A Fast and elegant method for forecast reconciliation using linear forecasting 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

5 December 2018

JEL classification: C10,C14,C22

A Fast and elegant method for forecast reconciliation using linear forecasting models

Abstract

Forecasting hierarchical or grouped time series includes 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). The reconceliation step is a linear process that adjusts the base forecasts. However using ETS or ARIMA for base forecasts can be computatioanaly challenging due to the large number of series forecastes. In this research we propose a set of linear models that alternates this computational problem. Our method provides a single-step reconceliation onapproach instead of the two-step approach. The proposed method is also more flexible in incorporating external data handeling missing values. We illustrate our approach using two real datasets, on Australian domestic tourism and Wikipedia pageviews. We compare our approach with reconceliation using ETS and ARIMA, and show that our approach is much faster and at the same time competes well in terms of forecasting accuracy.

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

1 Introduction

1.1 Hierarchical and grouped time series

The increasing internet usage and digitization has dramatically is increased the amount of collected time series data. For example, the Internet of Things (IoT) produces huge amount of series in short period of time. Forecasting large collections of time series is computationally heavy and challenging. In some cases these time series can be structured and disaggregated based on hierarchies or groups such as geographical location and gender. An example of hierarchical time series is sales in restaurant chains, which can be disaggregated into different branches and then different foods or drinks. Figure 1 shows a schematic os such a hierarchical time series structure. In this example the hierarchy includes three levels: Top level, level 0, is the total series which is the aggregation of all the bottom level series. In the middle level, level 1, series are aggregation of their own bottom level series. For instance, series A is the aggregation of AA and AB. Finally, the bottom level series, level 2, include the most disaggregated series.

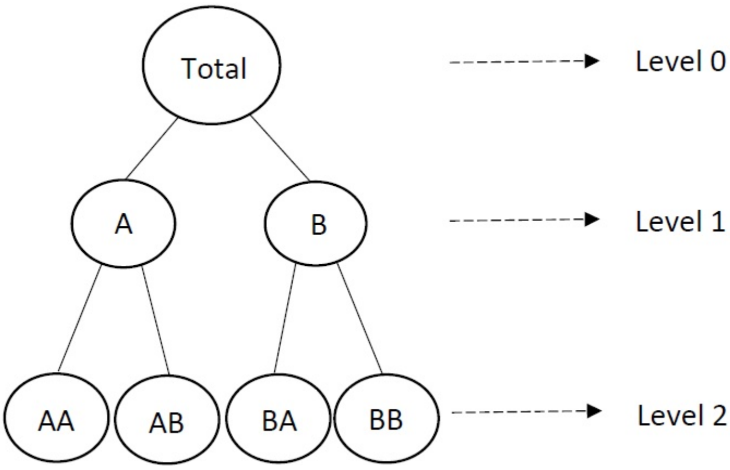

Figure 1: *An example of a two level hierarchy structure*

Grouped time series are more complicated aggregation structures compared with hierarchical time series. They can be defined as hierarchical time series without unique hierarchy structure (Hyndman et al. 2015). A schematic of a grouped time series structure is shown in Figure 2. Series in this structure can be split first into group A and B and then into C and D (left side) or first in C and D and then in A and B (right side) (Hyndman, Lee & Wang 2016). In the following, we use the same notation for both hierarchy and grouped time series.

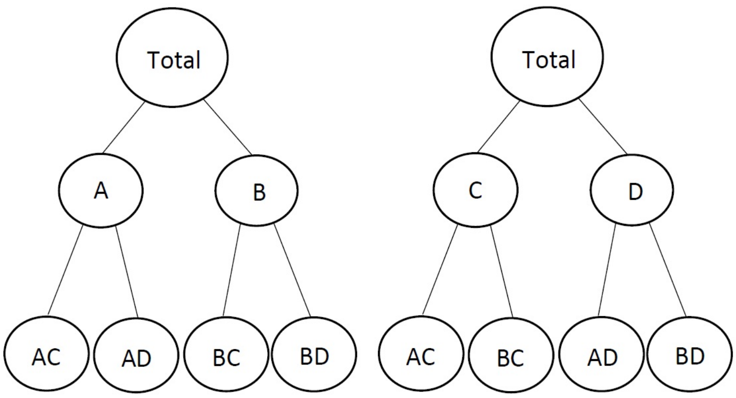

Figure 2: *An example of two level grouped structure*

Using the notations in Hyndman, Lee & Wang (2016), we denote the total series at time t by y_t, and the series in node Z at time t is denoted by $y_{Z,t}$. For generating different levels of series from the bottom level series, we use an $n \times n_k$ matrix, called the 'summing matrix', denoted by S, in which n is the overall number of nodes and n_k is the number of bottom level nodes.

For example in Figure 1, $n = 7$ and $n_k = 4$. This summing matrix can be partitioned based on different levels of the hierarchy. Using the 'summing matrix', generating a hierarchy structure at time t is given by $\mathbf{y_t} = \mathbf{Sb_t}$, where $\mathbf{y_t}$ is a vector of all the level nodes at time t and $\mathbf{b_t}$ is the vector of all the bottom level nodes at time t. For the example shown in Figure 1, the hierarchy equation involving the $S_{7\times4}$ matrix can be written as follows:

$$\begin{pmatrix} y_t \\ y_{A,t} \\ y_{B,t} \\ y_{AA,t} \\ y_{AB,t} \\ y_{BA,t} \\ y_{BB,t} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} y_{AA,t} \\ y_{AB,t} \\ y_{BA,t} \\ y_{BB,t} \end{pmatrix} \tag{1}$$

1.2 Forecasting hierarchical time series

In hierarchical time series, because the most disaggregated or bottom level series are highly noisy their forecasting results are less accurate. At the same time, the most aggregated (total) series is smoother and less noisy and therefore forecasting it is easier (Fliedner 2001). If we just forecast each series individually, we are ignoring the hierarchy or grouping structure (Hyndman, Lee & Wang 2016). The result is that, summing forecast values from lower levels does not necessarily coincide with the higher level (more aggregated) forecasts. Also, forecasting the most disaggregated level series and computing other level series by summing forecasts can result in poor forecasts at higher levels.

In the literature there are several methods that consider the hierarchy structure information in forecasting time series. These include the top-down (Gross & Sohl 1990), 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 lower level series 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. 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 hierarchy structure, and then uses regression models to reconcile the resulting forecasts. The advantage of the optimal combination method compared with the other methods is that it considers the interactions and correlation among the series in all the levels of hierarchy. This method also provides forecast uncertainty and is flexible for ad hoc adjustment.

In the optimal combination method, base forecasts can be computed using the following linear model:

$$\hat{\mathbf{y}}_{\mathbf{h}} = \mathbf{S}\boldsymbol{\beta}_{h} + \epsilon_h \tag{2}$$

where $\hat{\mathbf{y}}_{\mathbf{h}}$ represents a vector of h-step-ahead base forecasts for all levels of the hierarchy, $\boldsymbol{\beta}_h$ is the unknown conditional mean of the bottom level series, and ϵ_h is the aggregation error which has mean equal to zero and variance equal to $\sum_h$ (Hyndman, Lee & Wang 2016). Using Equation (2) and estimating $\boldsymbol{\beta}_h$, forecasts in all levels of the hierarchy can be computed. Since estimating $\boldsymbol{\beta}_h$ using Generalized Least Squares (GLS) requires knowledge of $\sum_h$, Ordinary Least Squares (OLS) can be used and then a vector of reconciled forecast can be calculated using Equation (3):

$$\tilde{\mathbf{y}}_{\mathbf{h}} = \mathbf{S}(\mathbf{S}'\mathbf{S})^{-1}\mathbf{S}'\hat{\mathbf{y}}_{\mathbf{h}} \tag{3}$$

1.3 Challenges and motivation

Hyndman et al. (2015) implemented a package called **hts** in R (R core Team (2013)) for forecasting hierarchical time series including all the approaches mentioned in Section Forecasting hierarchical time series. In this package, two functions, *hts* and *gts*, produce forecasts for hierarchical and grouped time series, respectively. Inputs for these functions include base forecasts and hierarchy and group structure. Here base forecasts are forecasts in all levels of hierarchy using only the history of each series and ignoring hierarchy or group structure. In the optimal combination method, there are two steps to determine the reconciled forecasts: first, computing base forecasts and second reconciling those forecasts. Currently options for computing the base forecasts are Random Walk (RW), Exponential Smoothing (ETS) and Autoregressive Integrated Moving Average (ARIMA).

When we encounter a large collection of time series computing base forecasts using ETS or ARIMA can be computationally expensive. ETS (function "ets()" in forecast package (Hyndman, Khandakar, et al. 2007)) optimizing the likelihood function and since we can not specify trend and seasonal components, it needs to search over all the possibilities of the trend and seasonal components (non, additive and multiplicative). Also ARIMA (function "auto.arima()" in forecast package (Hyndman, Khandakar, et al. 2007)) combines unit root tests, minimization of the Akaike information criterion (AICc) (Akaike 1998) and Maximum Likelihood Estimation (MLE) to obtain an estimated ARIMA model (Hyndman & Athanasopoulos 2018). Again, in ARIMA if we can not specify the order of the models for all the series, the procedure searches over all the order combination possibilities which make it computationally expensive. This computational challenges increases in the number of lower level series as well as in the number of hierarchies. 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

Our proposed approach is based on using linear regression models for computing base forecasts. We begin with partitioning the dataset into training and test sets. We denote by $\mathbf{y_t} = \{y_1, y_2, ..., y_t\}$ and $\mathbf{y_h} = \{y_{t+1}, y_{t+2}, ..., y_h\}$ the vector of time series in the training and test set for h-step-ahead forecasts in all the levels of hierarchy. The h-step-ahead base forecasts and reconciled vectors are denoted by $\widehat{\mathbf{y}}_\mathbf{h}$ and $\tilde{\mathbf{y}}_\mathbf{h}$, respectively. We also use $\mathbf{X_t}$ and $\mathbf{X_h}$ to denote the matrices of predictors in the training and test set, respectively.

The linear forecasting Ordinery Least Square (OLS) model is given by:

$$\mathbf{y_t} = \mathbf{X_t}\boldsymbol{\alpha}_h + \delta_h, \tag{4}$$

where $\boldsymbol{\alpha}_h$ is the vector of coefficients and δ_h is the error term with mean zero and constant variance. We can estimate the reconciled coefficients in two ways: in two steps, **two-step**, or in single step, **single-step**. The two-step approach we should first estimate $\boldsymbol{\alpha}_h$ using OLS estimation :

$$\hat{\boldsymbol{\alpha}}_h = (\mathbf{X_t'X_t})^{-1}\mathbf{X_t'y_t}, \tag{5}$$

and then using Equations (5) and (3), we can find the reconciled forecasts:

$$\tilde{y}_h = \mathbf{S}(\mathbf{S'S})^{-1}\mathbf{S'X_h}\hat{\boldsymbol{\alpha}}_h. \tag{6}$$

We can use Equations (5) and (6) to compute the base forecasts using an OLS forecasting model and then to apply the reconciled forecasts. However because both OLS and reconciliation steps are linear, we can combine these two steps and compute the reconciled forecasts in one step:

$$\tilde{y}_h = \mathbf{S}(\mathbf{S}'\mathbf{S})^{-1}\mathbf{S}'\mathbf{X_h}(\mathbf{X_t'}\mathbf{X_t})^{-1}\mathbf{X_t'}\mathbf{y_t} \tag{7}$$

This single-step reconciliation approach is more parsimoneous and elegant.

2.1 OLS predictors

As an example for the $\mathbf{X_t}$ matrix in Equation (4), we can refer to the set of predictors proposed in Ashouri, Shmueli & Sin (2018) for modeling trend, seasonality and autocrrolation by using laged values $(y_{t-1}, y_{t-2}, \ldots)$ trend variables and seasonal dummy variables as a set of predictors in the linear model. Equation (8) shows a linear equation of this type which models linear trend, additive seasonality with m seasons, autocorrelation and external data. t is the running index $(t = 1, 2, ...)$, and $Season_{jt}$ is a dummy variable taking value 1 if time t $(j = 1, 2, ..., m)$ is in season j, y_{t-k} is the kth lagged value for y_t and z_t is the external data. For instance, if we have daily data with day of week seasonality, m would be 7 (6 seasonal dummies and 7 the violaton of lags).

$$y_t = \alpha_0 + \alpha_1 t + \beta_1 Season_{1t} + \beta_2 Season_{2t} + \cdots + \beta_{m-1} Season_{(m-1)t} + \gamma_1 y_{t-1} + \gamma_2 y_{t-2} + \cdots + \gamma_m y_{t-m} + \delta z_t \epsilon_t \tag{8}$$

While OLS is popular in practice for forecasting time series, it is often frown upon due to its independence assumption. This can cause issue for parametric inference but is less of a problem for forecasting, in fact it offen performs sufficiently well for forecasting as can be seen by its popular use in practice.

3 Applications

In this section we illustrate our approach two examples, forecasting monthly Australian domestic tourism and forecasting daily Wikipedia pageviews. We compare the forecasting accuracy levels of ETS, ARIMA and the proposed linear OLS forecasting model, with and without the reconciliation step. For comparing these methods we use the average of Root Mean Square Error (RMSE) across all series and also display box and density plots for forecast errors along with the raw forecast errors.

Since we are using time series lags $(1, ..., m)$ in the linear forecasting model, we cant forecast multiple steps ahead. We therefore apply two methods for generating h-step-ahead forecast: In the first model we use 1-step-ahead forecasts and for forecasting the following periods $(t + 2, t + 3, ...)$ we replace the previous periods with the actual values. This value is known to us because it is in the test set. In our applications, we call this approach '1-step-ahead'. In the second method, we again use 1-step-ahead forecasts but for forecasting the following periods we use the earlier forecasted values. In our applications, we call this approach 'h-step-ahead forecast'. We also show the computation challenges in all the methods.

3.1 Australian domestic tourism

This dataset has 19 years of monthly visitor nights in Australia by Australian tourists. This measure is used as an indicator of tourism activity (Wickramasuriya, Athanasopoulos & Hyndman 2018). This data were collected by computer-assisted telephone interviews with 120000, Australians aged 15 and up (Research tourism Australia 2005). In total this dataset includes 304 time series with length 228 each. The hierarchy and grouping structure for this dataset is made using geographical and purpose of travel information.

In this dataset we have three levels of geographical divisions in Australia. In the first level, Australia was 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 geographical divisions see Figure 3 and Table 1). For 'Purpose' we have four groups: Holiday (Hol), Visiting (Vis), Business (Bis) and Others (Oth). Based on geographical hierarchy and purpose grouping, we end up with 8 levels of hierarchy with 555 series in total. The hierarchy structure which is used in this example includes the following levels:

- Level 0 = Total series
- Level 1 = State
- Level 2 = Zone
- Level 3 = Region
- Level 4 = Purpose
- Level 5 = State $\times$ Purpose
- Level 6 = Zone $\times$ Purpose
- Level 7 = bottom level series

We report the forecast results for all these hierarchy levels, as well as the average RMSE across all the levels of hierarchy.

In the predictor matrix, for the OLS forecasting model we apply linear trend, 11 dummy variables, and 12 time series lags[1]. This is intended to capture the monthly seasonality. In addition, before running the model, we partition the data into two parts, training and test sets. We keep the last 24 months periods (2 years) as our test set to forecast and we use the rest as our training set.

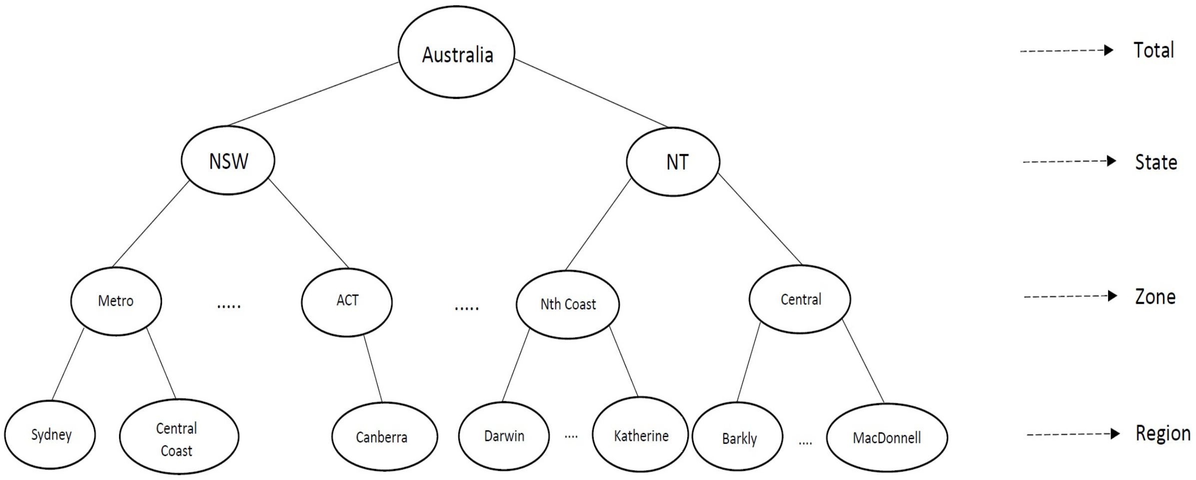

Figure 3: *Australia geographical hierarchy structure*

In Tables 3, 4 and 5, we have the average RMSE and computation time for the 24-month forecast period. Methods include ETS, ARIMA and our proposed OLS forecasting model. In Table 3 we forecast 24 periods by computing 1-step-ahead forecasts and rolling forward month by month. In Table 4 we genarated 24-step-ahead forecasts. In these tables we have two parts related to the forecast, with and without reconciliation.

The results in Table 3 and 4 show that our proposed OLS forecasting model produces forecast accuracy similar to ETS and ARIMA, which are computationally heavy for many time series. Also they show the usefulness of the reconciliation in decreasing the average RMSE in all the three methods. Except for the total series, reconciliation can help in forecasting all the hierarchy levels.

In Figures 4, 6, 5 and 7 we display the error box plots for both reconciled and unreconciled forecasts, and error density plots for reconciled forecasts using all three methods, for 1-step-ahead and 24-step-ahead. In all these figures we see the error distribution similarity across all the models, as well as usefulness of the reconciliation step in improving the forecasts. By comparing density plots 5 and 7, as could be expected, we see that by applying 1-step-ahead forecasts, the error densities are closer and more distributed around zero.

[1]Since the forecasting results are better without the lags, we just use a linear trend and dummy seasonality variables in our linear model for 24-step-aehad model.

Table 1: *Australia geographical hierarchy structure*

Series	Name	Lable	Series	Name	Lable
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	Merbourne 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	Barrosa	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

Table 2: *Number of Australian domestic tourism series in each level of hierarchy and group structure*

Geographical devision	# of series (geographical devision)	# of series (purpose of travel)	Total
Australia	1	4	5
State	7	28	35
Zone	27	108	135
Region	76	304	380
Total	111	444	555

Table 3: *Mean(RMSE) for ETS, ARIMA and OLS with and without reconciliation - 1-step-ahead - Tourism dataset*

	Mean(RMSE)					
	Unreconciled			Reconciled		
	ETS	ARIMA	OLS	ETS	ARIMA	OLS
Level 0	1516.40	1445.49	1415.06	1533.58	1453.44	1454.39
Level 1	511.37	493.14	510.83	495.88	457.65	488.33
Level 2	214.81	219.01	224.50	209.16	207.52	212.44
Level 3	122.91	125.08	123.97	118.67	120.52	119.52
Level 4	675.99	709.22	694.50	668.26	679.74	678.54
Level 5	213.06	220.08	216.11	210.64	209.39	211.13
Level 6	97.53	102.41	101.03	96.36	99.77	98.56
Level 7	56.17	58.20	58.17	55.98	57.68	57.20

Table 4: *Mean(RMSE) for ETS, ARIMA and OLS with and without reconciliation - 24-step-ahead - Tourism dataset*

	Mean(RMSE)					
	Unreconciled			Reconciled		
	ETS	ARIMA	OLS	ETS	ARIMA	OLS
Level 0	2238.58	3553.99	4194.26	2250.22	3179.39	4194.21
Level 1	593.57	570.13	827.67	553.76	626.32	827.67
Level 2	239.52	229.64	275.99	234.21	242.46	275.99
Level 3	132.58	129.40	144.01	126.74	129.40	144.02
Level 4	766.78	824.00	1274.00	795.48	958.24	1274.01
Level 5	226.74	241.18	285.63	222.48	236.94	285.63
Level 6	103.02	105.38	112.20	101.95	103.93	112.19
Level 7	59.12	58.81	62.54	58.54	58.71	62.55

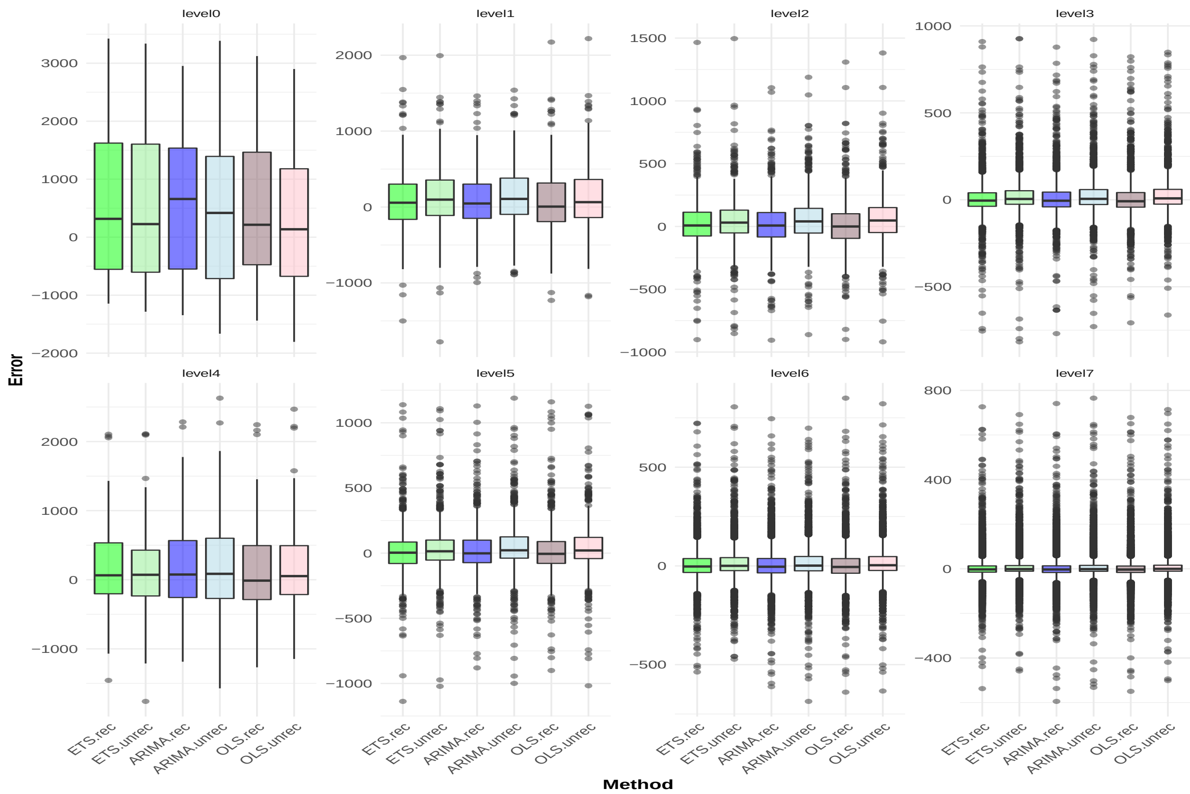

Figure 4: *Box plot for forecast errors - Reconciled and unreconciled ETS, ARIMA and OLS in each hierarchy level for 1-step-ahead tourism demand*

Table 5: *Computation time (seconds) for ETS, ARIMA and OLS with and without reconciliation - 1- and 24-step-ahead - Tourism dataset*

	Computation time (secs)			
	1-step-ahead		24-step-ahead	
	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	16.66	16.85

Figure 8 shows the 1-step-ahead and 24-step-ahead forecast results for one of the bottom level series, BACBus (Geelong - Business). In these plots we have both reconciled (solid lines) and unreconciled (dashed 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.

Table 5 compares the three methods, computation time for 1-step-ahead and 24-step-ahead forecasting. We see that the OLS forecasting model is much faster compared with the other methods. Also, since reconciliation is a linear process, in all methods, it is very fast and does not affect computation time significantly.

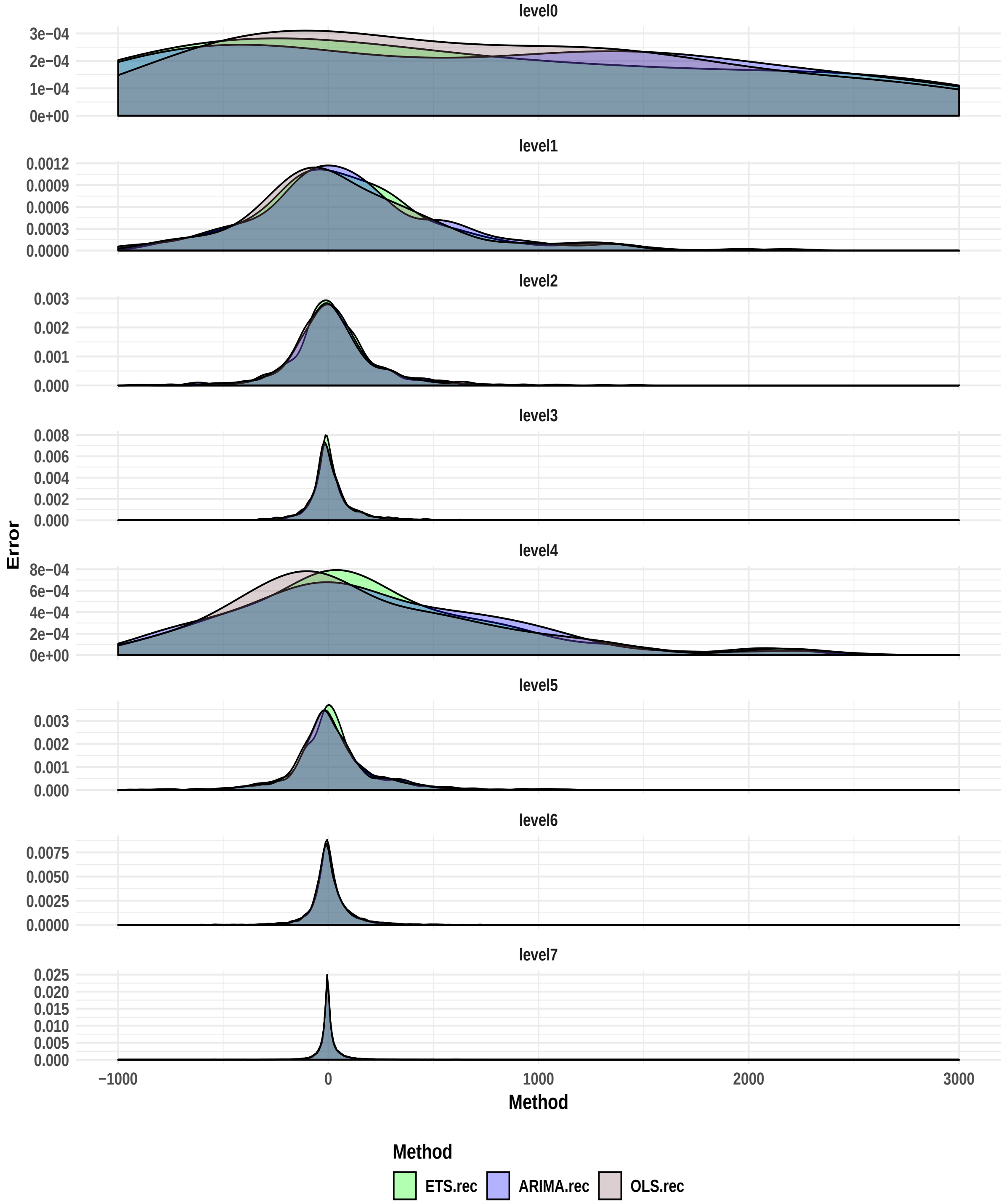

Figure 5: *Density plot for forecast errors - Reconciled and unreconciled ETS, ARIMA and OLS in each hierarchy level for 1-step-ahead tourism demand using interval (-1000,3000)*

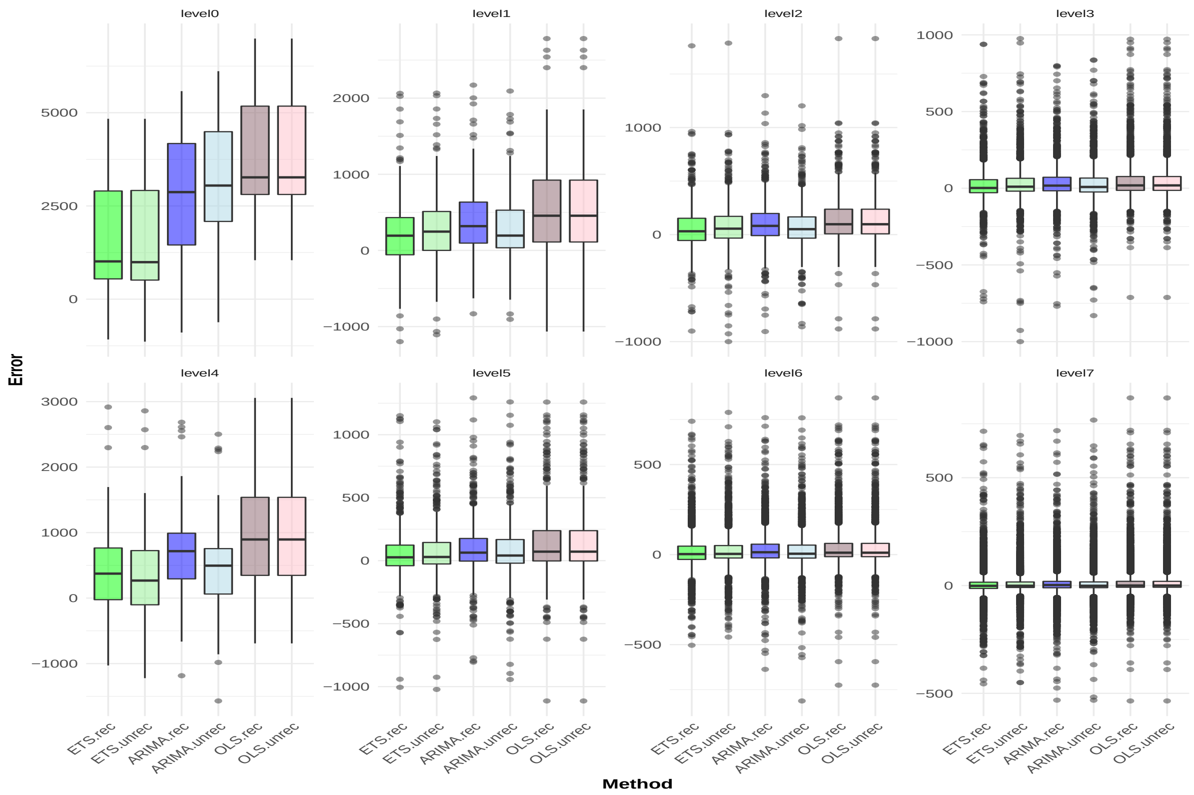

Figure 6: *Box plot for forecast errors - Reconciled and unreconciled ETS, ARIMA and OLS in each hierarchy level for 24-step-ahead tourism demand*

3.2 Wikipedia pageviews

The second dataset consists of 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 compared with the Australian monthly tourism data and forecasting its series is more challenging. It has a grouped structure, with grouping 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 (check Table 6). We display the group structure in Table 6 and Figure 9. In Figure 9 we use one possible hierarchy for this dataset, but the order of the hierarchy can be switched. The final dataset includes 913 time series, each with length 394. The group structure and different levels include:

- Level 0 = Total
- Level 1 = Agent
- Level 2 = Access
- Level 3 = Language
- Level 4 = Purpose

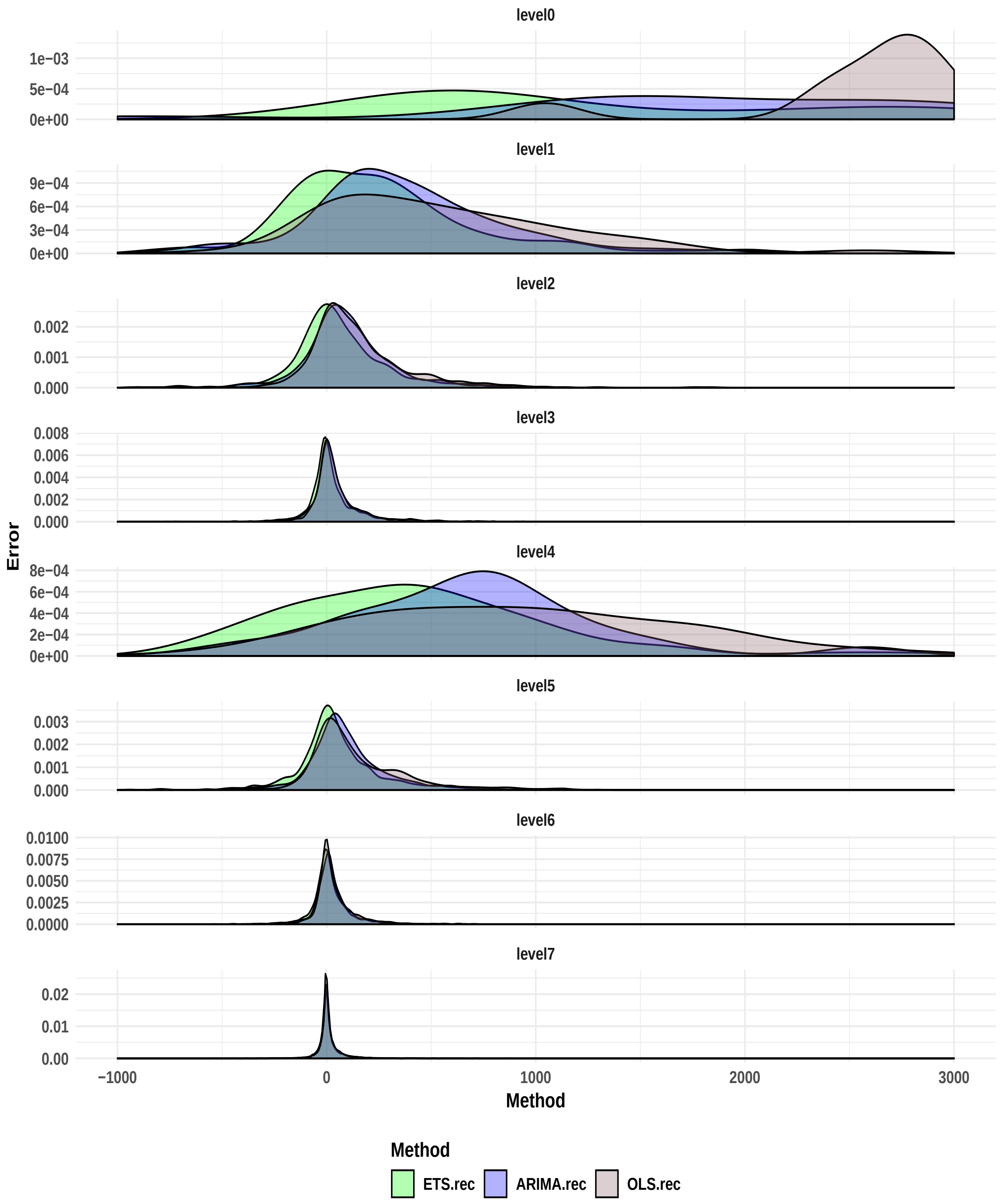

Figure 7: *Density plot for forecast errors - Reconciled and unreconciled ETS, ARIMA and OLS in each hierarchy level for 24-step-ahead tourism demand using interval (-1000,3000)*

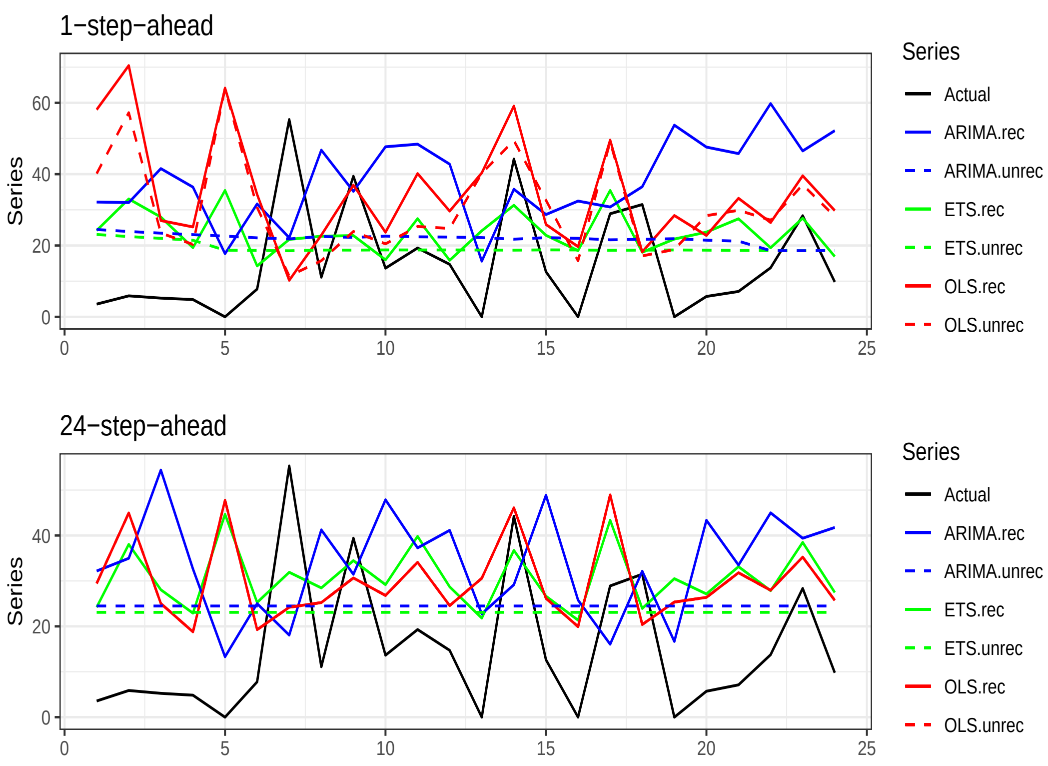

Figure 8: *Comparing Actual test set, Reconciled and unreconciled ETS, ARIMA and OLS for BACBus bottom level series for 1-step-ahead and 24-step-ahead tourism demand*

Table 6: *Social networking Wikipedia article grouping structure*

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

- Level 5 = bottom level series

For this daily dataset, in the OLS forecasting model we include in the predictor matrix a linear trend, 6 seasonal dummies and 7 lags. 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.

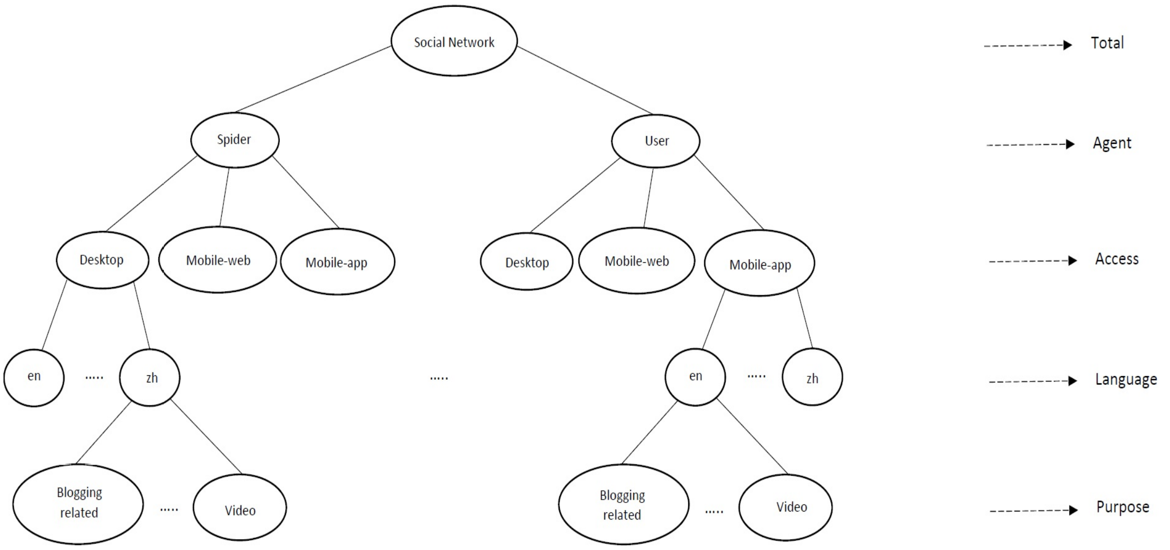

Figure 9: *One of the possile hierarchy structures for Wikipedia pageview dataset*

Table 7, 8 and 9 represent the RMSE results and computation time. Although these time series are noisier, still we 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 in the forecasted errors.

Figures 10 and 12 display the forecast error box plot. These plots are for 1-step-ahead and 28-step-ahead forecasts in each level of grouping. Further, we can see that the error distribution 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.

Figures 11 and 13 show the density plots for the forecast errors. For both 1-step-ahead and 28-step-ahead forecasts, we can see the density structure of the forecast errors across ETS, ARIMA and the OLS forecasting model. Except for the Total series which ETS works better, in all the other levels are the models have similar structure for the forecast errors.

In Figure 14, we display results for one of the bottom level series, desktopusenPho (desktop-user-english-photo sharing). The plot shows 1-step-ahead and 28-step-ahead forecast results for ETS, ARIMA and OLS, with (solid lines) and without (dashed lines) applying reconciliation. We see that the OLS forecasting model performs close to the other two methods, and reconciliation improves the forecasts.

Table 7: *Mean(RMSE) for ETS, ARIMA and OLS with and without reconciliation - 1-step-ahead - Wikipedia dataset*

	Mean(RMSE)					
	Unreconciled			Reconciled		
	ETS	ARIMA	OLS	ETS	ARIMA	OLS
Level 0	10773.66	15060.65	15748.18	11014.73	14276.47	15270.23
Level 1	8272.92	10196.34	10623.85	7736.88	9904.12	10673.98
Level 2	6524.72	6705.03	6979.58	6257.44	7142.49	7285.97
Level 3	4870.08	6333.02	7150.13	4981.91	6369.98	7106.11
Level 4	5233.50	4659.53	4675.18	5001.40	4586.53	4650.26
Level 5	358.90	238.97	254.98	362.25	241.60	256.11

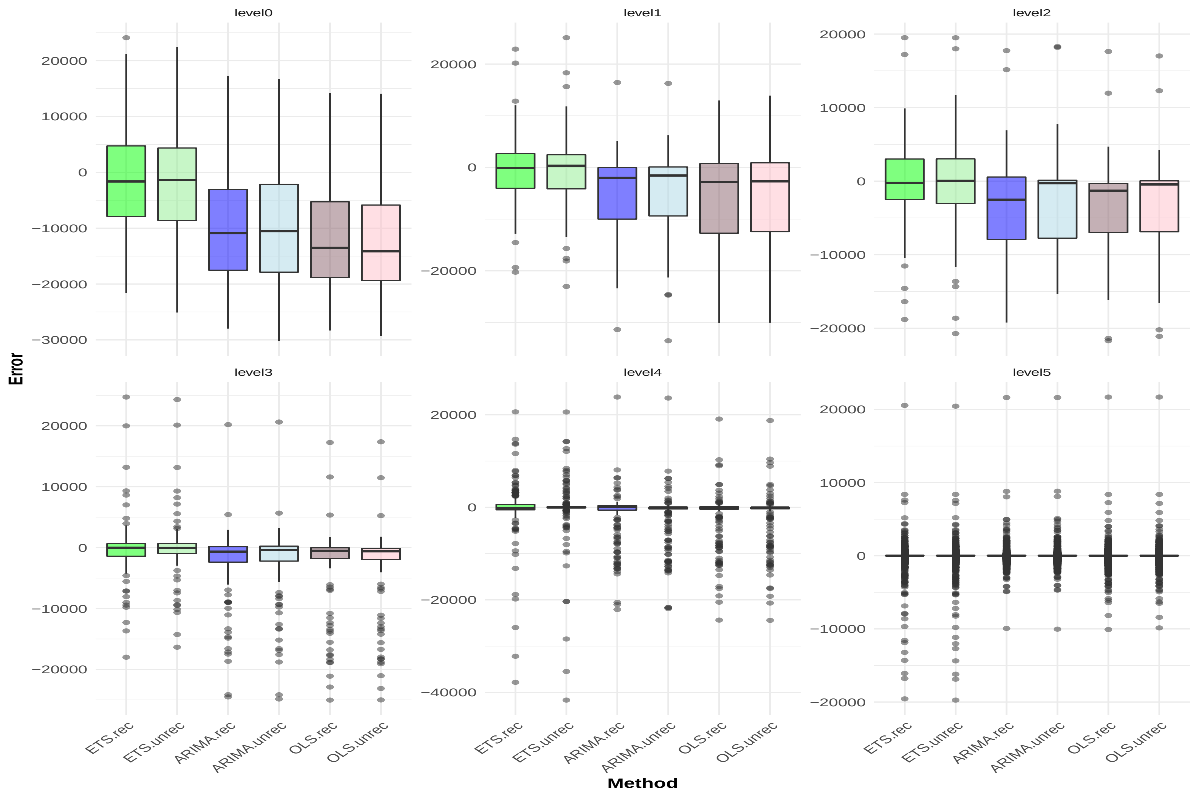

Figure 10: *Box plot for forecast errors - Reconciled and unreconciled ETS, ARIMA and OLS in each hierarchy level for 1-step-ahead Wikipedia pageviews*

Lastly, Table 9 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.

4 Conclusion

We proposed a single-step linear approach to forecast hierarchical or grouped time series in a much faster way, but 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

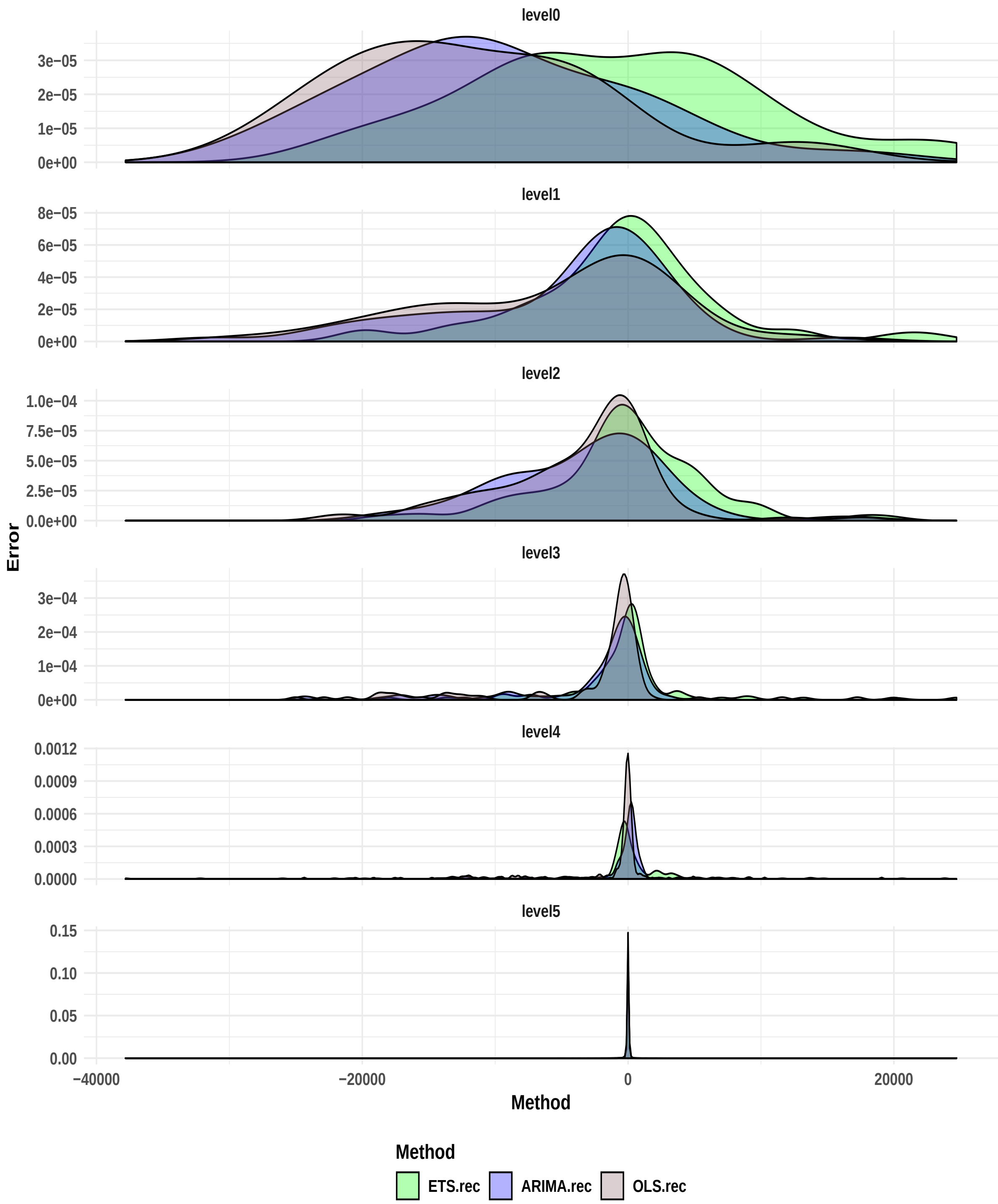

Figure 11: *Density plot for forecast errors - Reconciled and unreconciled ETS, ARIMA and OLS in each hierarchy level for 1-step-ahead Wikipedia pageviews*

Table 8: *Mean(RMSE) for ETS, ARIMA and OLS with and without reconciliation - 28-step-ahead - Wikipedia dataset*

	Mean(RMSE)					
	Unreconciled			Reconciled		
	ETS	ARIMA	OLS	ETS	ARIMA	OLS
Level 0	14846.93	24298.84	29840.58	14999.18	24649.91	29665.70
Level 1	13608.73	17277.01	21165.30	12240.30	16810.45	21048.06
Level 2	7117.43	10731.97	12678.89	7523.43	11068.81	12811.18
Level 3	6475.90	9580.38	12056.62	6509.03	9799.11	12112.46
Level 4	5302.74	8611.25	8451.09	5307.34	8239.77	8460.35
Level 5	435.64	390.05	389.41	437.67	391.22	390.97

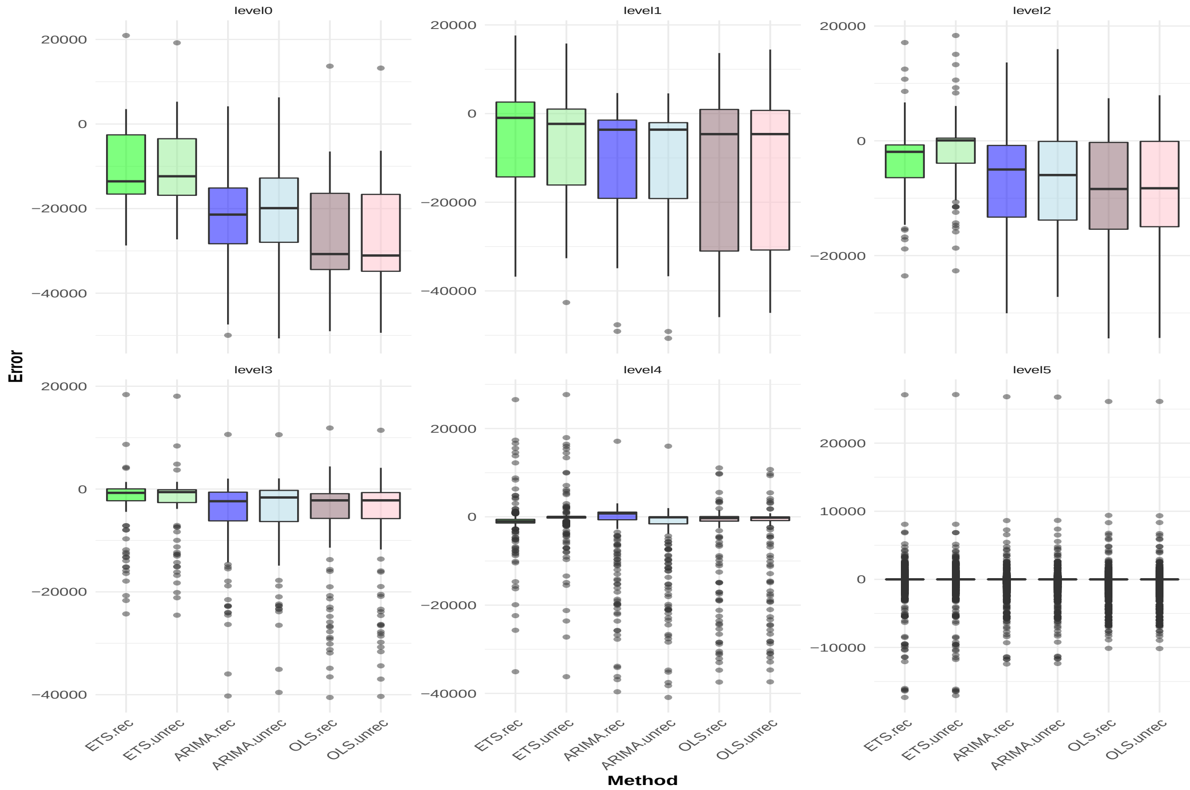

Figure 12: *Box plot for forecast errors - Reconciled and unreconciled ETS, ARIMA and OLS in each hierarchy level for 28-step-ahead Wikipedia pageviews*

Table 9: *Computation time (seconds) for ETS, ARIMA and OLS with and without reconciliation - 1- and 28-step-ahead - Wikipedia dataset*

	Computation time (secs)			
	1-step-ahead		28-step-ahead	
	Unreconciled	Reconciled	Unreconciled	Reconciled
ETS	13963.93	13963.96	450.89	450.92
ARIMA	10327.02	10327.15	670.40	670.44
OLS	82.55	82.62	35.39	35.43

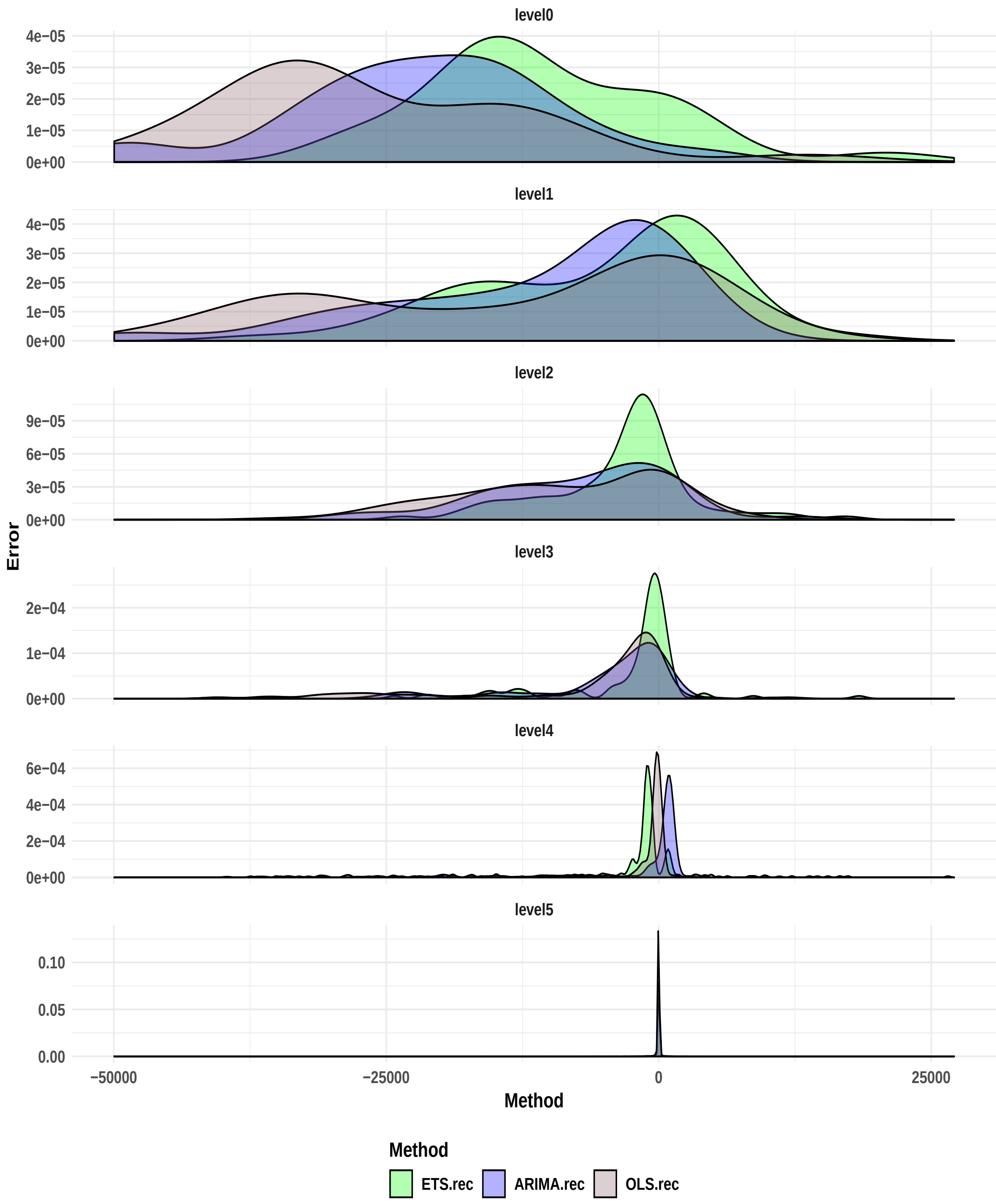

Figure 13: *Density plot for forecast errors - Reconciled and unreconciled ETS, ARIMA and OLS in each hierarchy level for 28-step-ahead Wikipedia pageviews*

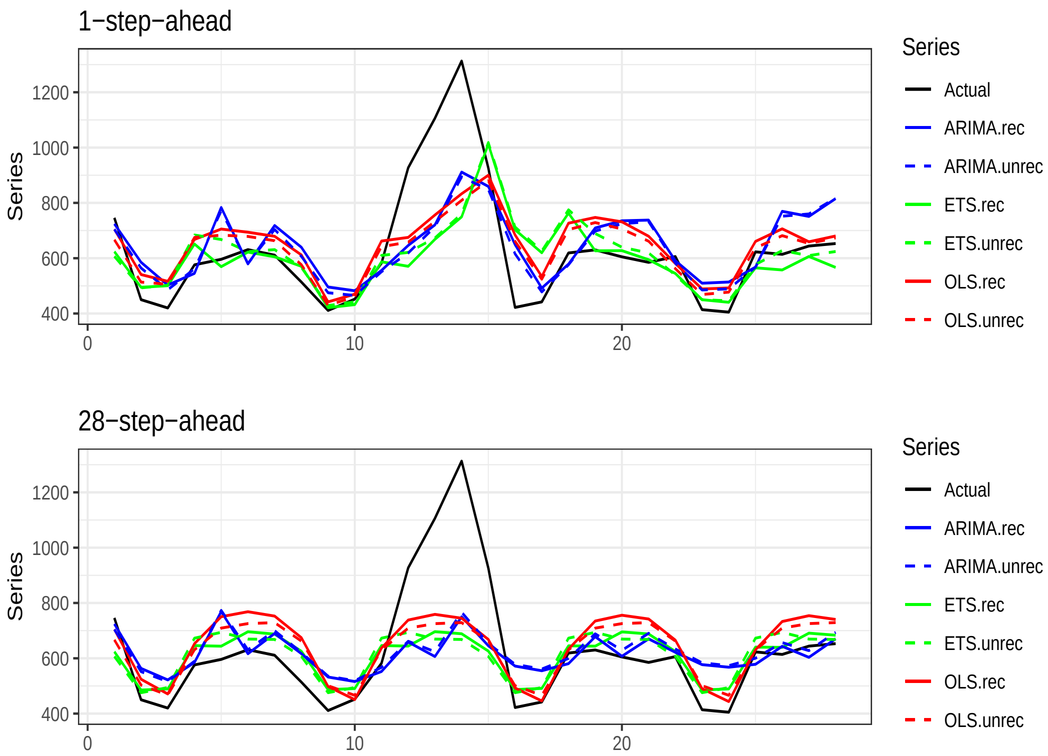

Figure 14: *Comparing Actual test set, Reconciled and unreconciled ETS, ARIMA and OLS for desktopusenPho (desktop-user-english-photo sharing) bottom level series- 1- and 28-step-ahead Wikipedia pageviews*

and grouped structures. Although ETS and ARIMA are good in terms of forecasting power and accuracy, they can be computationally heavy when facing large collections of time series in hierarchy. Adding another faster option for calculating base forecasts was our purpose in this research. Here we suggest a linear model, OLS, instead of ETS and ARIMA which is not computationally intensive. We also showed that OLS can compete ETS and ARIMA in terms of forecasting accuracy level. We also note that OLS has the additional practice feature in handling missing data while ETS and ARIMA requiers imputation. One more important feature of our model is the ability to easily include external information such as holiday dummies or other external series. In addition to the computation adjustment, our proposed approach forecasts hierarchical time series in a parsimoneas single step whereas other available methods all forecast in two-steps.

5 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].

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.

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

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, RA Ahmed, HL Shang & E Wang (2015). hts: Hierarchical and grouped time series. *R package version* **4**.

Hyndman, RJ & G Athanasopoulos (2018). *Forecasting: principles and practice*. OTexts.

Hyndman, RJ, Y Khandakar, et al. (2007). *Automatic time series for forecasting: the forecast package for R*. 6/07. Monash University, Department of Econometrics and Business Statistics.

Hyndman, RJ, AJ Lee & E Wang (2016). Fast computation of reconciled forecasts for hierarchical and grouped time series. *Computational Statistics & Data Analysis* **97**, 16–32.

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

Team, RC (2013). R: A language and environment for statistical computing. *URL http://www.R-project.org/.*

Wickramasuriya, SL, G Athanasopoulos & RJ Hyndman (2018). Optimal forecast reconciliation for hierarchical and grouped time series through trace minimization. *Journal of the American Statistical Association* (just-accepted), 1–45.