MONASH University

MONASH BUSINESS SCHOOL

ISSN 1440-771X

Department of Econometrics and Business Statistics

http://monash.edu/business/ebs/research/publications

Fast forecast reconciliation using linear models

Mahsa Ashouri, Rob J Hyndman, Galit Shmueli

July 2020

Working Paper 29/19

AACSB ACCREDITED EFMD EQUIS ACCREDITED ASSOCIATION OF MBAS ACCREDITED

Fast forecast reconciliation using linear models

Mahsa Ashouri
Institute of Service Science, National Tsing Hua University, Taiwan
Email: mahsa.ashouri@iss.nthu.edu.tw
Corresponding author

Rob J Hyndman
Monash University, Clayton VIC 3800, Australia
Email: rob.hyndman@monash.edu

Galit Shmueli
Institute of Service Science, National Tsing Hua University, Taiwan
Email: galit.shmueli@iss.nthu.edu.tw

25 July 2020

JEL classification: C10,C14,C22

Fast forecast reconciliation using linear models

Abstract

Forecasting hierarchical or grouped time series usually involves two steps: computing base forecasts and reconciling the forecasts. Base forecasts can be computed by popular time series forecasting methods such as Exponential Smoothing (ETS) and Autoregressive Integrated Moving Average (ARIMA) models. The reconciliation step is a linear process that adjusts the base forecasts to ensure they are coherent. However using ETS or ARIMA for base forecasts can be computationally challenging when there are a large number of series to forecast, as each model must be numerically optimized for each series. We propose a linear model that avoids this computational problem and handles the forecasting and reconciliation in a single step. The proposed method is very flexible in incorporating external data, handling missing values and model selection. We illustrate our approach using two datasets: monthly Australian domestic tourism and daily Wikipedia pageviews. We compare our approach to reconciliation using ETS and ARIMA, and show that our approach is much faster while providing similar levels of forecast accuracy.

Keywords: hierarchical forecasting, grouped forecasting, reconciling forecast, linear regression

1 Introduction

Modern data collection tools have dramatically increased the amount of available time series data (Januschowski et al. 2013). For example, the internet of things and point-of-sale scanning produce huge volumes of time series in a short period of time. Naturally, there is an interest in forecasting these time series, yet forecasting large collections of time series is computationally challenging.

1.1 Hierarchical and grouped time series

In many cases, these time series can be structured and disaggregated based on hierarchies or groups such as geographic location, product type, gender, etc. An example of hierarchical time series is sales in restaurant chains, which can be disaggregated into different stores and then different types of food or drinks. Figure 1 shows a schematic of such a hierarchical time series structure with three levels. The top level is the total series, formed by aggregating all the bottom level series. In the middle level, series are aggregations of their own child series; for instance, series A is the aggregation of AW and AX. Finally, the bottom level series, includes the most disaggregated series.

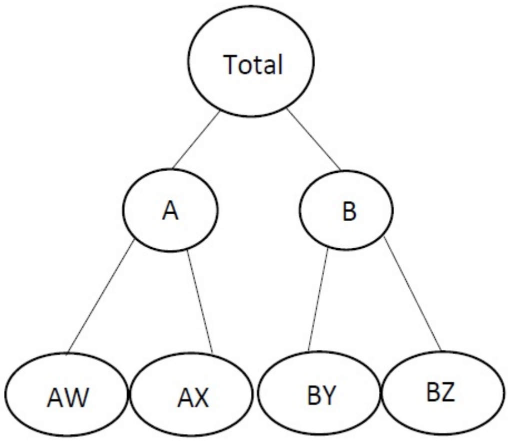

Figure 1: *An example of a two level hierarchical structure.*

Grouped time series involve more complicated aggregation structures compared to strictly hierarchical time series. To take the simplest example, suppose we have two grouping factors which are not nested: sex (Male/Female) and city (New York/San Francisco). The disaggregated series for each combination of sex and city can be combined to form city sub-totals, or sex sub-totals. These sub-totals can be combined to give the overall total. Both sub-totals are of interest.

We can think of such structures as hierarchical time series without a unique hierarchy. A schematic of this grouped time series structure is shown in Figure 2 with two grouping factors, each of two levels (A/B and C/D). The series in this structure can be split first into groups A and B and then subdivided further into C and D (left side), or split first into C and D and then subdivided into A and B (right side). The final disaggregation is identical in both cases, but the middle level aggregates are different.

We use the same notation (following Hyndman & Athanasopoulos 2018) for both hierarchical and grouped time series. We denote the total series at time t by y_t, and the series at node Z (subaggregation level Z) and time t by $y_{Z,t}$. For describing the relationships between series, we use an $N \times M$ matrix, called the "summing matrix", denoted by $\boldsymbol{S}$, in which N is the overall number of nodes and M is the number of bottom level nodes. For example in Figure 1, $N = 7$ and $M = 4$, while in Figure 2, $N = 9$ and $M = 4$. Then we can write $\boldsymbol{y}_t = \boldsymbol{S}\boldsymbol{b}_t$, where $\boldsymbol{y}_t$ is a vector of all the level nodes at time t and $\boldsymbol{b}_t$ is the vector of all the bottom level nodes at time t.

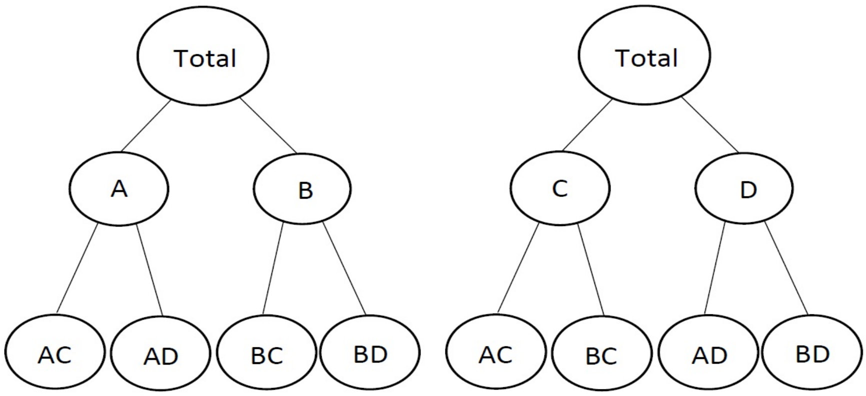

Figure 2: *An example of a two level grouped structure.*

For the example shown in Figure 2, the equation can be written as follows:

$$\begin{pmatrix} y_t \\ y_{A,t} \\ y_{B,t} \\ y_{C,t} \\ y_{D,t} \\ y_{AC,t} \\ y_{AD,t} \\ y_{BC,t} \\ y_{BD,t} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} y_{AC,t} \\ y_{AD,t} \\ y_{BC,t} \\ y_{BD,t} \end{pmatrix}.$$

1.2 Forecasting hierarchical time series

If we just forecast each series individually, we are ignoring the hierarchical or grouping structure, and the forecasts will not be "coherent" (they will not add up appropriately). This means we need the forecasts to add up in a way which is consistent with the aggregation structure of the time series collection (Hyndman & Athanasopoulos 2018).

There are several available methods that consider the hierarchical structure information when forecasting time series. These include the top-down (Gross & Sohl 1990; Fliedner 2001), bottom-up (Kahn 1998), middle-out and optimal combination (Hyndman et al. 2011) approaches. In the top-down approach, we first forecast the total series and then disaggregate the forecast to form lower level series forecasts based on a set of historical and forecasted proportions (for details see Athanasopoulos, Ahmed & Hyndman 2009). In the bottom-up approach, the forecasts in each level of the hierarchy can be computed by aggregating the bottom level series forecasts. However, we may not get good upper-level forecasts because the most disaggregated series can be noisy and so their forecasts are often inaccurate. In the middle-out approach, the process can be started from one of the middle levels and other forecasts can be computed using aggregation for upper levels and disaggregation for lower levels. Finally, optimal combination uses all the N forecasts for all of the series in the entire structure, and then uses an optimization process to reconcile the resulting forecasts. The advantage of the optimal combination method, compared with the other methods, is that it considers all information in the hierarchy, including any correlations among the series.

In the optimal combination method, reconciled forecasts can be computed using the following equation known as weighted least squares (WLS) (Wickramasuriya, Athanasopoulos & Hyndman 2019)

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

where $\hat{\boldsymbol{y}}_h$ represents a vector of h-step-ahead base forecasts for all levels of the hierarchy, and $\boldsymbol{W}_h$ is the covariance matrix of forecast errors for the h-step-ahead base forecasts.

Several possible simple methods for estimating $\boldsymbol{W}_h$ are available. Wickramasuriya, Athanasopoulos & Hyndman (2019) discuss a simple approximation whereby $\boldsymbol{W}_h = k_h\boldsymbol{\Lambda}$ with k_h being a positive constant, $\boldsymbol{\Lambda} = \text{diag}(\boldsymbol{S}\mathbf{1})$, and $\mathbf{1}$ being a column of 1s. Note that $\boldsymbol{\Lambda}$ simply contains the row sums of the summing matrix $\boldsymbol{S}$, and that k_h will cancel out in (1). Thus

$$\tilde{\boldsymbol{y}}_h = \boldsymbol{S}(\boldsymbol{S}'\boldsymbol{\Lambda}^{-1}\boldsymbol{S})^{-1}\boldsymbol{S}'\boldsymbol{\Lambda}^{-1}\hat{\boldsymbol{y}}_h. \tag{2}$$

The most computationally challenging part of the optimal combination method is to produce all the base forecasts that make up $\hat{\boldsymbol{y}}_h$. In many applications, there may be thousands or even millions of individual series, and each of them must be forecast independently. The most popular time series forecasting methods such as ETS and ARIMA models (Hyndman & Athanasopoulos 2018) involve non-linear optimization routines to estimate the parameters via maximum likelihood estimation. Usually, multiple models are fitted for each series, and the best is selected by minimizing Akaike's Information Citerion (Akaike 1998). This computational challenges increases with the number of lower level series as well as in the number of aggregations of interest.

We therefore propose a new approach to compute the base forecasts that is both computationally fast while maintaining an acceptable forecasting accuracy level.

2 Proposed approach: Linear model

Our proposed approach is based on using linear regression models for computing base forecasts. Suppose we have a linear model that we use for forecasting, and we wish to apply it to N different series which have some aggregation constraints. We have observations $y_{t,i}$ from times $t = 1, \dots, T$ and series $i = 1, \dots, N$. Then

$$y_{t,i} = \boldsymbol{\beta}_i' \boldsymbol{x}_{t,i} + \varepsilon_{t,i}$$

where $\boldsymbol{x}_{t,i} = \{1, x_{t,i,1}, \dots, x_{t,i,p}\}$ is a $(p+1)$-vector of regression variables. This equation for all the observations in matrix form can be written as follows:

$$\begin{pmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \\ \boldsymbol{y}_3 \\ \vdots \\ \boldsymbol{y}_N \end{pmatrix} = \begin{pmatrix} \boldsymbol{X}_1 & 0 & 0 & \dots & 0 \\ 0 & \boldsymbol{X}_2 & 0 & \dots & 0 \\ 0 & 0 & \boldsymbol{X}_3 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \dots & 0 & \boldsymbol{X}_N \end{pmatrix} \begin{pmatrix} \boldsymbol{\beta}_1 \\ \boldsymbol{\beta}_2 \\ \boldsymbol{\beta}_3 \\ \vdots \\ \boldsymbol{\beta}_N \end{pmatrix} + \begin{pmatrix} \boldsymbol{\varepsilon}_1 \\ \boldsymbol{\varepsilon}_2 \\ \boldsymbol{\varepsilon}_3 \\ \vdots \\ \boldsymbol{\varepsilon}_N \end{pmatrix}, \tag{3}$$

where $\boldsymbol{y}_i = \{y_{1,i}, y_{2,i}, \dots, y_{T,i}\}$ is a T-vector, $\boldsymbol{\beta}_i = \{\beta_{0,i}, \beta_{1,i}, \beta_{2,i}, \dots, \beta_{p,i}\}$ is a $(p+1)$-vector, $\boldsymbol{\varepsilon}_i = \{\varepsilon_{1,i}, \varepsilon_{2,i}, \dots, \varepsilon_{T,i}\}$ is a T-vector and $\boldsymbol{X}_i$ is the $T \times (p+1)$-matrix

$$\boldsymbol{X}_i = \begin{pmatrix} 1 & x_{1,i,1} & x_{1,i,2} & \dots & x_{1,i,p} \\ 1 & x_{2,i,1} & x_{2,i,2} & \dots & x_{2,i,p} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{T,i,1} & x_{T,i,2} & \dots & x_{T,i,p} \end{pmatrix}.$$

Equation (3) can be written as $\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{B} + \boldsymbol{E}$, with parameter estimates given by $\hat{\boldsymbol{B}} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{Y}$. Then the base forecasts are obtained using

$$\hat{\boldsymbol{y}}_{t+h} = \boldsymbol{X}^*_{t+h}\hat{\boldsymbol{B}}, \tag{4}$$

where $\hat{\boldsymbol{y}}_{t+h}$ is an N-vector of forecasts, $\hat{\boldsymbol{B}}$ comprises N stacked $(p+1)$-vectors of estimated coefficients, and $\boldsymbol{X}^*_{t+h}$ is the $N \times N(p+1)$ matrix

$$\boldsymbol{X}^*_{t+h} = \begin{pmatrix} \boldsymbol{x}'_{t+h,1} & 0 & 0 & \dots & 0 \\ 0 & \boldsymbol{x}'_{t+h,2} & 0 & \dots & 0 \\ 0 & 0 & \boldsymbol{x}'_{t+h,3} & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \dots & 0 & \boldsymbol{x}'_{t+h,N} \end{pmatrix}.$$

Note that we use $\boldsymbol{X}^*_t$ to distinguish this matrix, which combines $\boldsymbol{x}_{t,i}$ across all series for one time from $\boldsymbol{X}_i$ which combines $\boldsymbol{x}_{t,i}$ across all time for one series.

Finally, we can combine the two linear equations for computing base forecasts and reconciled forecasts (Equations (2) and (4)) to obtain the reconciled forecasts with a single equation:

$$\tilde{\boldsymbol{y}}_{t+h} = \boldsymbol{S}(\boldsymbol{S}'\boldsymbol{\Lambda}\boldsymbol{S})^{-1}\boldsymbol{S}'\boldsymbol{\Lambda}(\boldsymbol{X}^*_{t+h}\hat{\boldsymbol{B}}) = \boldsymbol{S}(\boldsymbol{S}'\boldsymbol{\Lambda}\boldsymbol{S})^{-1}\boldsymbol{S}'\boldsymbol{\Lambda}\boldsymbol{X}^*_{t+h}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{Y}. \tag{5}$$

2.1 Simplified formulation for a fixed set of predictors (X)

If we have the same set of predictor variables, $\boldsymbol{X}$, for all the series, we can write Equations (3) to (5) more easily using multivariate regression equations, and we can obtain all the reconciled forecasts for all the series in one equation. In that case, Equation (3) can be rearranged as follows:

$$\begin{pmatrix} y_{11} & \dots & y_{1N} \\ y_{21} & \dots & y_{2N} \\ \vdots & & \vdots \\ y_{T1} & \dots & y_{TN} \end{pmatrix} = \begin{pmatrix} 1 & X_{11} & \dots & X_{1p} \\ 1 & X_{21} & \dots & X_{2p} \\ \vdots & \vdots & & \vdots \\ 1 & X_{T1} & \dots & X_{Tp} \end{pmatrix} \begin{pmatrix} \beta_{01} & \dots & \beta_{0N} \\ \beta_{11} & \dots & \beta_{1N} \\ \vdots & & \vdots \\ \beta_{p1} & \dots & \beta_{pN} \end{pmatrix} + \begin{pmatrix} \varepsilon_{11} & \dots & \varepsilon_{1N} \\ \varepsilon_{21} & \dots & \varepsilon_{2N} \\ \vdots & & \vdots \\ \varepsilon_{T1} & \dots & \varepsilon_{TN} \end{pmatrix}, \tag{6}$$

where $\boldsymbol{Y}$, $\boldsymbol{X}$, $\boldsymbol{B}$ and $\boldsymbol{E}$ are now matrices of size $T \times N$, $T \times (p+1)$, $(p+1) \times N$ and $T \times N$, respectively. Equations (4) to (5) can be written accordingly using Equation (6) and here $\boldsymbol{X}^*_{t+h,i} = \boldsymbol{X}^*_{t+h}$, where $\boldsymbol{X}^*_{t+h}$ is an $h \times (p+1)$ matrix.

2.2 OLS predictors

As an example of the $\boldsymbol{X}_t$ matrix in Equation (3), we can refer to the set of predictors proposed in Ashouri, Shmueli & Sin (2018) for modeling trend, seasonality and autocorrelation by using lagged values ($y_{t-1}, y_{t-2}, \dots$), trend variables and seasonal dummy variables:

$$y_t = \alpha_0 + \alpha_1 t + \beta_1 s_{1,t} + \dots + \beta_{m-1} s_{m-1,t} + \gamma_1 y_{t-1} + \dots + \gamma_p y_{t-p} + \delta z_t + \varepsilon_t. \tag{7}$$

Here, $s_{j,t}$ is a dummy variable taking value 1 if time t is in season j ($j = 1, 2, \dots, m$), y_{t-k} is the kth lagged value for y_t and z_t is some external information at time t. The seasonal period m

depends on the problem; for instance, if we have daily data with day-of-week seasonality, then $m = 7$.

Because of using lags and external series as predictors in Equation (7), we do not have same set of predictors for all the series, y_t. However, if we just use trend and seasonality dummies as the predictors, then the simpler equations, Equation (6), can be written using multivariate regression models.

When there are many options for choosing predictors, such as many seasonal dummy variables, lags, or high order trend terms, we can consider applying a model selection approach such as leave-one-out cross-validation (LOOCV) for time series to select the best set of predictors in terms of prediction. In practice, LOOCV can be computationally heavy, and therefore using linear models provide a viable solution. Also, when the number of seasons m is large (e.g. in hourly data), Fourier terms can sometimes result in fewer predictors than dummy variables. The number of Fourier terms can also be determined using the same LOOCV approach (Hyndman & Athanasopoulos 2018).

While OLS is popular in practice for forecasting time series, it is often frowned upon due to its independence assumption. This can cause issues for parametric inference but is less of a problem for forecasting. In fact it often performs sufficiently well for forecasting as can be seen by its popular use in practice. Further, the use of autoregressive terms in the above model should model most of the autocorrelation in the data.

2.3 Computational considerations

There are two ways for computing the above forecasts. First, we could create the matrices $\boldsymbol{Y}$, $\boldsymbol{X}$ and $\boldsymbol{E}$, and then directly use the above equations (taking advantage of sparse matrix routines) to obtain the forecasts. Alternatively, we could use separate regression models to compute the coefficients for each linear model individually. Although the matrix, $\boldsymbol{X}'\boldsymbol{X}$, which we need to invert is sparse and block diagonal, it is still faster to use the second approach involving separate regression models.

2.4 Prediction intervals

For obtaining prediction intervals, we need to compute the variance of reconciled forecasts as follows (Wickramasuriya, Athanasopoulos & Hyndman 2019):

$$\mathrm{Var}(\tilde{\boldsymbol{y}}_{t+h}) = \boldsymbol{S}\boldsymbol{P}\boldsymbol{\Sigma}_{t+h}\boldsymbol{P}'\boldsymbol{S}', \tag{8}$$

where $\boldsymbol{P} = (\boldsymbol{S}'\boldsymbol{\Lambda}\boldsymbol{S})^{-1}\boldsymbol{S}'\boldsymbol{\Lambda}$ and $\boldsymbol{\Sigma}_{t+h}$ denotes the variance of the base forecasts given by the usual linear model formula (Hyndman & Athanasopoulos 2018)

$$\boldsymbol{\Sigma}_{t+h} = \sigma^2 \left[1 + \boldsymbol{X}^*_{t+h}(\boldsymbol{X}'\boldsymbol{X})^{-1}(\boldsymbol{X}^*_{t+h})'\right].$$

Where the σ^2 is the $N \times N$ diagonal matrix of forecasted error variances. Assuming normally distributed errors, we can easily obtain any required prediction intervals corresponding to elements of $\tilde{\boldsymbol{y}}_{t+h}$ using the diagonals of (8).

3 Applications

In this section we illustrate our approach using two real data examples and one simulated dataset example[1]. The real examples study includes forecasting monthly Australian domestic tourism and forecasting daily Wikipedia pageviews. In the simulation studies, we simulate series based on the monthly Australian domestic tourism data and modify the forecasting horizon, noise level, hierarchy levels, and number of series.. We compare the forecasting accuracy of ETS, ARIMA[2] and the proposed linear OLS forecasting model, with and without the reconciliation step. In these applications, we used the weighted reconciliation approach from Equation (2). For comparing these methods, we use the average of Root Mean Square Errors (RMSEs) across all series and also display box plots for forecast errors along with the raw forecast errors. For visibility, we suppress plotting the outliers.

The two real datasets differ in terms of structure, size and behavior. The tourism data contains 304 series with both hierarchical and grouped structure, while the Wikipedia pageviews dataset contains 913 series with grouped structure. The tourism dataset has strong seasonality while the Wikipedia data are noiser.

We apply two methods for generating forecasts that align with two different practical forecasting scenarios. The first approach is *rolling origin* forecasting, where we generate one-step-ahead forecasts ($\tilde{\boldsymbol{y}}_{t+h}$ where t changes). This assumes a scenario where data are refreshed every time period. In the second *fixed origin* method, forecasts are generated at fixed time t for h steps ahead: $\tilde{\boldsymbol{y}}_{t+1}$, $\tilde{\boldsymbol{y}}_{t+2}$,..., $\tilde{\boldsymbol{y}}_{t+h}$ (we replace lagged values of y by their forecasts if they occur at periods after the forecast origin).

[1] All methods were run on a Linux server with Intel Xeon Silver 4108 (1.80GHz / 8-Cores / 11MB Cache)*2 and 8GB DDR4 2666 DIMM ECC Registered Memory. R version 1.2.5019.

[2] For running ETS and ARIMA, we applied 'ets' and 'auto.arima' functions in 'forecast' package (Hyndman et al. 2020). Also, The two sets of functions were run independently and not immediately one after the other.

3.1 Australian domestic tourism

This dataset has 19 years of monthly visitor nights in Australia by Australian tourists, a measure used as an indicator of tourism activity (Wickramasuriya, Athanasopoulos & Hyndman 2019). The data were collected by computer-assisted telephone interviews with 120,000 Australians aged 15 and over (Tourism Research Australia 2005). The dataset includes 304 time series each of length 228 observations. The hierarchy and grouping structure for this dataset is made using geographic and purpose of travel information.

Table 1: *Australia geographic hierarchical structure.*

Series	Name	Label	Series	Name	Label
Total			Region		
1	Australia	Total	55	Lakes	BCA
State			56	Gippsland	BCB
2	NSW	A	57	Phillip Island	BCC
3	VIC	B	58	General Murray	BDA
4	QLD	C	59	Goulburn	BDB
5	SA	D	60	High Country	BDC
6	WA	E	61	Melbourne East	BDD
7	TAS	F	62	Upper Yarra	BDE
8	NT	G	63	Murray East	BDF
Zone			64	Wimmera+Mallee	BEA
9	Metro NSW	AA	65	Western Grampians	BEB
10	Nth Coast NSW	AB	66	Bendigo Loddon	BEC
11	Sth Coast NSW	AC	67	Macedon	BED
12	Sth NSW	AD	68	Spa Country	BEE
13	Nth NSW	AE	69	Ballarat	BEF
14	ACT	AF	70	Central Highlands	BEG
15	Metro VIC	BA	71	Gold Coast	CAA
16	West Coast VIC	BB	72	Brisbane	CAB
17	East Coast VIC	BC	73	Sunshine Coast	CAC
18	Nth East VIC	BD	74	Central Queensland	CBA
19	Nth West VIC	BE	75	Bundaberg	CBB
20	Metro QLD	CA	76	Fraser Coast	CBC
21	Central Coast QLD	CB	77	Mackay	CBD
22	Nth Coast QLD	CC	78	Whitsundays	CCA
23	Inland QLD	CD	79	Northern	CCB
24	Metro SA	DA	80	Tropical North Queensland	CCC
25	Sth Coast SA	DB	81	Darling Downs	CDA
26	Inland SA	DC	82	Outback	CDB
27	West Coast SA	DD	83	Adelaide	DAA

28	West Coast WA	EA	84	Barossa	DAB
29	Nth WA	EB	85	Adelaide Hills	DAC
30	Sth WA	EC	86	Limestone Coast	DBA
31	Sth TAS	FA	87	Fleurieu Peninsula	DBB
32	Nth East TAS	FB	88	Kangaroo Island	DBC
33	Nth West TAS	FC	89	Murraylands	DCA
34	Nth Coast NT	GA	90	Riverland	DCB
35	Central NT	GB	91	Clare Valley	DCC
Region			92	Flinders Range and Outback	DCD
36	Sydney	AAA	93	Eyre Peninsula	DDA
37	Central Coast	AAB	94	Yorke Peninsula	DDB
38	Hunter	ABA	95	Australia's Coral Coast	EAA
39	North Coast NSW	ABB	96	Experience Perth	EAB
40	Northern Rivers Tropical NSW	ABC	97	Australia's SouthWest	EAC
41	South Coast	ACA	98	Australia's North West	EBA
42	Snowy Mountains	ADA	99	Australia's Golden Outback	ECA
43	Capital Country	ADB	100	Hobart and the South	FAA
44	The Murray	ADC	101	East Coast	FBA
45	Riverina	ADD	102	Launceston, Tamar and the North	FBB
46	Central NSW	AEA	103	North West	FCA
47	New England North West	AEB	104	Wilderness West	FCB
48	Outback NSW	AEC	105	Darwin	GAA
49	Blue Mountains	AED	106	Kakadu Arnhem	GAB
50	Canberra	AFA	107	Katherine Daly	GAC
51	Melbourne	BAA	108	Barkly	GBA
52	Peninsula	BAB	109	Lasseter	GBB
53	Geelong	BAC	110	Alice Springs	GBC
54	Western	BBA	111	MacDonnell	GBD

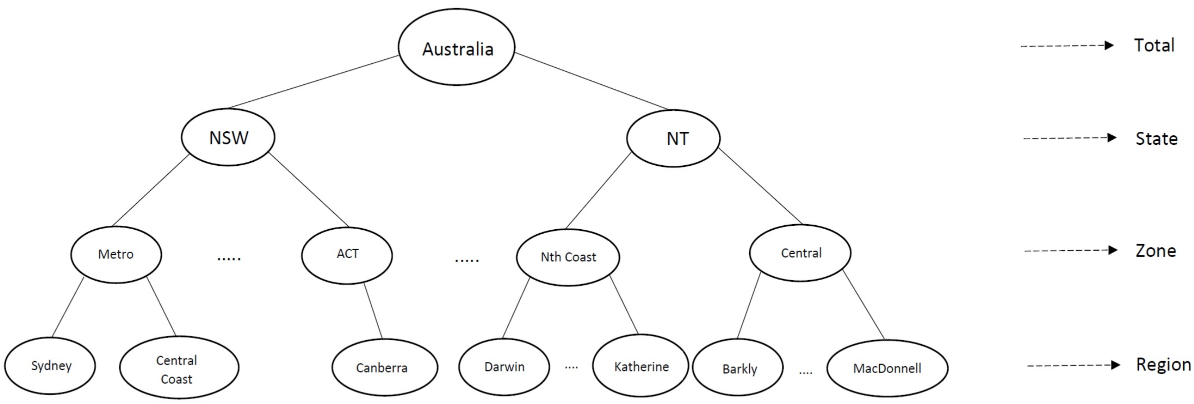

Figure 3: *Australian geographic hierarchical structure.*

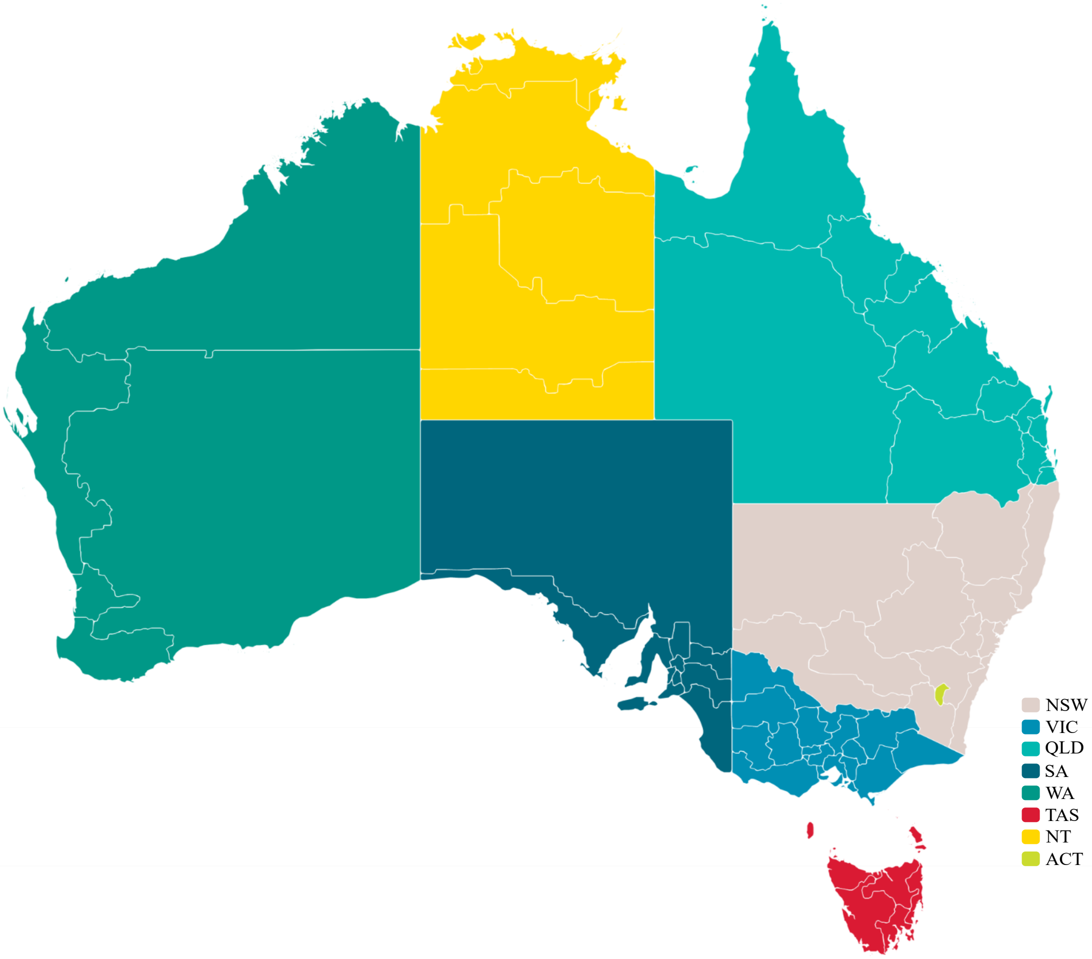

Figure 4: *Australia tourism region map - colors represent states.*

In this dataset we have three levels of geographic divisions in Australia. In the first level, Australia is divided into seven "States" including New South Wales (NSW), Victoria (VIC), Queensland (QLD), South Australia (SA), Western Australia (WA), Tasmania (TAS) and Northern Territory (NT). In the second and third levels, it is divided into 27 "Zones" and 76 "Regions" (for details about Australia geographic divisions see Figure 3 and Table 1 and also Figure 4 which shows Australia map divided by tourism region and colored by states[3]).

We have four purposes of travel: Holiday (Hol), Visiting friends and relatives (Vis), Business (Bus) and Other (Oth). So there are $76 \times 4 = 304$ series at the most disaggregate level. Based on the geographic hierarchy and purpose grouping, we end up with 8 aggregation levels with 555 series in total as shown in Table 2.

We report the forecast results for all these aggregation levels, as well as the average RMSE across all the levels of the hierarchy. We used different predictors in the OLS predictor matrix for the rolling and fixed origin approaches. For the rolling and fixed origin model, we include a

[3]www.tra.gov.au/tra/2016/Tourism_Region_Profiles/Region_profiles/index.html

Table 2: *Number of Australian domestic tourism series at each aggregation level.*

Division	Series
Australia	1
State	7
Zone	27
Region	76
Purpose	4
State x Purpose	28
Zone x Purpose	108
Region x Purpose	304
Total	555

linear trend, 11 dummy variables, and lags 1 and 12. This is intended to capture the monthly seasonality. In addition, before running the model, we partition the data into training and test sets, with the last 24 months (2 years) as our test set, and the rest as our training set.

Table 3: *Mean(RMSE) on 2 year test set for ETS, ARIMA and OLS with and without reconciliation - Rolling origin - Tourism dataset*

	Unreconciled			Reconciled		
Level	ETS	ARIMA	OLS	ETS	ARIMA	OLS
Total	1516.4	1445.5	2191.0	1517.2	1517.2	2194.2
State	511.4	493.1	593.9	499.9	499.9	561.0
Zone	214.8	219.0	233.7	209.6	209.6	219.1
Region	122.9	125.1	125.7	119.4	119.4	121.0
Purpose	676.0	709.2	780.8	674.2	674.2	785.6
State x Purpose	213.1	220.1	230.6	212.7	212.7	221.4
Zone x Purpose	97.5	102.4	101.5	96.8	96.8	98.1
Region x Purpose	56.2	58.2	57.4	56.2	56.2	56.3

Table 4: *Mean(RMSE) on 2 year test set for ETS, ARIMA and OLS with and without reconciliation - Fixed origin - Tourism dataset*

	Unreconciled			Reconciled		
Level	ETS	ARIMA	OLS	ETS	ARIMA	OLS
Total	2238.6	3554.0	3872.8	2232.8	3460.3	3876.8
State	593.6	570.1	788.5	555.7	658.5	776.7
Zone	239.5	229.6	273.1	235.3	249.8	265.2
Region	132.6	129.4	142.5	127.6	132.4	139.1
Purpose	766.8	824.0	1171.6	801.7	1019.3	1169.4
State x Purpose	226.7	241.2	277.0	224.5	245.6	268.8
Zone x Purpose	103.0	105.4	110.3	102.4	105.8	108.3
Region x Purpose	59.1	58.8	61.5	58.8	59.3	60.8

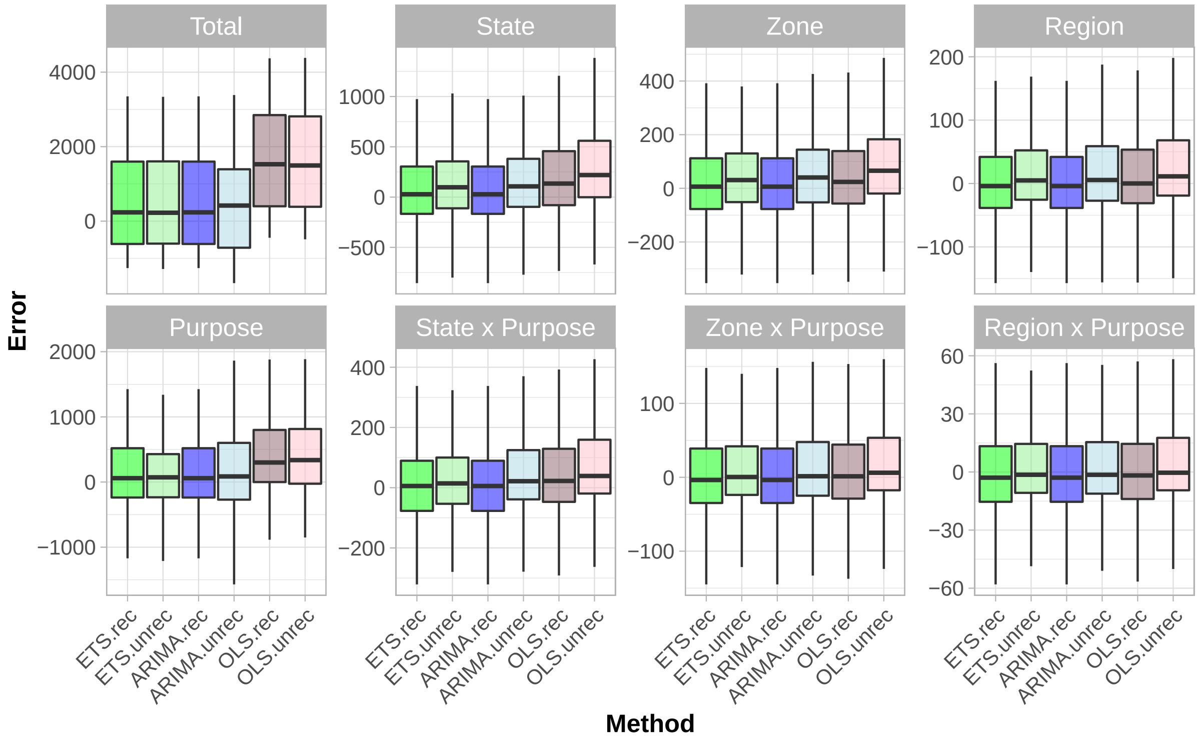

Figure 5: *Box plots of rolling origin forecast errors from reconciled and unreconciled ETS, ARIMA and OLS methods at each hierarchical level for tourism demand.*

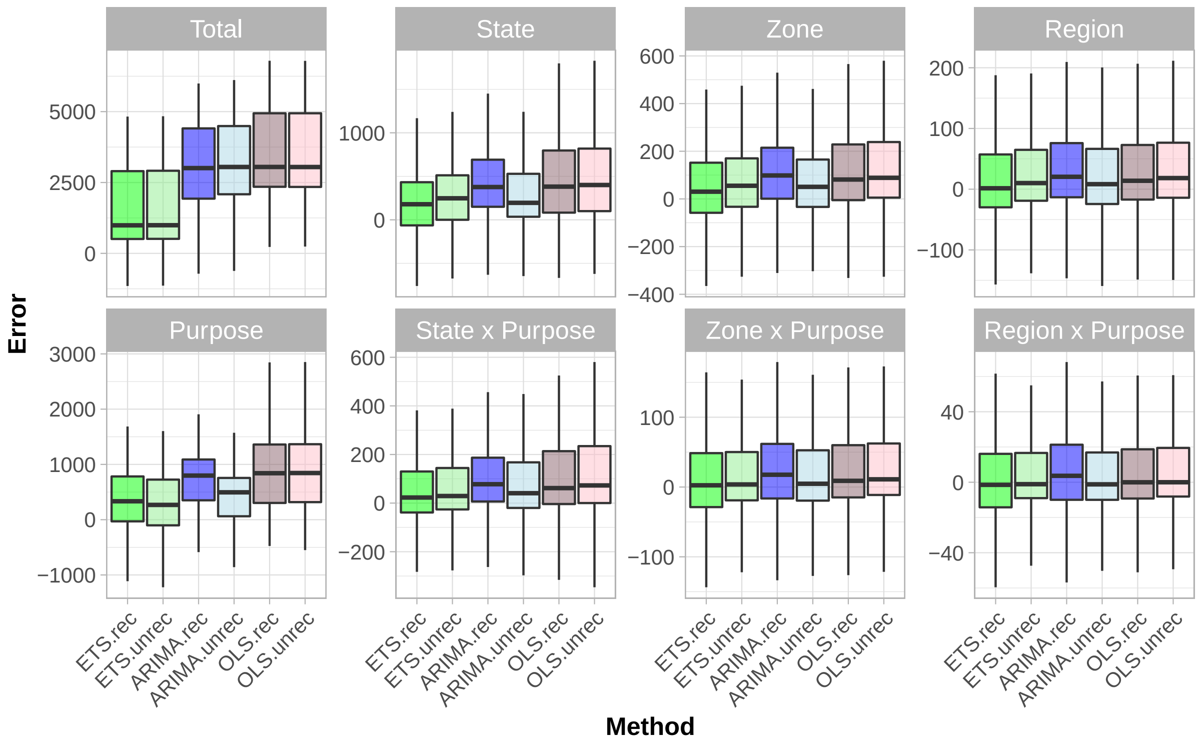

Figure 6: *Box plots of fixed origin forecast errors for reconciled and unreconciled ETS, ARIMA and OLS methods at each hierarchical level for tourism demand.*

In Figures 5 and 6 we display the error box plots for both reconciled and unreconciled forecasts using all three methods, for the rolling origin and fixed origin forecasts. In these figures we see the error distributions across all the models.

Together with Tables 3 and 4, results show that our proposed OLS forecasting model produces forecast accuracy similar to ETS and ARIMA, which are computationally heavy for many time series (see Table 5). We also see the usefulness of the reconciliation in decreasing the average RMSE in all three methods. Except for the total series, reconciliation improves forecasts in all the hierarchy levels. Also, because the higher level series have higher counts, the errors are larger in magnitude (Appendix A shows the box plots with scaled errors[4], to better compare errors across all the hierarchy levels). In addition, we see that (as expected) by applying rolling origin 1-step-ahead forecasts, the error densities are closer and more tightly distributed around zero than the fixed origin multi-step-ahead forecasts.

Figures 7 and 8 show the rolling and fixed origin forecast results for the total series and one of the bottom level series, BACBus (Geelong - Business). In these plots we have both reconciled (dashed lines) and unreconciled (dotted lines) forecasts and we see that the reconciliation step improves the forecasts in this series. We also see that the OLS model forecast accuracy is similar to the other two methods.

Figures 9 and 10 display the prediction interval for the OLS approach, with and without reconciliation forecasts for the total series and one of the bottom level series, BACBus (Geelong - Business).

Table 5 compares the computation time of the three methods for rolling and fixed origin forecasting. We see that the OLS forecasting model is much faster compared to the other methods. Also, since reconciliation is a linear process, in all methods it is very fast and does not affect computation time significantly.

Table 5: *Computation time (seconds) for ETS, ARIMA and OLS with and without reconciliation - Rolling and fixed origin forecasts on a 24 month test set - Tourism dataset*

	Rolling origin		Fixed origin	
	Unreconciled	Reconciled	Unreconciled	Reconciled
ETS	10924.57	10924.60	407.10	407.15
ARIMA	31146.38	31146.52	1116.15	1116.19
OLS	48.40	48.31	17.42	17.80

[4] Scaled errors are computed by subtracting the mean and dividing by the standard deviation.

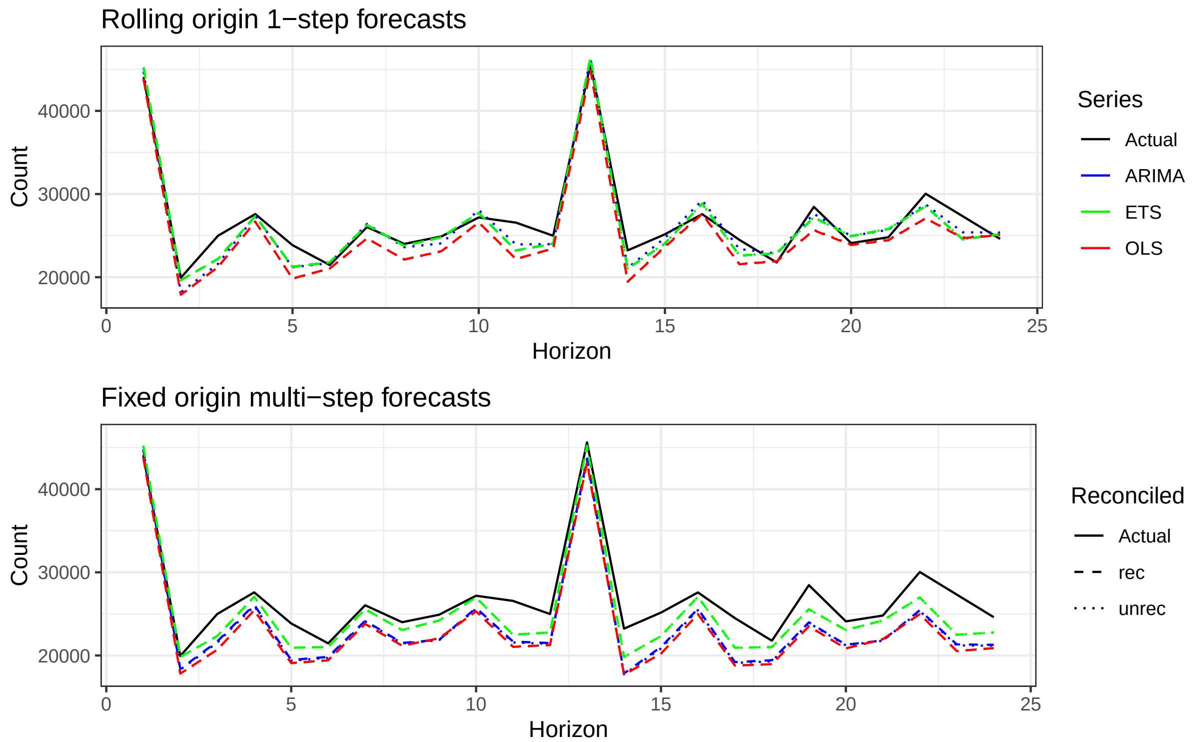

Figure 7: *The actual test set for the 'Total series' compared to the forecasts from reconciled and unreconciled ETS, ARIMA and OLS methods for rolling and fixed origin tourism demand.*

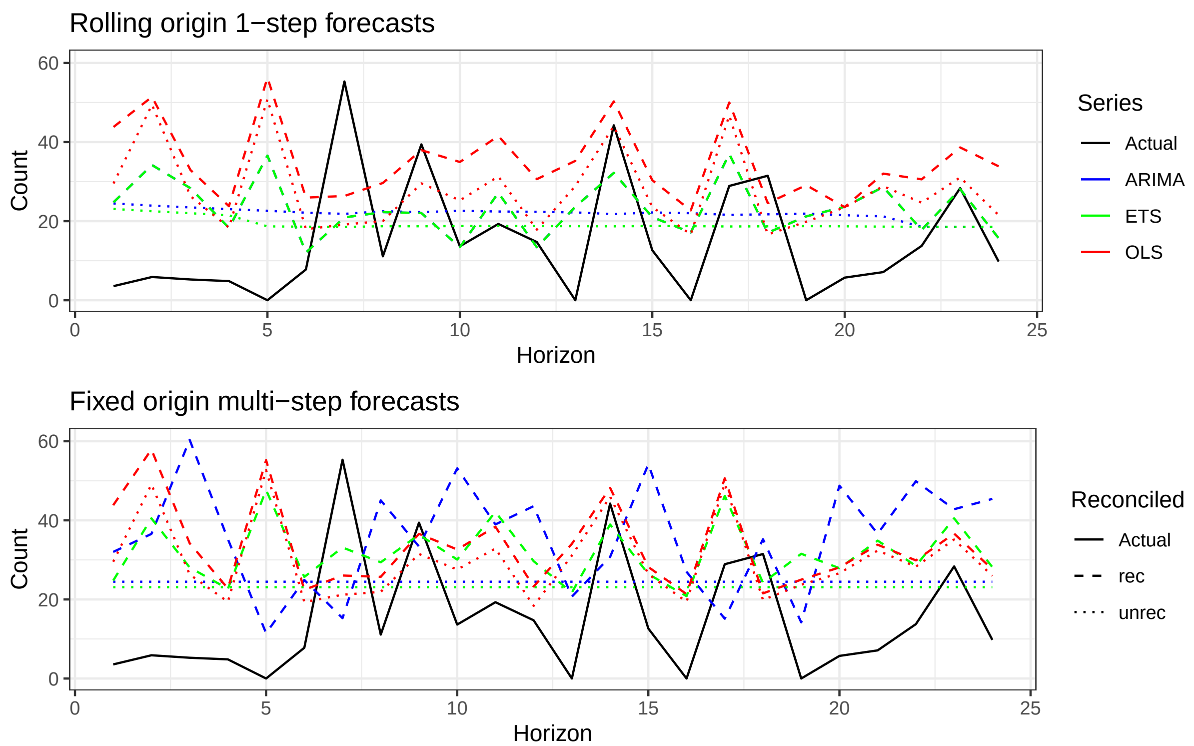

Figure 8: *The actual test set for the 'BACBus' bottom level series compared to the forecasts from reconciled and unreconciled ETS, ARIMA and OLS methods for rolling and fixed origin tourism demand.*

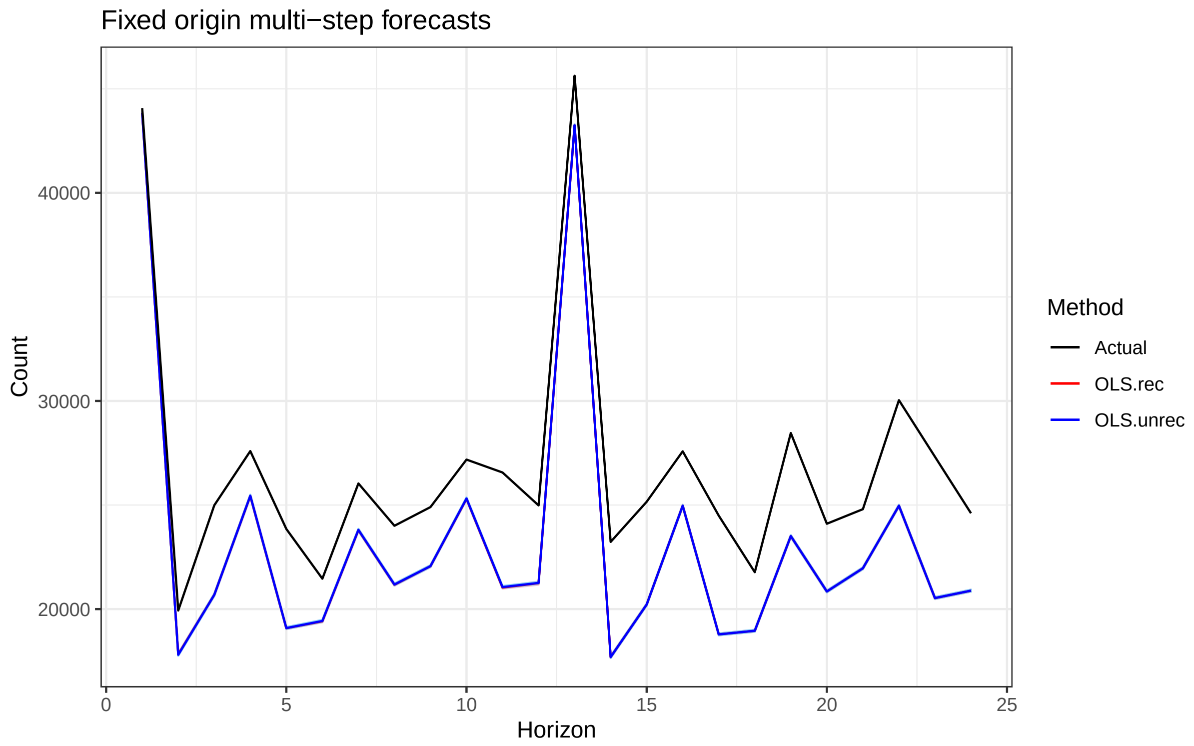

Figure 9: *The actual test set for the 'Total series' compared to the forecasts from reconciled and unreconciled OLS methods with prediction interval for fixed origin tourism demand.*

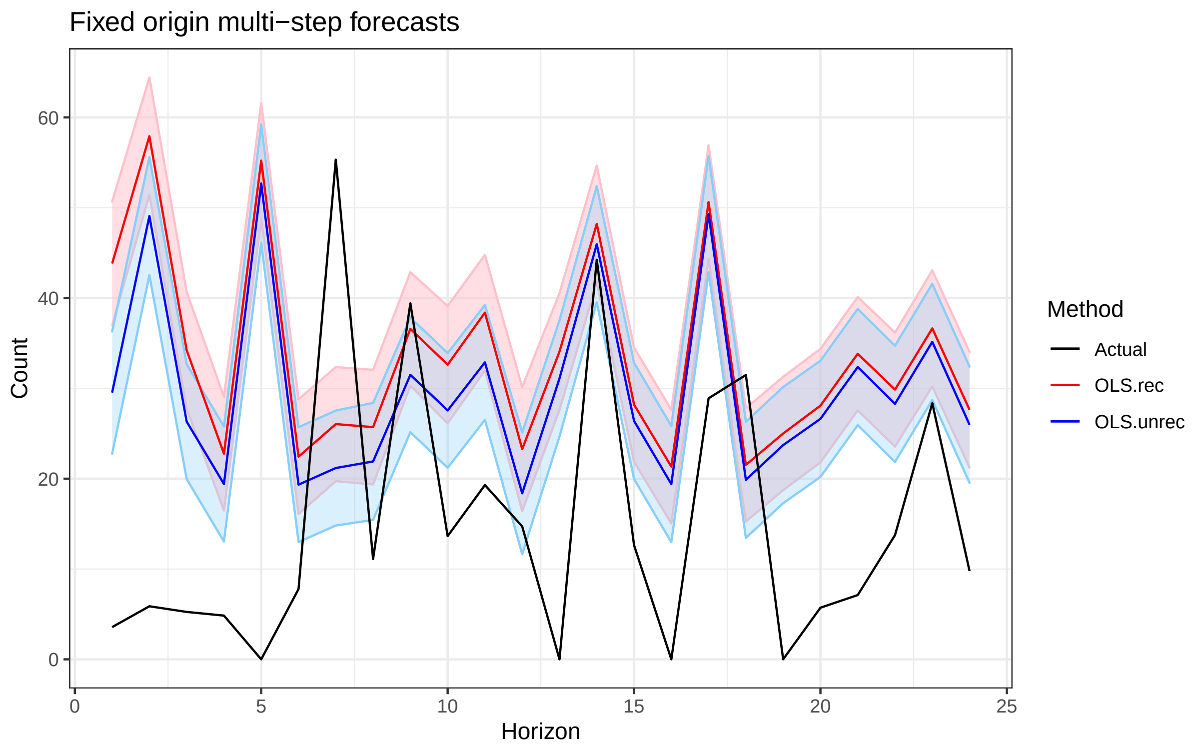

Figure 10: *The actual test set for the 'BACBus' bottom level series compared to the forecasts from reconciled and unreconciled OLS methods with prediction interval for fixed origin tourism demand.*

Since we are using a linear model, we can easily include exogenous variables which can often be helpful in improving forecast accuracy. In this application, we tried including an "Easter" dummy variable indicating the timing of Easter, but its affect on forecast accuracy was minimal, so it was omitted in the model reported here.

Finally, Table 6 shows that, as mentioned in Section 2.3, computation is faster using separate regression models compared to the matrix approach (even using sparse matrix algebra).

Table 6: *Computation time (seconds) for OLS using the matrix approach and separate regression models, with and without reconciliation, on a rolling and fixed origin for 24 steps ahead.*

	Rolling origin		Fixed origin	
	Unreconciled	Reconciled	Unreconciled	Reconciled
Matrix approach	202.06	209.84	87.73	105.69
Separate models	48.40	48.31	16.66	16.85

3.2 Australian domestic tourism simulation study

In this part, we provide results from two simulation studies based on the Australian domestic tourism dataset, to evaluate the sensitivity of our results to several factors. In the first study, we simulate bottom-level series similar to the real bottom-level series of the tourism data, with the same number of series and the same length. We then generate forecasts for four forecast horizons (12, 24, 36 and 48 months) with four different noise levels (standard deviation=0.01, 0.1, 0.5 and 1)[5].

Tables 7 and 8 display the average of the RMSEs for 12 to 48 month-ahead forecasts with different noise levels. Results are shown for the base and the reconciled forecasts for both rolling and fixed origin approaches. The results show that, as expected, by increasing the forecast horizon and/or noise level, the average RMSE increases in all the three methods. Also, the proposed OLS approach shows similar results compared with ETS and ARIMA. It should be noted that for both rolling and fixed origin forecasts in the OLS approach we use the same set of predictors as the Australian domestic tourism example.

Figures 11 and 12 show the computation time (seconds) for ETS, ARIMA and OLS methods on rolling and fixed origin forecasts. From these figures we see that increasing the forecast horizon from one to four years increases computation time almost linearly, while the noise level does not change the computation time. Also, the computation time for ARIMA and ETS is much longer

[5]Since the level of the series are different, we first scale the simulated series (subtracting by mean and dividing by standard deviation), add the white noise series and then we rescale the series.

Table 7: *Mean(RMSE) on one to four year test set with different error levels for ETS, ARIMA and OLS with and without reconciliation - Rolling origin - 304 bottom level series and 8 levels of hierarchy - Simulated tourism dataset*

Reconceliation	Error	Forecast horizon	ETS	ARIMA	OLS
rec	0.01	12	142.00	139.46	140.01
rec	0.01	24	167.27	164.42	162.38
rec	0.01	36	136.35	136.07	136.93
rec	0.01	48	127.89	127.27	130.77
rec	0.10	12	142.78	142.52	141.96
rec	0.10	24	146.74	148.59	146.53
rec	0.10	36	138.83	138.84	139.00
rec	0.10	48	129.74	130.07	132.12
rec	0.50	12	172.29	171.53	171.73
rec	0.50	24	178.38	173.88	175.94
rec	0.50	36	172.73	167.86	169.46
rec	0.50	48	162.54	159.47	159.70
rec	1.00	12	240.59	236.44	235.57
rec	1.00	24	240.48	238.01	236.91
rec	1.00	36	236.17	233.06	232.45
rec	1.00	48	224.14	224.45	219.61
unrec	0.01	12	140.91	139.74	139.45
unrec	0.01	24	144.07	143.98	144.37
unrec	0.01	36	135.85	135.79	136.82
unrec	0.01	48	127.78	127.11	130.55
unrec	0.10	12	141.87	142.47	141.36
unrec	0.10	24	145.78	147.91	146.38
unrec	0.10	36	138.22	138.71	138.90
unrec	0.10	48	129.54	129.74	131.93
unrec	0.50	12	172.47	170.83	171.13
unrec	0.50	24	177.98	173.32	175.79
unrec	0.50	36	172.20	167.19	169.42
unrec	0.50	48	161.89	158.78	159.74
unrec	1.00	12	239.70	235.82	235.09
unrec	1.00	24	240.02	237.39	237.01
unrec	1.00	36	235.57	232.80	232.43
unrec	1.00	48	223.57	223.95	219.84

than OLS. Note that the computation time for the reconciliation step is less than a second and therefore the computation time for base and reconciled forecasts is similar.

In the second simulation study, we fix the forecast horizon at h=24 and the noise at 0.5, and then create 4 different hierarchy levels (8, 10, 12 and 18) – these obtained using the hierarchy structures in Table 2, 9, 10 and 11 (8 = same as Australian domestic tourism data; 9= adding one hierarchy factor, resulting in 10 levels; 10= adding two hierarchy factors, resulting in 12 levels; and 11 = adding two hierarchy factors and one grouping factor, resulting in 18 levels)[6]).

[6]For simplicity we just include 2-way aggregation combinations.

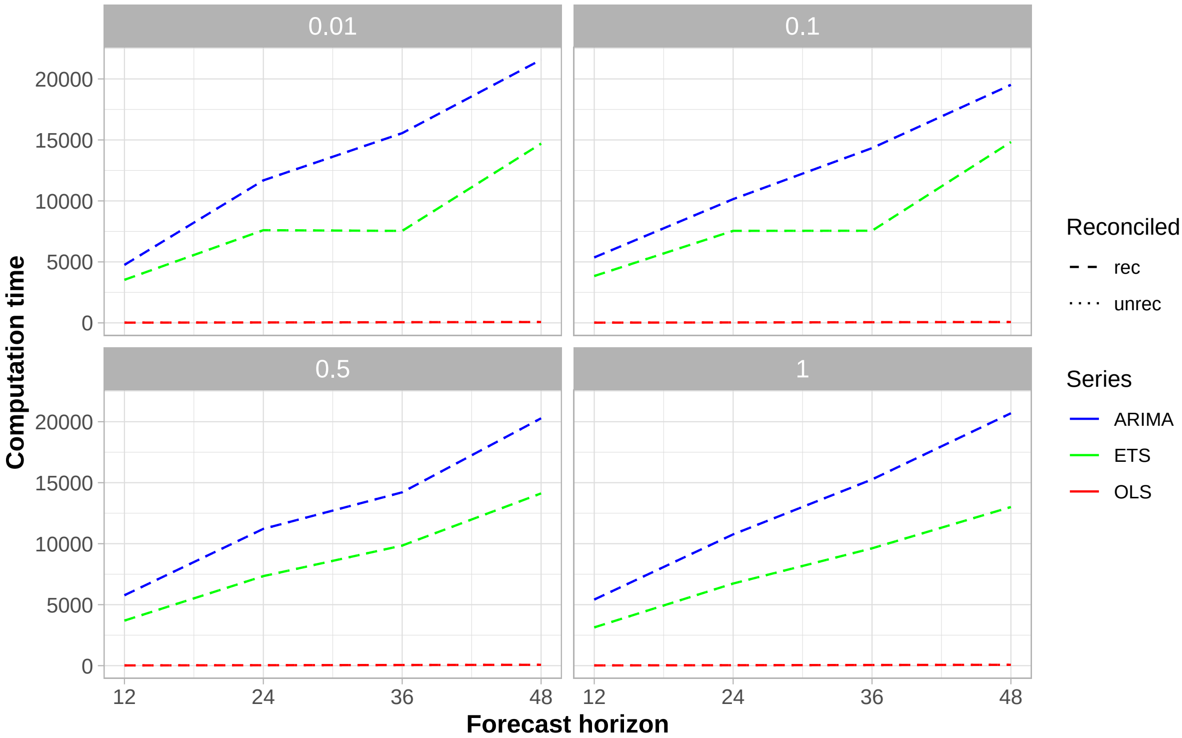

Figure 11: *Computation time (seconds) for ETS, ARIMA and OLS with and without reconciliation - Rolling origin forecasts on one to four year test set and different error values - 304 bottom level series and 8 levels of hierarchy - Simulated tourism dataset.*

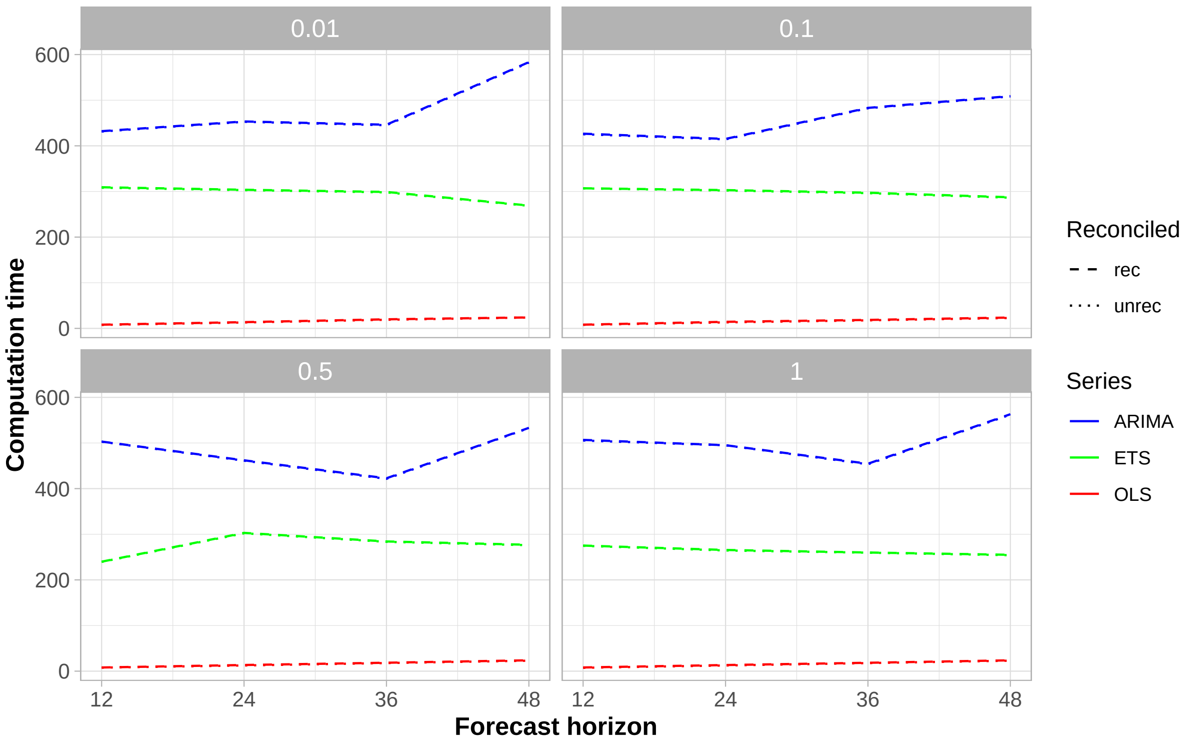

Figure 12: *Computation time (seconds) for ETS, ARIMA and OLS with and without reconciliation - Fixed origin forecasts on one to four year test set and different error values - 304 bottom level series and 8 levels of hierarchy - Simulated tourism dataset.*

Table 8: *Mean(RMSE) on one to four year test set with different error levels for ETS, ARIMA and OLS with and without reconciliation - Fixed origin - 304 bottom level series and 8 levels of hierarchy - Simulated tourism dataset*

Reconceliation	Error	Forecast horizon	ETS	ARIMA	OLS
rec	0.01	12	118.42	127.57	122.16
rec	0.01	24	112.53	124.74	121.05
rec	0.01	36	112.96	125.14	125.76
rec	0.01	48	111.72	122.82	125.48
rec	0.10	12	119.33	128.89	122.20
rec	0.10	24	113.42	126.26	121.47
rec	0.10	36	113.88	127.43	126.80
rec	0.10	48	112.77	125.29	126.81
rec	0.50	12	137.84	158.86	138.58
rec	0.50	24	136.13	162.82	141.93
rec	0.50	36	138.97	168.40	148.92
rec	0.50	48	139.39	168.50	150.52
rec	1.00	12	184.68	205.20	182.98
rec	1.00	24	196.02	215.29	193.08
rec	1.00	36	203.37	220.63	200.19
rec	1.00	48	206.59	222.98	203.78
unrec	0.01	12	119.77	128.42	122.54
unrec	0.01	24	113.59	125.86	121.03
unrec	0.01	36	113.99	126.36	125.63
unrec	0.01	48	112.55	124.18	125.29
unrec	0.10	12	120.62	129.55	122.57
unrec	0.10	24	114.43	127.02	121.45
unrec	0.10	36	114.94	128.28	126.69
unrec	0.10	48	113.64	126.21	126.65
unrec	0.50	12	138.22	158.09	138.62
unrec	0.50	24	136.80	161.06	141.89
unrec	0.50	36	139.94	166.11	148.89
unrec	0.50	48	140.20	166.00	150.49
unrec	1.00	12	185.06	205.52	182.78
unrec	1.00	24	196.41	214.87	193.00
unrec	1.00	36	204.04	219.96	200.16
unrec	1.00	48	207.53	222.26	203.77

We also simulated four sizes of bottom-level series (304, 608, 1520 and 3040). In order to add series, we change the number of 'Purpose' categories (grouping factor) in the Australian domestic tourism example. Table 12 displays the total number of series based on 304, 608, 1520 and 3040 bottom levels series with 8, 10, 12 and 18 hierarchy levels.

Table 9: *Number of simulated Australian domestic tourism series at each aggregation level - adding one hierarchy variable (Level 1)*

Division	Series
Australia	1
Level 1	3
State	7
Zone	27
Region	76
Purpose	4
Level 1 x Purpose	12
State x Purpose	28
Zone x Purpose	108
Region x Purpose	304
Total	570

Table 10: *Number of simulated Australian domestic tourism series at each aggregation level - adding two hierarchy variables (Level 1 and Level 2)*

Division	Series
Australia	1
Level 1	3
Level 2	5
State	7
Zone	27
Region	76
Purpose	4
Level 1 x Purpose	12
Level 2 x Purpose	20
State x Purpose	28
Zone x Purpose	108
Region x Purpose	304
Total	595

Table 11: *Number of simulated Australian domestic tourism series at each aggregation level - adding two hierarchy and one grouping variables (Level 1, Level 2 and Group 1)*

Division	Series
Australia	1
Level 1	3
Level 2	5
State	7
Zone	27
Region	76
Purpose	4
Group 1	5
Level 1 x Purpose	12
Level 2 x Purpose	20
State x Purpose	28
Zone x Purpose	108
Level 1 x Group 1	5
Level 2 x Group 1	6
State x Group 1	7
Zone x Group 1	27
Purpose x Group 1	20
Bottom level	304
Total	665

Table 12: *Total number of the series in the hierarchy structure based on the different number of series with 8, 10, 12 and 18 levels of the hierarchy.*

	Total series			
Bottom level series	8	10	12	18
304	555	570	595	665
608	999	1026	1071	1161
1520	2331	2394	2499	2649
3040	4551	4674	2643	5129

Table 13: *Mean(RMSE) on 8, 10, 12 and 18 levels of hierarchy with 304, 608, 1520 and 3040 number of bottom level series for ETS, ARIMA and OLS with and without reconciliation - two years forecast points with 0.5 error value - Fixed origin - Simulated tourism dataset*

Reconceliation	Series	Levels	ETS	ARIMA	OLS
rec	304	8	136.13	163.02	141.93
rec	304	10	150.25	179.03	157.95
rec	304	12	162.55	191.52	169.99
rec	304	18	188.41	221.29	196.63
rec	608	8	995.90	996.65	1025.54
rec	608	10	1114.37	1115.57	1149.27
rec	608	12	1187.76	1189.15	1225.63
rec	608	18	1364.58	1365.21	1410.01
rec	1520	8	1220.75	1221.09	1204.62
rec	1520	10	1369.87	1370.47	1350.59
rec	1520	12	1464.08	1464.66	1442.03
rec	1520	18	1696.08	1698.90	1670.58
rec	3040	8	1095.35	1129.84	1057.09
rec	3040	10	1229.98	1286.68	1185.05
rec	3040	12	1314.39	1368.99	1265.10
rec	3040	18	1525.32	1558.62	1463.65
unrec	304	8	136.80	161.36	141.89
unrec	304	10	151.40	177.64	157.75
unrec	304	12	163.99	190.19	169.84
unrec	304	18	190.66	220.24	196.42
unrec	608	8	993.18	991.88	1024.60
unrec	608	10	1110.89	1109.42	1148.23
unrec	608	12	1183.70	1182.20	1224.52
unrec	608	18	1359.06	1356.75	1409.00
unrec	1520	8	1218.72	1219.57	1204.58
unrec	1520	10	1368.48	1369.15	1350.56
unrec	1520	12	1463.22	1463.55	1442.00
unrec	1520	18	1694.60	1697.77	1670.59
unrec	3040	8	1096.51	1127.31	1057.13
unrec	3040	10	1231.37	1276.52	1185.10
unrec	3040	12	1316.10	1355.21	1265.17
unrec	3040	18	1528.15	1551.03	1463.73

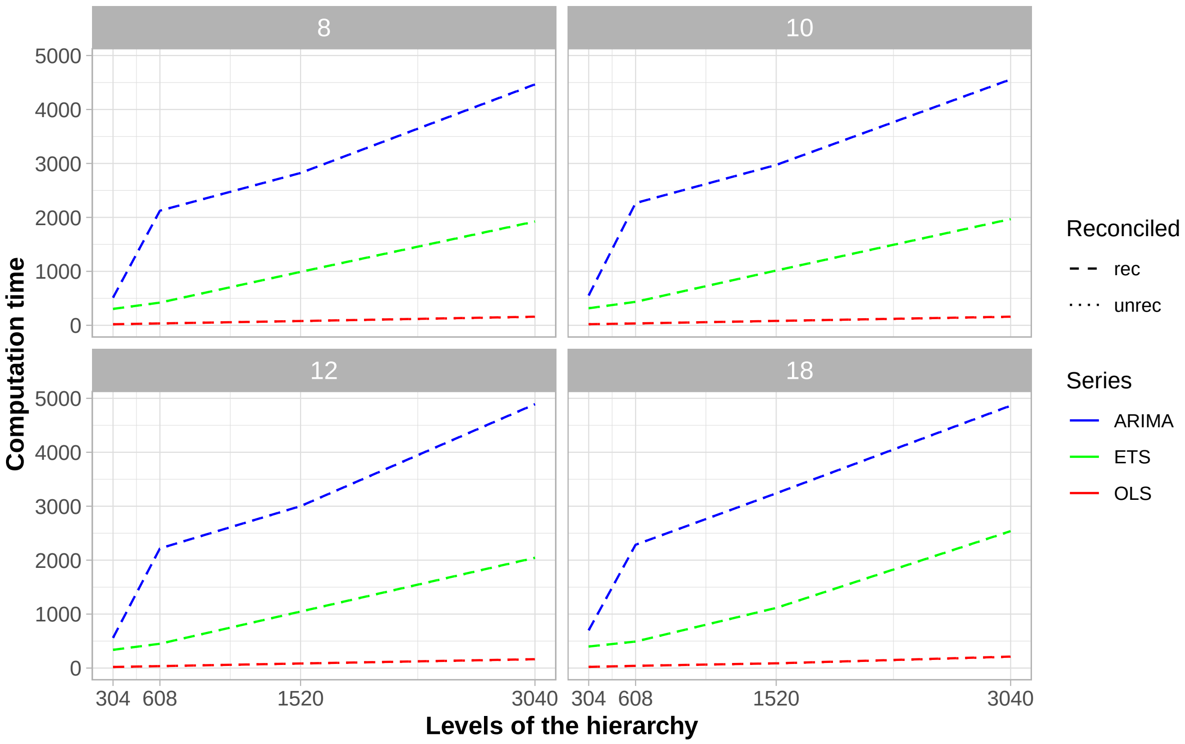

Figure 13: *Computation time (seconds) for ETS, ARIMA and OLS with and without reconciliation - Fixed origin forecasts with 8, 10, 12 and 18 levels of hierarchy with 304, 608, 1520 and 3040 number of bottom level series - two years forecast points with 0.5 error value - Simulated tourism dataset.*

3.3 Wikipedia pageviews: Grouped structure

The second dataset comprises one year of daily data (2016-06-01 to 2017-06-29) on Wikipedia pageviews for the most popular social networks articles (Ashouri, Shmueli & Sin 2018). This dataset is noisier than the Australian monthly tourism data, making forecasting more challenging. The data has a grouped structure with the following attributes: "Agent": Spider, User, "Access": Desktop, Mobile app, Mobile web, "Language": en (English), de (German), es (Spanish), zh (Chinese) and "Purpose": Blogging related, Business, Gaming, General purpose, Life style, Photo sharing, Reunion, Travel, Video (see Table 14). In Figure 20 (Appendix B), we show one possible hierarchy for this dataset, but the order of the hierarchy can be switched.

Table 14: *Social networking Wikipedia article grouping structure*

Grouping	Series	Grouping	Series
Total		Language	
	1. Social Network		10. zh (Chinese)
Access		Purpose	
	2. Desktop		11. Blogging related
	3. Mobile app		12. Business
Agent			13. Gaming
	4. Mobile web		14. General purpose
	5. Spider		15. Life style
	6. User		16. Photo sharing
Language			17. Reunion
	7. en (English)		18. Travel
	8. de (German)		19. Video
	9. es (Spanish)		

We consider the main aggregation factors and two-way combinations of them [7]. The final dataset includes 913 time series, each with length 394. Table 15 shows the group structure's different levels and the number of series in each level.

For this daily dataset, in the OLS forecasting model we include in the predictor matrix a quadratic trend, 6 seasonal dummies and lags 1 and 7 for rolling and fixed origin models. We partitioned the data into two parts training and test sets. We used the last 28 days for our test set and the rest for the training set. In this example, the results in tables and figures are represented for single groups although we applied all the above levels in the group structure for reconciliation.

[7] There are four more 3-way aggregation combinations that we do not include: Agent × Access × Language, Agent × Access × Purpose, Agent × Language × Purpose, and Access × Language × Purpose. Including these four additional aggregations might slightly improve the results but for simplicity, we excluded them.

Table 15: *Number of Wikipedia pageviews series at each aggregation level.*

Division	Series
Total pageviews	1
Access	3
Agent	2
Language	4
Purpose	9
Access x Agent	5
Access x Language	12
Access x Purpose	27
Agent x Language	8
Agent x Purpose	18
Language x Purpose	33
Bottom level	913
Total	1035

Tables 16 and 17 show the RMSE results. Although these time series are noisier, we still get acceptable results for the OLS forecasting model compared with ETS and ARIMA. In this case, we get similar results with and without the reconciliation step.

Table 16: *Mean(RMSE) for ETS, ARIMA and OLS with and without reconciliation - Rolling origin - Wikipedia dataset*

	Unreconciled			Reconciled		
Level	ETS	ARIMA	OLS	ETS	ARIMA	OLS
Total	10773.7	15060.7	12968.1	10851.0	14575.5	12779.1
Access	6524.7	6705.0	6021.1	6314.2	7316.6	6310.7
Agent	8272.9	10196.3	9372.2	7728.9	9896.3	9222.4
Language	4870.1	6333.0	5688.2	5134.3	6372.0	5778.9
Purpose	5233.5	4659.5	4106.9	4977.8	4525.3	4040.4
Bottom level	358.1	239.0	261.1	372.3	245.4	264.6

Table 17: *Mean(RMSE) for ETS, ARIMA and OLS with and without reconciliation - Fixed origin - Wikipedia dataset*

	Unreconciled			Reconciled		
Level	ETS	ARIMA	OLS	ETS	ARIMA	OLS
Total	14846.9	24298.8	20203.7	15251.3	24383.6	20088.2
Access	7117.4	10732.0	8866.4	7758.2	11013.9	8970.1
Agent	13608.7	17277.0	14985.7	12431.2	16564.9	14884.8
Language	6475.9	9580.4	7913.7	6728.7	9797.3	8094.7
Purpose	5302.7	8611.3	5694.1	5256.3	8121.8	5665.0
Bottom level	435.6	389.4	363.7	445.9	394.0	363.5

Figures 14 and 15 display the forecast error box plot. These plots are for rolling and fixed origin forecasts over 28 days in each level of grouping. Further, we can see that the error distribution

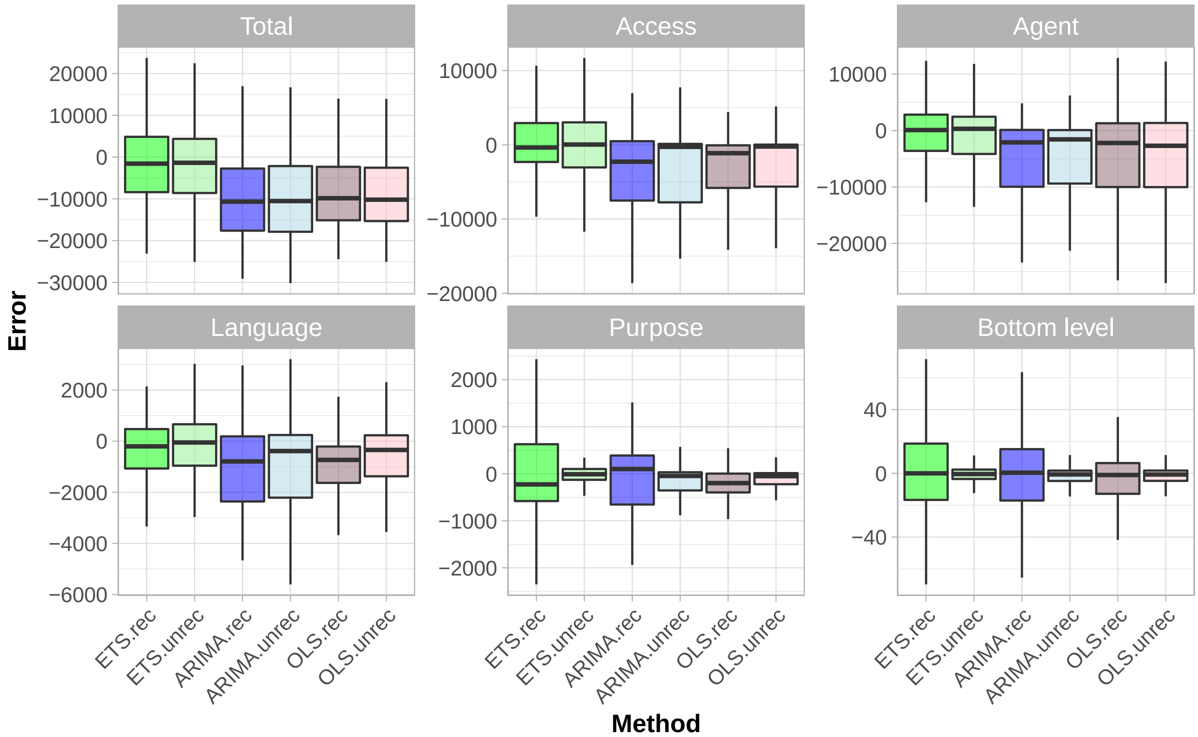

Figure 14: *Box plots of forecast errors for reconciled and unreconciled ETS, ARIMA and OLS methods at each hierarchical level for rolling origin forecasts of Wikipedia pageviews.*

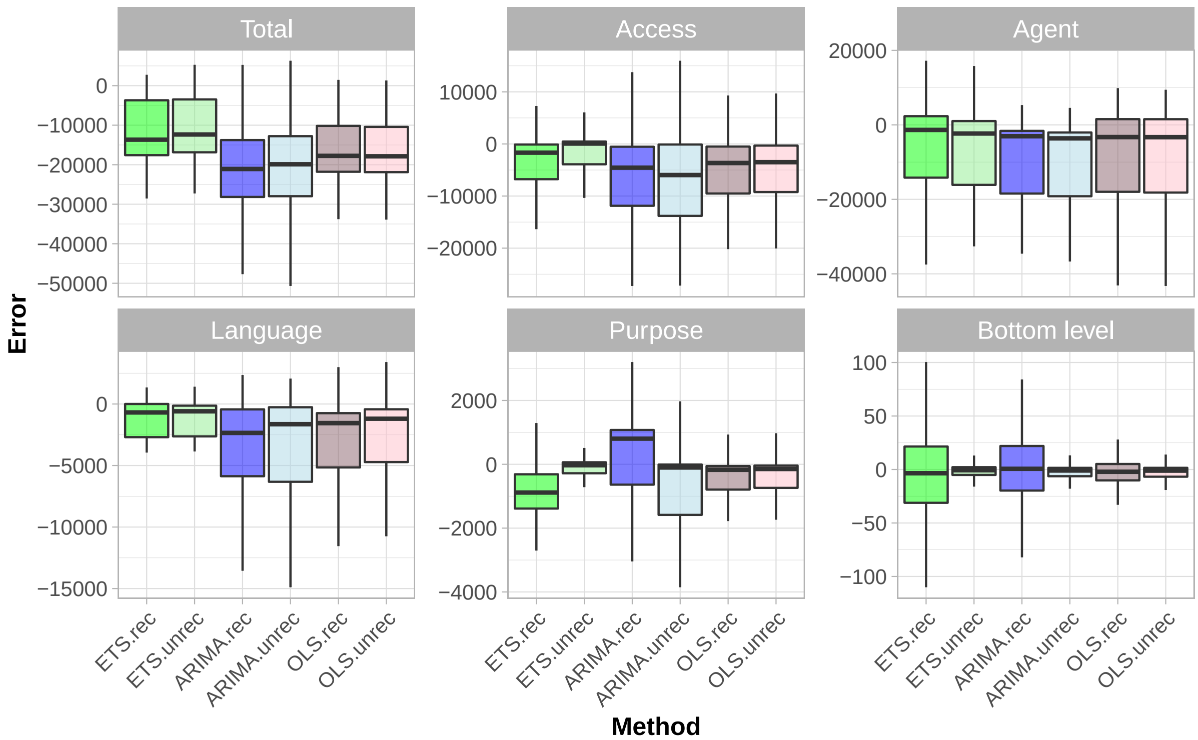

Figure 15: *Box plots of forecast errors for reconciled and unreconciled ETS, ARIMA and OLS methods at each hierarchical level for fixed origin forecasts of Wikipedia pageviews.*

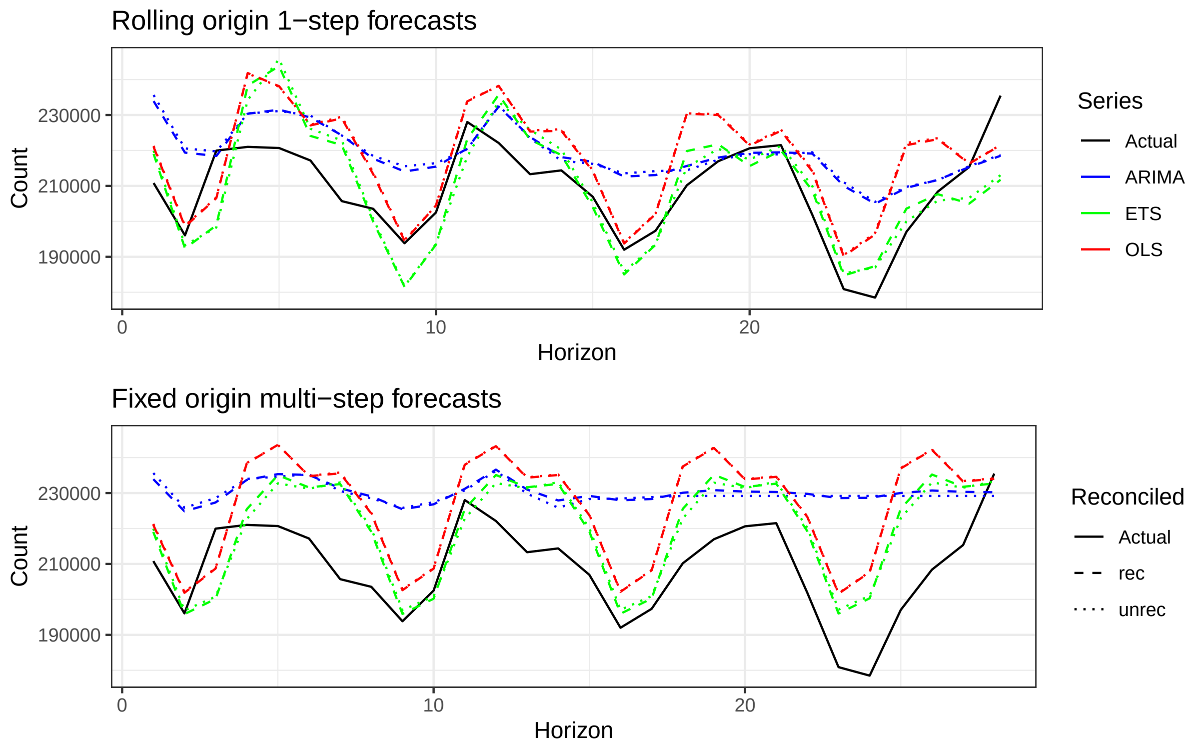

Figure 16: *The actual test set for the 'Total' series compared to the forecasts from reconciled and unreconciled ETS, ARIMA and OLS methods for rolling and fixed origin forecasts of Wikipedia pageviews.*

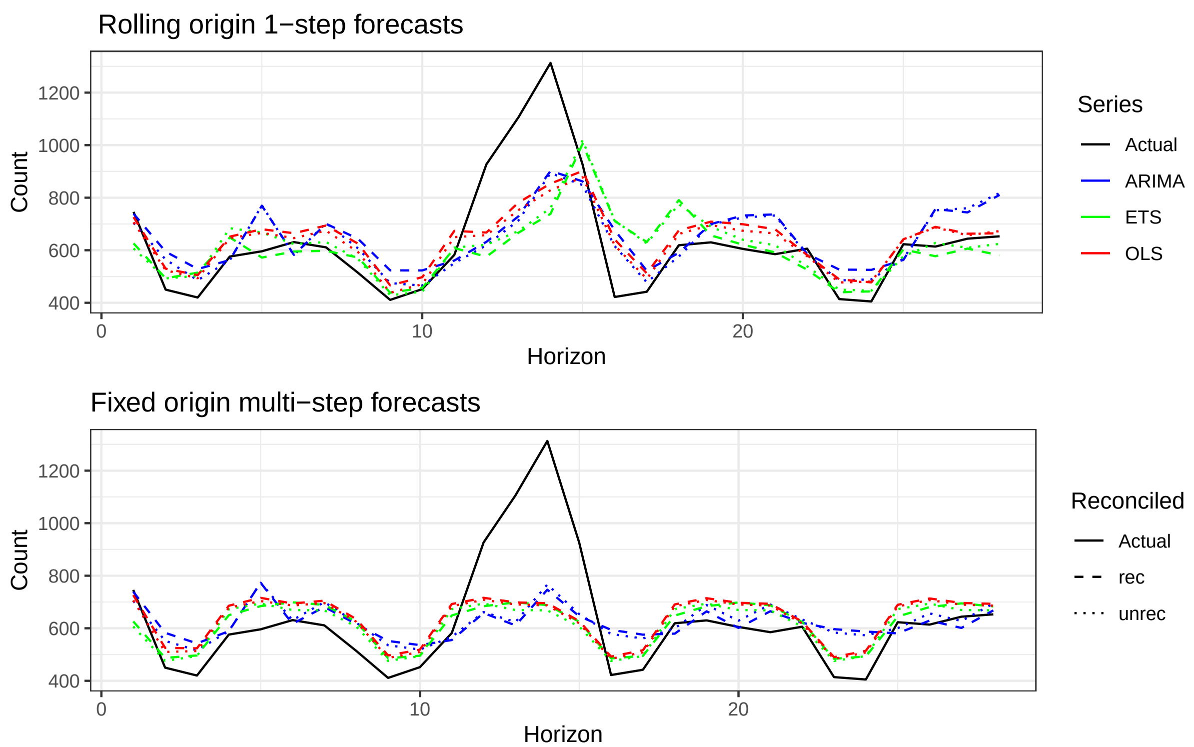

Figure 17: *The actual test set for the 'desktopusenPho04' bottom level series compared to the forecasts from reconciled and unreconciled ETS, ARIMA and OLS methods for rolling and fixed origin forecasts of Wikipedia pageviews.*

is almost similar in all levels across the different methods. The only exception is the Total series, where ETS performs significantly better than ARIMA and OLS. We also note that the reconciliation is less effective. As in the tourism example, in higher levels, series have higher counts and therefore their error magnitudes are larger.

In Figures 16 and 17, we display results for the total and one of the bottom level series, "desktopusenPho04" (desktop-user-english-photo sharing). The plot shows rolling and fixed origin forecast results over the 28 day test set for ETS, ARIMA and OLS, with (dashed lines) and without (dotted lines) applying reconciliation. We see that the OLS forecasting model performs close to the other two methods, and reconciliation improves the forecasts.

Table 18 presents the computation times for all three methods. ETS and ARIMA are clearly much more computationally heavy compared with OLS. As in the Australian tourism dataset, running reconciliation does not have much effect on computation time.

Table 18: *Computation time (seconds) for ETS, ARIMA and OLS with and without reconciliation - Rolling and fixed origin forecasts - Wikipedia dataset*

	Computation time (secs)			
	Rolling origin		Fixed origin	
	Unreconciled	Reconciled	Unreconciled	Reconciled
ETS	27613.08	27613.14	971.55	971.58
ARIMA	49419.36	49419.39	1769.52	1769.56
OLS	116.27	116.31	61.33	61.38

4 Conclusion

We have proposed a linear model approach to fast forecasting of hierarchical or grouped time series, with accuracy that nearly matches that of forecast methods such as ETS and ARIMA. This is especially useful in large collections of time series, as is typical in hierarchical and grouped structures. Although ETS and ARIMA are advantageous in terms of forecasting power and accuracy, they can be computationally heavy when facing large collections of time series in the hierarchy. An important feature of our model is its ability to easily include external information such as holiday dummies or other external series. We also note that OLS has the additional practical advantage of handling missing data while ETS requires imputation.

Another advantage of our approach is that it can be computed in a single matrix equation (5). This makes it extremely fast and easy to implement, and enables standard results to be derived with minimal effort (e.g., prediction intervals).

Pennings & Dalen (2017) proposed another approach for forecasting hierarchical time series using state space models. Although their approach is flexible in handling outliers, missing data and external features, it is less flexible to different kinds of datasets and it is computationally much more complex.

Acknowledgements

The first and third authors of this research were partially funded by Ministry of Science and Technology (MOST), Taiwan [Grant 106-2420-H-007-019].

Appendix A

We provide boxplots of the scaled forecasted errors for the tourism example. These plots are displayed for both rolling forward and multiple-step-ahead forecasts.

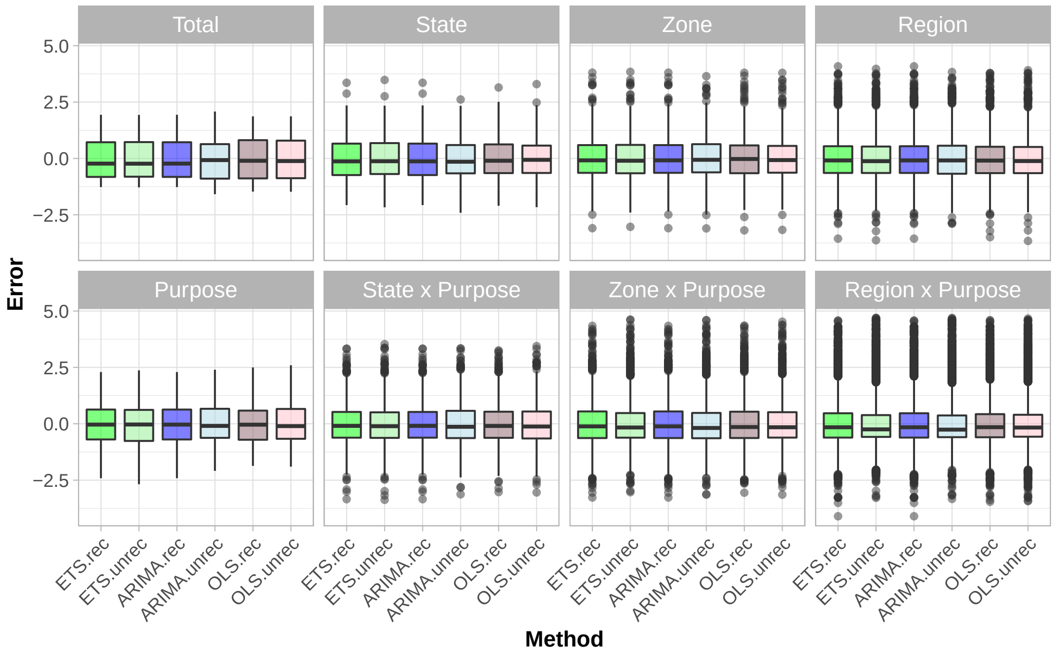

Figure 18: *Box plots of scaled forecast errors from reconciled and unreconciled ETS, ARIMA and OLS methods at each hierarchical level for rolling origin 1-step-ahead tourism demand.*

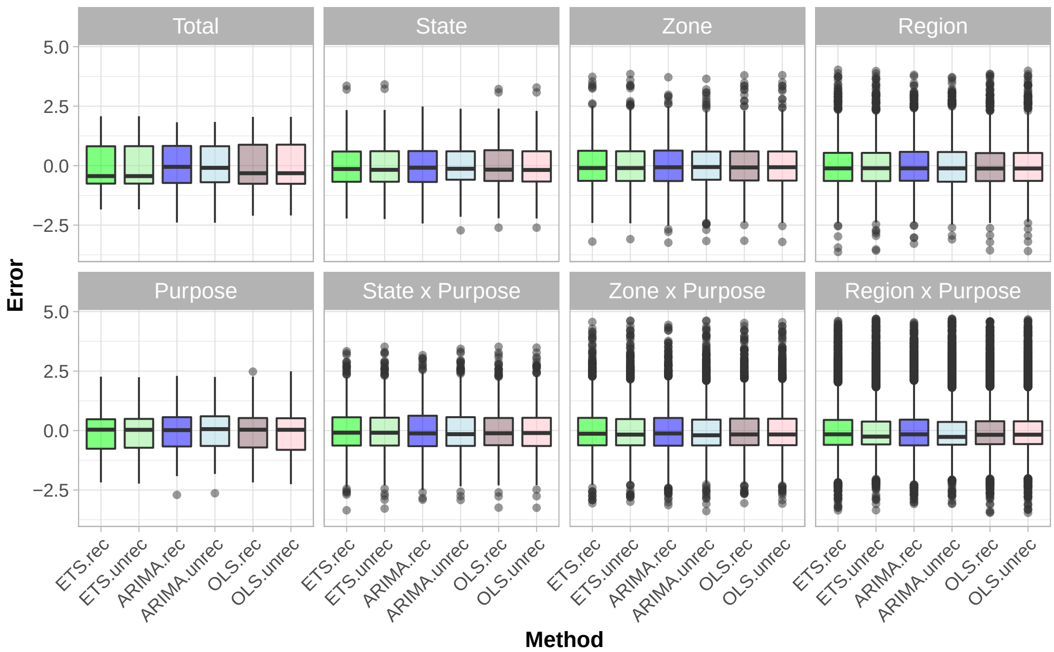

Figure 19: *Box plots of scaled forecast errors from reconciled and unreconciled ETS, ARIMA and OLS methods at each hierarchical level for fixed origin multi-step-ahead tourism demand.*

Appendix B

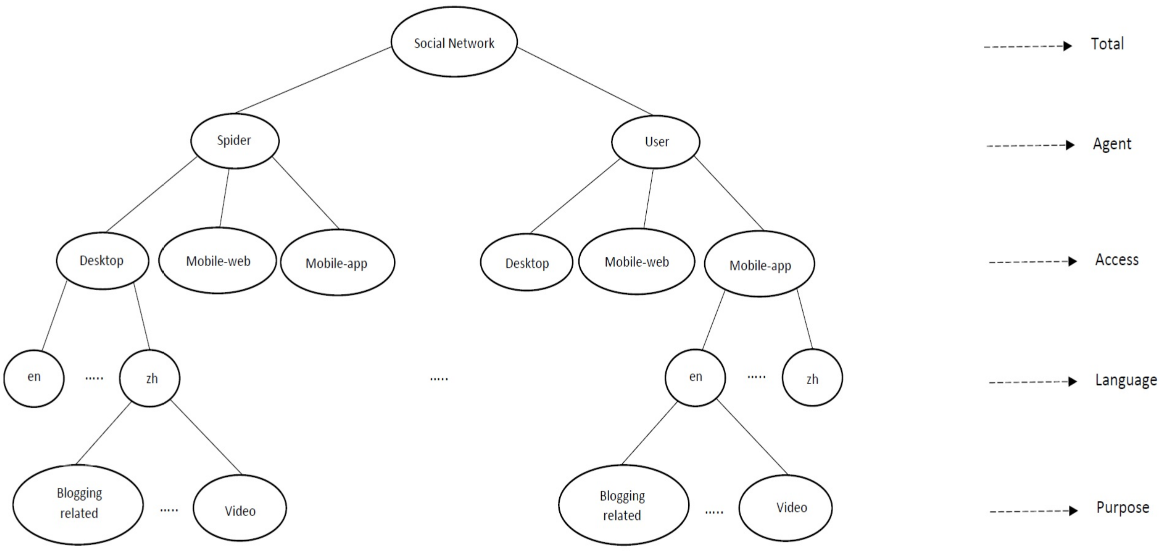

Figure 20: *One of the possible hierarchical structures for the Wikipedia pageview dataset.*

References

Akaike, H (1998). "Information theory and an extension of the maximum likelihood principle". In: *Selected Papers of Hirotugu Akaike*. Springer Series in Statistics (Perspectives in Statistics). Springer, pp.199–213.

Ashouri, M, G Shmueli & CY Sin (2018). Clustering time series by domain-relevant features using model-based trees. *Proceedings of the 2018 Data Science, Statistics & Visualization (DSSV)*.

Athanasopoulos, G, RA Ahmed & RJ Hyndman (2009). Hierarchical forecasts for Australian domestic tourism. *International Journal of Forecasting* **25**(1), 146–166.

Fliedner, G (2001). Hierarchical forecasting: issues and use guidelines. *Industrial Management & Data Systems* **101**(1), 5–12.

Gross, CW & JE Sohl (1990). Disaggregation methods to expedite product line forecasting. *Journal of Forecasting* **9**(3), 233–254.

Hyndman, RJ, RA Ahmed, G Athanasopoulos & HL Shang (2011). Optimal combination forecasts for hierarchical time series. *Computational Statistics & Data Analysis* **55**(9), 2579–2589.

Hyndman, RJ & G Athanasopoulos (2018). *Forecasting: principles and practice*. Melbourne, Australia: OTexts. https://OTexts.org/fpp2.

Hyndman, RJ, G Athanasopoulos, C Bergmeir, G Caceres, L Chhay, M O'Hara-Wild, F Petropoulos, S Razbash & E Wang (2020). Package 'forecast'. *Online] https://cran. r-project. org/web/packages/forecast/forecast. pdf*.

Januschowski, T, S Kolassa, M Lorenz, C Schwarz, et al. (2013). Forecasting with in-memory technology. *Foresight: The International Journal of Applied Forecasting* (31), 14–20.

Kahn, KB (1998). Revisiting top-down versus bottom-up forecasting. *The Journal of Business Forecasting* **17**(2), 14.

Pennings, CL & J van Dalen (2017). Integrated hierarchical forecasting. *European Journal of Operational Research* **263**(2), 412–418.

Tourism Research Australia (2005). Travel by Australians, September Quarter 2005. *Tourism Australia*.

Wickramasuriya, SL, G Athanasopoulos & RJ Hyndman (2019). Optimal forecast reconciliation for hierarchical and grouped time series through trace minimization. *Journal of the American Statistical Association* **14**(526), 804–819.