The M5 Accuracy competition: Results, findings and conclusions

Spyros Makridakis[a], Evangelos Spiliotis[b,*], Vassilios Assimakopoulos[b]

[a] *Institute For the Future, University of Nicosia, Cyprus*
[b] *Forecasting and Strategy Unit, School of Electrical and Computer Engineering, National Technical University of Athens, Greece*

Abstract

This paper describes the M5 "Accuracy" competition, the first of two parallel challenges of the latest M competition whose aim is to advance the theory and practice of forecasting. The main objective of the M5 "Accuracy" competition was to accurately predict 42,840 time series that represent the hierarchical unit sales of the largest retail company in the world by revenue, Walmart. To do so, the competition required the submission of 30,490 point forecasts for the lowest cross-sectional aggregation level of the data that could then be summed up accordingly to estimate the forecasts for the rest of the upward levels. The paper provides details on the implementation of the M5 "Accuracy" challenge, presents its results and the top performing methods, and summarizes its major findings and conclusions. Finally, it discusses the implications of its findings and suggests directions for future research.

Keywords: Forecasting Competitions, M Competitions, Accuracy, Time Series, Machine Learning, Retail Sales Forecasting

*Corresponding author
Email address: spiliotis@fsu.gr (Evangelos Spiliotis)

1. Introduction

Forecasts are indispensable for a great number of daily decisions we make, from what time to get up in the morning in order to not be late for work, to which brand of a TV to buy that would provide the best value for money. Supermarkets require forecasts on what products their clients will buy so that they will have them available when needed. What is important is that such forecasts should be as accurate as possible, since stocking too many products costs extra money while not having enough would mean lost sales and less profits. The M competitions, which have been taking place for almost 40 years (Makridakis et al., 1982, 1993; Makridakis & Hibon, 2000; Makridakis et al., 2020d), aim to identify ways to improve forecasting accuracy as much as possible. This is achieved by empirically evaluating several forecasting methods and determining the most accurate one(s). Such findings have significantly influenced the theory and practice of forecasting, offering valuable insights about the ways that accuracy can be improved (Hyndman, 2020). The first three competitions demonstrated the value of combining, the potential of automatic forecasting methods, and the merits of simplicity, among others, with the fourth competition proving that Machine Learning (ML) methods and a hybrid approach, utilizing "cross-learning", were more successful in their forecasts than the alternatives.

The M5 competition extended the objectives of the previous four by focusing on a retail sales forecasting application and using real-life, hierarchically structured sales data that display intermittency and erraticness (Syntetos & Boylan, 2005; Syntetos et al., 2005). The competition attracted a great number of participants, eager to experiment with effective forecasting solutions for such a real-life situation faced by numerous retail companies on a daily basis. LightGBM, a simple ML approach, reported superior forecasting performance over all other alternatives and was used by practically all top 50 competitors, indicating that the method can be adopted by retail firms to significantly improve the accuracy of their sales predictions and daily operation. In addition, the results of the M5 confirmed most of the key findings of the previous M competitions, further advancing the theory and practice of forecasting in the area of hierarchical retail sales. This paper presents the results of the competition and its top-performing methods, comparing their performance to statistical and other benchmarks while summarizing its key findings and conclusions.

2. Implementation and execution

The M5 "Accuracy" competition was organized following the general principles described by Makridakis et al. (2020b). The competition began on March 3$^{\text{rd}}$, 2020, when the initial train data set became available to download on the Kaggle platform[1], and ended on June 30$^{\text{th}}$, 2020, when the final leaderboard was announced. Moreover, the competition was chronologically divided into two phases which were used for the evaluation

[1] https://www.kaggle.com/c/m5-forecasting-accuracy

of the teams. The first, called the "validation" phase, was used to allow the teams to receive feedback about their performance and guide the development of their forecasting models. The second, called the "test" phase, was used for the final evaluation of the teams. The data set used involved the unit sales of 3,049 products sold by Walmart in the USA, organized in the form of grouped time series that are aggregated based on their type (category and department) and selling location (stores and states), thus consisting of a total of 42,840 series in 12 cross-sectional, aggregation levels.

The implementation of the M5 "Accuracy" competition differed from the "Uncertainty" one (Makridakis et al., 2020e) in the following four aspects: (i) Submission template, (ii) performance measure, (iii) prizes, and (iv) benchmarks. The first three aspects are described in the following subsections, while the last in the Appendix of the supplementary material.

2.1. Submission

All forecasts were submitted through the Kaggle platform using the template provided by the organizers. The template for the M5 "Accuracy" competition required the submission of the forecasts that corresponded just to the 30,490 series of the lowest cross-sectional aggregation level of the data set (level 12), not all 42,840 series of the competition. This was done because the M5 series are hierarchically structured and, as a result, we expect the corresponding forecasts to be coherent (forecasts at the lower levels have to sum up to the ones at the higher levels so that the forecasts across different levels are aligned; Spiliotis et al., 2019b). In other words, it was assumed that the forecasting approaches used by the contestants to predict all 42,840 series of the competition resulted in coherent forecasts and, as a result, the forecasts at the lowest aggregation level could be properly aggregated (summed up) to automatically compute the ones at the rest of the levels.

Note that the submission template did not affect the way the forecasts were produced and teams were completely free to use their forecasting method of choice to forecast the individual series. However, it ensured that the forecasts were coherent and, therefore, in an appropriate form to be evaluated in a direct way. A team, for instance, could just forecast the series at the most disaggregated level of the competition (level 12) and derive the remaining forecasts using the bottom-up approach. Another could just forecast the most aggregated series of the competition (level 1) and compute the remaining ones using proportions (top-down method). A mix of the previous two approaches was also possible (middle-out method). Finally, predicting the series of all levels and obtaining the ones of the lowest level through an appropriate weighting scheme was another option (Hyndman et al., 2011). The benchmarks of the competition apply some of these options, involving some indicative forecasting approaches that utilize the bottom-up and top-down methods, as well as a combination of the two (Abouarghoub et al., 2018).

Note that on the Kaggle platform, teams were allowed to submit a maximum of five entries per day. However, for their final evaluation in the test phase, each team had to select a single set of forecasts (one submission) since in real life forecasters face the same problem of choosing a single set of forecasts which

they believe will represent the future as adequately as possible. If no particular submission was selected, the one with the highest performance during the "validation" phase was automatically selected by the system.

2.2. Performance measure

The academic literature involves various measures for evaluating point forecast accuracy (Hyndman & Koehler, 2006). The first three M competitions considered several of these measures, while the M4 examined the overall weighted average of the symmetric mean absolute percentage error (sMAPE; Makridakis, 1993) and a variant of the mean absolute scaled error (MASE; Hyndman & Koehler, 2006). Undoubtedly, no measure is perfect as they all have advantages and drawbacks (Goodwin & Lawton, 1999; Kolassa, 2020). The comments made about the measures utilized in all previous M competitions by the invited commentators clearly demonstrate this lack of agreement, and also highlight that each forecaster has his/her own preferences (Makridakis et al., 2020c). We believe that from the measures commonly used in the literature to assess forecasting accuracy, those based on scaled errors probably display the most preferable statistical properties. For this reason, the M5 "Accuracy" competition utilized a variant of the MASE originally proposed by Hyndman & Koehler (2006), the Root Mean Squared Scaled Error (RMSSE). The measure is defined as follows:

$$RMSSE = \sqrt{\frac{\frac{1}{h}\sum_{t=n+1}^{n+h}(y_t - \hat{y}_t)^2}{\frac{1}{n-1}\sum_{t=2}^{n}(y_t - y_{t-1})^2}},$$

where y_t is the actual future value of the examined time series at point t, $\hat{y}_t$ the forecast of the method being evaluated, n the length of the training sample (number of historical observations), and h the forecasting horizon (28 days). Note that the denominator of RMSSE (in-sample, one-step-ahead mean squared error of the Naive method) is computed only for the periods during which the examined product(s) are actively sold, i.e. the periods following the first non-zero demand observed for the series under evaluation. This is done since many of the products included in the data set started being sold later than the first available date (Makridakis et al., 2020b).

Like MASE, RMSSE is independent of the scale of the data, has a predictable behavior, i.e. becomes infinite or undefined only when all the errors of the Naive method are equal to zero, has a defined mean and a finite variance, and is symmetric in the sense that it penalizes equally positive and negative forecast errors, as well as large and small ones. The choice for this particular measure can be further justified as follows:

- Many of the competition's series are characterized by intermittency, involving sporadic unit sales with many zeros. This means that absolute errors, which are optimized for the median (Schwertman et al.,

1990), would assign lower scores (better accuracy) to forecasting methods that derive forecasts close to zero. However, the objective of the competition is to accurately forecast the average sales. As a result, the accuracy measure used builds on squared errors, which are optimized for the mean (Kolassa, 2016).

- In contrast to other measures of similar statistical properties, such as relative errors and measures (Davydenko & Fildes, 2013), RMSSE can be safely computed for all M5 series as it does not rely on divisions with values that could be equal or close to zero. For example, this is typically the case in percentage errors when y_t is equal to zero or relative errors when the error of the benchmark used for scaling is zero.

After estimating RMSSE for all the 42,840 time series of the competition (average accuracy reported for each series across the complete forecasting horizon), the overall accuracy of the forecasting method is computed by averaging the RMSSE scores across all the series of the data set using appropriate weights. The measure, to be called Weighted RMSSE (WRMSSE), is defined as follows:

$$WRMSSE = \sum_{t=1}^{42,840} w_i \times RMSSE_i, \tag{1}$$

where w_i and $RMSSE_i$ is the weight and the RMSSE is the score of the i^{th} series of the competition, respectively. The weights are computed based on the last 28 observations of the final training sample of the data set, specifically based on the cumulative actual dollar sales that each series displayed in that particular period (sum of units sold multiplied by their respective price). Lower WRMSSE scores indicate more accurate forecasts. Note that the estimation of WRMSSE differs from the approaches adopted in the previous M competitions. In the first three competitions, all errors were computed both per series and per forecasting horizon, and then equally averaged together. In the M4, the errors were first averaged per series, exactly as done in M5, but then averaged again using equal weights.

We believe that the weighting scheme adopted in the M5 competition, involving the unit sales of various products of different selling volumes and prices that are organized in a hierarchical fashion, is more appropriate for successfully identifying forecasting methods that add significant value to retail companies interested in accurately forecasting the series that mostly translate to relatively higher revenues. Business-wise, in order for a forecasting method to be considered appropriate, it must provide accurate forecasts across all aggregation levels, especially for series of high importance, i.e. series that represent significant sales, measured in monetary terms. In other words, we expect the "best" performing forecasting methods to derive lower forecasting errors for the series that are more valuable to the company.

Note that, according to WRMSSE, all aggregation levels are equally weighted. The reason is that the total dollar sales of a product, measured across all three states, are equal to the sum of the dollar sales

of this product when measured across all ten stores. Similarly, the total dollar sales of a store's product category are equal to the sum of the dollar sales of the departments that this category consists of, as well as the sum of the dollar sales of the corresponding departments' products. Moreover, given that M5 does not focus on a particular decision-making problem, there is no obvious reason for weighting the individual levels unequally.

An indicative example for computing WRMSSE can be found in the Competitors' Guide of the competition, available on the M5 website[2]. The code for estimating WRMSSE, as well as the exact weight of each series, can be found in the GitHub repository of the competition.

2.3. Prizes

In order for a team to be eligible for a prize, point forecasts had to be provided for all 30,490 series of the competition's 12^{th} aggregation level (product-store level), which will consequently be aggregated (summed up) to produce forecasts for the rest of the levels. Moreover, winning teams had to provide code for reproducing the forecasts originally submitted to the competition, as well as some documentation for understanding the forecasting method used.

Just like in M4, objectivity and reproducibility was a prerequisite for collecting any prize (Makridakis et al., 2018a) and therefore, the winning teams, with the exception of companies providing forecasting services and those claiming proprietary software, had to upload their code onto the Kaggle platform no later than 14 days after the end of the competition (i.e. the 14^{th} of July, 2020). This material was later uploaded onto the M5 public GitHub repository, in order for individuals and companies interested in using the winning methods to be able to do so, while crediting the team that had developed them. Companies providing forecasting services and those claiming proprietary software had to provide the organizers with a detailed description of how their forecasts were made and a source or execution file for reproducing their forecasts.

After receiving the code and documentation from all the winning teams, the organizers evaluated the reproducibility of their results. Since ML algorithms typically involve random initializations, the organizers considered as fully reproducible any method that displayed a reproducibility rate, i.e. absolute percentage difference of WRMSSE between the original and reproduced forecasts, higher than 98%. Although all winning methods were found to be fully reproducible, if this were not true, the prizes would have been given to the next best-performing and fully reproducible submission.

The prizes of the M5 "Accuracy" competition are listed in Table 1. Note that there were no restrictions preventing a team from collecting both a regular and a student[3] prize. Moreover, there were no restrictions

[2]https://mofc.unic.ac.cy/m5-competition/

[3]A student team is one for which at least half of the team members are current full-time students. Teams that were eligible for the student prize have a name followed by "_STU".

preventing a team from collecting a prize in both the M5 "Accuracy" and the M5 "Uncertainty" competitions. The awards were given during the virtual, online M5 conference on October 29th, 2020.

Table 1: The six prizes of the M5 "Accuracy" competition.

Prize name	Description	Amount
1st prize	Best-performing method according to WRMSSE	$25,000
2nd prize	Second best-performing method according to WRMSSE	$10,000
3rd prize	Third best-performing method according to WRMSSE	$5,000
4th prize	Fourth best-performing method according to WRMSSE	$3,000
5th prize	Fifth best-performing method according to WRMSSE	$2,000
Student prize	Best-performing method among student teams according to WRMSSE.	$5,000
Total		$50,000

An amount of $40,000 was generously provided by Kaggle, that also waived the fees for hosting the M5 competition. In addition, Google and MOFC generously provided $20,000 each, while Walmart, apart from the M5 data set, also generously provided an amount of $10,000. Finally, the global transportation technology company Uber generously provided $5,000, while IIF generously provided another $5,000. The total amount of $100,000 was equally distributed between the accuracy and uncertainty challenges of the M5 competition.

3. Participating teams and submissions

The M5 "Accuracy" competition involved 7,092 participants on 5,507 teams from 101 countries. Of these teams, 4,373 entered the competition during the validation phase and 1,134 during the test phase. Moreover, 1,434 teams made submissions during both the validation and the test phase of the competition, while 2,939 only during the validation phase. In total, the participating teams made 88,136 submissions, most of which (about 78.3%) were submitted during the validation phase. Note that most of the teams made a single submission, while the majority of the rest made between three and 20 submissions. It is worth mentioning that for 1,563 participants, including 15 in the top 100, this was their first time participating in a Kaggle competition.

Unfortunately, due to privacy regulations, no information was made available about the background (academic, research, business or other) of the participating teams, their experience and skills, and the type of methods utilized (e.g. statistical, ML, combination or hybrid), with the exception of the winning teams and a few more that were willing to share this information with the organizers. However, based on the general characteristics of the Kaggle community, we assume that most of the teams had an adequate background

in statistics and computer science, and were also familiar with ML forecasting methods, such as Neural Networks (NNs) and Regression Trees (RTs).

Out of the participating teams, 2,666 (48.4%) managed to outperform the Naive benchmark, 1,972 (35.8%) outperformed the sNaive benchmark, and 415 (7.5%) beat the top-performing benchmark (ES_bu). However, it is important to note that these numbers refer to the forecasts selected by each team for the final evaluation of their performance and not to the "best" submission made per case while the competition was still running. In the latter case, 3,510 (63.7%), 2,685 (48.8%), and 672 (12.2%) teams would have managed to outperform the Naive, sNaive, and ES_bu benchmarks, respectively. This indicates that many teams failed to choose the best method developed, probably due to misleading validation scores.

Figure 1 summarizes this information, presenting the daily number of submissions made and the cumulative number of participating teams, the number of participants per country, the distribution of accuracy of the teams that did better than the Naive benchmark, and the accuracy of the teams that did better than the top-performing benchmark, along with their respective ranks.

By observing Figure 1 we find that:

- The majority of the teams made most of their submissions during the validation phase, when the public leaderboard was available and live feedback was received. During the test phase, most of the teams probably used their own, private cross-validation (CV) strategies to fine-tune their methods, which were mainly submitted four days before the competition ended.

- The majority of the participants originated from USA (17%), Japan (17%), India (10%), China (10%), and Russia (6%). Thus, we conclude that there is a large, active community interested in forecasting in both developed and developing countries.

- Only a limited number of teams managed to outperform the top performing benchmark of the competition, with the majority of the teams being outperformed by more than 13% by the top ES_bu.

- From the 415 teams that managed to outperform all the benchmarks of the competition, five displayed an improvement greater than 20%, 42 greater than 15%, 106 greater than 10%, and 249 greater than 5%. These improvements are substantial and demonstrate the superiority of the M5 methods over the standard forecasting benchmarks. Moreover, the five winners of the competition were the only teams to accomplish an accuracy improvement greater than 20%, thus achieving a clear victory.

The various tables presented in the remainder of this paper focus on the top 50 performing teams of the competition, as well as the benchmarks considered by the organizers. The reasoning is twofold: First, for practical reasons, as it would be impossible to analyze and report in detail the results from all the teams that participated in the competition. Second, given that very few teams were willing to share detailed

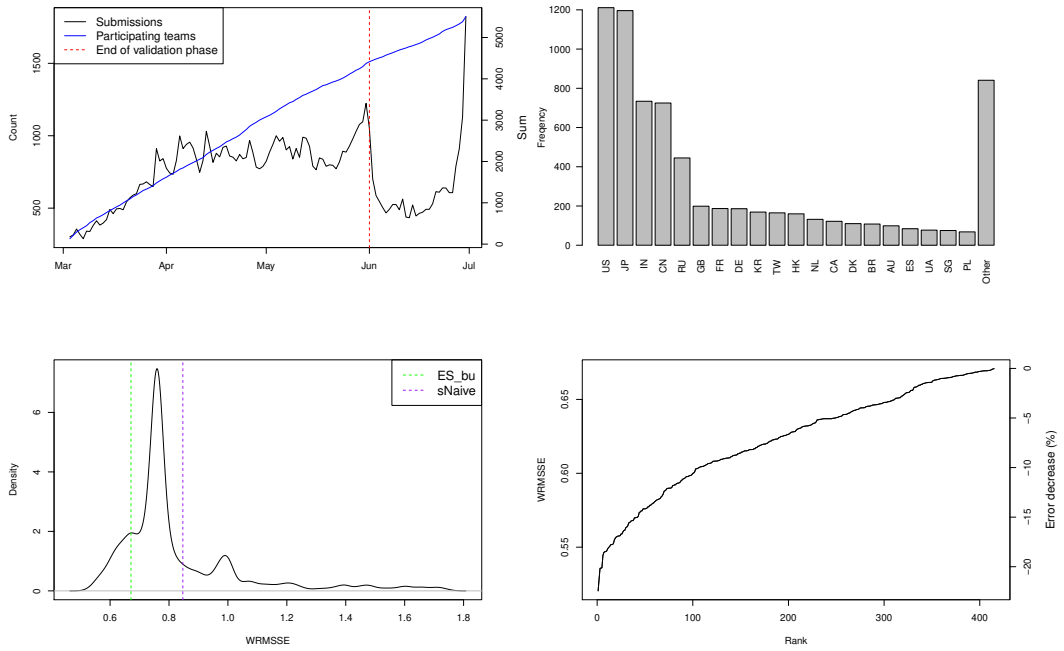

Figure 1: Summary of the participating teams and submissions made. Top left: The daily number of submissions made (black line) and the cumulative number of participating teams (blue line). The red dotted line indicates the end of the validation phase; Top right: Number of participants per country (top 20 in terms of participation), as estimated based on their IP address; Bottom left: The distribution of the accuracy (WRMSSE) achieved by the teams that did better than the Naive benchmark. The green dotted line indicates the accuracy of the ES_bu benchmark, while the purple dotted line the accuracy of sNaive; Bottom right: The accuracy (WRMSSE) and ranks of the teams that did better than the top-performing benchmark (ES_bu). Percentage improvements over ES_bu are also reported.

information about the methods utilized, we feel that there is more to learn from the top performers for which detailed information is available. Furthermore, given the complexity of the data and the competition in general, we believe it is safer to draw conclusions from methods that worked well rather than rationalizing why some methods performed poorly.

4. Results, winning submissions, and key findings

4.1. Results

Table 2 presents the accuracy (WRMSSE) achieved by the top 50 teams of the competition, both overall and across the 12 aggregation levels. The last column of the table displays the overall (42,840 series) percentage improvement of each team over the top-performing benchmark (ES_bu), whose performance is displayed at the bottom of the table.

Table 2: The performance of the top 50 teams of the M5 "Accuracy" competition in terms of WRMSSE. The results are presented both per aggregation level and overall. Overall percentage improvements are also reported in comparison to the top-performing benchmark (ES_bu).

Rank	Team	Aggregation level												Average	Improvement over ES_bu (%)
		1	2	3	4	5	6	7	8	9	10	11	12		
1	YJ_STU	0.199	0.310	**0.400**	0.277	0.365	0.390	**0.474**	**0.480**	**0.573**	0.966	0.929	0.884	**0.520**	22.4
2	Matthias	0.186	**0.294**	0.416	**0.246**	**0.349**	**0.381**	0.481	0.497	0.594	1.023	0.964	0.907	0.528	21.3
3	mf	0.236	0.319	0.421	0.308	0.397	0.405	0.496	0.505	0.600	**0.950**	**0.917**	0.875	0.536	20.2
4	monsaraida	0.254	0.340	0.418	0.302	0.377	0.411	0.483	0.490	0.579	0.963	0.928	0.886	0.536	20.1
5	Alan Lahoud	0.213	0.324	0.414	0.272	0.361	0.416	0.494	0.503	0.595	0.995	0.950	0.897	0.536	20.1
6	wyzJack_STU	0.248	0.367	0.431	0.319	0.396	0.436	0.502	0.502	0.584	0.953	0.918	**0.875**	0.544	18.9
7	RandomLearner	0.194	0.317	0.423	0.276	0.404	0.408	0.516	0.503	0.608	1.029	0.968	0.910	0.546	18.6
8	SHJ	0.279	0.357	0.419	0.336	0.406	0.429	0.498	0.497	0.586	0.956	0.922	0.878	0.547	18.5
9	gest 2	0.197	0.322	0.424	0.269	0.406	0.420	0.536	0.513	0.624	1.000	0.953	0.901	0.547	18.5
10	DenisKokosinskiy_STU	0.294	0.363	0.419	0.341	0.401	0.429	0.493	0.494	0.581	0.955	0.921	0.878	0.547	18.4
11	XueWang	0.288	0.358	0.417	0.348	0.423	0.424	0.501	0.490	0.582	0.959	0.921	0.876	0.549	18.2
12	yq_STU	0.226	0.320	0.456	0.294	0.403	0.399	0.496	0.526	0.614	1.010	0.954	0.899	0.550	18.1
13	PoHaoChou	0.212	0.317	0.459	0.322	0.402	0.413	0.504	0.539	0.630	0.968	0.940	0.898	0.550	18.0
14	Tsuru	0.257	0.335	0.402	0.325	0.416	0.421	0.506	0.503	0.608	0.994	0.951	0.900	0.552	17.8
15	bk_18	0.217	0.333	0.420	0.303	0.433	0.431	0.537	0.510	0.615	0.986	0.943	0.893	0.552	17.8
16	N60610	0.195	0.350	0.436	0.298	0.409	0.441	0.530	0.521	0.619	0.976	0.945	0.900	0.552	17.8
17	MonashSL_STU	0.247	0.342	0.446	0.308	0.404	0.412	0.501	0.520	0.622	0.992	0.944	0.892	0.552	17.7
18	leoclement	0.270	0.354	0.410	0.322	0.415	0.434	0.526	0.498	0.603	0.986	0.945	0.896	0.555	17.3
19	minghui_Tju	0.254	0.365	0.428	0.327	0.425	0.439	0.529	0.505	0.602	0.979	0.936	0.886	0.556	17.1
20	zfc613	0.236	0.343	0.470	0.291	0.387	0.412	0.506	0.539	0.627	1.008	0.959	0.907	0.557	17.0
21	Nodalpoints	0.312	0.376	0.423	0.353	0.416	0.443	0.508	0.500	0.587	0.963	0.925	0.880	0.557	17.0
22	CPUkiller	0.263	0.367	0.427	0.333	0.431	0.438	0.529	0.504	0.601	0.979	0.935	0.885	0.558	16.9
23	dont overfit	0.263	0.367	0.427	0.333	0.431	0.438	0.529	0.504	0.601	0.979	0.935	0.885	0.558	16.9
24	Dan Hargreaves	0.269	0.350	0.439	0.316	0.405	0.431	0.517	0.522	0.616	0.984	0.946	0.900	0.558	16.9
25	M0T0_STU	0.252	0.346	0.425	0.345	0.431	0.445	0.530	0.526	0.623	0.967	0.932	0.886	0.559	16.7
26	Genryu	0.295	0.368	0.436	0.340	0.410	0.445	0.515	0.523	0.611	0.961	0.924	0.880	0.559	16.7
27	Moscow Five	0.245	0.349	0.442	0.309	0.435	0.436	0.542	0.526	0.624	0.988	0.941	0.889	0.560	16.5
28	Daniela A	**0.162**	0.333	0.483	0.278	0.441	0.412	0.569	0.558	0.679	0.988	0.942	0.892	0.561	16.3
29	shuheioka	0.267	0.354	0.440	0.313	0.419	0.430	0.524	0.517	0.616	1.002	0.954	0.903	0.562	16.3
30	sk 2	0.191	0.381	0.511	0.263	0.364	0.470	0.552	0.585	0.661	0.962	0.932	0.887	0.563	16.1
31	nagao	0.279	0.382	0.443	0.328	0.413	0.456	0.531	0.523	0.609	0.975	0.936	0.889	0.564	16.0
32	AjayNagar	0.221	0.324	0.518	0.285	0.412	0.400	0.515	0.573	0.660	1.010	0.954	0.900	0.564	15.9
33	cjwh	0.248	0.348	0.449	0.314	0.420	0.442	0.544	0.535	0.639	1.002	0.955	0.903	0.566	15.6
34	CWD75	0.237	0.326	0.422	0.330	0.452	0.442	0.551	0.526	0.637	1.004	0.960	0.912	0.567	15.6
35	Groot	0.278	0.384	0.443	0.342	0.432	0.458	0.540	0.519	0.611	0.979	0.937	0.887	0.567	15.4
36	Astral	0.299	0.381	0.453	0.342	0.401	0.453	0.520	0.528	0.611	0.984	0.945	0.896	0.568	15.4
37	Logistic	0.278	0.386	0.445	0.344	0.436	0.457	0.541	0.518	0.610	0.979	0.936	0.886	0.568	15.3
38	jdsc_perceiving_team	0.262	0.372	0.461	0.326	0.433	0.445	0.532	0.531	0.623	0.990	0.948	0.897	0.568	15.3
39	Abzal	0.314	0.373	0.434	0.351	0.420	0.447	0.519	0.515	0.603	0.998	0.956	0.906	0.570	15.1
40	Pianus	0.287	0.383	0.473	0.342	0.435	0.451	0.535	0.536	0.626	0.964	0.926	0.880	0.570	15.1
41	NAU	0.277	0.366	0.456	0.310	0.425	0.440	0.537	0.532	0.633	1.002	0.957	0.906	0.570	15.0
42	shirokane_friends	0.300	0.387	0.454	0.347	0.429	0.461	0.540	0.534	0.619	0.965	0.926	0.880	0.570	15.0
43	Alexnet	0.301	0.390	0.444	0.353	0.435	0.463	0.540	0.520	0.610	0.975	0.934	0.885	0.571	14.9
44	Griffin_Series	0.317	0.380	0.469	0.361	0.442	0.448	0.527	0.529	0.618	0.971	0.933	0.887	0.574	14.5
45	Hiromitsu Kigure	0.291	0.380	0.462	0.342	0.428	0.449	0.533	0.535	0.629	0.991	0.950	0.895	0.574	14.5
46	YK	0.247	0.369	0.464	0.314	0.438	0.453	0.551	0.542	0.644	1.011	0.958	0.904	0.575	14.4
47	PASSTA	0.339	0.396	0.460	0.366	0.421	0.457	0.521	0.532	0.614	0.970	0.933	0.886	0.575	14.4
48	golubyatniks	0.359	0.413	0.455	0.387	0.434	0.466	0.519	0.521	0.600	0.956	0.922	0.879	0.576	14.2
49	belkasanek	0.184	0.329	0.538	0.260	0.427	0.416	0.549	0.608	0.701	1.028	0.964	0.905	0.576	14.2
50	Random_prediction	0.249	0.348	0.455	0.347	0.457	0.460	0.563	0.558	0.655	0.986	0.943	0.890	0.576	14.2
416	ES_bu - Benchmark	0.426	0.514	0.580	0.478	0.557	0.577	0.654	0.643	0.728	1.012	0.969	0.915	0.671	-

By observing Table 2 we find that all top 50 submissions improve the overall forecasting accuracy of the top-performing benchmark by more than 14%, while the improvements are higher than 20% for the top

five performing teams and an impressive 22.4% for the winning team. Taking into consideration that the improvements of the winning submissions of the M3 and M4 competitions over the corresponding benchmarks were less than 10% (Makridakis et al., 2020d), we can conclude that M5 included more accurate approaches that reduced the error over the most accurate benchmark by more than one fifth. This means that retail and logistic companies could benefit substantially from utilizing such innovative forecasting approaches in their forecasting practice, where small improvements in accuracy lead to considerable inventory reductions (Syntetos et al., 2010) and slight inaccuracies to higher stock holdings and lower service levels (Ghobbar & Friend, 2003; Pooya et al., 2019).

Another interesting finding is that the winning team (*YJ_STU*) does not display the most accurate forecasts across all 12 aggregation levels; it is the best approach at only levels 3, 7, 8, and 9, and the second best at levels 2 and 6. This is particularly true for the lowest three aggregation levels of the data set (10, 11, and 12) where, out of the 50 submissions, the *YJ_STU* is ranked 13$^{\text{rd}}$, 12$^{\text{th}}$, and 11$^{\text{th}}$, respectively. The same stands for the runner-up (*Matthias*), which is ranked 1$^{\text{st}}$ at levels 2, 4, 5, and 6, but displays almost the worst performance out of the 50 methods examined at levels 10, 11, and 12, being ranked 48$^{\text{rd}}$, 49$^{\text{th}}$, and 48$^{\text{th}}$, respectively. *Daniela A*, ranked 28$^{\text{th}}$ in total, displays the best performance at level 1, *mf*, ranked 3$^{\text{rd}}$, displays the best performance at levels 10 and 11, while *wyzJack_STU*, ranked 6$^{\text{th}}$, displays the best performance at level 12, which contains the vast majority of the series requested to be forecast. We can therefore conclude that, depending on the aggregation level, different forecasting methods are more appropriate and, as the literature suggests, there are indeed "horses for courses" (Petropoulos et al., 2014). Thus, depending on the forecasting task and the nature of the data, different forecasting methods should be used to support decisions and optimize forecasting performance at different aggregation levels.

We also find that the accuracy of the top-performing methods deteriorates at a lower aggregation level, as uncertainty increases when forecasting more disaggregated data where sales are volatile and patterns like trend and seasonality are difficult to capture (Kourentzes et al., 2014b). This finding can be better visualized in Figure 2 which presents the distribution of WRMSSE for the top 50 performing teams per aggregation level along with the accuracy of ES_bu. As seen, although the top benchmark is outperformed by all teams at levels 1 to 9, the improvements reported for the rest of the levels are less significant, with some teams performing even worse than the benchmark. For example, the average improvement of the methods over the benchmark is 40% at level 1, which drops to about 23% at levels 5, 6, and 7, and reaches 3% at levels 10, 11, and 12. Therefore, we can conclude that the gains of the top-performing methods mainly refer to the top and middle parts of the hierarchies, and are rather limited in terms of WRMSSE at product, product-state, and product-store levels.

In order to further investigate the differences reported between the top 50 submissions, as well as the top-performing benchmark, we employ multiple comparisons with the best (MCB) test (Koning et al., 2005). The test computes the average ranks of the forecasting methods according to RMSSE across the

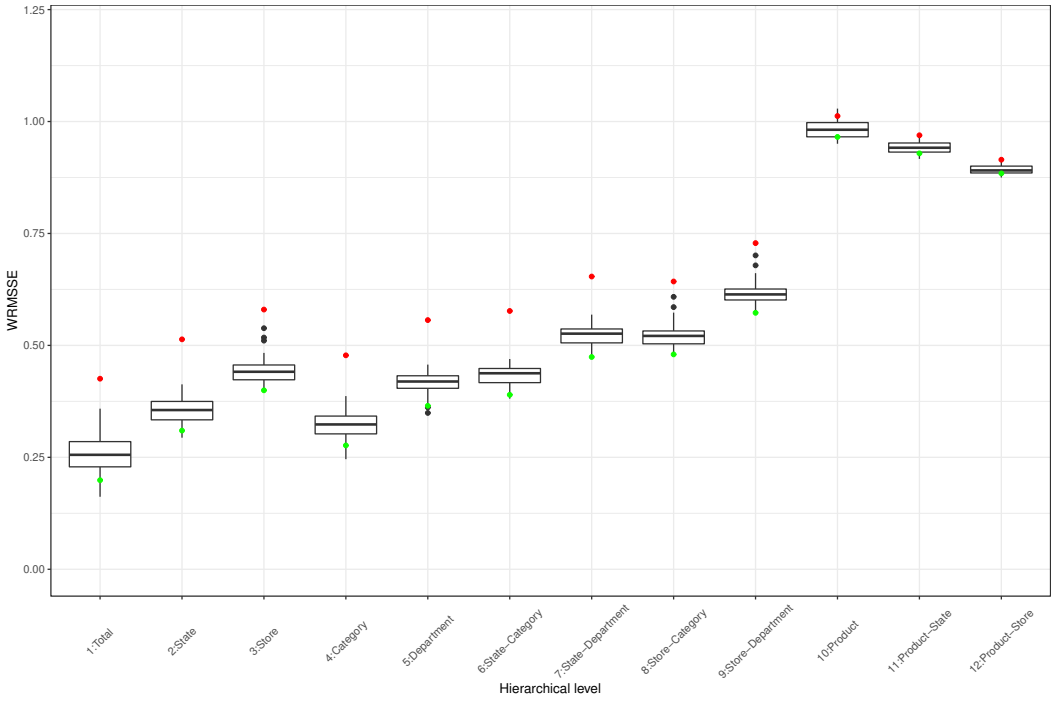

Figure 2: Forecasting accuracy (WRMSSE) of the top 50 performing teams of the M5 "Accuracy" competition. The results are reported per aggregation level and box-plots are used to display the distribution of the average errors recorded for the examined methods (minimum value, 1st quantile, median, 3rd quantile, maximum value, and outliers, noted with black dots). The red dots indicate the performance of the top-performing benchmark of the competition (ES_bu), while the green dots the performance of the winning team (*YJ_STU*).

complete data set of the competition and concludes whether or not these are statistically different. Figure 3 presents the results of the analysis. If the intervals of two methods do not overlap, this indicates a statistically different performance. Thus, methods that do not overlap with the gray interval of the figures are considered significantly worse than the best, and vice versa.

As seen, teams *SHJ* (ranked 8th), *DenisKokosinskiy_STU* (ranked 10th), and *XueWang* (ranked 11th) provide significantly better forecasts than the rest of the examined methods, and are more accurate for the majority of the series. Note also that apart from *mf* (ranked 3rd in total), none of the five winning teams performs equally as well as *SHJ*, while *wyzJack_STU*, which ranked 1st at level 12 according to WRMSSE, also displays a significantly worse performance. Based on this observation, we conclude that the winning teams developed methods that mostly focused on expensive and fast-moving products for which WRMSSE is minimized, thus providing less accurate results for the rest of the series which probably offer less value to the company. This highlights that the objective of the competition (minimizing the error measure across all aggregation levels and especially for high-valued series), expressed though the error measure used, was critical for determining the winning submissions and optimizing their parameters. Consequently, we find

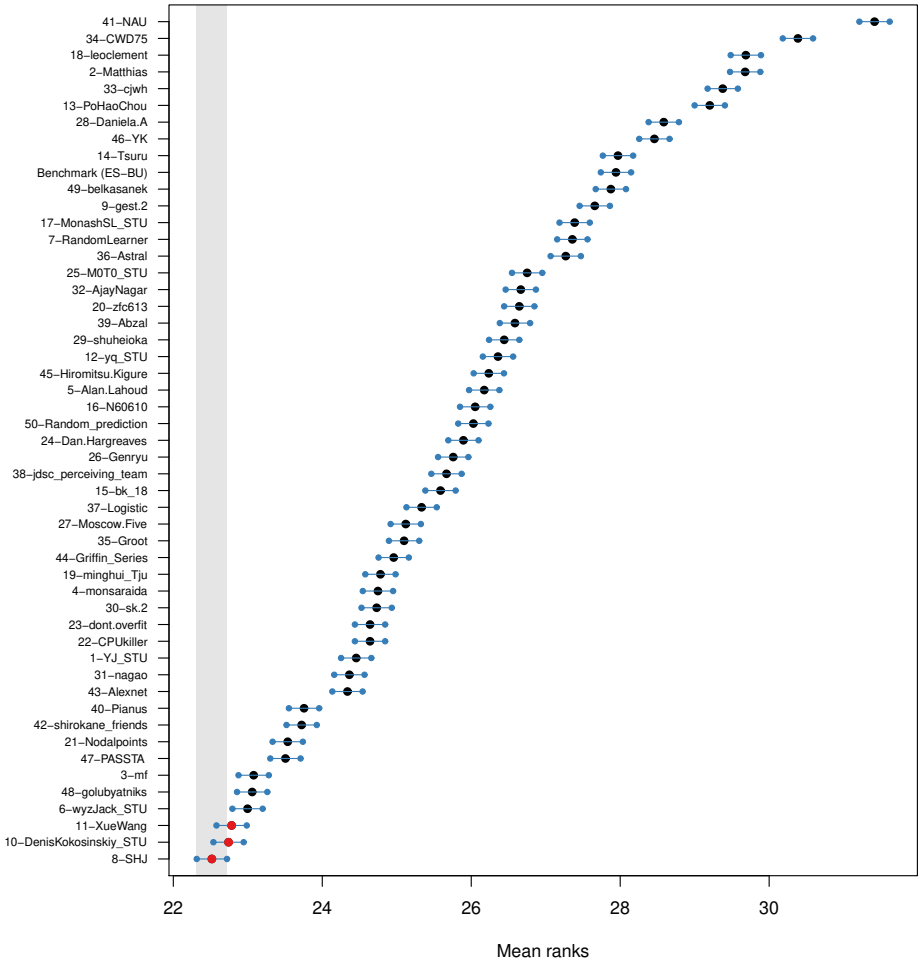

Figure 3: Average ranks and 95% confidence intervals of the top 50 performing teams of the M5 "Accuracy" competition, plus the top-performing benchmark (ES_bu) over all series: multiple comparisons with the best (RMSSE used for ranking the methods) as proposed by Koning et al. (2005). The overall rank of the teams in terms of WRMSSE is displayed to the left of their names.

that, in such weighting settings, the "best" forecasts depend on the accuracy measure used (Kolassa, 2020), especially when utilizing flexible ML methods whose loss function can be adjusted accordingly to optimize forecasts based on the selected measure.

Finally, we investigate the length of the forecasting horizon's impact on the accuracy achieved by the top 50 performing methods of the competition. To do so, we first compute the weighted root squared scaled error (WRSSE) of these methods for each forecasting horizon and series separately and then aggregate the results per aggregation level and horizon. A summary of the results is presented in Figure 4. As seen, although

in most of the cross-sectional levels the accuracy remains rather constant, and is even slightly reduced in some cases, this is not true for the lowest aggregation levels (10, 11, and 12) where the accuracy significantly deteriorates as the forecasting horizon increases. This finding is closely related to the characteristics displayed by the series of each level. At the higher levels, trend and seasonality dominate randomness that does not significantly affect forecasting accuracy, at least for the relatively short forecasting horizon of 28 days considered in the competition. On the other hand, at lower aggregation levels, intermittency, erraticness, and lack of trend and seasonality, increase randomness and negatively affect forecasting accuracy. Also, in many aggregation levels, and especially at the lowest ones, the errors display some sort of periodicity, e.g.,larger errors are observed during the weekends, indicating that part of the seasonality present in the data was not appropriately captured by the forecasting methods, even the top-ranked ones.

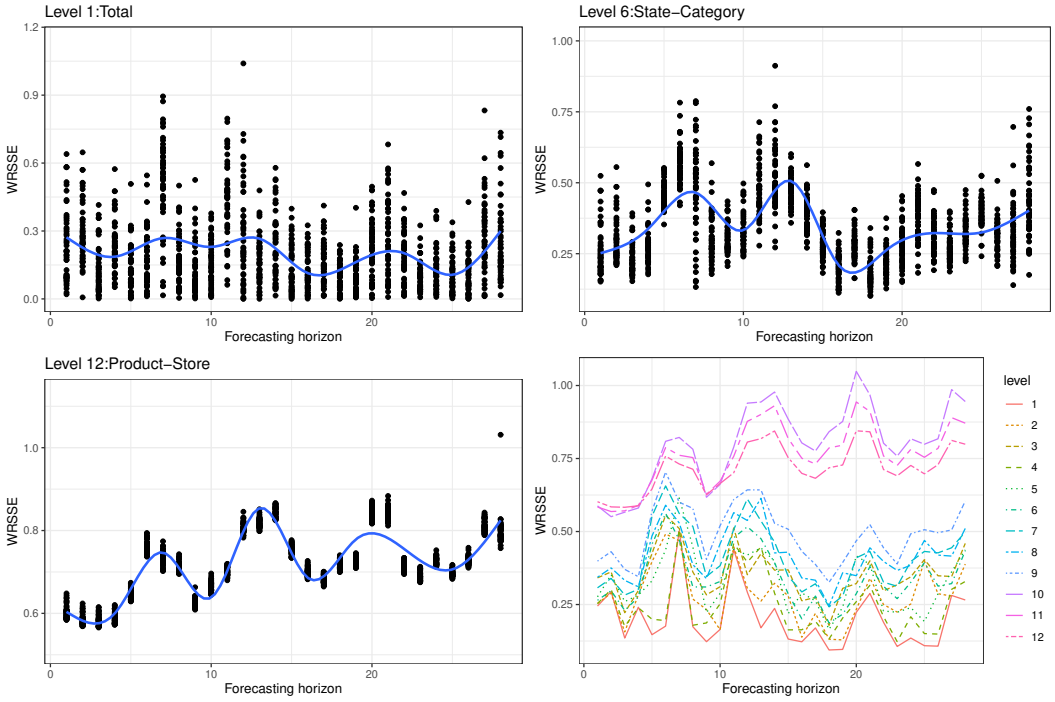

Figure 4: Forecasting horizon length's impact on forecasting accuracy. Top left: Forecasting accuracy (WRSSE) of the top 50 performing methods of the competition per forecasting horizon for the top level of the data set. The blue line represents LOESS (locally estimated scatter plot smoothing). Top right: Similar to the top left figure, but this time the results are reported for the middle aggregation level of the data set (state-category). Bottom left: Similar to the top left figure, but this time the results are reported for the lowest aggregation level of the data set (product-store); Bottom right: The average error of all top 50 performing methods across all 12 aggregation levels and forecasting horizons.

4.2. Winning submissions

Unfortunately, as previously mentioned, a very limited number of teams that participated in the M5 "Accuracy" competition were willing to share with the organizers and the Kaggle community the description of their methods, and even fewer to share their code. Although the organizers tried to reach at least the top 50 performing teams of the competition through e-mails (a template for describing the main features of the utilized methods was provided), such information was obtained for just 17 of them, either by receiving a direct reply, or by observing the public discussions and notebooks posted by these teams onto Kaggle. Nevertheless, we still believe that there are many lessons to be learned from these methods as they all provided significantly more accurate forecasts than the benchmarks considered and the thousands of other participating teams.

Before presenting the five winning methods, we should note that most of the methods examined utilized LightGBM[4], a ML algorithm for performing non-linear regression using gradient boosted trees (Ke et al., 2017). LightGBM displays several advantages over other ML alternatives in forecasting tasks, like those characterizing the M5 "Accuracy" competition, as it allows the effective handling of multiple features (e.g. past sales and exogenous/explanatory variables) of various types (numeric, binary, and categorical). In addition, it is fast to compute, compared to other gradient boosting (GBM) methods, does not depend on data pre-processing and transformations, and requires the optimization of only a relatively small number of parameters (e.g. learning rate, number of iterations, maximum number of bins, number of estimators, and loss functions). In this regard, LightGBM is very convenient to experiment with and develop solutions that can be accurately generalized for a large number of series that display cross-correlations. In fact, LightGBM can be considered as the standard method of choice in Kaggle's recent forecasting competitions, as the winners of the "Corporación Favorita Grocery Sales Forecasting" and "Recruit Restaurant Visitor Forecasting" competitions built their approaches using this method (Bojer & Meldgaard, 2020) and the discussion and notebooks posted on Kaggle for the M5 "Accuracy" competition focused on using LightGBM and variants of it.

The forecasting methods of the five winning teams can be summarized as follows:

- **First place (*YJ_STU*; YeonJun In):** The winner of the competition, a senior undergraduate student at Kyung Hee University, South Korea, considered an equal weighted combination (arithmetic mean) of various LightGBM models that were trained to produce forecasts for the product-store series using data per store (10 models), store-category (30 models), and store-department (70 models). Two variations were considered for each type of model, the first applying a recursive and the second a non-recursive forecasting approach (Bontempi et al., 2013). In this respect, a total of 220 models

[4]https://lightgbm.readthedocs.io/en/latest/index.html

were built and each series was forecast using the average of six models, each one exploiting a different learning approach and train set. The models were optimized without considering early stopping and by maximizing the negative log-likelihood of the Tweedie distribution (Zhou et al., 2020), which is considered an effective approach when dealing with data with a probability mass of zero and non-negative, highly right-skewed distribution. The method was fine-tuned using the last four 28-day-long windows of available data for CV and by measuring both the mean and the standard deviation of the errors produced by the individual models and their combinations. That way, the final solution was chosen so that it provided both accurate and robust forecasts. Regarding the features used, the models considered various identifiers, calendar-related information, special days, promotions, prices, and unit sales data, both in a recursive and a non-recursive format.

- **Second place (*Matthias*; Matthias Anderer):** This method was also based on an equally weighted combination of various LightGBM models, however, was externally adjusted through multipliers according to the forecasts produced by N-BEATS (deep-learning NN for time series forecasting; Oreshkin et al., 2019) for the top five aggregation levels of the data set. Essentially, LightGBM models were first trained per store (10 models) and then five different multipliers were used to adjust their forecasts and properly capture the trend. In this regard, a total of 50 models were built and each series of the product-store level of the data set was forecast using a combination of five different models. The loss function used was a custom, asymmetric one. The last four 28-day-long windows of available data were used for CV and model building. The LightGBM models were trained using only some basic features about calendar effects and prices (past unit sales were not considered), while the N-BEATS model was based solely on historical unit sales.

- **Third place (*mf*; Yunho Jeon & Sihyeon Seong):** This method involved an equally weighted combination of 43 deep-learning NNs (Salinas et al., 2020), each consisting of multiple LSTM layers that were used to recursively predict the product-store series. From the models trained, 24 considered dropout, while the remaining 19 did not. Note that these models originated from just 12 models and corresponded to the last, more accurate instances observed for these models while training, as specified through CV (last fourteen 28-day-long windows of available data). Similar to the winner, the method considered Tweedie regression, but was modified however to optimize weights based on sampled predictions instead of actual values. The Adam optimizer and the cosine annealing was used for the learning rate schedule. The NNs considered a total of 100 features of similar nature to those of the winning submission (sales data, calendar-related information, prices, promotions, special days, identifiers, and zero-sales periods).

- **Fourth place (*monsaraida*; Masanori Miyahara):** This method produced forecasts for the product-store series of the data set using non-recursive LightGBM models, trained per store (10 mod-

els). However, in contrast to the rest of the methods, each week of the forecasting horizon was forecast separately using a different model (four models per store). Thus, a total of 40 models were built to produce the forecasts. The features used as inputs were similar to those of the winning submission, with the exception of the recursive ones. Tweedie regression was considered for training the models, with no early stopping, and no optimization was performed in terms of training parameters. The last five 28-day-long windows of available data were used for CV.

- **Fifth place (*Alan Lahoud*; Alan Lahoud):** This method considered recursive LightGBM models, trained per department (seven models). After producing the forecasts for the product-store series, these were externally adjusted so that that the mean of each of the series at the store-department level was the same as the one of the previous 28 days. This was done using appropriate multipliers. The models were trained using Poisson regression with early stopping and validated using a random sample of 500 days. The features used as input were similar to those of the winning submission.

Regarding the rest of the top 50 performing methods for which a method description was available, we should mention that almost all of them adopted similar approaches to the winning submission, training recursive and non-recursive LightGBM models per store, department, or store-department. The main exceptions were: *N60610*, ranked 16$^{\text{th}}$, who predicted the product-store series of the data set using both LightGBM and a Kalman filter and selected the most appropriate approach per series, *MonashSL_STU*, ranked 17$^{\text{th}}$, who used an equal-weighted combination of LightGBM and a Pooled Regression Model, *Nodalpoints*, ranked 21$^{\text{st}}$, who employed a weighted combination of LightGBM and NNs trained across all series or per store, and Astral, ranked 36$^{\text{th}}$, who considered a non-recursive Prophet-like model that mixes classical statistics practices with non-linear optimization ML techniques, namely XGBoost and LightGBM.

4.3. Key findings

Below is a summary of the findings related to the performance of the top five methods:

Finding 1: The superiority of relatively simple ML methods. For many years it has been empirically found that simple methods are as accurate as complex or statistically sophisticated ones (Makridakis et al., 2020d). Limited data availability, inefficiency of algorithms, the need for preprocessing, and restricted computational power, were just some of the factors that deteriorated the accuracy of ML methods in comparison to statistical ones (Makridakis et al., 2018b). M4 was the first forecasting competition identifying two ML based approaches significantly more accurate than simple, statistical ones, highlighting the potential value of ML methods to forecast more accurately (Makridakis et al., 2020d). The first method that won the M4 competition was a hybrid approach that mixed recurrent NNs and exponential smoothing (Smyl, 2020), while the second, ranked 2$^{\text{nd}}$, was a method that used XGBoost to optimally weight the forecasts

produced by standard time series forecasts (Montero-Manso et al., 2020). Although both of the M4 winning submissions were ML in nature, they both built on statistical, series-specific functionalities, while their accuracy was also close to a simple combination of the median of four statistical methods (Petropoulos & Svetunkov, 2020). M5 is, therefore, the first competition where all top-performing methods were both "pure" ML ones and significantly better than all statistical benchmarks and their combinations. LightGBM proved that it can be used effectively to process numerous, correlated series and exogenous/explanatory variables and reduce forecast errors. Moreover, deep learning methods like DeepAR and N-BEATS, using advanced, state-of-the-art ML implementations, have shown forecasting potential, motivating further research in this direction.

Finding 2: The value of combining. The M5 "Accuracy" competition confirmed the findings of the previous four M competitions and those of numerous other studies, suggesting that combining forecasts of different methods, even relatively simple ones (Petropoulos & Svetunkov, 2020), results in improved accuracy. The winner of the M5 "Accuracy" competition employed a very simple, equal-weighted combination, involving six models, each one exploiting a different learning approach and train set. Similarly, the runner-up utilized an equal-weighted combination of five models, each one of a different estimate for trend, while the third best-performing method, an equal-weighted combination of 43 NNs. Simple combinations of models were also reported for the methods ranked 14^{th}, 17^{th}, 21^{st}, 24^{th}, 25^{th}, and 44^{th}. Of these combination approaches, only the one ranked 25^{th} considered unequally weighting the individual methods. The value of combining is also supported by the comparisons made between the benchmarks of the competition. As shown in the Appendix of the supplementary material, the combination of exponential smoothing and ARIMA models performed better than the individual methods, while the combination of a top-down and bottom-up reconciliation method outperformed both top-down and bottom-up.

Finding 3: The value of "cross-learning": In the previous M competitions, most of the series were uncorrelated, of a different frequency and domain, and chronologically unaligned. Therefore, although both of the top-performing submissions of M4 utilized "cross-learning" from multiple series concurrently, instead of one series at a time, their approach was difficult to implement effectively in practice, and did not demonstrate the full potential of "cross-learning". In contrast, since the M5 consisted of aligned, highly-correlated series structured in a hierarchical fashion, "cross-learning" was made much easier to apply, achieving superior results when compared to methods that were trained in a series-by-series fashion. Note that, apart from resulting in more accurate forecasts, "cross-learning" implies the use of a single model instead of multiple ones, each trained on the data of a different series, thus reducing overall computational cost and mitigating difficulties related to limited historical observations (Semenoglou et al., 2020). Essentially, all top 50 performing methods in M5 utilized "cross-learning", exploiting all the information being offered by the data set.

Finding 4: The significant differences between the winning methods and benchmarks used for sales forecasting. As noted, the M5 "Accuracy" competition considered 24 benchmarks of various types that are typically used in sales forecasting applications, including traditional and state-of-the-art statistical methods, ML methods, and combinations. As shown in Figure 3 and Table 2, the winning submissions provided significantly more accurate forecasts in terms of ranks when compared to these benchmarks and were also, on average, more than 20% better in terms of WRMSSE. Although the differences were smaller at lower aggregation levels, the results clearly demonstrate their superiority and motivates additional research in the area of ML forecasting methods that can be used to predict complex, non-linear relationships between the series, as well as to include exogenous/explanatory variables.

Finding 5: The beneficial effect of external adjustments. Forecast adjustments are typically used when forecasters exploit external information, as well as inside knowledge and their expertise to improve forecasting accuracy (Davydenko & Fildes, 2013). Such adjustments were applied in the M2 competition where it was found that they did not improve the accuracy of pure statistical methods (Makridakis et al., 1993). In the M5 "Accuracy" competition, some of the top-performing methods, namely the ones ranked 2^{nd} and 5^{th}, utilized such adjustments in the form of multipliers to enhance the forecasts derived by the ML models. Although they were not completely based on judgment but rather on the analytical alignment of the forecasts produced at the lowest aggregation levels with those at the higher ones, these adjustments proved to be beneficial, helping the models to reduce bias and better account for the longer-term trends that are easier to observe at higher aggregation levels (Kourentzes et al., 2014b). Even though the actual value of such adjustments requires further investigation, the concept of reconciling forecasts produced at different aggregation levels is not new in the field of forecasting, with numerous studies empirically proving its benefits, especially when forecasts and information from the complete hierarchy are exploited (Hyndman et al., 2011; Spiliotis et al., 2020c).

Finding 6: The value added by effective CV strategies. When dealing with complex forecasting tasks, adopting effective CV strategies is critical for objectively capturing post-sample accuracy, avoiding overfitting, and mitigating uncertainty. The importance of adopting such strategies is demonstrated by the results of the M5 "Accuracy" competition, indicating that a significant number of teams failed to select the most accurate set of forecasts from those submitted while the competition was still running (see section 3). Yet, various CV strategies can be adopted and, based on their design, different conclusions can be drawn. Selecting the time period during which the CV will be performed, the size of the validation windows, the way these windows will be updated, and the criteria that will be used to summarize forecasting performance, are just some of the factors that forecasters have to consider. In the M5 "Accuracy" competition, the top

four performing methods and the vast majority of the top 50 submissions considered a CV strategy where at least the last four 28-day-long windows of available data were used to assess forecasting performance, thus providing a reasonable approximation of post-sample accuracy. What the winner did in addition to this CV scheme, was that he measured both the mean and the standard deviation of the models he had developed. According to his validations, the recursive models of his approach were found to be, on average, more accurate than the non-recursive ones but of greater instability. As such, he decided to combine those two models to make sure that the forecasts produced would be both accurate and stable. Spiliotis et al. (2019a) stressed the necessity of accounting for the full distributions of forecasting errors and especially their tails when evaluating forecasting methods, indicating that robustness is a prerequisite for achieving high accuracy. It is our hope that the results of M5 will encourage more research in this area and contribute to the development of more powerful CV strategies.

Finding 7: The importance of exogenous/explanatory variables. Time series methods are usually sufficient for identifying and capturing historical data patterns (level, trend, and seasonality) and produce accurate forecasts by extrapolating them. However, methods that solely rely on identifying and extrapolating historical data fail to effectively account for the effect of holidays, special days, promotions, prices, and possibly the weather. Moreover, given that such factors can affect the historical data, they can distort the time series pattern unless removed before the data is used to forecast. In such settings, the information from exogenous/explanatory variables becomes of critical importance to improve forecasting accuracy (Ma et al., 2016). In the M5 "Accuracy" competition all winning submissions utilized external information to improve the forecasting performance of their models. For example, *monsaraida* and other top teams found that several price-related features were of significant importance for improving the accuracy of their results. Furthermore, the importance of exogenous/explanatory variables is also supported by the comparisons made between the benchmarks of the competition, as shown in the Appendix of the supplementary material. For instance, ESX, which used information about promotions and special days as exogenous variables within exponential smoothing models, was 6% better than ES_td which employed the same exponential smoothing models but without considering exogenous variables. The same was true in the case of the ARIMA models, where ARIMAX was found to be 13% more accurate than ARIMA_td.

5. Discussion, limitations, advantages, and directions for future research

5.1. Discussion

What became clear from the M5 "Accuracy" competition, is that ML methods have entered the mainstream of forecasting applications, at least in the area of retail sales forecasting. The potential benefits are

substantial and there is little doubt that retail firms will need to adopt them to improve the accuracy of their forecasts and better support decisions related to their operations and supply chain management.

Table 3 provides a simple comparison of the Croston's method (CRO), widely used for forecasting intermittent demand data, with the sNaive, SES, ES_bu, ES_td, and ESX benchmarks (for more information about the benchmarks, please see the Appendix of the supplementary material). As seen, sNaive (a naive method that accounts for seasonality) is on average 11.5% more accurate than CRO, but at the same time its improvements are extremely uneven across various cross-sectional levels. These improvements start at 37.8% at the highest aggregation level, drop to 9.7% at level 9, and become negative at levels 10, 11, and 12, with CRO being more accurate than sNaive by 13.0%, 20.2%, and 27.0%, respectively. These results prove the value of CRO in forecasting intermittent demand data, while also highlighting its limitations when it comes to continuous series characterized by seasonality and trend. A similar comparison of CRO with SES (a simple exponential smoothing method that does not account for seasonality) further proves the value added by CRO as, in this case, SES is on average 1.3% less accurate than CRO, providing slightly better forecasts only at level 10. Finally, by comparing the accuracy of CRO over the three top-performing exponential smoothing benchmarks of the competition (ES_bu, ES_td, and ESX), all capable of accounting for seasonality, we observe an average improvement of about 28%, starting at 54% at the top level and dropping to 1.1% at the lowest, product-store level. Note that the improvements are consistent for the three exponential smoothing methods across all aggregation levels.

Table 3: Percentage improvements (according to WRMSSE) reported between indicative benchmarks of the competition, namely CRO, sNaive, SES, ES_bu, ES_td, and ESX.

Methods	Aggregation level												Average
compared	1	2	3	4	5	6	7	8	9	10	11	12	
CRO vs. sNaive	37.8%	26.4%	22.2%	31.5%	27.0%	19.2%	16.0%	14.8%	9.7%	-13.0%	-20.2%	-27.0%	11.5%
CRO vs. SES	-2.3%	-2.6%	-2.3%	-2.0%	-1.3%	-2.0%	-1.5%	-1.7%	-1.1%	1.1%	0.0%	-0.6%	-1.3%
CRO vs. ES_bu	52.7%	43.8%	37.1%	47.4%	42.7%	38.7%	33.8%	31.6%	25.9%	6.5%	3.2%	1.2%	29.9%
CRO vs. ES_td	47.8%	39.9%	28.0%	41.7%	34.1%	33.1%	26.4%	23.7%	18.5%	5.0%	2.8%	1.2%	24.7%
CRO vs. ESX	61.1%	46.0%	32.0%	51.8%	41.5%	37.3%	29.9%	26.4%	20.7%	5.2%	2.7%	1.0%	29.0%
ES_td vs. ESX	25.5%	10.1%	5.6%	17.3%	11.3%	6.2%	4.7%	3.4%	2.7%	0.2%	-0.1%	-0.2%	5.7%

It becomes evident that the majority of the improvements reported between CRO and the three top-performing exponential smoothing benchmarks came from the ability of the latter to adequately capture seasonality, as well as their capacity to exploit explanatory/exogenous variables. In order to separate the effect of these two influencing factors, we compare ES_td with ESX, as both of these methods employ the same exponential smoothing models, but the latter also considers some indicative explanatory/exogenous variables. As seen, the average improvement of ESX over ES_td is 5.7%, starting with 25.5% at the top level, dropping to 2.7% at level 9, and turning negative at levels 10, 11, and 12. Thus, we find that, although

external information can improve forecasting accuracy, seasonality, observed mainly at higher aggregation levels, is the most critical factor for improving overall forecasting performance.

Table 4: Percentage improvements (according to WRMSSE) reported between the winning submission (YJ_STU) and the Croston's method.

Methods	Aggregation level												Average
compared	1	2	3	4	5	6	7	8	9	10	11	12	
Winning team	0.199	0.310	0.400	0.277	0.365	0.390	0.474	0.480	0.573	0.966	0.929	0.884	0.520
CRO	0.900	0.915	0.923	0.909	0.971	0.941	0.987	0.940	0.983	1.083	1.002	0.926	0.957
Improvement	77.9%	66.1%	56.7%	69.6%	62.4%	58.6%	52.0%	49.0%	41.7%	10.8%	7.3%	4.5%	45.6%

The improvements become much more substantial when CRO is compared to the top-performing method of the competition. The improvements start at 77.9% at the top level, dropping to 10.8%, 7.3%, and 4.7% at levels 10, 11, and 12, respectively. However, the average improvement overall is 45.6% with superior values up to the eighth level. With such numbers the superiority of the winning method is not likely to be disputed, particularly up to level 8, at least until new, more accurate ML methods for handling similar forecasting tasks are developed. At the same time, CRO and other standards used for forecasting intermittent demand, like SBA, TSB, ADIDA, and iMAPA (for more details about these methods, please see the Appendix of the supplementary material), seem to hold some value for low cross-sectional levels, especially at the product-store level.

Tables 3 and 4 provide useful information. The ML method employed by the winning team of the competition that was based on the LightGBM algorithm, reported substantial improvements over the Croston's method, with significantly better forecasts at the higher aggregation levels but also accurate ones at the product-store level, which is the hardest one to predict due to its higher randomness. A notable part of these improvements is the proper modeling of seasonality and the exploitation of useful explanatory/exogenous variables. The rest of the improvements can probably be attributed to the "cross-learning" approach adopted, as well as to the non-linear nature of the models exploited by the winning team. Two interesting questions that need to be answered are whether LightGBM is truly the most suitable ML method for applying "cross-learning" in such forecasting applications, and how ML methods could possibly be adjusted so that they provide significantly better forecasts than the existing approaches at both high and low aggregation levels.

The biggest advantage of the ML methods used by all top 50 performing methods, was probably their versatility in accurately predicting all 30,490 product-store series of the competition concurrently using "cross-learning", and their flexibility to be fine-tuned based on the idiosyncrasies of the data set. At the same time, few teams tried to identify and use a single, "best" model to predict the series. Instead, there were many alternative models built that were subsequently averaged to forecast, with the winner developing

220 different models and using six models to predict each series. Furthermore, the approaches used by the top-performing teams to determine the most accurate forecasting method were unstructured, data-driven ones that were based on CV strategies, thus requiring little knowledge about forecasting and statistics. This is in contrast to the structured approaches widely used before the M5 competition, which focused on identifying a statistical method or a combination of statistical methods that could provide the most accurate forecasts for each series, as well as the hybrid approaches employed by the winning submissions of M4, which mixed elements of both statistical and ML methods. The unstructured, agnostic approaches used in the M5 required less experience about modeling and even less knowledge about the forecasting application considered, mostly depending on experimentation and data mining, without the need to understand the data itself and its characteristics. This can be asserted by the fact that the winning method was developed by a student with little forecasting knowledge and little experience in building sales forecasting models. Yet, he effectively managed to win the competition and outperform thousands of competitors, including experienced Kaggle grandmasters among others. As ML and computer science in general gain more acceptance in the field of forecasting, it would not be surprising to see the value of knowledge and experience become less important in developing and using forecasting models.

In the M4 competition, the five methods that achieved the most accurate results were also among the top five in terms of average ranks, as determined by the MCB test (Makridakis et al., 2020d), indicating a high degree of correspondence between these two evaluation measures; i.e. the methods that provided the most accurate forecasts on average were also the ones that most of the times provided the most accurate forecasts separately for each series. However, this has not been the case with the M5 "Accuracy" competition. As can be seen in Figure 3, the winning method was ranked 13^{th} according to the MCB test, the runner-up was ranked 47^{th}, while the third, fourth, and fifth were ranked 6^{th}, 17^{th}, and 29^{th}, respectively. It seems that the top five winning methods of M5 achieved their objective by minimizing the overall WRMSSE, weighting the more expensive and fast-moving products more heavily to achieve their single objective, rather than trying to provide accurate forecasts for every single series of the competition. As noted earlier, it remains to be seen whether these methods can be effectively adjusted to accurately predict all the series of the competition equally, especially the ones at the most disaggregated level.

Regarding the applicability of the competition's results, it would be interesting to see how long it will take until LightGBM and other ML methods are accepted by academics and widely utilized in practice by retail sales firms. In the academic world, exploring and adopting new methods does not usually take long as information is disseminated fast through journals and conferences. Of course, the results of relevant studies will have to be replicated by other researchers and unless they disagree with those of the M5, they will hopefully be accepted with little delay. However, in the business world things move slower. First, it will take some time until practitioners learn the results of the M5 "Accuracy" competition, who will then need to be persuaded of their superior value. Second, a software program will have to be developed, either in-

house or by some consulting firm, to implement the competition's winning methods or their variants. Third, it is necessary that the software will not require any special fine-tuning to produce forecasts of the same accuracy to that reported in the competition. Fourth, the computational cost should not be prohibitively expensive so that hundreds of thousands or even millions of forecasts can be produced on a weekly basis (Seaman, 2018), and finally, the software should be easy to integrate with the rest of the firms' ERP systems in order to retrieve the raw data and provide the corresponding forecasts (Petropoulos, 2015). In case these requirements are impossible to meet, then the most accurate benchmarks of the competition, which are relatively simple to implement and computationally cheap, would be utilized instead.

5.2. Limitations

The M5 "Accuracy" competition, as well as other empirical studies conducted in the past, provide valuable information about the accuracy of various forecasting approaches to guide academic research and offer precious advice to practitioners on what methods to use to improve their forecasting performance and make better decisions. The value of such information, however, strongly depends on the extent to which the data used for conducting the empirical comparison is representative of reality. It is hard to argue otherwise, when 100,000 times series that cover most data frequencies and various domains are utilized (Spiliotis et al., 2020a), but it is still possible even for such large data sets to differ on average to those used in particular forecasting applications, for instance involving high-frequency data or cross-correlated series. At the same time, when certain data sets cover a specific aspect of reality, like daily Wikipedia page visits or daily retail sales, their findings are likely more representative of the application examined but cannot be generalized beyond the specific area covered by the data, except to draw some general conclusions about the methods used or how they were selected or evaluated. Therefore, there is a fundamental difference between the data of M4, covering six different data frequencies and six different domains, and those of M5, referring to retail sales data, structured across 12 cross-sectional levels. Although the M5 data set refers to such a specific forecasting application, we still believe that it will be of great interest to a large number of retail firms that are specifically concerned with how to best forecast their daily sales and determine their inventory levels accordingly (Seaman, 2018). This has also been the case for past Kaggle competitions that involved daily and weekly product sales of large retail firms (Bojer & Meldgaard, 2020).

Another limitation of the M5 "Accuracy" competition, is that it focused on the point forecast accuracy of the submitted methods, which were not directly linked to Walmart's underlying operational costs. Although it has been empirically found that minor improvements in accuracy can lead to substantial reductions in stock holding and higher service levels (Syntetos et al., 2010; Ghobbar & Friend, 2003; Pooya et al., 2019), making such a connection and translating forecasting error reduction into cost savings is far from trivial. This is because, depending on the supply chain of each company, its facilities, its holding costs, and replenishment policies, different savings can be assumed for the same gains in accuracy. Moreover, many assumptions must

be made, particularly about the backlog and lost sales costs. Unfortunately, this detailed information was not made available to the M5 participants, making it impossible for the organizers to evaluate the implications of the accuracy improvements reported by the winning submissions into monetary terms. However, we hope that the results of the competition will inspire such studies and motivate relevant research in the field.

As noted, the train and test data of the competition was made publicly available at the end of the competition, ensuring openness and objectivity by allowing anyone who wants to replicate the results of the competition, test alternative forecasting methods, and propose new, more accurate ones, to do so. This type of openness and objectivity is not possible in Kaggle competitions where the test data is not made available after their competition has ended, while the participants are not required to reveal the methods developed to base the forecasts or share their code for others to use (Bojer & Meldgaard, 2020). Contrary to the first four M competitions, this has also been a serious problem with the M5 "Accuracy" competition as only 17 of the 50 top performing methods shared information about their forecasting approaches, even after the organizers had sent several emails requesting this information. On the positive side, the competition's rules stated that in order for the winners to receive their prizes, they had to reveal the method they used and make their code available in order for the organizers to be able to reproduce their results and compare the accuracy achieved to that of the originally submitted forecasts. This has been done and the forecasts have been successfully reproduced, allowing us to provide the detailed description for the five top-performing methods presented in subsection 4.2 and ensure that the respective code for those participants that do not belong to a business firm will be uploaded onto GitHub, from where it can be downloaded and used for free. In addition, just as was done with the M4 competition, a special issue of the *IJF*, exclusively devoted to the M5 competition, will be published presenting and discussing all its findings in detail while also including discussion papers about all aspects of the competition.

5.3. Directions for future research

The findings of the M4 and M5 "Accuracy" competitions, as well as those of the latest two Kaggle sales forecasting competitions (Bojer & Meldgaard, 2020), indicate that ML methods are becoming more accurate than statistical ones, and therefore require a reassessment of their theoretical value and potential usage by organizations. On the academic side, more research is needed to verify that the results of greater accuracy apply to other areas beyond hierarchical, retail sales forecasting and that other ML methods are not more accurate than the winning methods of these competitions.

On the practical side there is a need to determine the extra cost of running such ML methods, versus the standard, statistical ones, and whether their accuracy improvements would justify such higher cost (Nikolopoulos & Petropoulos, 2018; Fry & Brundage, 2020). If both concerns could be satisfied, there would be two additional that would require further investigation. The first relates to understanding how ML methods create their forecasts and account for factors like price, promotions, and special days. Managers

are typically unwilling to make decisions when they cannot understand the logic of the methods they are going to use. This is a big problem affecting all ML models that would eventually need to be solved. Until then, some interim solution must be found by comparing the forecasts of ML methods to those of known benchmarks, as shown in Tables 3 and 4, that would allow them to indirectly determine the contribution of each factor. The second concern relates to which and how many ML models will have to be combined to achieve such improvements in accuracy. Perhaps, instead of developing ensembles of hundreds of models, it could be the case that eliminating the worst ones, or those less likely to improve forecasting performance, could improve overall accuracy without exaggerating in terms of computational cost.

Another alternative to be further explored is the concept of "horses for courses" (Petropoulos et al., 2014), i.e. the idea that different methods could potentially be used to forecast the various aggregation levels separately based on their corresponding performances. If *Daniela*'s (ranked 28$^{\text{th}}$) method was used to forecast the top level, followed by *Matthias* (ranked 2$^{\text{nd}}$) for levels 2 to 6, *YJ_STU* for levels 7 to 9 (ranked 1$^{\text{st}}$), *mf* for levels 10 and 11 (ranked 3$^{\text{rd}}$) and *wyzJack_STU* for level 12 (ranked 6$^{\text{th}}$), the overall accuracy of the winning submission would have been improved by an additional 2.3%. Such a selective approach, however, would have required that the best-performing method at each level be effectively identified beforehand. Moreover, it would have also required the forecasts produced by these methods to be reconciled so that they become coherent across the various aggregation levels. This task proved much more challenging than expected in the M5 "Accuracy" competition, with many teams trying to apply well-known reconciliation methods (Hyndman et al., 2011) but failing to do so, probably due to the size of the data set, the complexity of the underlying hierarchy, and other present limitations (non-negative forecasts and additional computational cost). These insights indicate that there is a lot of potential in this particular area of forecasting and that developing new hierarchical forecasting methods, capable of reconciling forecasts so that the forecast error is minimized, not only on average but separately at each aggregation level, could provide substantial accuracy improvements. The potential of such methods is highlighted if we consider that the simple, equal weighted combination of the above-mentioned five submissions, which directly provides coherent forecasts, results on average in 2% more accurate forecasts than the winning submission, which, however, are not always the best ones for each individual level, as shown in Table 5.

6. Conclusions

It has been almost 40 years since the first M forecasting competition took place, which was the first of its kind in a scientific field still in its infancy (Makridakis et al., 1982). At that time, there were just seven contestants pitting their methods against each other and predicting up to 1,001 time series to determine the most accurate one that, contrary to expectations, was a simple exponential smoothing method rather than the statistically sophisticated Box-Jenkins methodology to ARIMA models, the "king" of that era.

Table 5: Percentage improvements (according to WRMSSE) reported between the winning submission (*YJ_STU*), the best-performing submission of each aggregation level, i.e. *Daniela* for level 1, *Matthias* for levels 2 to 6, *YJ_STU* for levels 7 to 9, *mf* for levels 10 and 11, and *wyzJack_STU* for level 12, and the simple, equal weighted combination of these five methods.

Methods compared	Aggregation level												Average
	1	2	3	4	5	6	7	8	9	10	11	12	
Winning team	0.199	0.310	0.400	0.277	0.365	0.390	0.474	0.480	0.573	0.966	0.929	0.884	0.520
Best team per level	0.162	0.294	0.400	0.246	0.349	0.381	0.474	0.480	0.573	0.950	0.917	0.875	0.508
Combination (COMB)	0.181	0.283	0.390	0.260	0.365	0.371	0.473	0.472	0.572	0.960	0.921	0.876	0.510
Improvement of COMB over winner	8.9%	8.5%	2.5%	6.1%	0.1%	4.7%	0.3%	1.7%	0.1%	0.6%	0.9%	0.9%	2.0%
Improvement of COMB over the best team per level	-11.9%	3.5%	2.5%	-5.7%	-4.5%	2.6%	0.3%	1.7%	0.1%	-1.0%	-0.5%	-0.1%	-0.4%

In addition, the competition established the value of combining, proving empirically that combining the forecasts of more than one method improved accuracy and reduced uncertainty. That was an important finding at a time when it was advocated that a single, appropriate model existed for each time series and that such a model had to be identified judgmentally by inspecting the characteristics of the series. The M2 (Makridakis et al., 1993) was also a small-scale competition with five participants that took place in real-time between 1987 and 1989, so that the contestants could incorporate their judgment by adjusting the statistical forecasts using inside company information and knowledge about the current economy. The competition discovered that, as opposed to expectations, human judgment did not improve the accuracy of the statistical forecasts and combining was the most accurate way to predict the 29 series of the competition. In the M3 (Makridakis & Hibon, 2000), run in 2000, the number of time series increased substantially to 3,003 and the number of participants grew to fifteen, covering both simple and statistically sophisticated methods, as well as rule-based and NN ones. Still, simple methods outperformed the relatively more complex ones, with a new simple method, called Theta (Assimakopoulos & Nikolopoulos, 2000), being the most accurate of all others on average, and forecast combinations continuing to produce more accurate results than the individual methods being combined.

The M4 (Makridakis et al., 2020d), conducted in 2018, witnessed a dramatic increase to 100,000 time series and 49 participants and, in addition to the accuracy competition, it also included the requirement to estimate uncertainty by asking participants to provide the 95% prediction interval around their point forecasts for each of the 100,000 series. As was mentioned, the M4 ended a long forecasting winter by reversing the findings of the previous three competitions and concluding that two sophisticated methods, using a mixture of statistical and ML concepts, outperformed all others, both in terms of accuracy and uncertainty. The forecasting spring continued with the M5 that proved the superiority of ML methods, particularly the LightGBM, with the 50 top-performing ones achieving a superior performance of more than 14% over the most accurate statistical benchmark and the top five more than 20%. What has remained constant in all five M competitions is the finding that combining improves forecasting accuracy. What has changed is the finding of M1, M2, and M3 that simple statistical methods were more accurate than more

complex, sophisticated ones. In M4, only two sophisticated methods were found to be more accurate than simple, statistical ones, with the latter dominating the top positions of the competition. On the contrary, in the M5 all 50 top-performing methods were ML. Therefore, M5 is the first M competition where all top-performing methods were both ML ones and significantly better than all statistical benchmarks and their combinations. LightGBM proved that it can be used to effectively process numerous, correlated series and exogenous/explanatory variables and reduce forecast error. Moreover, deep learning methods, like DeepAR and N-BEATS, that provide advanced, state of the-art ML implementations, showed potential for further improving forecasting accuracy in hierarchical retail sales applications.

In summary, the M5 "Accuracy" competition provided the forecasting community with the following three, new, important findings:

- The superior accuracy of the LightGBM method for predicting hierarchical retail sales that resulted in substantial improvements over the benchmarks considered.
- The benefits of external adjustments that some methods utilized to improve the forecasting accuracy of the baseline forecasting models.
- The importance of exogenous/explanatory variables to improve the forecasting accuracy of time series methods.

In addition, M5 reaffirms the value of the following three findings of the previous M competitions in improving forecasting accuracy:

- Combining
- "Cross-learning"
- Cross-validation

The exceptional performance of statistical methods versus ML ones found by Makridakis et al. (2018b), as well as in the early Kaggle competitions (Bojer & Meldgaard, 2020), first shifted towards both ML and statistical methods in the M4 competition, and then to exclusively ML methods as in the Kaggle competitions after 2018 and the M5 described in this paper. It will be of great interest to see if ML methods continue to dominate statistical ones in the future, particularly for other types of data that are not exclusively related to hierarchical, retail sales applications.

Finally, what is important for the forecasting area is the integration of statistics and data science into a unique field covering all academic aspects of forecasting and uncertainty, while determining how to increase the Usage of Forecasting in Organizations (UFO), by persuading executives of the benefits of systematic forecasting for improving their bottom line (Makridakis et al., 2020a).

References

Abouarghoub, W., Nomikos, N. K., & Petropoulos, F. (2018). On reconciling macro and micro energy transport forecasts for strategic decision making in the tanker industry. *Transportation Research Part E: Logistics and Transportation Review*, *113*, 225–238. Making connections: Supply chain innovation research collaboration;.

Assimakopoulos, V., & Nikolopoulos, K. (2000). The theta model: a decomposition approach to forecasting. *International Journal of Forecasting*, *16*, 521–530.

Bergmeir, C., & Benítez, J. M. (2012). Neural networks in R using the stuttgart neural network simulator: RSNNS. *Journal of Statistical Software*, *46*, 1–26.

Bojer, C. S., & Meldgaard, J. P. (2020). Kaggle forecasting competitions: An overlooked learning opportunity. *International Journal of Forecasting*, (pp. 1–17).

Bontempi, G., Ben Taieb, S., & Le Borgne, Y.-A. (2013). Machine learning strategies for time series forecasting. In M.-A. Aufaure, & E. Zimányi (Eds.), *Business Intelligence: Second European Summer School, eBISS 2012, Brussels, Belgium, July 15-21, 2012, Tutorial Lectures* (pp. 62–77). Berlin, Heidelberg: Springer Berlin Heidelberg.

Breiman, L. (2001). Random forests. *Machine Learning*, *45*, 5–32.

Croston, J. D. (1972). Forecasting and Stock Control for Intermittent Demands. *Journal of the Operational Research Society*, *23*, 289–303.

Davydenko, A., & Fildes, R. (2013). Measuring forecasting accuracy: The case of judgmental adjustments to sku-level demand forecasts. *International Journal of Forecasting*, *29*, 510–522.

Fry, C., & Brundage, M. (2020). The M4 forecasting competition – A practitioner's view. *International Journal of Forecasting*, *36*, 156–160.

Gardner Jr., E. S. (1985). Exponential smoothing: The state of the art. *Journal of Forecasting*, *4*, 1–28.

Ghobbar, A. A., & Friend, C. H. (2003). Evaluation of forecasting methods for intermittent parts demand in the field of aviation: a predictive model. *Computers & Operations Research*, *30*, 2097–2114.

Goodwin, P., & Lawton, R. (1999). On the asymmetry of the symmetric MAPE. *International Journal of Forecasting*, *15*, 405–408.

Hyndman, R., Athanasopoulos, G., Bergmeir, C., Caceres, G., Chhay, L., O'Hara-Wild, M., Petropoulos, F., Razbash, S., Wang, E., & Yasmeen, F. (2020). *forecast: Forecasting functions for time series and linear models*. R package version 8.12.

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

Hyndman, R. J. (2020). A brief history of forecasting competitions. *International Journal of Forecasting*, *36*, 7–14.

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

Hyndman, R. J., & Koehler, A. B. (2006). Another look at measures of forecast accuracy. *International Journal of Forecasting*, *22*, 679–688.

Hyndman, R. J., Koehler, A. B., Snyder, R. D., & Grose, S. (2002). A state space framework for automatic forecasting using exponential smoothing methods. *International Journal of Forecasting*, *18*, 439–454.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems 30* (pp. 3146–3154). Curran Associates, Inc.

Kolassa, S. (2016). Evaluating predictive count data distributions in retail sales forecasting. *International Journal of Forecasting*, *32*, 788–803.

Kolassa, S. (2020). Why the "best" point forecast depends on the error or accuracy measure. *International Journal of Forecasting*, *36*, 208–211.

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

Kourentzes, N., Barrow, D. K., & Crone, S. F. (2014a). Neural network ensemble operators for time series forecasting. *Expert Systems with Applications*, *41*, 4235–4244.

Kourentzes, N., Petropoulos, F., & Trapero, J. R. (2014b). Improving forecasting by estimating time series structural components across multiple frequencies. *International Journal of Forecasting*, *30*, 291–302.

Liaw, A., & Wiener, M. (2002). Classification and regression by randomforest. *R News*, *2*, 18–22.

Ma, S., Fildes, R., & Huang, T. (2016). Demand forecasting with high dimensional data: The case of sku retail sales forecasting with intra- and inter-category promotional information. *European Journal of Operational Research*, *249*, 245–257.

Makridakis, S. (1993). Accuracy measures: theoretical and practical concerns. *International Journal of Forecasting*, *9*, 527–529.

Makridakis, S., Andersen, A., Carbone, R., Fildes, R., Hibon, M., Lewandowski, R., Newton, J., Parzen, E., & Winkler, R. (1982). The accuracy of extrapolation (time series) methods: Results of a forecasting competition. *Journal of Forecasting*, *1*, 111–153.

Makridakis, S., Assimakopoulos, V., & Spiliotis, E. (2018a). Objectivity, reproducibility and replicability in forecasting research. *International Journal of Forecasting*, *34*, 835–838.

Makridakis, S., Bonnell, E., Clarke, S., Fildes, R., Gilliland, M., Hoover, J., & Tashman, L. (2020a). The benefits of systematic forecasting for organizations: The ufo project. *Foresight: The International Journal of Applied Forecasting*, (pp. 45–56).

Makridakis, S., Chatfield, C., Hibon, M., Lawrence, M., Mills, T., Ord, K., & Simmons, L. F. (1993). The M2-competition: A real-time judgmentally based forecasting study. *International Journal of Forecasting*, *9*, 5–22.

Makridakis, S., & Hibon, M. (2000). The M3-Competition: results, conclusions and implications. *International Journal of Forecasting*, *16*, 451–476.

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2018b). Statistical and machine learning forecasting methods: Concerns and ways forward. *PLOS ONE*, *13*, 1–26.

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2020b). *The M5 competition: Background, organization and implementation*. Working paper.

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2020c). Responses to discussions and commentaries. *International Journal of Forecasting*, *36*, 217–223.

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2020d). The M4 Competition: 100,000 time series and 61 forecasting methods. *International Journal of Forecasting*, *36*, 54–74.

Makridakis, S., Spiliotis, E., Assimakopoulos, V., Chen, Z., Gaba, A., Tsetlin, I., & Winkler, R. L. (2020e). *The M5 Uncertainty competition: Results, findings and conclusions*. Working paper.

Møller, M. F. (1993). A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks*, *6*, 525–533.

Montero-Manso, P., Athanasopoulos, G., Hyndman, R. J., & Talagala, T. S. (2020). Fforma: Feature-based forecast model averaging. *International Journal of Forecasting*, *36*, 86–92.

Nikolopoulos, K., & Petropoulos, F. (2018). Forecasting for big data: Does suboptimality matter? *Computers & Operations Research*, *98*, 322–329.

Nikolopoulos, K., Syntetos, A. A., Boylan, J. E., Petropoulos, F., & Assimakopoulos, V. (2011). An aggregate–disaggregate intermittent demand approach (ADIDA) to forecasting: an empirical proposition and analysis. *Journal of the Operational Research Society*, *62*, 544–554.

Oreshkin, B. N., Carpov, D., Chapados, N., & Bengio, Y. (2019). N-BEATS: neural basis expansion analysis for interpretable time series forecasting. *CoRR*, *abs/1905.10437*.

Petropoulos, F. (2015). Forecasting support systems: Ways forward. *Foresight: The International Journal of Applied Forecasting*, (pp. 5–11).

Petropoulos, F., & Kourentzes, N. (2015). Forecast combinations for intermittent demand. *Journal of the Operational Research Society*, *66*, 914–924.

Petropoulos, F., Makridakis, S., Assimakopoulos, V., & Nikolopoulos, K. (2014). 'horses for courses' in demand forecasting. *European Journal of Operational Research*, *237*, 152–163.

Petropoulos, F., & Svetunkov, I. (2020). A simple combination of univariate models. *International Journal of Forecasting*, *36*, 110–115.

Pooya, A., Pakdaman, M., & Tadj, L. (2019). Exact and approximate solution for optimal inventory control of two-stock with reworking and forecasting of demand. *Operational Research: An International Journal*, *19*, 333–346.

Salinas, D., Flunkert, V., Gasthaus, J., & Januschowski, T. (2020). DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, *36*, 1181–1191.

Schwertman, N. C., Gilks, A. J., & Cameron, J. (1990). A simple noncalculus proof that the median minimizes the sum of the absolute deviations. *The American Statistician*, *44*, 38–39.

Seaman, B. (2018). Considerations of a retail forecasting practitioner. *International Journal of Forecasting*, *34*, 822 – 829.

Semenoglou, A.-A., Spiliotis, E., Makridakis, S., & Assimakopoulos, V. (2020). Investigating the accuracy of cross-learning time series forecasting methods. *International Journal of Forecasting*, . Accepted.

Smyl, S. (2020). A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting. *International Journal of Forecasting*, *36*, 75–85.

Spiliotis, E., Kouloumos, A., Assimakopoulos, V., & Makridakis, S. (2020a). Are forecasting competitions data representative of the reality? *International Journal of Forecasting*, *36*, 37–53.

Spiliotis, E., Makridakis, S., Semenoglou, A.-A., & Assimakopoulos, V. (2020b). Comparison of statistical and machine learning methods for daily SKU demand forecasting. *Operational Research: An International Journal*, (pp. 1–25).

Spiliotis, E., Nikolopoulos, K., & Assimakopoulos, V. (2019a). Tales from tails: On the empirical distributions of forecasting errors and their implication to risk. *International Journal of Forecasting*, *35*, 687–698.

Spiliotis, E., Petropoulos, F., & Assimakopoulos, V. (2019b). Improving the forecasting performance of temporal hierarchies. *PLOS ONE*, *14*, 1–21.

Spiliotis, E., Petropoulos, F., Kourentzes, N., & Assimakopoulos, V. (2020c). Cross-temporal aggregation: Improving the forecast accuracy of hierarchical electricity consumption. *Applied Energy*, *261*, 114339.

Svetunkov, I. (2020). *smooth: Forecasting Using State Space Models*. R package version 2.5.6.

Syntetos, A. A., & Boylan, J. E. (2005). The accuracy of intermittent demand estimates. *International Journal of Forecasting*, *21*, 303–314.

Syntetos, A. A., Boylan, J. E., & Croston, J. D. (2005). On the categorization of demand patterns. *Journal of the Operational Research Society*, *56*, 495–503.

Syntetos, A. A., Nikolopoulos, K., & Boylan, J. E. (2010). Judging the judges through accuracy-implication metrics: The case of inventory forecasting. *International Journal of Forecasting*, *26*, 134–143.

Teunter, R. H., & Duncan, L. (2009). Forecasting intermittent demand: a comparative study. *Journal of the Operational Research Society*, *60*, 321–329.

Teunter, R. H., Syntetos, A. A., & Babai, M. Z. (2011). Intermittent demand: Linking forecasting to inventory obsolescence. *European Journal of Operational Research*, *214*, 606–615.

Zhang, G., Patuwo, B. E., & Hu, M. Y. (1998). Forecasting with artificial neural networks:: The state of the art. *International Journal of Forecasting*, *14*, 35–62.

Zhou, H., Qian, W., & Yang, Y. (2020). Tweedie gradient boosting for extremely unbalanced zero-inflated data. *Communications in Statistics - Simulation and Computation*, *0*, 1–23.

Appendix: Description and performance of benchmarks

This appendix provides information about the 24 forecasting methods selected to serve as benchmarks to compare the accuracy of the methods submitted by the participating teams and presents their performance. In summary, the benchmarks include 16 statistical (benchmarks 1 to 16), four ML (benchmarks 17 to 20), and four combination (benchmarks 21 to 24) methods. Note that most of the benchmarks considered in the M5 "Accuracy" competition were previously tested in a different data set, involving the daily product sales of a large Greek retail company (Spiliotis et al., 2020b), suggesting that "cross-learning" ML methods can provide less biased and more accurate results than traditional approaches.

Unless otherwise specified, the benchmarks are used to predict the product-store series (level 12) and the bottom-up method is then used to forecast the rest of the series to ensure that the forecasts derived across the various aggregation levels are coherent. In this respect, benchmark methods 1-10, 12, 15, 17-21, and 24 utilize the bottom-up approach. On the contrary, benchmark methods 11, 13, 14, 16, and 22 are used to predict total sales (level 1) and the top-down method is then used to obtain forecasts for the rest of the series (levels 2-12), considering historical proportions that are estimated for the last 28 days of the train set. Finally, benchmark 23 considers a mixture of the bottom-up and top-down method.

1. **Naive**: The forecasts at time t, $\hat{y}_t$, are equal to the last known observation of the time series, y, as follows:

$$\hat{y}_t = y_{t-1}.$$

2. **Seasonal Naive (sNaive)**: The forecasts at time t are equal to the last known observation of the same period, $t - m$, as follows:

$$\hat{y}_t = y_{t-m},$$

where m is the frequency of the series. In M5, m is set equal to 7 since the series are daily. Contrary to the Naive method, sNaive can capture possible seasonal variations. Although sales do not usually display strong seasonality at low cross-sectional levels, this is very likely at higher aggregation levels.

3. **Simple Exponential Smoothing (SES)**: The simplest exponential smoothing model, aimed at predicting series without a trend (Gardner Jr., 1985). Forecasts are calculated using weighted averages that decrease exponentially across time, specified through the smoothing parameter a as follows:

$$\hat{y}_t = ay_t + (1-a)\hat{y}_{t-1}.$$

Typically, in an intermittent demand context, low smoothing parameter values are recommended in the literature (Syntetos & Boylan, 2005; Teunter & Duncan, 2009), with a ranging from 0.1 to 0.3.

Thus, the optimal value from this range is selected by minimizing the in-sample mean squared error (MSE) of the model, initialized using the first observation of the series.

4. **Moving Averages (MA)**: Moving averages are often used in practice to forecast sales (Syntetos & Boylan, 2005). Forecasts are computed by averaging the last k observations of the series as follows:

$$\hat{y}_t = \frac{\sum_{i=1}^{k} y_{t-i}}{k}.$$

The order of the MA ranges between 2 and 14 and is specified by minimizing the in-sample MSE of the method.

5. **Croston's method (CRO)**: Croston (1972) proposed forecasting intermittent demand time series by separating them into two components and extrapolating them individually: the non-zero demand size, z_t, and the inter-demand intervals, p_t. The forecasts are given as follows

$$\hat{y}_t = \frac{\hat{z}_t}{\hat{p}_t}$$

and are only updated when demand occurs. Both z_t and p_t are forecast by SES, originally using a smoothing parameter of 0.1 and an initial value equal to the first observation of each series. Croston's method is regarded as the standard method for forecasting intermittent demand.

6. **Optimized Croston's method (optCRO)**: Like CRO, but this time the smoothing parameter is selected from the range 0.1 to 0.3, as is done with SES, in order to allow for more flexibility. The non-zero demand size and the inter-demand intervals are smoothed separately using (potentially) different a parameters.

7. **Syntetos-Boylan Approximation (SBA)**: Syntetos & Boylan (2005) showed that Croston's method is biased, depending on the value of the parameter a used for smoothing the inter-demand intervals. In this regard, they proposed utilizing the Croston's method along with a debiasing factor as follows:

$$\hat{y}_t = (1 - \frac{a}{2})\frac{\hat{z}_t}{\hat{p}_t}.$$

As is done for CRO, a is set equal to 0.1 and the first observations of z_t and p_t are used for initializing.

8. **Teunter-Syntetos-Babai method (TSB)**: Teunter et al. (2011) showed that Croston's method is inappropriate for dealing with obsolescence issues since its updating only occurs in non-zero demand periods. In this respect, they proposed replacing the inter-demand intervals component of the Croston's method with the demand probability, d_t, being 1 if demand occurs at time t and 0 otherwise. Similar to CRO, d_t is forecast using SES. The forecasts are given as follows:

$$\hat{y}_t = \hat{d}_t \hat{z}_t.$$

9. **Aggregate-Disaggregate Intermittent Demand Approach (ADIDA)**: Nikolopoulos et al. (2011) proposed the utilization of temporal aggregation for reducing the presence of zero observations and mitigating the undesirable effect of the variance observed in the intervals. ADIDA uses equally sized time buckets to perform non-overlapping temporal aggregation and predict the demand over a pre-specified lead time. The time bucket is set equal to the mean inter-demand interval (Petropoulos & Kourentzes, 2015) and SES is used to obtain the forecasts.
10. **Intermittent Multiple Aggregation Prediction Algorithm (iMAPA)**: Petropoulos & Kourentzes (2015) suggested another approach for implementing temporal aggregation in demand forecasting. In contrast to ADIDA, considering a single aggregation level, iMAPA considers multiple ones, aimed at capturing different dynamics of the data. Thus, iMAPA averages the derived point forecasts at each temporal level, in this implementation generated by SES (iMAPA originally involves a selection between the Croston's method, SBA, and SES). The maximum aggregation level is set equal to the maximum inter-demand interval.
11. **Exponential Smoothing with top-down reconciliation (ES_td)**: An algorithm is used to automatically select the most appropriate exponential smoothing model for predicting total sales (level 1), indicated through information criteria (Hyndman et al., 2002). The top-down method is then used to forecast the rest of the series (levels 2-12).
12. **Exponential Smoothing with bottom-up reconciliation (ES_bu)**: The same algorithm used in ES_td is employed for forecasting the product-store series of the data set (level 12). Then, the rest of the series (levels 1-11) are predicted using the bottom-up method.
13. **Exponential Smoothing with eXogenous/eXplanatory variables (ESX)**: Similar to ES_td, but this time two exogenous/explanatory variables are used as regressors in addition to historical data to improve forecasting accuracy by providing additional information about the future. The first variable is discrete and takes values 0, 1, 2 or 3, based on the number of states that allow SNAP purchases on the examined date. The second variable is binary and indicates whether or not the examined date includes a special day.
14. **AutoRegressive Integrated Moving Average with top-down reconciliation (ARIMA_td)**: An algorithm is used to automatically select the most appropriate ARIMA model for predicting total sales (level 1), indicated through information criteria (Hyndman & Khandakar, 2008). Then, the rest of the series (levels 1-11) are predicted using the bottom-up method.
15. **AutoRegressive Integrated Moving Average with bottom-up reconciliation (ARIMA_bu)**: The same algorithm used in ARIMA_td is employed for forecasting the product-store series of the data

set (level 12). Then, the rest of the series (levels 1-11) are predicted using the bottom-up method.

16. **AutoRegressive Integrated Moving Average with eXogenous/eXplanatory variables (ARIMAX)**: Similar to ARIMA, but this time two external variables are used as regressors to improve forecasting accuracy by providing additional information about the future, exactly as done for the case of ESX. The top-down method is used for reconciliation.
17. **Multi-Layer Perceptron (MLP)**: A single hidden layer NN of 14 input nodes, 28 hidden nodes, and one output node, inspired by the work done by Makridakis et al. (2018b) and Spiliotis et al. (2020b). The networks are trained using the standard approach of constant size, rolling input and output windows (Smyl, 2020). The produced one-step-ahead forecasts are used to predict all 28 periods. The Scaled Conjugate Gradient and weight decay backpropagation (Møller, 1993) is used for estimating the weights of the network, that are randomly initialized. The maximum iterations are set equal to 500. The activation functions of the hidden and output layers are the logistic and linear ones, respectively. In total, 10 networks are trained to forecast each series and then the median operator is used to average the individual forecasts in order to mitigate possible variations due to poor weight initializations (Kourentzes et al., 2014a). Due to the nonlinear activation functions used, the data is scaled before training between 0 to 1 to avoid computational problems, meet algorithm requirement, and facilitate faster learning (Zhang et al., 1998).
18. **Random Forest (RF)**: This is a combination of multiple regression trees, each one depending on the values of a random vector sampled independently and with the same distribution (Breiman, 2001). Given that RF averages the predictions of multiple trees, it is more robust to noise and less likely to overfit the training data. We consider a total of 500 non-pruned trees and four randomly sampled variables at each split. Bootstrap sampling is done with replacement. As done in MLP, the last 14 observations of the series are considered for training the model, using constant size, and rolling input and output windows, while the produced one-step-ahead forecasts are used to predict all 28 periods.
19. **Global Multi-Layer Perceptron (gMLP)**: Like MLP, but this time, instead of training multiple models, one for each series, a single model that learns across all series is constructed. This is done given that M4 indicated the beneficial effect of "cross-learning". A random sample of three 14-long windows are selected from each of the product-store series of the data set and used as inputs, along with information about the coefficient of variation of non-zero demands and the average number of time-periods between two successive non-zero demands. This additional information is used to facilitate learning across series of different characteristics.
20. **Global Random Forest (gRF)**: Like gMLP, but instead of using an MLP to obtain the forecasts, a RF is exploited.
21. **Average of ES and ARIMA, as computed using the bottom-up approach (Com_b)**: The simple arithmetic mean of ES_bu and ARIMA_bu.

22. **Average of ES and ARIMA, as computed using the top-down approach (Com_t)**: The simple arithmetic mean of ES_td and ARIMA_td.
23. **Average of the two ES methods, the first computed using the top-down approach and the second using the bottom-up approach (Com_tb)**: The simple arithmetic mean of ES_td and ES_bu. The combination of the top-down and bottom-up approach has been previously proposed by (Abouarghoub et al., 2018).
24. **Average of the global and local MLPs (Com_lg)**: The simple arithmetic mean of MLP ("local" method) and gMLP ("global method").

The code for implementing the benchmarks is publicly available in the M5 GitHub repository (https://github.com/Mcom methods). The code developed by the organisers of the competition utilizes the *randomForest* (Liaw & Wiener, 2002), *RSNNS* (Bergmeir & Benítez, 2012), *smooth* (Svetunkov, 2020), and *forecast* (Hyndman et al., 2020) packages for R.

Table A presents the accuracy of the considered benchmarks, both per aggregation level and overall. Moreover, Figure A displays the results of the multiple comparisons with the best (MCB) test, which evaluates the differences observed between the ranks of the methods in terms of significance using RMSSE for ranking the methods.

By observing Table A and Figure A we find that:

- Naive methods are significantly less accurate than the rest of the benchmarks considered for sales forecasting.
- Seasonality is critical for producing more accurate forecasts, especially at higher aggregation levels. For example, the improvements of sNaive over the Naive benchmark range from about 6%, 11%, and 17% for levels 12, 11, and 10, respectively, and reach 72% for level 1 (average improvement of 52%). Similarly, ES_bu, that captures possible seasonality, is on average 31% more accurate than SES.
- Methods which are typically considered superior for forecasting sales, and especially intermittent demand, do not perform significantly better than, theoretically, less appropriate methods. For example, TSB, SBA, and optCRO display similar performance to CRO, with the latter also performing equally well with SES and MA.
- Combinations perform better or equally well with the individual methods that they consist of.
- Combinations provide significantly better forecasts in terms of ranks, especially when the top-down and the bottom-up reconciliation approaches are mixed (Com_td) or different base forecasting models (ETS and ARIMA) are used along with the bottom-up method (Com_b).

Table A: The accuracy of the 24 benchmarks in terms of WRMSSE. The results are presented both overall and per aggregation level.

Method	Aggregation level												Average
	1	2	3	4	5	6	7	8	9	10	11	12	
Naive	1.967	1.904	1.880	1.947	1.914	1.881	1.878	1.798	1.764	1.479	1.360	1.253	1.752
sNaive	0.560	0.673	0.718	0.623	0.708	0.760	0.829	0.801	0.888	1.223	1.205	1.176	0.847
SES	0.921	0.938	0.944	0.927	0.983	0.959	1.002	0.956	0.994	1.071	1.002	0.932	0.969
MA	0.891	0.918	0.931	0.900	0.960	0.940	0.986	0.944	0.988	1.070	1.006	0.938	0.956
CRO	0.900	0.915	0.923	0.909	0.971	0.941	0.987	0.940	0.983	1.083	1.002	0.926	0.957
optCRO	0.902	0.916	0.926	0.910	0.970	0.940	0.987	0.942	0.985	1.084	1.004	0.928	0.958
SBA	0.902	0.917	0.926	0.914	0.983	0.943	0.993	0.940	0.983	1.073	0.994	0.919	0.957
TSB	0.911	0.926	0.935	0.918	0.975	0.949	0.994	0.948	0.988	1.068	0.997	0.928	0.961
ADIDA	0.902	0.917	0.924	0.912	0.969	0.943	0.987	0.941	0.982	1.063	0.993	0.922	0.955
iMAPA	0.909	0.925	0.932	0.917	0.973	0.948	0.992	0.946	0.986	1.065	0.996	0.925	0.960
ES_td	0.470	0.550	0.664	0.530	0.640	0.629	0.727	0.717	0.801	1.029	0.973	0.915	0.720
ES_bu	0.426	0.514	**0.580**	0.478	**0.557**	**0.577**	**0.654**	**0.643**	**0.728**	1.012	0.969	0.915	**0.671**
ESX	**0.350**	**0.494**	0.627	**0.438**	0.567	0.590	0.692	0.692	0.779	1.026	0.974	0.917	0.679
ARIMA_td	0.615	0.673	0.753	0.656	0.768	0.725	0.810	0.785	0.856	1.027	0.969	0.910	0.796
ARIMA_bu	0.829	0.850	0.870	0.844	0.905	0.882	0.932	0.893	0.938	1.048	0.981	0.917	0.908
ARIMAX	0.374	0.514	0.638	0.459	0.606	0.604	0.707	0.700	0.787	1.019	0.968	0.912	0.691
MLP	0.892	0.942	0.974	0.910	0.972	0.965	1.016	0.984	1.026	1.084	1.014	0.943	0.977
RF	0.960	0.989	1.026	0.962	1.012	1.003	1.047	1.023	1.057	1.085	1.010	0.940	1.010
gMLP	0.882	0.914	0.919	0.923	0.996	0.967	1.013	0.953	0.997	1.063	0.991	0.920	0.961
gRF	1.062	1.073	1.081	1.071	1.108	1.096	1.116	1.075	1.089	1.078	1.001	0.932	1.065
Com_b	0.522	0.591	0.644	0.561	0.641	0.647	0.718	0.696	0.771	1.012	0.963	0.907	0.723
Com_t	0.517	0.592	0.693	0.571	0.688	0.661	0.755	0.738	0.819	1.026	0.970	0.912	0.745
Com_tb	0.444	0.520	0.598	0.496	0.587	0.588	0.673	0.658	0.743	**1.008**	**0.960**	**0.905**	0.682
Com_lg	0.886	0.922	0.936	0.898	0.959	0.948	0.989	0.948	0.986	1.058	0.989	0.921	0.953

- Utilizing exogenous/explanatory variables significantly improves the performance of the methods that only depend on historical time series data. For example, ESX is on average 6% more accurate than ES_td, while ARIMAX is 13% more accurate than ARIMA_td.
- Temporal aggregation generally provides more accurate forecasts than traditional methods used in sales forecasting, like SES, CRO, SBA, and TSB. However, the improvements are minor in terms of WRMSSE, as ADIDA and iMAPA are just 1.5% more accurate on average than SES.

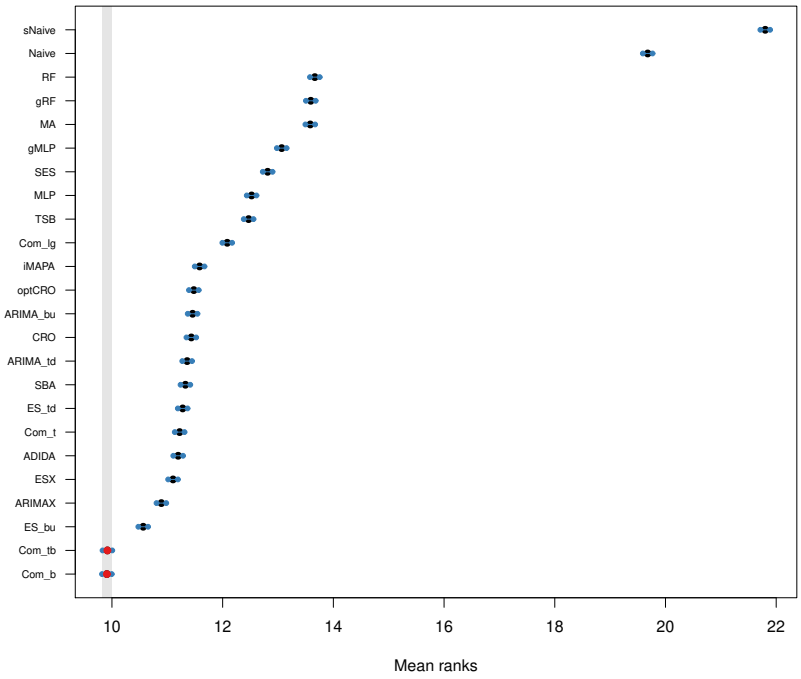

Figure A: Average ranks and 95% confidence intervals of the 24 benchmarks of the M5 "Accuracy" competition over all M5 series: multiple comparisons with the best (RMSSE used for ranking the methods) as proposed by Koning et al. (2005).