errors [21]. MAE measures the average absolute forecast error. RMSE penalizes larger errors more than MAE. RMSSE and MASE normalize errors by the in-sample naïve forecast error, i.e. naïve forecast residuals, which makes them robust to scale differences. We used a seasonal naïve version, therefore the scaled error is less than one if it arises from a better forecast than the seasonal naïve forecast residuals, and greater than one if it arises from a worse forecast than the seasonal naïve forecast residuals [26]. Finally, DS is given by

$$DS = \frac{100}{n-1}\sum_{t=2}^{n} d_t,$$

where

$$d_t = \begin{cases} 1 & \text{if } (y_t - y_{t-i})(\hat{y}_t - \hat{y}_{t-i}) > 0 \\ 0 & \text{otherwise.} \end{cases}$$

and evaluates whether the forecast correctly predicts the direction of change and gives us an estimate of the shape [43] .

Scale-dependent metrics, such as MAE and RMSE, are expressed in the same units as the original data. This makes them helpful in understanding absolute error magnitudes. However, they cannot be directly compared across series with different scales. MAE and RMSE help evaluate overall forecast accuracy on an absolute scale. Scale-independent metrics, such as MASE and RMSSE, adjust for differences in scale by normalizing errors using a reference value. Thus, time series across different hierarchies are comparable. Whether scale-dependent or independent metrics are more appropriate for forecasting electricity demand is not apparent. On the one hand, if one forecast error is big for the respective balancing authority's scale, it might still be insignificant relative to the other balancing authority's scale, especially regarding costs. On the other side, one balancing authority that generally has lower electricity flow might also have down-scaled infrastructure. Thus, the robustness of the infrastructure could depend on the scale of the respective balancing authority, and thereby, the scale-independent error would be again more significant. In addition, scale differences become less important when comparing models on the same dataset (such as per level) since all models are evaluated on the same data distribution. In such cases, scale-dependent metrics can still be informative as they still preserve the relative performance ranking of the models. However, if some series have significantly different scales, scale-independent metrics ensure a fairer comparison across hierarchical levels. Therefore, a combination of both types provides a balanced view of forecast performance. We see the necessity of this approach when observing Figure 3.2. Most of the balancing authorities are on a lower scale. Whereas simultaneously, the outliers exhibit a significant scale difference, even higher than many regional aggregates.

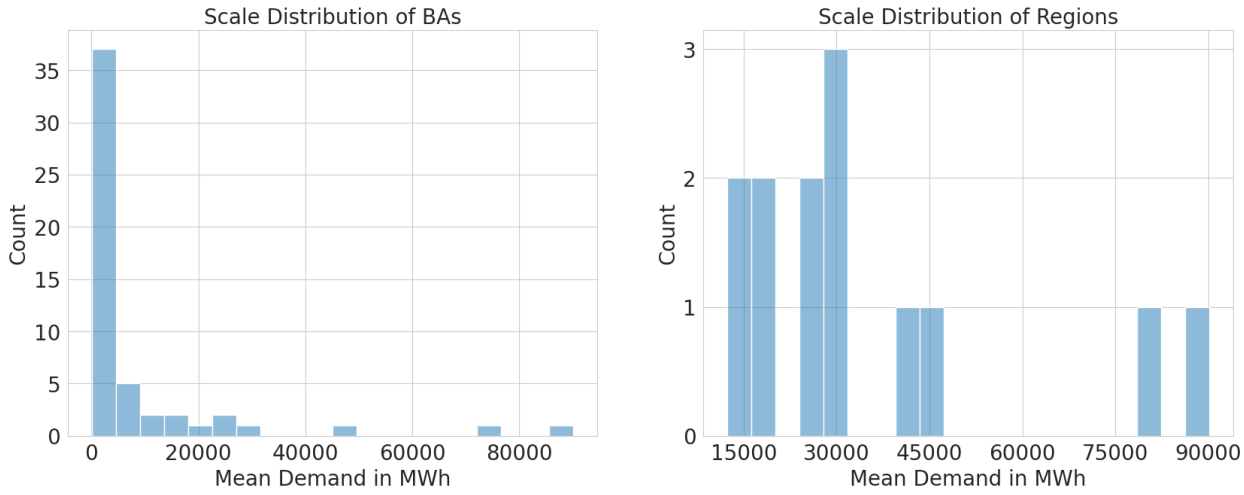

Figure 3.2 Distribution of the scale of all balancing authorities on the left and all regional aggregates on the right.

We refrained from using percentage errors, as they asymmetrically penalize errors and have trouble with values closer to zero, which affects the lower-scaled balancing authorities [31].

For each of the 365 cross-validation steps, we computed all selected metrics over the 24-step forecast horizon, and then we took the average across the 365 steps.

Hierarchical time series forecasting requires careful metric selection due to varying scales and decision-making needs at different levels. We utilized a mix of absolute, scaled, and directional metrics to ensure thorough evaluation. Cross-validation ensured that our evaluation accounts for temporal variations and generalizes across different forecast horizons.

3.2.2 Technical Setup

A broad selection of tools was utilized to conduct the experiments in this study. The necessary code was mainly implemented in Python, with supplementary use of R for specific statistical analyses. Python (version 3.9.19) served as the primary environment for data preparation, modeling, and evaluation. R (version 4.3.3) was used for statistical tests and model selection tasks uniquely offered by the forecast package (version 8.23.0).

We leveraged a vast set of libraries in the Python environment to handle various tasks. For reproducibility, the specific versions of the libraries used are listed in Table 1.

The model training phases and cross-validation on this large dataset were computationally intense; thus, the support of computational resources was needed. We executed the experiments on two computational platforms. The CPU-bound training was primarily conducted on a high-performance cluster provided by inovex GmbH featuring an AMD EPYC Milan Processor with 8 cores, operating at 2.0 GHz, and 32 GB of RAM. Additionally, we utilized Google Colab to speed up training processes by running computations in parallel.

MLflow was employed to streamline and manage the experimental workflow. It facilitated tracking data preparation steps, experiment runs, model configurations, and evaluation metrics. This ensured that the experimentation process was both structured and reproducible. Moreover, a data preparation and imputation pipeline was developed to handle new incoming data and enable easy updates without significant manual intervention. This ensures consistency in preprocessing steps, improving the reliability of the experimental results over time.

Table 3.5 Python libraries and their versions

Library	Version
geopandas	1.0.1
hierarchicalforecast	1.0.1
utilsforecast	0.2.10
mlflow	2.14.2
matplotlib	3.7.3
numpy	1.23.5
pandas	2.2.2
scikit-learn	1.5.0
scikit-learn-extra	0.3.0
scipy	1.13.1
seaborn	0.13.2
statsforecast	1.7.5
statsmodels	0.14.2
tslearn	0.6.3

We also used Screen (version 4.09.01), which is a practical tool for managing multiple terminal sessions on remote servers. This terminal multiplexer allowed us to robustly handle long-running processes during the computational experiments.

3.3 Modeling

The goal of the experiments was to compare two base models, namely ETS and ARIMA, and their reconciliations using various methods with a coherent baseline model.

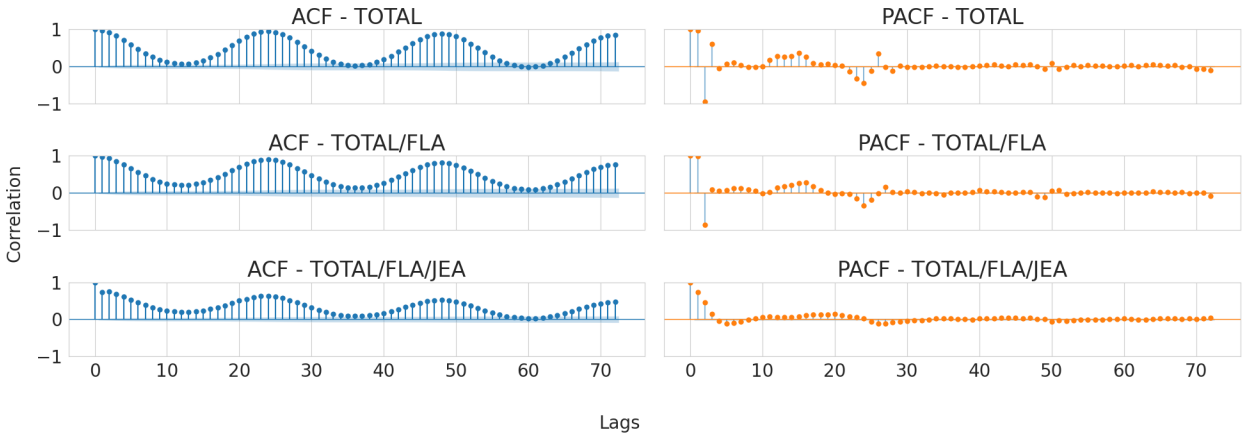

Figure 3.3 ACF and PACF plots for TOTAL, TOTAL/FLA, and TOTAL/FLA/JEA, respectively.

As a preparation, we analyzed the ACF and PACF plots of all time series on all levels. Figure 3.3 shows, as a demonstration, one BA with the respective region and total level series. We observed:

1. All time series show a sinusoidal pattern in the ACF plot with peaks at lag 24, indicating a daily seasonality.

2. The gradual decline in ACF suggests that an autoregressive (AR) component is likely present.

3. The sharp drop after the first few lags of the PACF plot suggests a low-order AR(p) process.

We concluded that all-time series have a daily seasonal pattern. This implies that they are non-stationary. To confirm, we applied a variety of statistical unit-root tests and seasonal unit-root tests, which agreed with our assumption. Furthermore, the tests showed that both a first-order and seasonal differencing stationarize the time series. According to Hyndman and Athanasopoulos (2021), seasonal differencing should be prioritized over first-order differencing Hyndman and Athanasopoulos (2021), because "if first differencing is done first, there will still be seasonality present" [27]. After seasonally differencing the series, all tests agreed that the series are stationary. Figure 3.4 illustrates how mean and variance change over time in the original series and that this behavior vanishes after seasonally differencing. The variability over time of the first-order difference is damped, but still existent.

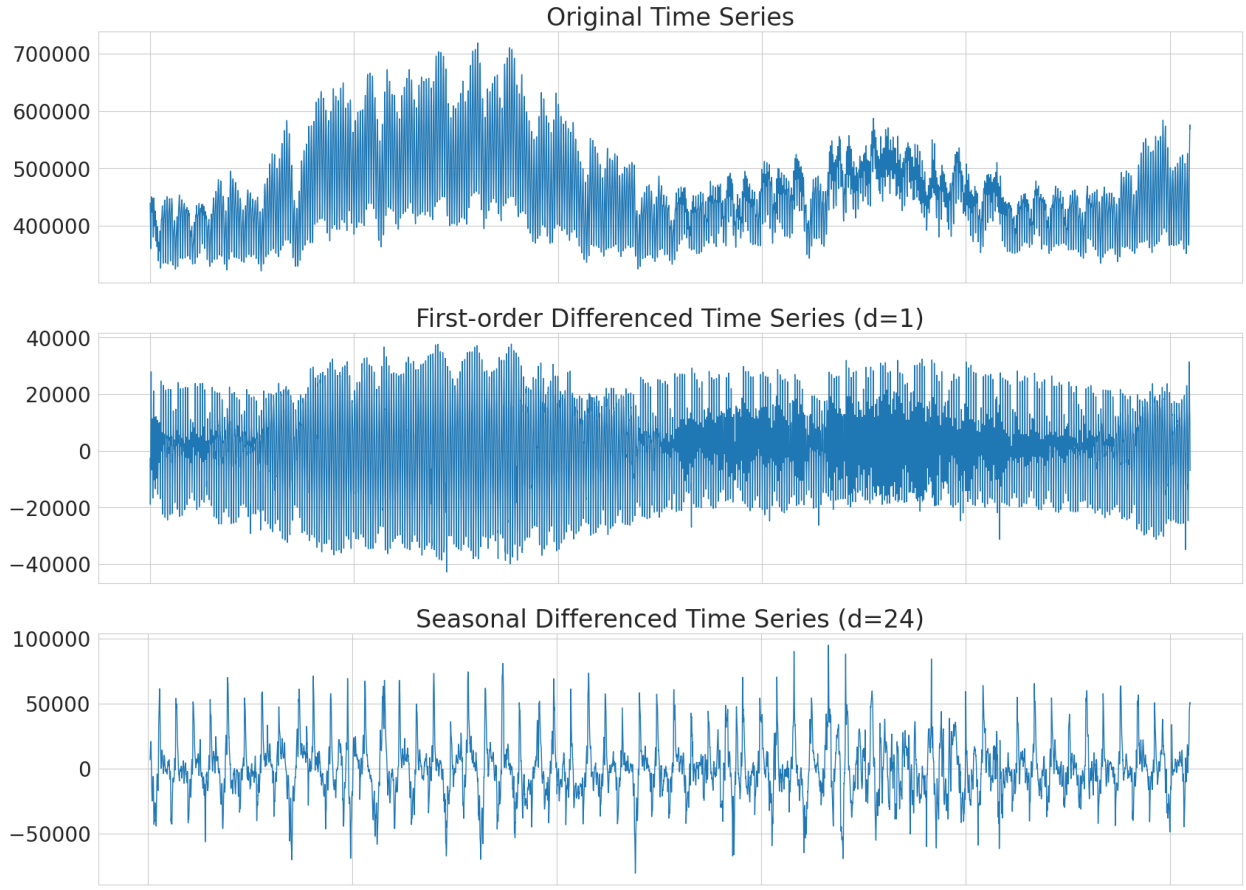

Figure 3.4 First-order and seasonal differencing compared to the original TOTAL series of the training data.

Despite having analyzed the ACF and PACF plots of all time series, they did not yield definitive conclusions in our case to support the parameter selection for ARIMA. To systematically identify the optimal model parameters, we utilized the auto.arima() function from the forecast package in R. This function automates the model selection process by minimizing the Akaike Information Criterion (AIC), which balances model fit and complexity. The AIC is defined as

$$AIC = -2 \ln L + 2k$$

where L is the likelihood of the model, and k is the number of estimated parameters. A lower AIC value indicates a model that achieves a good fit with fewer parameters. As L tends to overfit for short series of length T, a bias-corrected version

$$AICc = AIC + \frac{2(k+1)(k+2)}{T-k}$$

was developed.

The auto.arima() function employs the Hyndman-Khandakar algorithm, which integrates unit root tests, minimization of the corrected AIC (AICc), and Maximum Likelihood Estimation (MLE) to determine the optimal model [28]. The algorithm operates as follows:

1. The algorithm conducts unit root tests and determines the necessary differencing (d and D) to achieve stationarity
2. An initial ARIMA model is selected based on the determined differencing parameters, by comparing 4 initial models and choosing the one with the smallest AIC value.
3. The algorithm performs a stepwise search over possible model configurations, each time evaluating 16 combinations of AR and MA terms for both seasonal and non-seasonal components.
4. Each candidate model is estimated using MLE, and its AICc is computed.
5. The algorithm finishes if no model close to the currently best one is found with a lower AIC.

Further restrictions are imposed by the algorithm, which we did not relax. The values p and q are not allowed to exceed 5, and the values of P and Q are not allowed to exceed 2. By specifying the seasonal period in the auto.arima() function, we ensured that the model selection process accounted for the seasonality present in our data. Furthermore, since we already established that one seasonal differencing suffices to stationarize the series, we restricted the model to use $d = 0$ and $D = 1$.

After identifying the optimal models, we grouped time series with identical model specifications to facilitate parallelized training, thereby reducing computational time. Unique time series were trained individually to ensure accuracy.

For modeling using Exponential Smoothing (ETS), we employed the AutoETS implementation from the statsforecast.models package in Python. Similar to the ARIMA modeling process, we fixed the seasonality and allowed the model to select the optimal configuration for the remaining parameters.

Behind the scenes, AutoETS explores different combinations of ETS models, evaluating configurations based on their likelihood and information criteria, such as AICc. Some combinations of model parameters can lead to unstable models, making them statistically infeasible or impossible to fit. AutoETS automatically excludes these unstable combinations during the selection process to ensure model reliability.

Similar to the ARIMA models, we grouped time series with identical ETS model specifications to parallelize the training process. This grouping significantly reduced computation time. Furthermore, for regions with only one balancing authority, we generated forecasts only for the region and not for the balancing authority since the series are exactly the same.

Following the recommendation of Athanasopoulos and Kourentzes (2023), we chose a baseline model for comparison and decided for the the Seasonal Naïve method. Given the strong daily seasonality of our data, the expectation is that the Seasonal Naïve will already perform quite well as a baseline, posing a challenging benchmark to beat. Based on the examination of the ACF plots, we chose 24 as the seasonal period. Given the nature of this method, it generates already coherent forecasts that do not need to be reconciled.

For the reconciliation of the base forecasts, we chose a variety of methods. We decided to compare all single-level approaches: Bottom-Up, Top-Down, and Middle-Out. For the disaggregation of both Top-Down and Middle-Out, we compared different approaches and found the disaggregation by forecast proportions to yield consistently superior results. Thus, we disregarded the remaining ones. We selected Mint-Shrink, WLS-Struct, Mint-Shrink and WLS-Var among the Minimum Trace methods. In addition, we included Empirical Risk Minimization. The reconciliation was also performed at each cross-validation step.

3.4 Clustering the Hierarchical Structure

In our hierarchical time series data, the regional level is given by the geographical locations of the balancing authorities. For example, balancing authorities in Florida belong to the same region in the data set. Since balancing authorities of the same region are geographically close, they will likely also share a common climate zone. Generally, air temperature is expected to have a strong correlation to electricity demand [7]. Therefore, the expectation is that balancing authorities within the same region have more similar electricity demand patterns than other balancing authorities. We want to examine if this is the case by clustering the time series and comparing if the clustered hierarchy matches the original hierarchical structure that originates from the data set. Are balancing authorities that are geographically close clustered together? Or are potentially balancing authorities on the East and West Coast sharing common electricity demand patterns? Consequently, if the clustered hierarchy differs from the original, an interesting question arises regarding reconciliation. Will reconciling this new hierarchy yield more accurate bottom-level forecasts than the original hierarchy? We focussed here on the bottom-level, since

1. the rest of the hierarchy cannot be directly compared, and
2. according to Athanasopoulos and Kourentzes (2023) improving bottom-level forecasts implicitly supports the whole hierarchy [3].

3.4.1 Theoretical Analysis

We will motivate this research question by analyzing a simplified hierarchical system and will start by defining some necessary notions.

Definition 3.4.1 (Hierarchical Version) *Let an HST with bottom level series b_i, $i = 1, ..., n_b$ be given. An alternative mapping matrix $S^v \in \mathbb{R}^{(1+k+n_b)\times n_b}$ is a mapping matrix that aggregates the same set of bottom level series b_i in a different way than the original summing matrix. Here, the dimensions of the matrix are determined as follows: The top row is the total level, the next k rows are the in-between level series, where $k \geq 0$ can be an arbitrary natural number, and the n_b bottom rows represent the bottom levels. To be more precise, from the definition of a mapping matrix (Definition 2.2.1), we know that S^v satisfies the following properties:*

1. *The first row of S^v is a row vector of 1s*
2. *The last n_b rows of S^v form the identity matrix $\mathbb{I}_{n_b}$*

For an alternative mapping matrix S^v, we obtain the associated hierarchical version by using this mapping matrix for the aggregation structure, i.e. we aggregate by multiplying the bottom level vector b with S^v from the left. For short, we denote this HTS by $\mathcal{H}_v$. The hierarchical version that is originally given by the HTS is referred to as natural hierarchical version and is denoted as $\mathcal{H}_{nat}$.

Note that the number n_a from Definition 2.2.1 corresponds to $1 + k$ here. Clearly, the identity matrix at the bottom of the alternative mapping matrix leaves the bottom level series unchanged. Moreover, the top level series is always the sum of all of the bottom level entries. In other words, the hierarchical version only specifies the aggregation that leads to the k in-between level series. We hence arrive to the following corollary.

Corollary 3.4.2 *For any hierarchical version $\mathcal{H}(S)$ the top level series stays the same. Furthermore, for the Bottom-Up approach, the reconciled forecasts of the bottom and top level are not affected by the hierarchical version.*

To illustrate the theory that we developed above, we turn towards an example, where we analyze the following simple natural hierarchical version

$$\begin{aligned} y_t &= y_{A,t} + y_{B,t} \\ y_{A,t} &= y_{AA,t} + y_{AB,t} \\ y_{B,t} &= y_{BA,t} + y_{BB,t} \end{aligned}$$

and compare it with the alternative hierarchical version

$$\begin{aligned} y_t &= y_{A',t} + y_{B',t} \\ y_{A',t} &= y_{AA,t} + y_{BA,t} \\ y_{B',t} &= y_{AB,t} + y_{BB,t}. \end{aligned}$$

The natural hierarchical version $\mathcal{H}_{nat}$ and the alternative hierarchical version $\mathcal{H}_v$ emerge from the mapping matrices

$$S^{nat} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad S^{v} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

respectively. In both hierarchical versions, we have four bottom levels (meaning $n_b = 4$) and two in-between level series (i.e. $k = 2$) as is visualized in Figure 3.5.

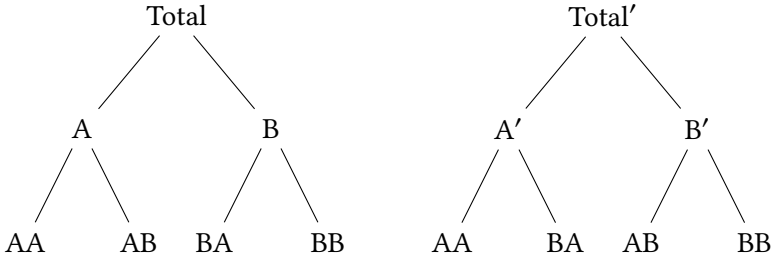

Figure 3.5 On the left side the natural hierarchical version, on the right side an alternative hierarchical version

Assume we are given observations y_t and are interested in the 1-step-ahead forecasts $\hat{y}_t$ and their reconciliations $\tilde{y}_t$ for $t = 1, ..., T$. Specifically we want to analyze the bottom-up and top-down approach and choose the disaggregation based on the forecast proportions. This generally means that for $y_t = y_{A,t} + y_{B,t}$ and $y_{A,t} = y_{AA,t} + y_{AB,t}$ an in-between level node is reconciled as

$$\tilde{y}_{A,T+1} = \left(\frac{\hat{y}_{A,T+1}}{\hat{y}_{A,T+1} + \hat{y}_{B,T+1}} \right) \cdot \hat{y}$$

and a bottom level node as

$$\tilde{y}_{AA,T+1} = \left(\frac{\hat{y}_{AA,T+1}}{\hat{y}_{AA,T+1} + \hat{y}_{AB,T+1}} \right) \cdot \hat{y}_{A,T+1} = \left(\frac{\hat{y}_{AA,T+1}}{\hat{y}_{AA,T+1} + \hat{y}_{AB,T+1}} \right) \cdot \left(\frac{\hat{y}_{A,T+1}}{\hat{y}_{A,T+1} + \hat{y}_{B,T+1}} \right) \cdot \hat{y}_{T+1}.$$

We compare the different reconciliation approaches in four scenarios using two forecasting methods. The first forecasting model is the naive forecast, where we set the forecast for the next time step simply as the

value of the current timestep, i.e., $\hat{y}_{T+1} = y_T$. In addition, we also consider the average forecasting, i.e., setting the forecast to be the average of the observations, i.e., $\hat{y}_{T+1} = \frac{1}{T}\sum_{t=1}^{T} y_t$.

Different forecasting methods may be used on the different levels of the aggregation structure, affecting the resulting reconciliation. In scenario 1, we first consider the simplest case where the same forecasting method is used on all levels. After that, we also consider the cases where the total level is using one way of forecasting and all the bottom and in-between levels are using a different way (scenario 2), as well as the case in which a different forecasting method is used on the bottom level than on the aggregated levels (scenario 3). Finally, we also consider the case where the bottom and total levels use one way of forecasting while only the intermediate levels use a different way.

Scenario 1 - All levels use the same forecasting method

Assume first that all forecasts are given by the average of the observations. Then the bottom level base forecasts are

$$\hat{y}_{AA,T+1} = \frac{1}{T}\sum_{t=1}^{T} y_{AA,t} \qquad \hat{y}_{AB,T+1} = \frac{1}{T}\sum_{t=1}^{T} y_{AB,t}$$
$$\hat{y}_{BA,T+1} = \frac{1}{T}\sum_{t=1}^{T} y_{BA,t} \qquad \hat{y}_{BB,T+1} = \frac{1}{T}\sum_{t=1}^{T} y_{BB,t}$$

and the forecasts of the in-between levels for the natural hierarchical version $\mathcal{H}_{nat}$ are

$$\hat{y}_{A,T+1} = \frac{1}{T}\sum_{t=1}^{T} y_{A,t} = \frac{1}{T}\sum_{t=1}^{T} y_{AA,t} + y_{AB,t} = \frac{1}{T}\sum_{t=1}^{T} y_{AA,t} + \frac{1}{T}\sum_{t=1}^{T} y_{AB,t} = \hat{y}_{AA,T+1} + \hat{y}_{AB,T+1}$$
$$\hat{y}_{B,T+1} = \frac{1}{T}\sum_{t=1}^{T} y_{B,t} = \frac{1}{T}\sum_{t=1}^{T} y_{BA,t} + y_{BB,t} = \frac{1}{T}\sum_{t=1}^{T} y_{BA,t} + \frac{1}{T}\sum_{t=1}^{T} y_{BB,t} = \hat{y}_{BA,T+1} + \hat{y}_{BB,T+1}$$
$$\hat{y}_{T+1} = \frac{1}{T}\sum_{t=1}^{T} y_t = \frac{1}{T}\sum_{t=1}^{T} y_{A,t} + y_{B,t} = \hat{y}_{A,T+1} + \hat{y}_{B,T+1} = \hat{y}_{AA,T+1} + \hat{y}_{AB,T+1} + \hat{y}_{BA,T+1} + \hat{y}_{BB,T+1}.$$

Analogously, the forecasts of the hierarchical version $\mathcal{H}_v$ are

$$\hat{y}_{A',T+1} = \hat{y}_{AA,T+1} + \hat{y}_{BA,T+1}$$
$$\hat{y}_{B',T+1} = \hat{y}_{AB,T+1} + \hat{y}_{BB,T+1}$$
$$\hat{y}_{T+1} = \hat{y}_{A',T+1} + \hat{y}_{B',T+1} = \hat{y}_{AA,T+1} + \hat{y}_{AB,T+1} + \hat{y}_{BA,T+1} + \hat{y}_{BB,T+1}.$$

We make the following observations:

1. The forecasts are already coherent and the bottom-up and top-down approaches are therefore redundant.

2. Since $\tilde{y}_{T+1} = \hat{y}_{T+1}$, the hierarchical version has no effect on the top level and bottom level forecasts.

The same holds if the naive model $\hat{y}_{T+1} = y_T$ is chosen for all forecasts.

Scenario 2 - Forecasting method for the total level differs from the other levels

Assume now that the naive model is chosen for the top level, while the in-between and bottom level forecasts are again chosen as the average of the observations. Then the in-between level forecasts

$$\hat{y}_{A,T+1} = \hat{y}_{AA,T+1} + \hat{y}_{AB,T+1}$$
$$\hat{y}_{B,T+1} = \hat{y}_{BA,T+1} + \hat{y}_{BB,T+1}$$

remain coherent while in general, the top level forecast $\hat{y}_{T+1} = y_T \neq \hat{y}_{A,T+1} + \hat{y}_{B,T+1}$ will now be incoherent. The bottom-up approach yields the reconciled forecasts

$$\tilde{y}_{T+1} = \hat{y}_{A,T+1} + \hat{y}_{B,T+1} = \frac{1}{T}\sum_{t=1}^{T} y_t$$

for the top level with forecast error $\epsilon_{T+1} = y_{T+1} - \tilde{y}_{T+1} = y_{T+1} - \frac{1}{T}\sum_{t=1}^{T} y_t$. There is again no difference between the two hierarchical versions at the top level for the bottom-up approach.

For the top-down approach we get without loss of generality for a bottom level series

$$\begin{aligned}
\tilde{y}_{AA,T+1} &= \frac{\hat{y}_{A,T+1}}{\hat{y}_{A,T+1} + \hat{y}_{B,T+1}} \cdot \frac{\hat{y}_{AA,T+1}}{\hat{y}_{AA,T+1} + \hat{y}_{AB,T+1}} \cdot \hat{y}_{T+1} \\
&= \frac{\hat{y}_{A,T+1}}{\hat{y}_{A,T+1} + \hat{y}_{B,T+1}} \cdot \frac{\hat{y}_{AA,T+1}}{\hat{y}_{A,T+1}} \cdot y_T \\
&= \frac{\hat{y}_{AA,T+1}}{\hat{y}_{A,T+1} + \hat{y}_{B,T+1}} \cdot y_T \\
&= \frac{\hat{y}_{AA,T+1}}{\hat{y}_{A',T+1} + \hat{y}_{B',T+1}} \cdot y_T
\end{aligned}$$

because of the coherence of $\hat{y}_{A,T+1} = \hat{y}_{AA,T+1} + \hat{y}_{AB,T+1}$ and the fact that

$$\begin{aligned}
\hat{y}_{A,T+1} + \hat{y}_{B,T+1} &= \frac{1}{T}\sum_{t=1}^{T} y_{AA,t} + y_{AB,t} + \frac{1}{T}\sum_{t=1}^{T} y_{BA,t} + y_{BB,t} \\
&= \frac{1}{T}\sum_{t=1}^{T} y_{AA,t} + y_{BA,t} + \frac{1}{T}\sum_{t=1}^{T} y_{AB,t} + y_{BB,t} \\
&= \hat{y}_{A',T+1} + \hat{y}_{B',T+1}.
\end{aligned}$$

Again, we see that the reconciliation of the bottom level forecast does not depend on the in-between level structure. The same also holds if the top level forecast is the average and the in-between and bottom level forecasts are chosen as the naive model instead. Furthermore, the sums of the in-between level forecast errors of the two hierarchical versions

$$\begin{aligned}
\epsilon_{A,T+1} + \epsilon_{B,T+1} &= \left(y_{A,T+1} - \frac{1}{T}\sum_{t=1}^{T} y_{AA,t} + y_{AB,t}\right) + \left(y_{B,T+1} - \frac{1}{T}\sum_{t=1}^{T} y_{BA,t} + y_{BB,t}\right) \\
&= \left(y_{A,T+1} - \frac{1}{T}\sum_{t=1}^{T} y_{AA,t} + y_{BA,t}\right) + \left(y_{B,T+1} - \frac{1}{T}\sum_{t=1}^{T} y_{AB,t} + y_{BB,t}\right) \\
&= \epsilon_{A',T+1} + \epsilon_{B',T+1}
\end{aligned}$$

agree.

Scenario 3 - Forecasting method for the bottom level differs from the other levels

For our next scenario we assume that the in-between and top level series are forecasted with the naive model and the bottom level series are again the average of the observations. Then the top level forecast $\hat{y}_{T+1} = y_T = y_{A,T} + y_{B,T} = \hat{y}_{A,T+1} + \hat{y}_{B,T+1}$ is now coherent while the in-between level forecasts

$$\hat{y}_{A,T+1} = y_{A,T} = y_{AA,T} + y_{AB,T} \neq \frac{1}{T}\sum_{t=1}^{T} y_{AA,t} + \frac{1}{T}\sum_{t=1}^{T} y_{AB,t} = \hat{y}_{AA,T+1} + \hat{y}_{AB,T+1}$$
$$\hat{y}_{B,T+1} = y_{B,T} = y_{BA,T} + y_{BB,T} \neq \frac{1}{T}\sum_{t=1}^{T} y_{BA,t} + \frac{1}{T}\sum_{t=1}^{T} y_{BB,t} = \hat{y}_{BA,T+1} + \hat{y}_{BB,T+1}$$

are in general not coherent. The bottom-up approach yields the same reconciled forecasts

$$\tilde{y}_{T+1} = \tilde{y}_{A,T+1} + \tilde{y}_{B,T+1} = \frac{1}{T}\sum_{t=1}^{T} y_t$$

for the top level as in scenario 2, since the forecasting method of the in-between level series does not affect the bottom-up reconciliation, which only uses the bottom level forecasts. Thus, the bottom-up reconciliation of the top level does not depend on the hierarchical version.

Using that $\hat{y}_{T+1} = \hat{y}_{A,T+1} + \hat{y}_{B,T+1}$ and $\hat{y}_{AA,T+1} + \hat{y}_{AB,T+1} = \frac{1}{T}\sum_{t=1}^{T} y_{AA,t} + \frac{1}{T}\sum_{t=1}^{T} y_{AB,t} = \frac{1}{T}\sum_{t=1}^{T} y_{A,t}$ we obtain

$$\begin{aligned}
\tilde{y}_{AA,T+1} &= \frac{\hat{y}_{A,T+1}}{\hat{y}_{A,T+1} + \hat{y}_{B,T+1}} \cdot \frac{\hat{y}_{AA,T+1}}{\hat{y}_{AA,T+1} + \hat{y}_{AB,T+1}} \cdot \hat{y}_{T+1} \\
&= \frac{\hat{y}_{A,T+1}}{\hat{y}_{T+1}} \cdot \frac{\hat{y}_{AA,T+1}}{\hat{y}_{AA,T+1} + \hat{y}_{AB,T+1}} \cdot \hat{y}_{T+1} \\
&= \frac{\hat{y}_{A,T+1}}{\hat{y}_{AA,T+1} + \hat{y}_{AB,T+1}} \cdot \hat{y}_{AA,T+1} \\
&= \frac{y_{A,T}}{\frac{1}{T}\sum_{t=1}^{T} y_{A,t}} \cdot \hat{y}_{AA,T+1}
\end{aligned}$$

and can see that the top-down reconciliation of a bottom-level series is a scaling of its base forecast $\hat{y}_{AA,T+1}$ which only depends on the observations of its parent node. The scaling is the relation of the naive and average forecast of $y_{A,t}$ and will differ depending on the hierarchical version. For $\mathcal{H}_{nat}$ the reconciliation of $\hat{y}_{AA,T+1}$ is

$$\tilde{y}_{AA,T+1} = \frac{y_{AA,T} + y_{AB,T}}{\frac{1}{T}\sum_{t=1}^{T} y_{AA,t} + y_{AB,t}} \cdot \hat{y}_{AA,T+1}$$

and for $\mathcal{H}_v$ it is

$$\tilde{y}_{AA,T+1} = \frac{y_{AA,T} + y_{BA,T}}{\frac{1}{T}\sum_{t=1}^{T} y_{AA,t} + y_{BA,t}} \cdot \hat{y}_{AA,T+1}.$$

We arrive at similar results if, instead, the bottom level series is forecasted with the naive model and the in-between and total level series are the average of the observations. The difference lies in $\hat{y}_{AA,T+1} + \hat{y}_{AB,T+1} = y_{AA,T} + y_{AB,T} = y_{A,T}$ and we get

$$\begin{aligned}
\tilde{y}_{AA,T+1} &= \frac{\hat{y}_{A,T+1}}{\hat{y}_{A,T+1} + \hat{y}_{B,T+1}} \cdot \frac{\hat{y}_{AA,T+1}}{\hat{y}_{AA,T+1} + \hat{y}_{AB,T+1}} \cdot \hat{y}_{T+1} \\
&= \frac{\hat{y}_{A,T+1}}{\hat{y}_{AA,T+1} + \hat{y}_{AB,T+1}} \cdot \hat{y}_{AA,T+1} \\
&= \frac{\frac{1}{T}\sum_{t=1}^{T} y_{A,t}}{y_{A,T}} \cdot \hat{y}_{AA,T+1},
\end{aligned}$$

which is again a scaling that differs depending on the hierarchical version. The scaling factor is the inverse of the previous scaling factor. Since the bottom-level forecasting method changed, it affects the reconciled forecast for the top level, which now is

$$\tilde{y}_{T+1} = \tilde{y}_{A,T+1} + \tilde{y}_{B,T+1} = \tilde{y}_{AA,T+1} + \tilde{y}_{AB,T+1} + \tilde{y}_{BA,T+1} + \tilde{y}_{BB,T+1} = y_{AA,T} + y_{AB,T} + y_{BA,T} + y_{BB,T} = y_T$$

Scenario 4 - Forecasting for the intermediate level differs from the other levels

In this final scenario, we assume that the bottom level and top level series are forecasted with the naive model and the in-between series are the average of the observations. In this case, none of the forecasts are coherent, since in general

$$\hat{y}_{T+1} = y_T = y_{A,T} + y_{B,T} \neq \frac{1}{T}\sum_{t=1}^{T} y_{A,t} + \frac{1}{T}\sum_{t=1}^{T} y_{B,t} = \hat{y}_{A,T+1} + \hat{y}_{B,T+1}$$
$$\hat{y}_{A,T+1} = \frac{1}{T}\sum_{t=1}^{T} y_{A,t} = \frac{1}{T}\sum_{t=1}^{T} y_{AA,t} + y_{AB,t} \neq y_{AA,T} + y_{AB,T} = \hat{y}_{AA,T+1} + \hat{y}_{AB,T+1}.$$

The bottom-level reconciliation is then

$$\begin{aligned}
\tilde{y}_{AA,T+1} &= \frac{\hat{y}_{A,T+1}}{\hat{y}_{A,T+1} + \hat{y}_{B,T+1}} \cdot \frac{\hat{y}_{AA,T+1}}{\hat{y}_{AA,T+1} + \hat{y}_{AB,T+1}} \cdot \hat{y}_{T+1} \\
&= \frac{\frac{1}{T}\sum_{t=1}^{T} y_{A,t}}{\frac{1}{T}\sum_{t=1}^{T} y_{A,t} + \frac{1}{T}\sum_{t=1}^{T} y_{B,t}} \cdot \frac{y_T}{y_{AA,T} + y_{AB,T}} \cdot \hat{y}_{AA,T+1} \\
&= \frac{\sum_{t=1}^{T} y_{A,t}}{\sum_{t=1}^{T} y_t} \cdot \frac{y_T}{y_{A,T}} \cdot \hat{y}_{AA,T+1} \\
&= \frac{\sum_{t=1}^{T} y_{A,t}}{y_{A,T}} \cdot \frac{y_T}{\sum_{t=1}^{T} y_t} \cdot \hat{y}_{AA,T+1}.
\end{aligned}$$

If instead, we look at the opposite case, where the in-between level forecasts are the naive model while the top level and bottom level series are average, then we arrive at

$$\begin{aligned}
\tilde{y}_{AA,T+1} &= \frac{\hat{y}_{A,T+1}}{\hat{y}_{A,T+1} + \hat{y}_{B,T+1}} \cdot \frac{\hat{y}_{AA,T+1}}{\hat{y}_{AA,T+1} + \hat{y}_{AB,T+1}} \cdot \hat{y}_{T+1} \\
&= \frac{y_{A,T}}{y_{A,T} + y_{B,T}} \cdot \frac{\frac{1}{T}\sum_{t=1}^{T} y_t}{\frac{1}{T}\sum_{t=1}^{T} y_{AA,t} + \frac{1}{T}\sum_{t=1}^{T} y_{AB,t}} \cdot \hat{y}_{AA,T+1} \\
&= \frac{y_{A,T}}{y_T} \cdot \frac{\sum_{t=1}^{T} y_t}{\sum_{t=1}^{T} y_{A,t}} \cdot \hat{y}_{AA,T+1} \\
&= \frac{y_{A,T}}{\sum_{t=1}^{T} y_{A,t}} \cdot \frac{\sum_{t=1}^{T} y_t}{y_T} \cdot \hat{y}_{AA,T+1}
\end{aligned}$$

Summary

We summarize our findings in Table 3.6.

Table 3.6 Summary of top-down for different scenarios.

Top Level	In-between Level	Bottom Level	$\tilde{y}_{AA,T+1}$	$\tilde{y}_{A,T+1}$
Naive	Naive	Naive	$\hat{y}_{AA,T+1}$	$\hat{y}_{A,T+1}$
Average	Average	Average	$\hat{y}_{AA,T+1}$	$\hat{y}_{A,T+1}$
Naive	Average	Average	$\frac{y_T}{\frac{1}{T}\sum_{t=1}^{T} y_t} \cdot \hat{y}_{AA,T+1}$	$\frac{y_T}{\frac{1}{T}\sum_{t=1}^{T} y_t} \cdot \hat{y}_{A,T+1}$
Average	Naive	Naive	$\frac{\frac{1}{T}\sum_{t=1}^{T} y_t}{y_T} \cdot \hat{y}_{AA,T+1}$	$\frac{\frac{1}{T}\sum_{t=1}^{T} y_t}{y_T} \cdot \hat{y}_{A,T+1}$
Naive	Naive	Average	$\frac{y_{A,T}}{\frac{1}{T}\sum_{t=1}^{T} y_{A,t}} \cdot \hat{y}_{AA,T+1}$	$\hat{y}_{A,T+1}$
Average	Average	Naive	$\frac{\frac{1}{T}\sum_{t=1}^{T} y_{A,t}}{y_{A,T}} \cdot \hat{y}_{AA,T+1}$	$\hat{y}_{A,T+1}$
Naive	Average	Naive	$\frac{\sum_{t=1}^{T} y_{A,t}}{y_{A,T}} \cdot \frac{y_T}{\sum_{t=1}^{T} y_t} \cdot \hat{y}_{AA,T+1}$	$\frac{y_T}{\frac{1}{T}\sum_{t=1}^{T} y_t} \cdot \hat{y}_{A,T+1}$
Average	Naive	Average	$\frac{y_{A,T}}{\sum_{t=1}^{T} y_{A,t}} \cdot \frac{\sum_{t=1}^{T} y_t}{y_T} \cdot \hat{y}_{AA,T+1}$	$\frac{\frac{1}{T}\sum_{t=1}^{T} y_t}{y_T} \cdot \hat{y}_{A,T+1}$

Overall, we see that the decisive factor for the reconciled forecasts on the intermediate level is whether the forecasting at the top level is performed in the same way as on the intermediate level or not, independent of which forecasting model is actually chosen on the intermediate level itself. When the forecasting models for the top and intermediate levels agree, the reconciled value is the same as the base forecast, while else, we obtain a scaling of the forecast.

Conversely, the reconciliation at the bottom level is a lot more sensitive to which forecasting method is chosen on which level. In particular, the scaling factor for the reconciliation differs between how often we change the way of forecasting between the levels and at which point we change the forecasting model. However, note that the hierarchical version only plays a role in reconciling the bottom level if the forecasting methods for the bottom and intermediate levels are not the same.

From this general example, it is not evident which time series and methods of the alternative aggregation structure will improve the reconciliation. We now turn towards our application case of electricity demand, where we empirically analyze this question. We want to cluster the time series and examine wether that will yield a different hierarchy than originally imposed by the regions.

3.4.2 Clustering

The general approach to cluster time series consists of selecting a similarity measure and a clustering algorithm. Since time series data is high-dimensional, it can initially be transformed into a lower-dimensional representation by extracting global features of the time series [40]. Alternatively, one can choose similarity metrics that consider the time dependence of the data.

We first wanted to analyze the objective of our clustering and determine what kind of similarity is important for our case. Balancing authorities forecast the next 24 hours on a daily basis. Consequently, a daily granularity is significant for them. Furthermore, all observations up to that point are available to them. Given the strong autocorrelation of the series to the preceding days, recent data is most influential to the forecasting process. If we cluster the complete time series, we expect this to instead align time series on long-term patterns like yearly seasonality. Two time series might have similar peaks over the year, but not when the comparison is restricted to a weekly granularity. Similarities on low frequencies might be more representative of the overall series, but they will be less relevant for forecasting on higher frequencies.

Based on those assumptions, we decided to segment the clustering on a weekly basis to capture similarities over all weekdays. The final goal remains to obtain a single clustering. However, through the segmentation, we aim to focus on the similarities in the short-term patterns.

We followed the following approach. We utilized training data from 2021-06-01 01:00:00 until 2024-05-31 00:00:00 and, hence, included each week of a year exactly once. We separated this data into 52 segments. We standardized each balancing authority time series per segment to navigate the scale differences but preserve each series' characteristics. We then applied PCA to reduce the dimensionality of each segment while preserving 80% of the variance. Figure 3.6 illustrates the number of PCA components chosen per segmentation. Then, given the number of clusters K, we performed K-Means with Euclidean distance, resulting in a clustering of the time series of that segment.

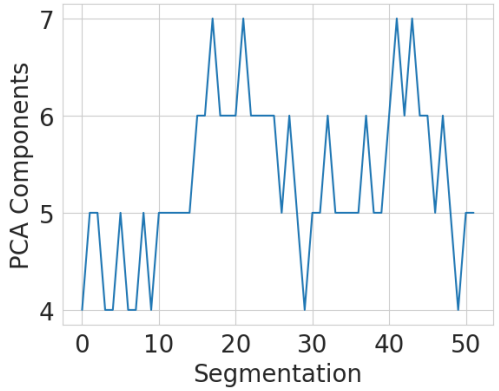

Figure 3.6 Number of PCA-components per segmentation while preserving 80% of the variance.

It remains to decide how many clusters to choose and how to obtain the final clustering. We wanted to analytically determine the number of clusters and considered cluster numbers between 2 and 16. Then, for each cluster size, we clustered on each segment. To evaluate the resulting clusters, we used WCSS, as introduced in Section 3.4 and the Silhouette Score. For the Silhouette Score, we need to first define for a data point i in the cluster C_I

$$a(i) = \frac{1}{|C_I| - 1} \sum_{j \in C_I, j \neq i} d(i, j),$$

which is the mean distance between i and all other data points in the same cluster, and $|C_I|$ is the number of points belonging to cluster C_I. $d(i, j)$ is the distance between data points i and j in the cluster C_I. Furthermore, we define

$$b(i) = \min_{J \neq I} \frac{1}{|C_J|} \sum_{j \in C_J} d(i, j),$$

which is the smallest mean distance of i to all points in any other cluster, i.e., the next best-fit cluster for point i. Finally, we define the Silhouette Score as

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, \quad \text{if } |C_I| > 1,$$

which measures how similar a data point is to its own cluster compared to other clusters [35]. The higher the value, the better. For WCSS, we aim for smaller values. That said, WCSS will be the smallest if each time series is clustered alone. By only examining WCSS, we will not be able to definitively tell how many clusters we should choose, which is why we combine this decision with the Silhouette Score. Averaging the score values over each cluster number yields the results in Figure 3.7, where we can observe two peaks at 5 and 8, respectively. While the peak at 5 is slightly greater, it still has a significantly higher WCSS score than at 8. Thus, we chose $K = 8$ and can see in Figure 3.8 the pairwise plots of all PCA components for one segment with the respective clustering.

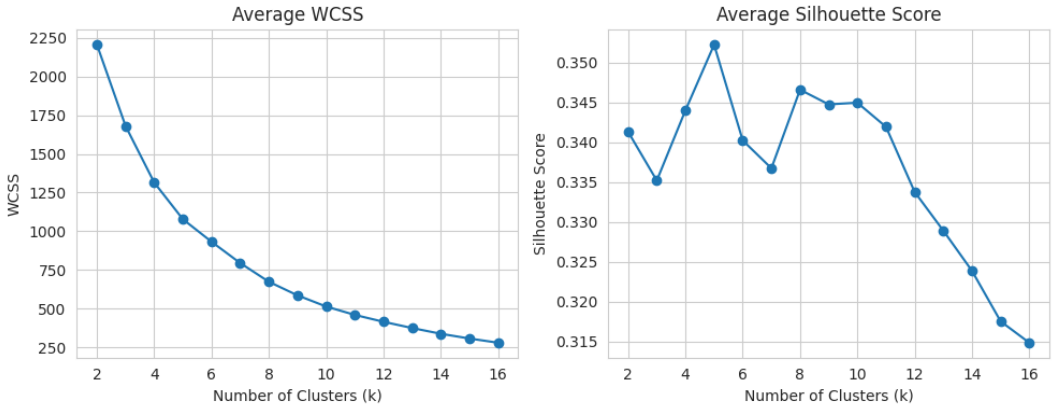

Figure 3.7 Average WCSS and Silhouette Score per number of clusters.

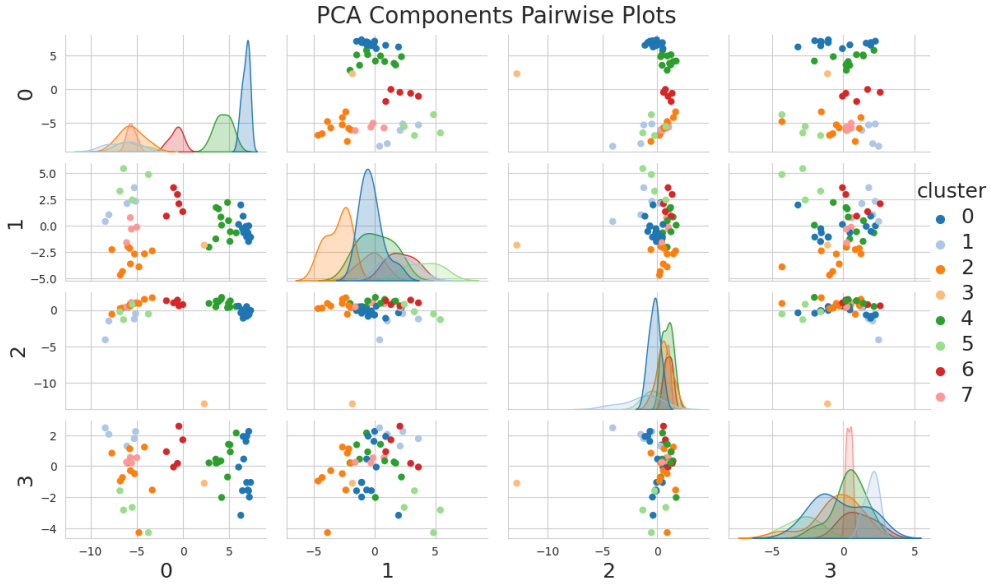

Figure 3.8 PCA-Components preserving 80% of the variance for one segmentation.

Now that we determined the number of clusters, we repeated the above-described process and obtained a clustering of the time series at each segment with K clusters each. In order to result with one final clustering, we counted for each pair of time series how often they are clustered together over all segments and created, thereby a dissimilarity metric, as can be seen in Figure 3.11. Using K-Medoids clustering, we resulted in one final clustering. This process is summarized in Algorithm 1. To visually compare the natural hierarchy with the clustered hierarchy, we can refer to Figure 3.10. It is important to note that this plot serves foremost as visualization since the borders of the balancing authorities are not perfectly accurate. The regions' slightly transparent filling was selected to visualize the overlapping borders of the balancing authorities' areas. Refer to Table 3.7 for an accurate cluster assignment.

We can make three major observations:

1. Neighboring balancing authorities are more likely clustered together,
2. new clusters are rather connected than split, and
3. the natural hierarchical structure has changed significantly.

Algorithm 1: Segment-Based Time Series Clustering

Data: Cluster number $K \in \mathbb{N}$ and set of time series y_t for $t = 1, ..., T$.
Result: Cluster assignment $y_t \mapsto \{C_i\}_{i=0}^{K-1}$
Split $\{1, ..., T\}$ in consecutive segments $\{S\}$ of same size;
$D = [0]_{T \times T}$;
foreach S **do**
    Standardize each time series $y_{t,S}$ separately;
    Apply PCA on all time series $y_{t,S}$, preserving 80% of variance;
    Perform K-Means;
    for $i, j \in \{1, ..., T\}$ **do**
        if y_i *and* y_j *are in the same cluster* **then**
            $D[i][j] += 1$
        end
    end
end
Perform K-Medoids on D for K clusters;

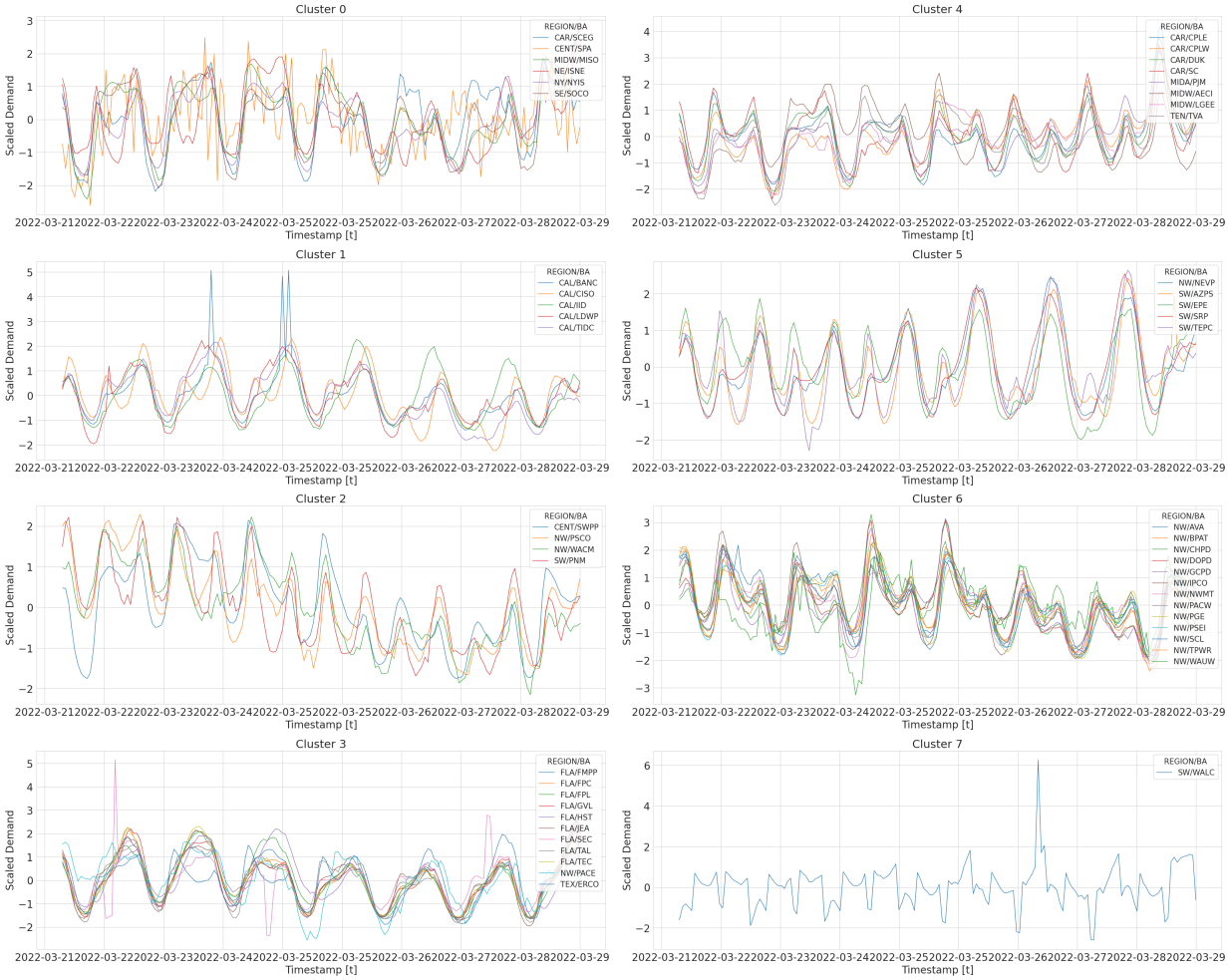

Figure 3.9 Clustering of all bottom-level series for one segment into 8 clusters.

Table 3.7 Comparison of the region groupings given by the natural hierarchical version and the clustering groupings.

BA	Region	New Cluster
PJM	MIDA	0
ISNE	NE	0
NYIS	NY	0
SPA	CENT	1
SWPP	CENT	1
AECI	MIDW	1
LGEE	MIDW	1
MISO	MIDW	1
ERCO	TEX	1
TVA	TEN	1
PGE	NW	2
PSEI	NW	2
SCL	NW	2
TPWR	NW	2
AVA	NW	2
BPAT	NW	2
NWMT	NW	2
IPCO	NW	2

BA	Region	New Cluster
PACW	NW	2
PSCO	NW	3
WAUW	NW	3
PACE	NW	3
WACM	NW	3
BANC	CAL	4
CISO	CAL	4
TIDC	CAL	4
LDWP	CAL	4
IID	CAL	4
WALC	SW	4
TEPC	SW	4
EPE	SW	4
PNM	SW	4
AZPS	SW	4
SRP	SW	4
NEVP	NW	4
GCPD	NW	5

BA	Region	New Cluster
DOPD	NW	5
CHPD	NW	5
CPLE	CAR	6
CPLW	CAR	6
DUK	CAR	6
SC	CAR	6
SCEG	CAR	6
SOCO	SE	6
FMPP	FLA	7
JEA	FLA	7
FPL	FLA	7
GVL	FLA	7
HST	FLA	7
TAL	FLA	7
TEC	FLA	7
FPC	FLA	7
SEC	FLA	7

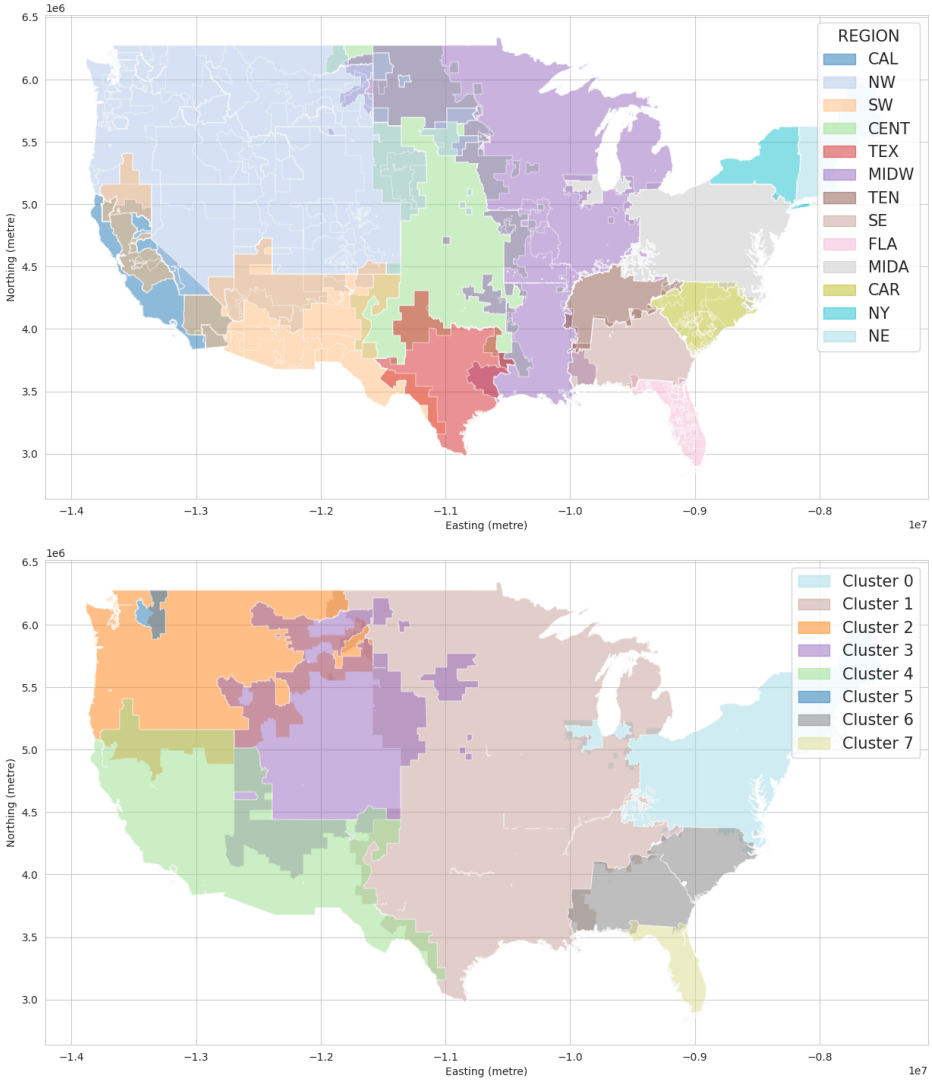

Figure 3.10 Comparison of the natural hierarchical structure and the hierarchical structure that emerged through clustering using the method described in Algorithm 1. The boarders are overlapping, due to the overlapping regions of the balancing authorities.

Figure 3.11 Dissimilarity matrix, showing the number of shared clusters between BAs over all segments.

Furthermore, balancing authorities in the center of the U.S. have widely merged, leaving a significantly lower number of clusters than in the natural hierarchy. Nevertheless, there are also regions from the natural hierarchical structure, which were split up into several clusters, most importantly the northwestern region. Moreover, the new clusters do still roughly follow the time zones in the United States which seems reasonable, as the time zone impacts the electricity demand. Compared to the natural hierarchical structure, the new clusters in the new structure are of relatively similar size, except for two small clusters. In particular, notably Florida is in its own cluster, as was already the case in the natural hierarchical structure.

3.5 Results

Now we can finally take a look at the performance of the different metrics.

Table 3.8 Average MAE over all cross-validation steps by model and level.

MODEL	BA	REGION	TOTAL	OVERALL
SeasonalNaive	470.27	1801.34	15834.64	957.86
ARIMA	3677502.68	1807.39	16167.21	2909661.27
ARIMA/BottomUp	3677502.68	14992769.42	194896324.43	8727014.46
ARIMA/MiddleOut	612.12	1807.39	15529.98	1066.69
ARIMA/TopDown	622.68	1855.25	16167.21	1093.84
ARIMA/OLS	6921486.59	1425550.74	1123850.46	5768579.10
ARIMA/WLSs	5888691.54	8395485.55	64974600.01	7256963.64
ARIMA/WLSv	3531609.58	4175571.43	9638510.43	3747705.17
ARIMA/MinTs	8826053.13	24470805.18	34587179.54	12246096.45
ARIMA/ERM	12548.49	49660.99	528537.94	27450.76
ETS	469.94	1804.31	15356.06	951.03
ETS/BottomUp	469.94	1793.36	15457.06	950.41
ETS/MiddleOut	472.43	1804.31	15540.33	955.75
ETS/TopDown	472.01	1804.23	15356.06	952.65
ETS/OLS	479.28	1811.81	**15341.49**	959.66
ETS/WLSs	473.93	1802.86	15372.35	954.15
ETS/WLSv	470.09	1793.80	15445.29	950.44
ETS/MinTs	**467.01**	**1783.20**	15614.28	**948.47**
ETS/ERM	727.96	2546.72	19044.61	1354.23

Table 3.9 Average RMSE over all cross-validation steps by model and level.

MODEL	BA	REGION	TOTAL	OVERALL
SeasonalNaive	554.47	2112.86	18271.18	1121.27
ARIMA	4195250.35	2140.77	19188.34	3319332.64
ARIMA/BottomUp	4195250.35	17103551.32	222334315.29	9955653.00
ARIMA/MiddleOut	715.28	2140.77	17896.19	1248.30
ARIMA/TopDown	726.41	2190.92	19188.34	1286.12
ARIMA/OLS	7895846.32	1626577.47	1282864.82	6580719.80
ARIMA/WLSs	6717688.96	9577662.16	74122093.09	8278645.01
ARIMA/WLSv	3999730.78	4744530.21	11003618.29	4248779.74
ARIMA/MinTs	9934889.72	27446262.98	40220143.04	13784637.56
ARIMA/ERM	13976.38	55266.74	581314.80	30455.68
ETS	555.12	2127.08	17901.81	1119.03
ETS/BottomUp	555.12	2115.48	17962.31	1117.68
ETS/MiddleOut	557.70	2127.08	18050.11	1123.29
ETS/TopDown	557.41	2127.68	17901.81	1120.96
ETS/OLS	565.51	2135.34	**17883.87**	1128.59
ETS/WLSs	559.53	2125.78	17889.43	1122.08
ETS/WLSv	555.27	2115.92	17952.26	1117.74
ETS/MinTs	**551.50**	**2102.80**	18126.82	**1114.82**
ETS/ERM	866.61	3036.94	22760.07	1614.49

Table 3.10 Average MASE over all cross-validation steps by model and level.

MODEL	BA	REGION	TOTAL	OVERALL
SeasonalNaive	0.99	0.97	0.96	0.99
ARIMA	28646.53	0.98	0.98	22660.89
ARIMA/BottomUp	28646.53	11636.82	11819.47	25094.99
ARIMA/MiddleOut	2.38	0.98	0.95	2.08
ARIMA/TopDown	2.39	1.00	0.98	2.10
ARIMA/OLS	130414.11	1090.42	68.16	103375.99
ARIMA/WLSs	94325.68	6605.17	3940.38	75956.25
ARIMA/WLSv	20013.46	2530.53	584.36	16331.27
ARIMA/MinTs	26473.41	15965.26	2095.76	24070.67
ARIMA/ERM	28.40	28.21	32.16	28.42
ETS	**0.97**	0.97	0.94	**0.97**
ETS/BottomUp	**0.97**	**0.96**	0.94	**0.97**
ETS/MiddleOut	0.98	0.97	0.95	0.98
ETS/TopDown	0.99	0.97	0.94	0.98
ETS/OLS	1.52	0.98	**0.93**	1.41
ETS/WLSs	1.22	0.97	0.94	1.17
ETS/WLSv	**0.97**	**0.96**	0.94	**0.97**
ETS/MinTs	**0.97**	**0.96**	0.95	**0.97**
ETS/ERM	2.27	1.45	1.16	2.09

Table 3.11 Average RMSSE over all cross-validation steps by model and level.

MODEL	BA	REGION	TOTAL	OVERALL
SeasonalNaive	0.83	**0.81**	0.81	0.83
ARIMA	23336.81	0.82	0.85	18460.63
ARIMA/BottomUp	23336.81	9773.01	9849.83	20503.73
ARIMA/MiddleOut	1.81	0.82	**0.79**	1.60
ARIMA/TopDown	1.82	0.84	0.85	1.62
ARIMA/OLS	106233.21	893.37	56.84	84209.41
ARIMA/WLSs	76704.29	5478.04	3283.75	61788.44
ARIMA/WLSv	16129.98	2104.96	487.51	13175.23
ARIMA/MinTs	21062.19	12510.54	1776.19	19115.06
ARIMA/ERM	22.00	22.31	25.84	22.11
ETS	**0.81**	0.82	**0.79**	**0.81**
ETS/BottomUp	**0.81**	**0.81**	0.80	**0.81**
ETS/MiddleOut	**0.81**	0.82	0.80	**0.81**
ETS/TopDown	0.82	0.82	**0.79**	0.82
ETS/OLS	1.27	0.83	**0.79**	1.17
ETS/WLSs	1.01	0.82	**0.79**	0.97
ETS/WLSv	**0.81**	**0.81**	0.80	**0.81**
ETS/MinTs	**0.81**	**0.81**	0.80	**0.81**
ETS/ERM	1.85	1.23	1.01	1.72

Table 3.12 Average DS over all cross-validation steps by model and level.

MODEL	BA	REGION	TOTAL	OVERALL
SeasonalNaive	80.64	**85.12**	**88.82**	81.63
ARIMA	81.35	85.11	88.49	82.19
ARIMA/BottomUp	81.35	84.78	84.03	82.06
ARIMA/MiddleOut	80.33	85.11	88.92	81.38
ARIMA/TopDown	80.14	84.88	88.49	81.18
ARIMA/OLS	73.98	80.59	84.47	75.42
ARIMA/WLSs	75.34	80.94	83.92	76.55
ARIMA/WLSv	78.54	81.36	83.94	79.17
ARIMA/MinTs	77.09	80.06	83.65	77.76
ARIMA/ERM	69.03	77.99	81.50	70.96
ETS	**81.73**	85.00	87.88	**82.46**
ETS/BottomUp	**81.73**	85.03	87.95	**82.46**
ETS/MiddleOut	81.67	85.00	88.01	82.41
ETS/TopDown	81.63	84.99	87.88	82.37
ETS/OLS	79.77	84.96	87.84	80.90
ETS/WLSs	80.59	85.00	87.99	81.56
ETS/WLS	**81.73**	85.03	87.88	**82.46**
ETS/MinTs	81.71	85.05	87.92	82.45
ETS/ERM	73.68	79.19	84.35	74.90

In regard of both MAE and RMSE, we can see that ETS models generally perform better than ARIMA. Surprisingly, ARIMA-based hierarchical methods produce extremely high errors. We can especially observe that the ARIMA base forecasts are exhibiting extreme errors on the BA level, which could be caused by single forecasts performing very badly and skewing the overall average. Thus, it seems that the reconciliation methods have a hard time correcting the BA-level error, and instead decreasing the performance across all levels. Only MiddleOut and TopDown could increase the performance within the ARIMA models, since they disregard the other levels. ERM is not the worst performing within ARIMA, but still badly performing. Looking at the ETS models, we can see that on the bottom-level only one reconciliation method could increase the base forecast. Also the other levels are not consistently improved over the reconciliation methods. However, the best performing models on a per level basis were consistently ETS models. Also, ETS/MinTs managed to improve the error on both the bottom and regional level. Furthermore, the SeasonalNaive might not be the best performing at any level, but consistently well performing.

Also in regard of MASE and RMSSE, ARIMA is performing very poorly. Again we can see, that within ARIMA, TopDown and MiddleOut are the winners.

For RMSSE, ARIMA/MiddleOut even managed to be along the best performing models for the total level. But again we can see that ETS is highly dominating. Interestingly, ETS, ETS/BottomUp, ETS/WLSv and ETS/MinTs managed to have the best performance on two levels and overall. The same in regard of MASE, except of ETS, where the regional level was slightly worse.

Looking at the DS metric, we can see that the Seasonal Naive is the best in predicting the direction of the series at a regional and total level. On the BA level, three ETS-based models perform the best. ETS-based models again consistently perform well across all levels, and also achieving the best overall DS. In general, ETS-based models consistently perform well across all levels. For ARIMA the difference are this time not as extreme as for other metrics, which means that ARIMA still managed to capture the directions, but failed at the magnitude.

All in all we can say, that ARIMA performed very poorly, and the reconciliation methods even decreased its performance. For ETS, the overall performance was quite good. There was no clear winner, and sometimes the SeasonalNaive was still better, but the differences were sometimes slight.

Now we can answer our question, if a clustered hierarchy will improve the bottom-level series performance. Since ETS consistently outperformed ARIMA, we will disregard ARIMA and only compare ETS, and with ETS models that were applied on the clustered hierarchy.

Table 3.13 Comparing the bottom-level series of the natural hierarchical version with the clustered version across different metrics. The average of all cross-validation steps is taken.

MODEL	DS	MAE	MASE	RMSE	RMSSE
SeasonalNaive	80.64	470.27	0.99	554.47	0.83
ETS	81.73	469.94	0.97	555.12	0.81
ETS/BottomUp	81.73	469.94	0.97	555.12	0.81
ETS/MiddleOut	81.67	472.43	0.98	557.70	0.81
ETS/TopDown	81.63	472.01	0.99	557.41	0.82
ETS/OLS	79.77	479.28	1.52	565.51	1.27
ETS/WLSs	80.59	473.93	1.22	559.53	1.01
ETS/WLSv	81.73	470.09	0.97	555.27	0.81
ETS/MinTs	81.71	467.01	0.97	551.50	0.81
ETS/ERM	73.68	727.96	2.27	866.61	1.85
ETS-CLUSTER/BottomUp	81.73	469.94	0.97	555.12	0.81
ETS-CLUSTER/MiddleOut	81.64	472.83	0.98	558.08	0.81
ETS-CLUSTER/TopDown	81.64	472.42	0.98	557.70	0.82
ETS-CLUSTER/OLS	79.31	482.90	1.80	569.57	1.50
ETS-CLUSTER/WLSs	80.76	474.02	1.27	559.56	1.06
ETS-CLUSTER/WLSv	81.72	469.89	0.97	555.10	0.81
ETS-CLUSTER/MinTs	**81.75**	**462.58**	**0.96**	**546.80**	**0.80**
ETS-CLUSTER/ERM	73.83	710.50	2.25	847.00	1.84

Since we only compare the bottom-level series, we can see all metrics at once. We can observe that ETS-CLUSTER/MinTs is best performing over all metrics. Furthermore, the clustered models are often performing better than their unclustered equivalent.

We can in general draw the following conclusions:

1. ETS-based models generally outperformed ARIMA models across all levels and metrics.
2. clustering the hierarchy, further increased performance on the bottom level.

4 Conclusion

In this work, we applied hierarchical time series forecasting to the dataset of electricity demain in the United States. First, we gave an introduction to all of the relevant data science concepts as well as an overview over the application case of electricity demand in the United States. We then conducted two experiments.

In the first experiment, we compared the performance of two forecasting models, ETS and ARIMA, in combination with various reconciliation techniques. As a result, we observed that ETS performed clearly better than ARIMA. The reason for this is that, in most of the cases, already the base forecast of ETS was superior to the one of ARIMA.

Our second experiment aimed at finding a different hierarchical structure and comparing it to the natural hierarchical structure. We obtained this hierarchical structure using a clustering technique that we developed. We then applied ETS to this new hierarchical structure and as a result, the forecasting accuracy improved compared to the natural hierarchy. This suggests that our clustering technique captures useful short-term time series information.

As a future line of research, we could see the examination of a local-global tradeoff in the clustering technique. In particular, for a concrete hierarchical time series, one could obtain several new hierarchical structures by using various ways of clustering. The clustering methods could be the local clustering that was used here, or the traditional clustering technique which captures global similarity of time series, or any combination of these two. This would give insight into how the local versus global clustering compare to each other.

Bibliography

[1] D. Aloise et al. “NP-hardness of euclidean sum-of-squares clustering”. In: *Mach. Learn.* 75.2 (May 2009). Publisher: Springer Science and Business Media LLC, pp. 245–248.

[2] G. Athanasopoulos, R. A. Ahmed, and R. J. Hyndman. “Hierarchical forecasts for Australian domestic tourism”. In: *International Journal of Forecasting* 25.1 (Jan. 2009), pp. 146–166. ISSN: 01692070.

[3] G. Athanasopoulos and N. Kourentzes. “On the evaluation of hierarchical forecasts”. In: *International Journal of Forecasting* 39.4 (Oct. 2023), pp. 1502–1511. ISSN: 01692070.

[4] G. Athanasopoulos et al. “Forecast reconciliation: A review”. In: *International Journal of Forecasting* 40.2 (Apr. 2024), pp. 430–456. ISSN: 01692070.

[5] G. Athanasopoulos et al. “Forecasting with temporal hierarchies”. In: *European Journal of Operational Research* 262.1 (Oct. 2017), pp. 60–74. ISSN: 03772217.

[6] G. Athanasopoulos et al. “Hierarchical Forecasting”. In: *Macroeconomic Forecasting in the Era of Big Data*. Ed. by P. Fuleky. Vol. 52. Series Title: Advanced Studies in Theoretical and Applied Econometrics. Cham: Springer International Publishing, 2020, pp. 689–719. ISBN: 978-3-030-31149-0 978-3-030-31150-6.

[7] S. Avdakovic, A. Ademovic, and A. Nuhanovic. “Correlation between air temperature and electricity demand by linear regression and wavelet coherence approach: UK, Slovakia and Bosnia and Herzegovina case study”. In: *Archives of Electrical Engineering* 62.4 (Dec. 1, 2013), pp. 521–532. ISSN: 0004-0746.

[8] *Balancing Authorities*. U.S. Energy Atlas. URL: `https://atlas.eia.gov/datasets/eia::balancing-authorities/about` (visited on 02/07/2025).

[9] S. Ben Taieb and B. Koo. “Regularized Regression for Hierarchical Forecasting Without Unbiasedness Conditions”. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. KDD ’19: The 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. Anchorage AK USA: ACM, July 25, 2019, pp. 1337–1347. ISBN: 978-1-4503-6201-6.

[10] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006. ISBN: 0-387-31073-8.

[11] G. E. P. Box, G. M. Jenkins, and G. C. Reinsel. *Time Series Analysis*. 1st ed. Wiley Series in Probability and Statistics. Wiley, June 12, 2008. ISBN: 978-0-470-27284-8 978-1-118-61919-3.

[12] M. Charlton and A. Caimo. *Time Series Analysis*. Research Report. ESPON | Inspire Policy Making with Territorial Evidence, 2012.

[13] A. Cini, D. Mandic, and C. Alippi. *Graph-based Time Series Clustering for End-to-End Hierarchical Forecasting*. May 30, 2023. arXiv: `2305.19183[cs]`.

[14] J. D. Cryer and K.-S. Chan. *Time Series Analysis*. Red. by G. Casella, S. Fienberg, and I. Okin. Springer Texts in Statistics. New York, NY: Springer New York, 2008. ISBN: 978-0-387-75958-6 978-0-387-75959-3.

[15] B. J. Dangerfield and J. S. Morris. “Top-down or bottom-up: Aggregate versus disaggregate extrapolations”. In: *International Journal of Forecasting* 8.2 (Oct. 1992), pp. 233–241. ISSN: 01692070.

[16] *Delivery to consumers - U.S. Energy Information Administration (EIA)*. URL: `https://www.eia.gov/energyexplained/electricity/delivery-to-consumers.php` (visited on 01/27/2025).

[17] “EIA-930 HOURLY AND DAILY BALANCING AUTHORITY OPERATIONS REPORT DATA FORMAT AND TRANSMITTAL INSTRUCTIONS”. In: 1905 ().

[18] E. S. Gardner and E. Mckenzie. “Forecasting Trends in Time Series”. In: *Management Science* 31.10 (Oct. 1985), pp. 1237–1246. ISSN: 0025-1909, 1526-5501.

[19] C. W. Gross and J. E. Sohl. "Disaggregation methods to expedite product line forecasting". In: *Journal of Forecasting* 9.3 (May 1990), pp. 233–254. ISSN: 0277-6693, 1099-131X.

[20] X. Han et al. "Efficient Forecasting of Large-Scale Hierarchical Time Series via Multilevel Clustering". In: *ITISE 2023*. ITISE 2023. MDPI, June 29, 2023, p. 31.

[21] H. Hewamalage, K. Ackermann, and C. Bergmeir. "Forecast evaluation for data scientists: common pitfalls and best practices". In: *Data Mining and Knowledge Discovery* 37.2 (Mar. 1, 2023), pp. 788–832. ISSN: 1573-756X.

[22] K. Hitchens. *Understanding the Role of Balancing Authorities*. PCI. Dec. 20, 2023. URL: https://www.pcienergysolutions.com/2023/12/20/a-high-wire-act-understanding-the-role-of-balancing-authorities/ (visited on 01/27/2025).

[23] C. C. Holt. "Forecasting trends and seasonal by exponentially weighted averages". In: *ONR Memorandum* vol. 52 (1957).

[24] "How it Works: The Role of a Balancing Authority". In: ().

[25] S. H. Huddleston, J. H. Porter, and D. E. Brown. "Improving forecasts for noisy geographic time series". In: *Journal of Business Research* 68.8 (Aug. 2015), pp. 1810–1818. ISSN: 01482963.

[26] R. Hyndman. "Another Look at Forecast Accuracy Metrics for Intermittent Demand". In: *Foresight: The International Journal of Applied Forecasting* 4 (Jan. 2006), pp. 43–46.

[27] R. J. Hyndman and G. Athanasopoulos. *Forecasting: principles and practice*. Third print edition. Melbourne, Australia: Otexts, Online Open-Access Textbooks, 2021. 440 pp. ISBN: 978-0-9875071-3-6.

[28] R. J. Hyndman and Y. Khandakar. "Automatic Time Series Forecasting: The **forecast** Package for *R*". In: *Journal of Statistical Software* 27.3 (2008). ISSN: 1548-7660.

[29] R. J. Hyndman, A. J. Lee, and E. Wang. "Fast computation of reconciled forecasts for hierarchical and grouped time series". In: *Computational Statistics & Data Analysis* 97 (May 2016), pp. 16–32. ISSN: 01679473.

[30] R. J. Hyndman et al. "Optimal combination forecasts for hierarchical time series". In: *Computational Statistics & Data Analysis* 55.9 (Sept. 2011), pp. 2579–2589. ISSN: 01679473.

[31] N. Kourentzes and G. Athanasopoulos. "Elucidate structure in intermittent demand series". In: *European Journal of Operational Research* 288.1 (Jan. 2021), pp. 141–152. ISSN: 03772217.

[32] A. Panagiotelis et al. "Forecast reconciliation: A geometric view with new insights on bias correction". In: *International Journal of Forecasting* 37.1 (Jan. 2021), pp. 343–359. ISSN: 01692070.

[33] Y. Pang et al. "Hierarchical electricity time series prediction with cluster analysis and sparse penalty". In: *Pattern Recognition* 126 (June 2022), p. 108555. ISSN: 00313203.

[34] *Real-time Operating Grid - U.S. Energy Information Administration (EIA)*. URL: https://www.eia.gov/electricity/gridmonitor/index.php (visited on 02/07/2025).

[35] P. J. Rousseeuw. "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis". In: *Journal of Computational and Applied Mathematics* 20 (Nov. 1987), pp. 53–65. ISSN: 03770427.

[36] R. H. Shumway and D. S. Stoffer. *Time Series Analysis and Its Applications: With R Examples*. Springer Texts in Statistics. Cham: Springer International Publishing, 2017. ISBN: 978-3-319-52451-1 978-3-319-52452-8.

[37] *U.S. electric system is made up of interconnections and balancing authorities - U.S. Energy Information Administration (EIA)*. URL: https://www.eia.gov/todayinenergy/detail.php?id=27152 (visited on 01/27/2025).

[38] O. US EPA. *U.S. Electricity Grid & Markets*. US EPA. Aug. 30, 2017. URL: https://19january2021snapshot.epa.gov/greenpower/us-electricity-grid-markets (visited on 02/06/2025).

[39] X. Wang, R. J. Hyndman, and S. L. Wickramasuriya. "Optimal forecast reconciliation with time series selection". In: *European Journal of Operational Research* (Dec. 2024), S0377221724009469. ISSN: 03772217.

[40] X. Wang, K. Smith, and R. Hyndman. "Characteristic-Based Clustering for Time Series Data". In: *Data Mining and Knowledge Discovery* 13.3 (Sept. 25, 2006), pp. 335–364. ISSN: 1384-5810, 1573-756X.

[41] S. L. Wickramasuriya, G. Athanasopoulos, and R. J. Hyndman. "Optimal Forecast Reconciliation for Hierarchical and Grouped Time Series Through Trace Minimization". In: *Journal of the American Statistical Association* 114.526 (Apr. 3, 2019), pp. 804–819. ISSN: 0162-1459, 1537-274X.

[42] P. R. Winters. "Forecasting Sales by Exponentially Weighted Moving Averages". In: *Management Science* 6.3 (Apr. 1960), pp. 324–342. ISSN: 0025-1909, 1526-5501.

[43] T. Xiong, Y. Bao, and Z. Hu. "Beyond one-step-ahead forecasting: Evaluation of alternative multi-step-ahead forecasting models for crude oil prices". In: *Energy Economics* 40 (Nov. 2013), pp. 405–415. ISSN: 01409883.

[44] B. Zhang, A. Panagiotelis, and H. Li. "Constructing hierarchical time series through clustering: Is there an optimal way for forecasting?" In: *International Journal of Forecasting* (Nov. 2024), S0169207024001031. ISSN: 01692070.