Constructing hierarchical time series through clustering: Is there an optimal way for forecasting?

Bohan Zhang[a,*], Anastasios Panagiotelis[b], Han Li[c]

[a] *School of Economics and Management, Beihang University, Beijing, China*
[b] *The University of Sydney Business School, NSW 2006, Australia*
[c] *Department of Economics, The University of Melbourne, VIC 3010, Australia*

Abstract

Forecast reconciliation has attracted significant research interest in recent years, with most studies taking the hierarchy of time series as given. We extend existing work that uses time series clustering to construct hierarchies, with the goal of improving forecast accuracy, in three ways. First, we investigate multiple approaches to clustering, including not only different clustering algorithms, but also the way time series are represented and how distance between time series is defined. We find that cluster-based hierarchies lead to improvements in forecast accuracy relative to two-level hierarchies. Second, we devise an approach based on random permutation of hierarchies, keeping the structure of the hierarchy fixed, while time series are randomly allocated to clusters. In doing so, we find that improvements in forecast accuracy that accrue from using clustering do not arise from grouping together similar series but from the structure of the hierarchy. Third, we propose an approach based on averaging forecasts across hierarchies constructed using different clustering methods, that is shown to outperform any single clustering method. All analysis is carried out on two benchmark datasets and a simulated dataset. Our findings provide new insights into the role of hierarchy construction in forecast reconciliation and offer valuable guidance on forecasting practice.

Keywords: Forecast reconciliation, Hierarchical time series, Clustering, Hierarchy construction, Forecast combination

[*] Corresponding author.
Email address: zhangbohan@buaa.edu.cn (Bohan Zhang)

1. Introduction

Applications where some variables are aggregates of one another, or so-called *hierarchical time series (HTS)*, are found in many forecasting problems ranging from supply chain management (Syntetos et al., 2016) to tourism planning (Kourentzes and Athanasopoulos, 2019), electrical load forecasting (Jeon et al., 2019), and retail demand forecasting (Makridakis et al., 2022). In recent decades, there has been an increasing interest in hierarchical forecasting, primarily driven by the success of the optimal reconciliation framework (Hyndman et al., 2011; Wickramasuriya et al., 2019; Panagiotelis et al., 2023). The original motivation for forecast reconciliation was to ensure forecasts are *coherent*, that is they respect the aggregation constraints implied by the hierarchical structure. Coherent forecasts facilitate aligned decisions by agents acting upon different variables within the hierarchy. For example, consider a retail setting, where a warehouse manager supplies stock to individual store managers within their region. Forecasts could be incoherent when the warehouse manager forecasts low total demand while store managers forecast high demand, leading to supply shortages. Numerous case studies in the literature demonstrate that reconciliation approaches not only yield coherent forecasts but also enhance overall forecast performance (Athanasopoulos et al., 2023).

A limitation in the overwhelming majority of forecast reconciliation studies is that the structure of the hierarchy is taken as *given*. This structure usually includes *bottom level series*, an overall aggregate or *top level series*, with various aggregation schemes used to construct *middle level series*. Typically, middle levels are formed according to inherent attributes of the bottom-level series, such as geographical location, gender, product category, travel purpose, and others. We refer to this type of structure as the *natural hierarchy*. While in some forecasting applications, decisions must be made with respect to the natural hierarchy, in other settings there might be some flexibility in determining how bottom levels are aggregated into middle levels. Moreover, a predefined natural hierarchy may not always exist in hierarchical settings, underscoring the importance of a hierarchical construction framework for forecasting purposes. It should be noted that very little attention has been paid to whether middle level series can be constructed in a way that leads to further improvements in forecast accuracy relative to a given natural hierarchy. To the best of our knowledge, only Pang et al. (2018), Li et al. (2019), Pang et al. (2022), and Mattera et al. (2023) have attempted to *construct* middle level series in a data-driven way which ultimately improved forecast accuracy. However, all of these works use time series clustering to construct hierarchies in a manner that is somewhat ad hoc. Our work conducts a more thorough investigation of issues faced when constructing hierarchical structures in forecast reconciliation. In particular, we address the following three research questions:

RQ1. In terms of forecast performance, can the use of middle level series lead to improvement compared to a two-level hierarchy (consisting of only top and bottom time series)? If so, is it possible to construct hierarchies in a data-driven way that leads to further improvements in forecast accuracy?

To investigate these questions, we consider two widely used empirical HTS datasets; the first is Australian tourism demand, the second, cause-of-death mortality data. Throughout the paper, all evaluations are carried out using the series common to all hierarchies; namely the top and bottom level series. In both datasets, we find that the natural hierarchy outperforms the two-level hierarchy, and data-driven hierarchy via clustering can further improve forecast performance compared to natural hierarchy.

The rationale behind the data-driven approach lies in grouping time series with similar patterns together, thereby creating middle-level series with enhanced signals and consequently, improved forecastability. Such arguments have been put forward by Pang et al. (2018), Li et al. (2019), Pang et al. (2022), and Mattera et al. (2023). They all demonstrate superior forecast performance from hierarchies constructed via clustering, relative to the natural hierarchy or the two-level hierarchy. However, these studies for the most part focus on a small number of (in some cases, a single) clustering techniques. In this paper, we take a more systematic approach by clustering time series using different representations (the original time series, forecast errors, features of both), different distance metrics (Euclidean, dynamic time warping), and different clustering paradigms (k-medioids, hierarchical). The models used to obtain base forecasts and the reconciliation method are fixed throughout the experiments. Using both empirical datasets, as well as a simulation study, we find evidence that constructing hierarchies via clustering can lead to improved forecasting performance, although the optimal clustering method depends on the dataset characteristics.

While the idea behind time series clustering is intuitively appealing, the increased accuracy when using clustering-based methods may be attributed to two factors. The first, which we refer to as "grouping" is the idea that some correct subsets of series are chosen to form new middle-level series. This is the argument commonly made to support clustering-based hierarchy construction (see *e.g.* Li et al., 2019; Pang et al., 2022; Mattera et al., 2023). The second factor, which we refer to as the "structure" of the hierarchy, includes the number of middle-level series, the depth of the hierarchy, and the distribution of group sizes in the middle layer(s). Evidence showing that clustering within a forecast reconciliation framework leads to improved forecast accuracy, does not disentangle contributions from these two factors. Indeed, clustering methods may only work in so far as they generate a larger number of base forecasts. This argument would be consistent with the interpretation of reconciliation as a forecast combination of "direct" and "indirect" forecasts (Hollyman et al., 2021), since more middle-level series implies a greater number of indirect forecasts in the combination. This leads to our second research question:

RQ2. Should the improved accuracy of clustering-based methods be attributed to grouping together similar time series, or to the structure of the hierarchy?

To investigate this question, we devise the following approach. We take a hierarchy found using a certain clustering method (or even the natural hierarchy), and then randomly permute the bottom level series (*i.e.*, the leaf nodes of the hierarchical tree). Multiple new "twin" hierarchies are formed with an identical structure to the original hierarchy, but with permuted leaves. In this way, we keep the hierarchical structure fixed, but alter how series are combined. This method can be thought of as an informal "permutation type" test (Welch, 1990). Our main finding is that hierarchies constructed using clustering methods do not significantly outperform their random "twins", leading to the conclusion that the driver of forecast improvement is the enlarged number of series in the hierarchy and/or its structure, rather than similarities between the time series.

Finally, from a practical perspective, we investigate the role of forecast combination in cluster-based hierarchical forecasting. With multiple hierarchies available and inspired by the forecast combination literature (Wang et al., 2023), we consider the last research question

RQ3. Does an equally-weighted combination of reconciled forecasts derived from multiple hierarchies improve forecast reconciliation performance?

Note that forecast combination here differs from that of Hollyman et al. (2021), in that our approach averages not only different coherent forecasts, but also across hierarchies with completely different middle level series. This is possible since only coherent bottom and top level forecasts are averaged and evaluated.

In summary, this paper presents four main contributions:

- We introduce a novel hierarchical forecast reconciliation framework centered on hierarchy construction. Within this framework, we introduce and compare three distinct approaches: cluster-based hierarchies, hierarchies based on random permutation, and forecast combinations across different hierarchies.

- In contrast to existing literature that often focuses on a single clustering technique, our study systematically investigates the effectiveness of various time series clustering implementations. This investigation involves the incorporation of four time series representations, two distance measures, and two clustering algorithms.

- We conduct experiments using two empirical datasets - the Australian tourism dataset and the U.S. cause-of-death mortality dataset as well as a synthetic dataset. The results allow for a comparison of different approaches to constructing hierarchies.

- By constructing random hierarchies through permutation of leaf nodes, we show that the hierarchical structure is the primary contributor to improvements in forecast reconciliation performance, rather than the grouping of similar bottom level series.

The rest of the paper is organized as follows. Section 2 describes the trace minimization reconciliation methods employed and the clustering-based hierarchical time series augmentation techniques considered. Section 3 first introduces the two datasets used, and then investigates RQ1, in particular the performance of cluster hierarchy compared to natural hierarchy and two-level hierarchy. Section 4 introduces the novel permutation approach, followed by the investigation of RQ2 via evaluating the performance of natural hierarchy and the best performing cluster hierarchy compared to their respective random twins. To avoid the concern that clusters found in the empirical datasets are spurious, a simulation study is considered in Section 5. Section 6 covers the forecast combination approach raised in RQ3. Section 7 concludes this paper with discussions on the findings and outlines future research directions. Data and code for reproducing the results in this paper are available at https://github.com/AngelPone/project_hierarchy.

2. Methodology

2.1. Hierarchical forecasting and reconciliation methods

Hierarchical data can be characterized as being made up of bottom level series and their aggregates. In general, we consider a hierarchy with n time series stacked into a vector $\boldsymbol{y}_t$. Let $\boldsymbol{b}_t$ denote the m bottom level series, $\boldsymbol{a}_t$ denote the top level series, and $\boldsymbol{c}_t$ denote k middle level series. The top level and bottom level will be common to all hierarchies we consider, and are augmented by middle level series. These middle level series can be formed according to attributes of the time series or in a data-driven fashion using time series clustering. The series are linked through an $(m + k + 1) \times m$ summing matrix

$$\boldsymbol{y}_t = \boldsymbol{S}\boldsymbol{b}_t = \begin{bmatrix} \boldsymbol{A} \\ \boldsymbol{C} \\ \boldsymbol{I}_m \end{bmatrix} \boldsymbol{b}_t = \begin{bmatrix} \boldsymbol{a}_t \\ \boldsymbol{c}_t \\ \boldsymbol{b}_t \end{bmatrix},$$

where, $\boldsymbol{C}$ consists of zeros and ones that encode the aggregation, *i.e.*, $c_{ij} = 1$ if bottom level series j is included in middle level series i, and zero otherwise. The top level series aggregates all bottom levels, *i.e.*, $\boldsymbol{A} = \mathbf{1}_{1\times m}$, which is a row of ones.

For the purposes of this paper, all forecasting is carried out as a two-step process. First, so-called *base* forecasts are produced for all series in the hierarchy and stacked into an n-vector $\hat{\boldsymbol{y}}$, where subscripts are suppressed for brevity. For base forecasts, we use the Exponential Smoothing (ETS) method (Hyndman et al., 2008), implemented using the `forecast` (Hyndman and Khandakar, 2008) package in R (R Core Team, 2022). Alternative methods such as ARIMA models were also considered, but this choice had very little impact on the overall conclusions.

The base forecasts generated in this first step, do not have the property that bottom level forecasts add up to forecasts of the aggregates, *i.e.*, they are *incoherent*. Therefore, forecast reconciliation is used as a post-forecasting step to ensure coherence of forecasts for all series in the hierarchy. In general, reconciliation takes the form of projecting the base forecasts as

$$\tilde{\boldsymbol{y}} = \boldsymbol{S}(\boldsymbol{S}'\boldsymbol{W}_h^{-1}\boldsymbol{S})^{-1}\boldsymbol{S}'\boldsymbol{W}_h^{-1}\hat{\boldsymbol{y}},$$

where $\tilde{\boldsymbol{y}}$ are the *reconciled* forecasts and $\boldsymbol{W}_h$ is the covariance matrix of h-step-ahead forecast errors. For reconciliation, we use the MinT method (Wickramasuriya et al., 2019), which involves plugging in a shrinkage estimator for $\boldsymbol{W}_h$. Although alternatives were considered, including approaches that assume $\boldsymbol{W}_h$ is diagonal or an identity matrix (Hyndman et al., 2011), this choice has little impact on our main conclusions.

2.2. Time series clustering

To the best of our knowledge, four studies have attempted to improve forecast accuracy in a reconciliation setting by constructing middle levels of the hierarchy using time series clustering. Pang et al. (2018) detect consumption patterns of electricity smart meter data based on X-means clustering algorithm, while Pang et al. (2022) propose several alternative clustering methods to group similar electricity and solar power time series. Li et al. (2019) apply agglomerative hierarchical clustering to cause-of-death time series, and Mattera et al. (2023) utilize Partition Around Medoids algorithms to unveil underlying structures in stock price indexes. However, these studies are limited in scope as they focus on a few clustering techniques. Inspired by the comprehensive overview of time series clustering by Aghabozorgi et al. (2015), we consider various approaches based on three key components, namely *time series representations*, *distance measures*, and *clustering algorithms*.

Time series representations. The time series representation refers to the object that acts as an input for time series clustering. Our first candidate for the representation is raw time series itself, due to its simplicity and broad applicability. We also consider the in-sample one-step-ahead forecast error as a representation of the time series, since a key step in MinT reconciliation is to estimate the $\boldsymbol{W}_h$ matrix. It is important to note that raw time series and in-sample error representations are standardized to eliminate the impact of scale variations.

A potential shortcoming to using the time series or forecast error as a representation is their high dimensionality, which is equal to the sample size of the training data. To address this concern, low dimension summaries of "features" can be considered. Features have been used in the context time series clustering by Tiano et al. (2021), and for forecasting by Wang et al. (2022) and Li et al. (2023). We consider features of both the raw time series and the in-sample forecast error as representations. After filtering out the features that are constant across all series, we select 56 features.[1] These time series features are calculated by the `tsfeatures`

[1] The list and descriptions of features are available in the online GitHub repository.

(Hyndman et al., 2022) package in R. To the best of our knowledge, we are the first to utilize in-sample forecast error and time series features as representations in the context of forecast reconciliation.

Distance measures. All clustering algorithms we consider require a distance to be defined between the objects that act as inputs to the algorithm. We consider two widely applied distance measures: Euclidean distance and dynamic time warping (DTW). When employing Euclidean distance, dimension reduction is necessary due to the curse of dimensionality. To address this, we perform Principal Component Analysis (PCA), extracting the first few principal components that collectively explain at least 80% of the variance within the representations.

Unlike Euclidean distance, DTW is not as sensitive to the curse of dimensionality (Sakoe and Chiba, 1978). Instead of performing one-to-one point comparisons, DTW accommodates time series of varying lengths through many-to-one comparisons. This flexible approach allows for the recognition of time series with similar shapes, even in the presence of signal transformations such as shifting and/or scaling.

Clustering algorithms. In this paper, we focus on two clustering algorithms, namely k-medoids and agglomerative hierarchical clustering. These algorithms are implemented using the `cluster` (Maechler et al., 2022) package in R. The k-medoids algorithm aims to minimize the total distance between all observations within a cluster and their respective cluster median. This is implemented using the partitioning around medoids (PAM) method (Kaufman and Rousseeuw, 1990). Following the recommendation of Kaufman and Rousseeuw (1990), we determine the optimal number of clusters using the average silhouette width (ASW), which is a commonly used index in cluster validation (see *e.g.*, Shutaywi and Kachouie, 2021).

Agglomerative hierarchical clustering initializes each observation in its own cluster, and then merges the nearest two clusters in a stepwise fashion until all observations form a single cluster. We employ Ward's linkage (Murtagh and Legendre, 2014) which defines the nearest clusters as those that minimize the increase in within-cluster variance at each step. Applying hierarchical clustering to m bottom-level series results in a binary hierarchical tree with $(2m-1)$ nodes, all of which are retained as middle level series.

Figure 1 illustrates two examples of hierarchies generated by k-medoids and agglomerative clustering algorithms, on the left and right panels respectively. The corresponding two-level

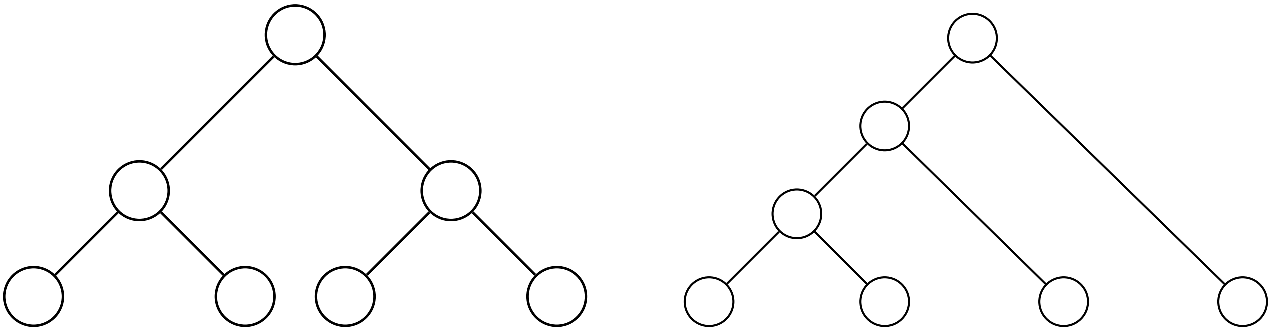

Figure 1: Example clustering results of two clustering algorithms. Left panel displays example for k-medoids algorithm, and right panel displays example for agglomerative hierarchical clustering algorithm.

hierarchy would have four bottom-level series and one top-level series. Note that, k-medoids constructs a simple hierarchy with a single middle level, while hierarchical clustering generates multiple nested middle levels. In general, k-medoids produces a hierarchy with fewer middle levels and middle level series compared to hierarchical clustering. As the number of bottom-level series increases, these differences become increasingly pronounced, with potential implications for forecast reconciliation.

In summary, we employ 12 time series clustering approaches which are derived from combinations of four time series representations, two distance measures, and two clustering algorithms. The names and details of these approaches are listed in Table 1. Note that all methods using DTW, have either the raw time series or forecast errors as representations, since DTW is not compatible with time series features.

Table 1: Details of the 12 clustering approaches considered.

Approach	Dimension reduction	Representation	Distance measure	Clustering algorithm
TS-EUC-ME	Yes	Time series	Euclidean	k-medoids
ER-EUC-ME	Yes	In-sample error	Euclidean	k-medoids
TSF-EUC-ME	Yes	Time series features	Euclidean	k-medoids
ERF-EUC-ME	Yes	In-sample error features	Euclidean	k-medoids
TS-EUC-HC	Yes	Time series	Euclidean	Hierarchical
ER-EUC-HC	Yes	In-sample error	Euclidean	Hierarchical
TSF-EUC-HC	Yes	Time series features	Euclidean	Hierarchical
ERF-EUC-HC	Yes	In-sample error features	Euclidean	Hierarchical
TS-DTW-ME	No	Time series	DTW	k-medoids
TS-DTW-HC	No	In-sample error	DTW	Hierarchical
ER-DTW-ME	No	Time series	DTW	k-medoids
ER-DTW-HC	No	In-sample error	DTW	Hierarchical

3. Improving forecast performance via hierarchy augmentation

3.1. Data description

We conduct our experiments on two empirical datasets throughout this paper. The first is the monthly Australian domestic tourism dataset, covering the period from January 1998 to December 2016.[2] The data is recorded as "visitor nights", representing the total number of nights spent by Australians away from home. In this dataset, the total visitor nights of Australia is geographically disaggregated into seven states and territories, which are further divided into 27 zones, and then into 76 regions. Additionally, each regional-level series is divided by four travel purposes. Overall, this dataset comprises a total of 555 time series with 304 of those at the bottom level. In the case of tourism data, the first two or three letters of the series name indicate geographical zones or regions, and the last three denote travel purposes. For example, "*AAAHol*" represents the visitor nights spent for holiday in the "Sydney" region.

The second dataset focuses on cause-of-death mortality in the U.S. We obtain monthly cause-specific death count data from the Center for Disease Control and Prevention (CDC) for the period between January 1999 and December 2019. The dataset, organized based on the 10th revision of the International Classification of Diseases (ICD) 113 Cause List[3], forms a hierarchy containing 120 time series, with 98 of those being bottom-level series[4]. The top-level series represents the aggregated deaths from all causes, while the middle-level series are constructed based on major cause-of-death groups. As an example, *Diseases of heart* (ICD code: I00–I09, I11, I13, I20–I51; 113 Cause List: GR113-054) is a middle-level series in the hierarchy, which contains bottom-level series *Hypertensive heart disease* (I11; GR113-056) and *Heart failure* (I50; GR113-067), among other circulatory diseases.

The top-level series, one selected middle-level series, and three selected bottom-level series are illustrated in Figures 2 and 3 for the tourism and mortality datasets, respectively. Both figures exhibit more regular and apparent trend and seasonality for series at a higher level of aggregation, while bottom-level series are noisier and prone to outliers. Comparing the

[2] Please refer to Section 4 of Wickramasuriya et al. (2019) for an in-depth explanation of this dataset.

[3] For more detailed information on the dataset, please refer to `https://wonder.cdc.gov/ucd-icd10-expanded.html`.

[4] To address the data suppression issue, we combined certain ICD codes to ensure all death counts are no less than 10.

datasets to one another, the mortality dataset generally exhibits stronger seasonality and trend, whereas the tourism dataset displays greater volatility. Table 2 summarizes three features of each dataset, with larger values indicating more "signal" relative to "noise". These are: the Holt Winters seasonal smoothing parameter; the lag 12 autocorrelation coefficient; and the strength of trend measured as the proportion of variance explained by the trend component in an STL decomposition. Table 2 supports the conclusions made by visualizing the time series of the data, which is that the tourism data are noisier and less regular with respect to trend and seasonality.

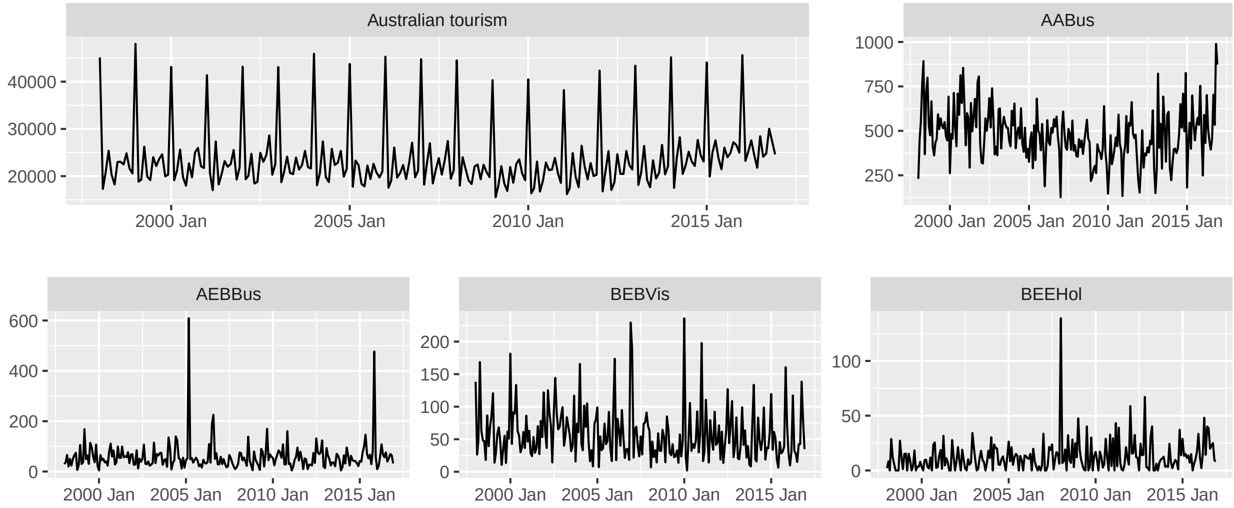

Figure 2: Visualization of selected time series from the tourism dataset. "AA", "AEB", "BEB", and "BEE" represent the zone "Metro NSW", and the regions "New England North West", "Western Grampians", and "Spa Country", respectively. "Bus", "Vis", and "Hol" denote travel purposes "Business", "Visit", and "Holiday", respectively.

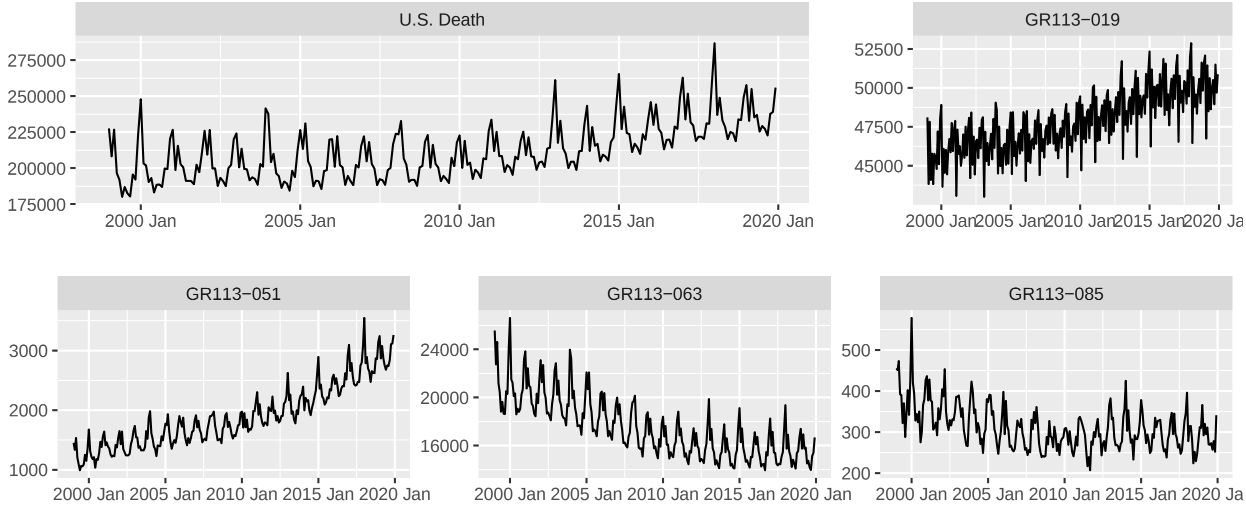

Figure 3: Visualization of selected time series from the death count dataset. "GR113-019", "GR113-051", "GR113-063", and "GR113-085" denote "Malignant neoplasms", "Parkinson disease", "All other forms of chronic ischemic heart disease", and "Asthma", respectively.

Table 2: Trend and seasonality features for the tourism and mortality dataset.

Feature	tourism	mortality
Strength of trend	0.1559	0.7574
Seasonality smoothing parameter	0.0002	0.0202
Seasonal auto-correlation coefficient	0.1814	0.6523

3.2. Evaluation of forecast accuracy

To measure forecast accuracy, we first calculate the Root Mean Squared Scaled Error (RMSSE, Makridakis et al., 2022), for each series. Letting $\breve{y}_t$ be any forecast of y_t, we define RMSSE as

$$RMSSE = \sqrt{\frac{\frac{1}{h}\sum_{t=T+1}^{T+h}(y_t - \breve{y}_t)^2}{\frac{1}{T-12}\sum_{t=13}^{T}(y_t - y_{t-12})^2}}.$$

RMSSE is symmetric, independent of the data scale, and thus suitable for evaluating hierarchical forecasts (Athanasopoulos and Kourentzes, 2023). It should be noted that the denominator of our RMSSE measure slightly differs from that used in Makridakis et al. (2022); we replace the naive forecast, with the seasonal naive forecast since the time series in our applications exhibit monthly seasonality. As a single measure of accuracy, we take the average RMSSE across all series. Since we will be comparing hierarchies with different structures (thus different middle level series), this average only includes top and bottom level series, both of which are guaranteed to be present in all hierarchies.

We utilize the expanding window strategy to evaluate the performance of different approaches with a forecast horizon of 12 months (one year). For both datasets, the first window contains 96 training observations. The training window is then increased by one observation and new 12-step ahead forecasts are obtained. This procedure is repeated until the training window comprises all but the last 12 observations. This results in 121 12-step-ahead forecasts (January 2006 - January 2016) for the tourism dataset and 145 12-step-head forecasts (January 2007 - January 2019) for the mortality dataset.

To assess whether differences in forecast performance are statistically significant, we employ the Multiple Comparison with the Best (MCB) test (Koning et al., 2005). This test is based on the average ranks of different approaches across all evaluation windows and controls for multiple comparisons.

3.3. Cluster hierarchies vs benchmarks

Table 3 compares the accuracy of reconciled forecasts when using hierarchies obtained from the 12 clustering-based hierarchies outlined in Table 1. The base forecasts, as well as the reconciled forecasts from the two-level hierarchy (only containing top- and bottom- level time series) and from the natural hierarchy, are included as benchmarks. The MCB test results are presented in Figure 4.

Table 3: Performance of cluster hierarchies and benchmark hierarchies in terms of average RMSSE across all evaluation windows on both datasets. Column-wise minimum values are displayed in bold.

Approach	tourism	mortality
Base	0.6945	0.7530
Two-level	0.6944	0.7528
Natural	0.6913	0.7501
TS-EUC-ME	0.6939	0.7528
ER-EUC-ME	0.6938	0.7530
TSF-EUC-ME	0.6938	0.7549
ERF-EUC-ME	0.6942	0.7532
TS-EUC-HC	0.6922	0.7540
ER-EUC-HC	0.6920	0.7507
TSF-EUC-HC	**0.6909**	0.7509
ERF-EUC-HC	0.6910	0.7501
TS-DTW-ME	0.6940	0.7528
TS-DTW-HC	0.6911	**0.7496**
ER-DTW-ME	0.6942	0.7531
ER-DTW-HC	0.6912	0.7532

We have the following observations from Table 3 and Figure 4. In terms of average RMSSE, for both datasets, the base forecasts provide the worst forecast performance. The natural hierarchies provide better results than the base forecasts and the two-level hierarchies, and comparable results with cluster hierarchies. In the case of the tourism dataset, all twelve clustering-based hierarchies outperform the simple two-level hierarchy. For ten out of twelve of these methods, the prediction intervals for the average ranks do not overlap with the two-level hierarchy, indicating significantly superior performance. For the mortality dataset, five cluster hierarchies surpass the two-level hierarchy in terms of average RMSSE. However, not even the best clustering method is significantly more accurate than the two-level hierarchy based on the MCB test.

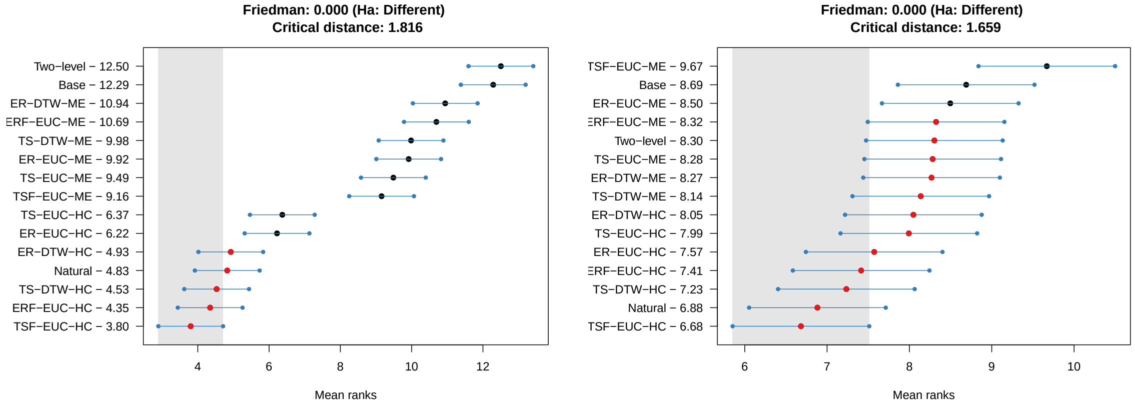

Figure 4: Average ranks and 95% confidence intervals for twelve cluster hierarchies and three benchmarks on tourism dataset (left) and mortality dataset (right) based on MCB test.

The varying performance of cluster hierarchies across two datasets can be attributed to the unique characteristics of their bottom-level series. The tourism dataset, as shown in Figure 2, predominately contains volatile and noisy bottom-level time series with weak trend and seasonality (see also in Table 2). Arguably, creating new middle-level time series in this context helps elucidate the underlying pattern which can not be easily captured by bottom-level base forecasting models due to a low signal-to-noise ratio. On the other hand, the bottom-level series in mortality dataset exhibit stronger trend and seasonality patterns, meaning that the addition of middle-level series is less beneficial.

Table 4: Average number of middle-level time series resulting from k-medoids clustering, hierarchical clustering, and the natural hierarchy for both datasets.

Approach	tourism	mortality
Natural	250	21
k-medoids clustering	21	7
Hierarchical clustering	302	96

We also observe that the hierarchies constructed via hierarchical clustering algorithms outperform the hierarchies based on k-medoids when using the same representation and distance metric. As an example, “TSF-EUC-HC” outperforms “TSF-EUC-ME”. This superior performance can be attributed to hierarchical clustering generating a greater number of middle-level time series than k-medoids. Table 4 summarizes the average number of middle-level series across all evaluation windows for natural, k-medoids, and hierarchical clustering hierarchies. Interestingly, the natural hierarchy shows competitive accuracy compared to hierarchical clustering methods on both datasets, despite having fewer middle-level series. However, it should be noted that natural hierarchies may not always exist. Regarding the superiority of any specific representation or distance metric, no consistent findings emerge. It shows that while it is possible to improve forecast accuracy via clustering, the performance of different clustering methods largely depends on the specific data in consideration.

4. Disentangling grouping and structure

These results in Section 3.3 raise the question of why hierarchies augmented with middle levels improve upon the two-level hierarchy. There are two possible explanations. On the one hand, it could be argued that by "grouping" together series with similar characteristics, certain signals are enhanced, leading to improved forecasting performance. Alternatively, it could be argued that the improved forecasting performance is a by-product of the "structure" of the hierarchy, *i.e.* the number of middle-level series, the depth of the hierarchy, the distribution of group sizes in the middle layer(s), or a combination of all these factors.

4.1. Permutation hierarchy construction

To assess whether "grouping" or "structure" has relatively more importance, we consider a procedure based on permutation. For ease of exposition, we will describe this procedure for the natural hierarchy, although it is applied equally to hierarchies where middle level series are constructed using clustering algorithms. First, the structure of the natural hierarchy is kept fixed. A new "twin" hierarchy is constructed by randomly permuting the bottom level series[5]. An example is shown in Figure 5.

Suppose the forecast performance of the natural hierarchy can be explained by "grouping". In this case, the "twin" hierarchy, with a grouping formed at random, should perform significantly worse than the natural hierarchy. Alternatively, suppose the "twin" performs similarly to the natural hierarchy. In this case, the critical factor in improved forecast performance is the structure of the hierarchy, which is the same for both the natural hierarchy and its twin.

[5] This can also be achieved by shuffling the columns of $\boldsymbol{C}$.

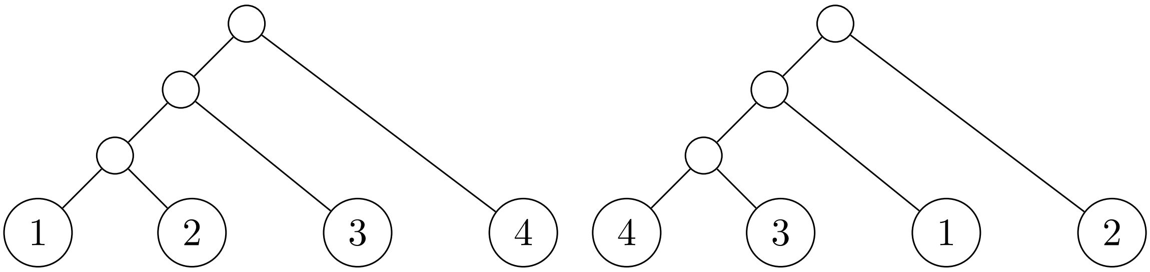

Figure 5: Examples of a given hierarchy and its "twin".

To rule out the possibility that a random twin with exceptionally good (or bad) grouping is generated by chance, in all cases we consider 100 random twin hierarchies.

4.2. Natural hierarchy vs its twins

Figure 6 compares the natural hierarchy to 100 twins using the MCB test. For brevity, we only display the average rank labels for the natural hierarchy and 5 of its twin hierarchies[6]. The figure is adjusted so that the gray band is around the 95% confidence interval for the natural hierarchy rather than the best performing twin. Any hierarchies whose confidence interval overlaps with the gray zone is not significantly better or worse than the natural hierarchy.

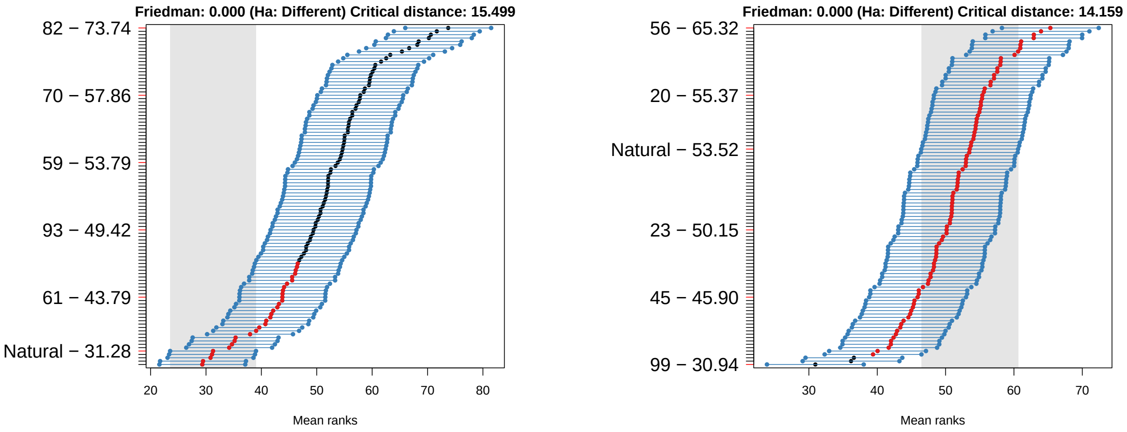

Figure 6: Average ranks and 95% confidence intervals for natural hierarchy and its 100 twins, tourism dataset (left) and mortality dataset(right).

It is clear that for both datasets, the natural hierarchy does not significantly outperform a large proportion of its random twins. On the tourism dataset, the natural hierarchy ranks the 5th. Its performance is statistically indistinguishable from 32 of its twins, but significantly

[6] Note that "82 - 73.74" represents that the 82nd permutation of the hierarchy which has an average rank of 73.74. The same convention applies to the labels of the y-axis for Figures 7, 10, and 13.

better than the remaining 68. The difference in forecast performance for the natural hierarchy and its twins is even less pronounced for the mortality dataset. As shown in right panel of Figure 6, the performance of the natural hierarchy is statistically indistinguishable from most of its twins and, there are three twin hierarchies significantly better than the natural hierarchy. In both datasets, and particularly for the mortality dataset, we conclude that the "structure" of the natural hierarchy is the primary contributor to the improvement in forecast accuracy over the two-level hierarchy.

4.3. Cluster hierarchy vs its twins

The result in Section 4.2 suggesting that "structure" is a more important contributor to "grouping" may arise since the grouping for the natural hierarchy is not selected in a data-driven fashion. To assess whether clustering methods select a better "grouping", we compare the best-performing clustering-based hierarchies (TSF-EUC-HC and TS-DTW-HC for tourism and mortality, respectively) with their random twins. Recall that Section 3.3 demonstrates that the clustering methods can outperform the natural hierarchy and two-level hierarchy.

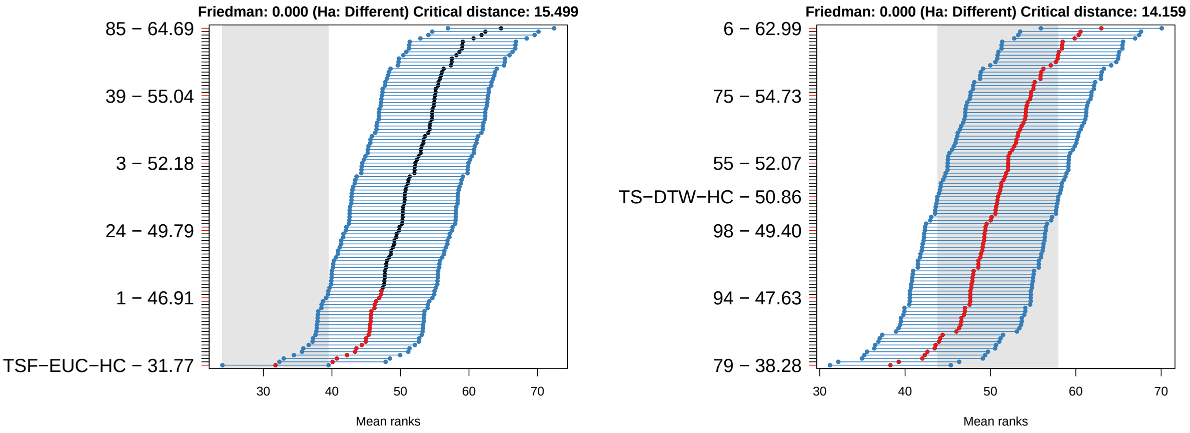

Figure 7: Average ranks and 95% confidence intervals for best performing clustering approach and its 100 twins, tourism dataset (left) and mortality dataset (right).

The MCB test results for the tourism dataset and mortality dataset are shown in Figure 7. We observe that in both datasets, the best performing clustering approach does not yield significantly better results than its random twins. The best cluster approach of mortality dataset ranks nearly in the middle of its random twins, indicating that once again structure rather than grouping is the main driver of improvement in forecast accuracy. On the other hand, the best performing clustering method for the tourism data is statistically indistinguishable from

30 of its random twins, despite being the best in terms of the mean ranks. One can argue that for tourism dataset, a data-driven method for grouping time series plays a more prominent role in forecast improvement. This may be attributed to the noisier nature of bottom level tourism data, suggesting that similar weak signals are strengthened when aggregated. However, there is still roughly a 30% chance that a random twin performs similarly, once again highlighting that structure is the main contributor to improved forecast performance.

5. Simulation study

The main conclusion of Section 4 is that hierarchies with more middle level series improve forecast accuracy due to structure rather than grouping. In so far as grouping may be a factor, this may be due to aggregating similar weak signals into stronger signals. To further test this conjecture, we consider a simulation study. Time series are generated that form clear clusters according to their trend and seasonality. These are then aggregated into middle level series, based on the known characteristics of each time series. The purpose of this is two-fold. First, in a simulated setting, the "true" clusters can be known, which guards against the risk that a given clustering algorithm fails to identify the correct clusters. Second, a simulation study guards against the shortcoming of any cluster analysis, namely that clusters will always be found even where they are not present. Such spurious clusters may explain the similar forecasting performance of cluster-based hierarchies with their randomly permuted twins. The simulation study thus sets up an ideal scenario, where grouping can potentially dominate structure as the factor explaining improved forecast performance.

5.1. Simulation design

We construct $m = 120$ bottom-level time series in an additive manner, each following a data generating process described as follows:

$$\begin{aligned} Y_t &= L_t + S_t + \xi_t, \\ L_t &= \alpha t + \varepsilon_t, \\ S_t &= \begin{cases} \beta & \text{if } t-\delta \text{ is even} \\ \gamma & \text{if } t-\delta \text{ is odd} \end{cases} \quad , \end{aligned} \tag{1}$$

where L_t represents the trend term that increases or decreases over time with slope α. We set α to 0.001, -0.002, and 0 so that exactly one third of the bottom level series have increasing, decreasing, and no trend respectively. The seasonal pattern is determined by S_t. It is deterministic with a seasonal period of 2, hitting a peak β drawn uniformly from [2, 3], and a trough γ drawn uniformly from [0, 1]. The parameter δ controls whether a time series has its seasonal peak for odd values of t or even values of t, which we refer to as "odd" and "even" seasonality respectively. We set $\delta = 0$ (even seasonality) for exactly half of the series and $\delta = 0$ (odd seasonality) for the other half of the series. Both ξ_t and ε_t are white noise with the variance of ξ_t set to 0.25 and the variance of ε_t set to 2.5×10^{-5} and 4.9×10^{-5} for increasing trend and decreasing trend, respectively. The combination of three different trends with two different patterns of seasonality leads to six clusters as described in Table 5.

Table 5: Parameter setting for all clusters in the simulation experiments.

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6
Trend	Increase	Increase	None	None	Decrease	Decrease
Seasonality	Odd	Even	Odd	Even	Odd	Even

For each series, we generate 144 observations, with the last 12 observations reserved for evaluation. Figure 8 displays typical time series from each cluster, while Figure 9 is a scatterplot of the first two principal components of the series. It is clear that the simulation design generates distinct clusters.

Various schemes for constructing middle level series can be considered and are summarized in Table 6. Middle level series can be constructed according to the value of the α parameter, leading to three clusters, or according to whether the series has a trend or no trend, leading to two clusters. In Table 6, these are referred to as Cluster-trend1 and Cluster-trend2 respectively. Middle level series can also be formed on the basis of seasonality (Cluster-season), leading to two clusters, or according to both trend and seasonality leading to six clusters (Cluster-trend-season).

5.2. Results

We replicate the simulation 500 times and follow the evaluation procedure introduced in Section 3.2. Table 7 reports the average RMSSE across all hierarchies, and Figure 10 presents the MCB test results. The results reveal that most approaches perform better than the base

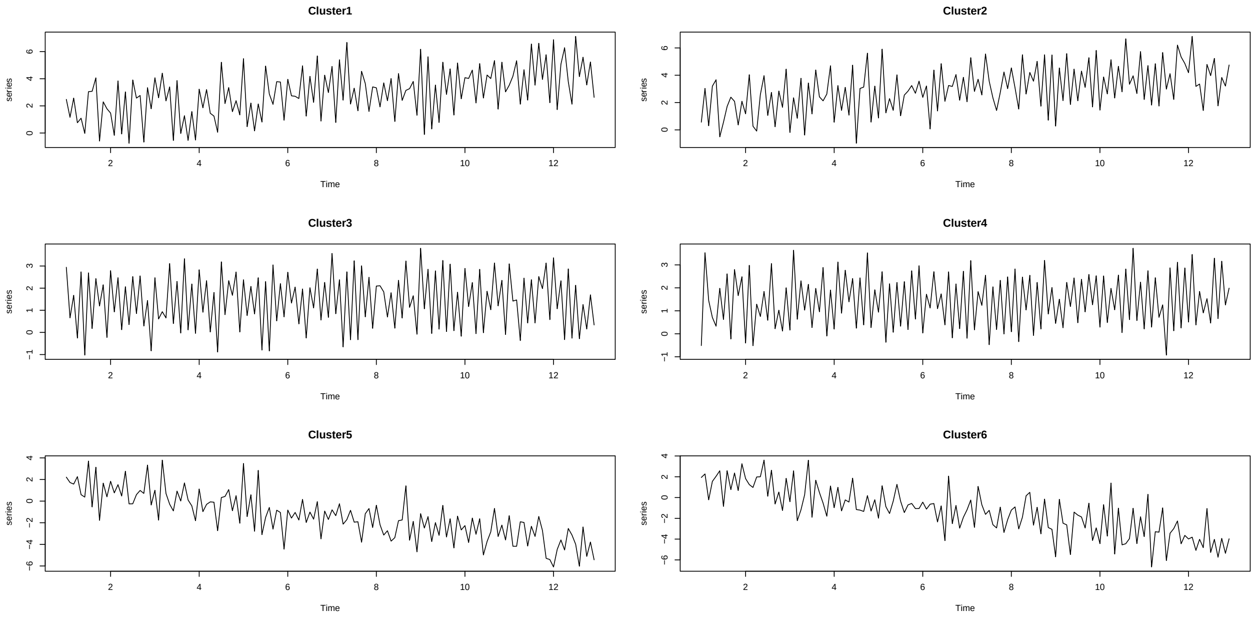

Figure 8: Example time series for each cluster in the simulation experiments.

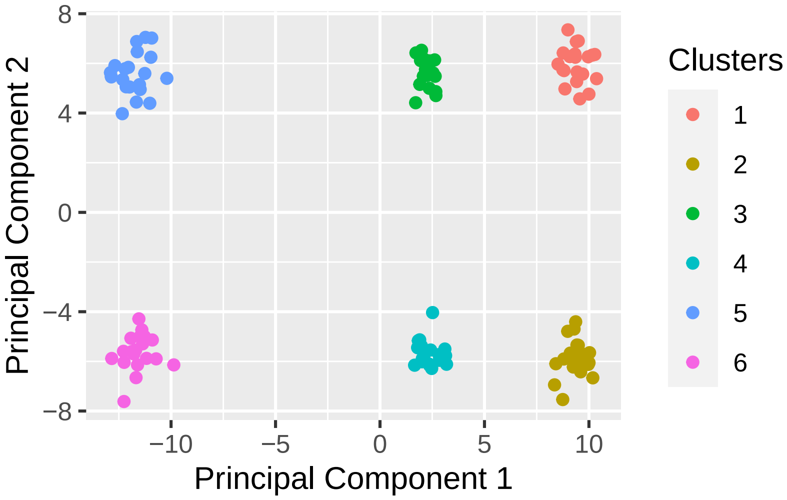

Figure 9: Visualization of the generated time series in the simulation experiments.

Table 6: Four clustering approaches used in the simulation experiments.

Approach	Description
Cluster-trend-season	Hierarchy based on trend and seasonal pattern.
Cluster-trend1	Hierarchy based on trend (positive/negative/none).
Cluster-trend2	Hierarchy based on trend/no trend.
Cluster-season	Hierarchy based on seasonal pattern (odd/even).

forecasts and the two-level hierarchy. This outcome indicates that hierarchy construction generally improves forecast reconciliation performance, corroborating our findings reported in

Section 3.3. Clustering only on the basis of trend yields the best performance, however all clustering schemes are statistically indistinguishable from one another.

Table 7: Performance of cluster hierarchies and benchmark hierarchies in terms of average RMSSE in simulation. Column-wise minimum values are displayed in bold.

Approach	Average RMSSE
Base	0.7764
Two-level	0.5971
Cluster-trend-season	0.5963
Cluster-trend1	**0.5962**
Cluster-trend2	0.5965
Cluster-season	0.5965

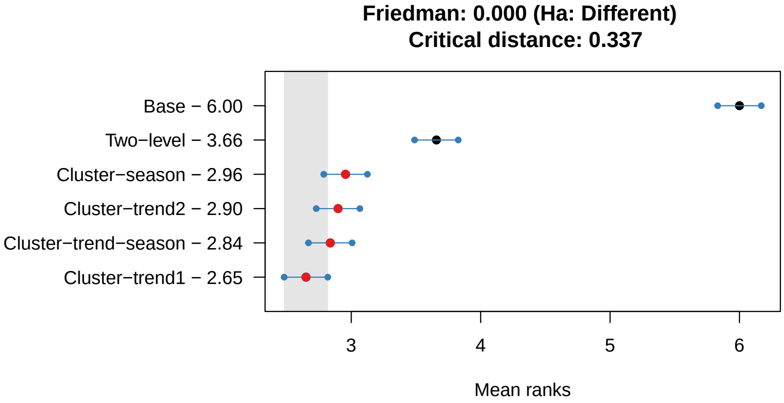

Figure 10: Average ranks and 95% confidence intervals for four cluster hierarchies and two benchmarks in simulation based on MCB test.

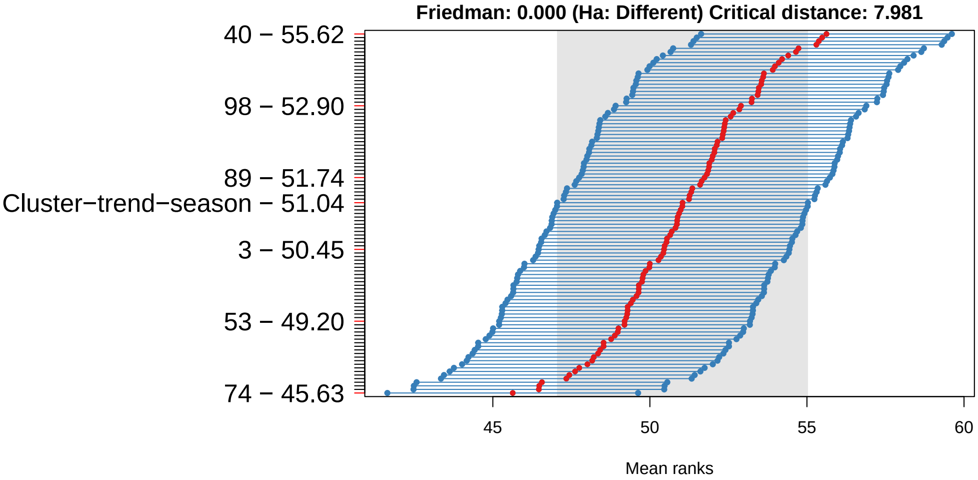

Figure 11: Average ranks and 95% confidence intervals for four the ideal hierarchy and its 100 random twins in simulation based on MCB test.

Having constructed clearly demarcated clusters, we now compare the six cluster hierarchy against 100 of its random twins. The MCB test result is shown in Figure 11. Using the known clusters yields a performance somewhere in the middle of the 100 twins that is statistically indistinguishable from nearly all twins. This arguably provides the most compelling evidence so far on whether improvements in forecast accuracy can be attributed to structure or grouping. It is the process of forming any middle-level series that is more important than aggregating similar time series. This result holds not only for our empirical studies, but also for an example specifically designed to have distinct, known clusters as illustrated in this section.

6. Forecast combination

The results in Sections 3 and 5 highlight the potential of improving forecast reconciliation performance through the construction of new middle levels series using time series clustering. The selection of the best performing combination of time series representation, distance measure, and clustering algorithm remains an open question. Since the best cluster-based method will heavily depend on the characteristics of the data, we consider averaging forecasts across different hierarchies, as an alternative to hierarchy selection. This is supported by substantial research and empirical evidence in favor of forecast combination over selection (see *e.g.*, Elliott and Timmermann, 2016). Specifically, the reconciled forecasts from multiple hierarchies are combined using equal weights (Wang et al., 2023), *i.e.*,

$$\tilde{\boldsymbol{y}}_{T+h}^{\text{comb}} = \frac{1}{l}\sum_{j=1}^{l} \tilde{\boldsymbol{y}}_{T+h}^{j}.$$

We note that this average can only be carried out using series that are common to all hierarchies, in our case, the top and bottom levels. In this case, since all elements of the average are coherent, any linear combination of these forecasts will also be coherent.

Table 8 presents the accuracy in terms of average RMSSE across all evaluation windows for both datasets. Note that for brevity we only present results for three benchmarks (base, two-level, and natural), the best cluster hierarchy (TS-DTW-HC for the mortality data and TSF-EUC-HC for the tourism data) and the combination hierarchy. The MCB test results are shown in Figure 12. As expected, we observe that on both datasets, forecast combination improves forecast performance compared to any single hierarchy. The improvement on the

Table 8: Average RMSSE across six approaches. Column-wise minimum values are displayed in bold.

Approach	tourism	mortality
Base	0.6945	0.7530
Two-level	0.6944	0.7528
Natural	0.6913	0.7501
TS-DTW-HC	0.6911	0.7496
TSF-EUC-HC	0.6909	0.7509
Combination	**0.6902**	**0.7423**

mortality dataset is more pronounced, with forecast combination significantly outperforming all other approaches.

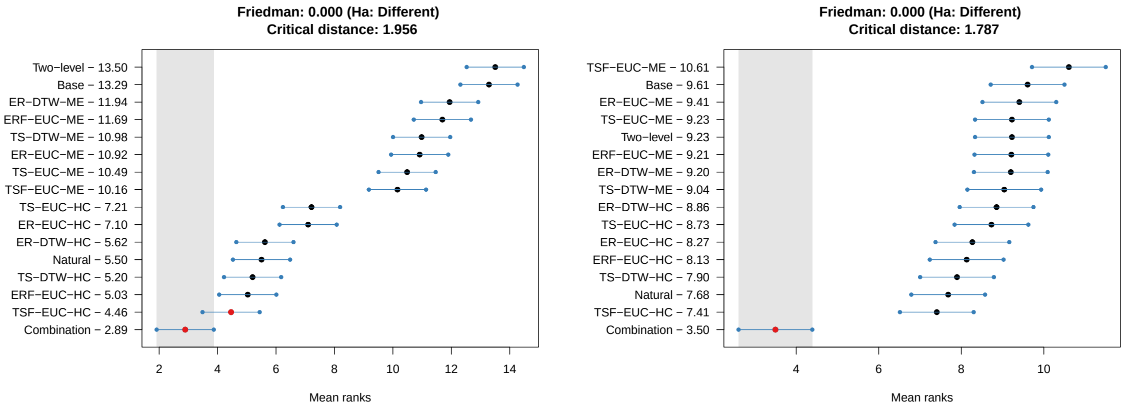

Figure 12: Average ranks and 95% confidence intervals for all approaches on tourism dataset(left) and mortality dataset(right) based on MCB test.

6.1. Combination hierarchy vs its random twins

We also consider how the question of grouping versus structure plays out in the case of combinations by extending the permutation approach introduced in Section 4.1. We do so by first permuting the labels of the bottom level series while keeping the hierarchy fixed. The same permutation is used for all hierarchies in a combination. This is then repeated 100 times, yielding 100 “twins” of the combination forecast.

The results of MCB test for the combination and its 100 random twins are presented in Figure 13 for the mortality dataset[7]. Similar to the results in Section 4, we find evidence that a large number of random twins do not perform significantly worse than an approach based on

[7] Note that this experiment has been conducted only on the mortality dataset, as computing random twins for all cluster hierarchies on the tourism dataset is too computationally expensive.

clustering. In fact, out of the 100 twins, the combination only performs better than a single twin. This provides the last piece of compelling evidence that the grouping of similar time series, contributes less to improved forecast performance than structure.

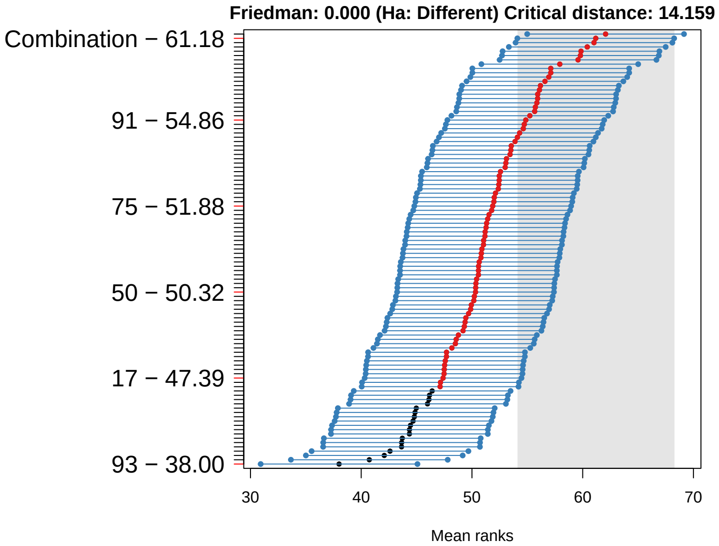

Figure 13: Average ranks and 95% confidence intervals for combination of twelve cluster hierarchies and its 100 random twins on mortality dataset based on MCB test.

7. Conclusion

This paper thoroughly investigates the issue of constructing hierarchies for forecast reconciliation, with the goal of improving forecast accuracy. This issue is particularly important in the absence of a predefined natural hierarchy. Rather than focus on a single clustering algorithm, we consider a more general framework, incorporating three distinct approaches: cluster hierarchies, permutation hierarchies, and combination hierarchies. Unsurprisingly, no single method emerges as superior for all datasets, although hierarchical clustering, which by construction leads to a larger number of clusters, tends to outperform k-mediods clustering.

This naturally begs the question, of why adding more middle level series can improve forecast performance. We devise a method that keeps the hierarchy fixed while permuting the bottom level series, yielding “twin” hierarchies. The bottom-level series that aggregate to a single middle level series are thus chosen at random. Across different datasets and settings, an overwhelmingly large number of twins, perform not significantly worse, or even better, than a hierarchy selected using clustering. This suggests that the improved forecast performance arises not from grouping similar times series together, but rather from features related to the structure of the hierarchy, *e.g.* the larger number of middle level series. We find similar results from a simulation study where clusters are predefined. This eliminates the possibility that our empirical results arise due to spurious clusters or the inadequate performance of a single clustering method.

Our main practical recommendation is to use multiple clustering methods and combine forecasts across these methods using equal weights combination. This mitigates the uncertainty of selecting the best clustering approach and is shown to significantly outperform all benchmarks across both datasets that we consider. One could also extend this idea to averaging over random twins. However, it's worth noting that any averaging approach incurs a computation cost as it requires obtaining base forecasts for additional middle-level series.

Future research based on our study could proceed in several promising directions. For example, while we used equally-weighted combinations in this study, there is potential to apply more sophisticated forecast combination methods to improve performance. The extensive literature on forecast combination, including advanced methods for calculating weights, could also be considered (refer to Wang et al., 2023 for an in-depth review). Also, our results are based on cross-sectional data, this could be extended to explore temporal (Athanasopoulos et al., 2017) and cross-temporal hierarchies (Girolimetto et al., 2023). Finally, more work could be carried out to understand whether some middle levels contribute more to forecast accuracy than others, and accordingly "pruning" the less useful middle level series.

References

Aghabozorgi, S., Seyed Shirkhorshidi, A. and Ying Wah, T. (2015), 'Time-series clustering – A decade review', Information Systems **53**, 16–38.

Athanasopoulos, G., Hyndman, R. J., Kourentzes, N. and Panagiotelis, A. (2023), 'Forecast reconciliation: A review'.
URL: *https://robjhyndman.com/publications/hfreview.html*

Athanasopoulos, G., Hyndman, R. J., Kourentzes, N. and Petropoulos, F. (2017), 'Forecasting with temporal hierarchies', European Journal of Operational Research **262**(1), 60–74.

Athanasopoulos, G. and Kourentzes, N. (2023), 'On the evaluation of hierarchical forecasts', International Journal of Forecasting **39**(4), 1502–1511.

Elliott, G. and Timmermann, A. (2016), 'Forecasting in Economics and Finance', Annual Review of Economics **8**(1), 81–110.

Girolimetto, D., Athanasopoulos, G., Di Fonzo, T. and Hyndman, R. J. (2023), 'Cross-temporal probabilistic forecast reconciliation: Methodological and practical issues', International Journal of Forecasting .

Hollyman, R., Petropoulos, F. and Tipping, M. E. (2021), 'Understanding forecast reconciliation', European Journal of Operational Research **294**(1), 149–160.

Hyndman, R. J., Ahmed, R. A., Athanasopoulos, G. and Shang, H. L. (2011), 'Optimal combination forecasts for hierarchical time series', Computational Statistics & Data Analysis **55**(9), 2579–2589.

Hyndman, R. J., Kang, Y., Montero-Manso, P., Talagala, T., Wang, E., Yang, Y. and O'Hara-Wild, M. (2022), tsfeatures: Time Series Feature Extraction. R package version 1.1.0.9000.
URL: *https://pkg.robjhyndman.com/tsfeatures/*

Hyndman, R. J. and Khandakar, Y. (2008), 'Automatic time series forecasting: the forecast package for R', Journal of Statistical Software **26**(3), 1–22.

Hyndman, R. J., Koehler, A. B., Ord, K. and Snyder, R. D. (2008), Forecasting with Exponential Smoothing: the State Space Approach, Springer Series in Statistics, Springer.

Jeon, J., Panagiotelis, A. and Petropoulos, F. (2019), 'Probabilistic forecast reconciliation with applications to wind power and electric load', European Journal of Operational Research **279**(2), 364–379.

Kaufman, L. and Rousseeuw, P. J. (1990), Partitioning Around Medoids (Program PAM), in 'Finding Groups in Data', John Wiley & Sons, Ltd, chapter 2, pp. 68–125.

Koning, A. J., Franses, P. H., Hibon, M. and Stekler, H. O. (2005), 'The M3 competition: Statistical tests of the results', International Journal of Forecasting **21**(3), 397–409.

Kourentzes, N. and Athanasopoulos, G. (2019), 'Cross-temporal coherent forecasts for Australian tourism', Annals of Tourism Research **75**, 393–409.

Li, H., Li, H., Lu, Y. and Panagiotelis, A. (2019), 'A forecast reconciliation approach to cause-of-death mortality modeling', Insurance: Mathematics and Economics **86**, 122–133.

Li, L., Kang, Y., Petropoulos, F. and Li, F. (2023), 'Feature-based intermittent demand forecast combinations: accuracy and inventory implications', International Journal of Production Research **61**(22), 7557–7572.

Maechler, M., Rousseeuw, P., Struyf, A., Hubert, M. and Hornik, K. (2022), cluster: Cluster Analysis Basics and Extensions. R package version 2.1.4.
URL: *https://cran.r-project.org/web/packages/cluster/index.html*

Makridakis, S., Spiliotis, E. and Assimakopoulos, V. (2022), 'M5 accuracy competition: Results, findings, and conclusions', International Journal of Forecasting **38**(4), 1346–1364.

Mattera, R., Athanasopoulos, G. and Hyndman, R. J. (2023), 'Improving out-of-sample forecasts of stock price indexes with forecast reconciliation and clustering'.
URL: *https://robjhyndman.com/publications/dow_hts.html*

Murtagh, F. and Legendre, P. (2014), 'Ward's Hierarchical Agglomerative Clustering Method: Which Algorithms Implement Ward's Criterion?', Journal of Classification **31**(3), 274–295.

Panagiotelis, A., Gamakumara, P., Athanasopoulos, G. and Hyndman, R. J. (2023), 'Probabilistic forecast reconciliation: Properties, evaluation and score optimisation', European Journal of Operational Research **306**(2), 693–706.

Pang, Y., Yao, B., Zhou, X., Zhang, Y., Xu, Y. and Tan, Z. (2018), Hierarchical Electricity Time Series Forecasting for Integrating Consumption Patterns Analysis and Aggregation Consistency, in 'Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence', pp. 3506–3512.

Pang, Y., Zhou, X., Zhang, J., Sun, Q. and Zheng, J. (2022), 'Hierarchical electricity time series prediction with cluster analysis and sparse penalty', Pattern Recognition **126**, 108555.

R Core Team (2022), R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austria.
URL: *https://www.R-project.org/*

Sakoe, H. and Chiba, S. (1978), 'Dynamic Programming Algorithm Optimization for Spoken Word Recognition', IEEE Transactions on Acoustics, Speech, and Signal Processing **26**(1), 43–49.

Shutaywi, M. and Kachouie, N. N. (2021), 'Silhouette analysis for performance evaluation in machine learning with applications to clustering', Entropy **23**(6), 759.

Syntetos, A. A., Babai, Z., Boylan, J. E., Kolassa, S. and Nikolopoulos, K. (2016), 'Supply chain forecasting: Theory, practice, their gap and the future', European Journal of Operational Research **252**(1), 1–26.

Tiano, D., Bonifati, A. and Ng, R. (2021), FeatTS: Feature-based Time Series Clustering, in 'Proceedings of the 2021 International Conference on Management of Data', SIGMOD '21, Association for Computing Machinery, New York, NY, USA, pp. 2784–2788.

Wang, X., Hyndman, R. J., Li, F. and Kang, Y. (2023), 'Forecast combinations: An over 50-year review', International Journal of Forecasting **39**(4), 1518–1547.

Wang, X., Kang, Y., Petropoulos, F. and Li, F. (2022), 'The uncertainty estimation of feature-based forecast combinations', Journal of the Operational Research Society **73**(5), 979–993.

Welch, W. J. (1990), 'Construction of permutation tests', Journal of the American Statistical Association **85**(411), 693–698.

Wickramasuriya, S. L., Athanasopoulos, G. and Hyndman, R. J. (2019), 'Optimal Forecast Reconciliation for Hierarchical and Grouped Time Series Through Trace Minimization', Journal of the American Statistical Association **114**(526), 804–819.