International Journal of Forecasting 38 (2022) 1386–1399

Contents lists available at ScienceDirect

International Journal of Forecasting

journal homepage: www.elsevier.com/locate/ijforecast

Simple averaging of direct and recursive forecasts via partial pooling using machine learning

YeonJun In[1], Jae-Yoon Jung[*]

Department of Industrial and Management Systems Engineering, Kyung Hee University, 1732 Deogyeong-daero, Giheung-gu, Yongin-si, Gyeonggi-do 17104, Republic of Korea

ARTICLE INFO

Keywords:
Direct and recursive multi-step forecasting
Multi-level data
Forecast averaging
Machine learning
LightGBM

ABSTRACT

This article introduces the winning method at the M5 Accuracy competition. The presented method takes a simple manner of averaging the results of multiple base forecasting models that have been constructed via partial pooling of multi-level data. All base forecasting models of adopting direct or recursive multi-step forecasting methods are trained by the machine learning technique, LightGBM, from three different levels of data pools. At the competition, the simple averaging of the multiple direct and recursive forecasting models, called DRFAM, obtained the complementary effects between direct and recursive multi-step forecasting of the multi-level product sales to improve the accuracy and the robustness.

1. Introduction

M-competitions, named after the organizer Spyros Makridakis, are traditional forecasting competitions that have been held since 1982 (Hyndman, 2020). The fifth competition, called the M5 Accuracy competition, was opened at the Kaggle site on March 2, 2020 and considered the problem of point forecasting the product sales of Walmart stores in the USA (Kaggle, 2020; MOFC, 2020). Given the historical data of product sales for a 5-year period, participants were required to accurately predict the daily sales amounts of each product in ten stores. This article introduces the winning method of this M5 Accuracy competition.

The presented forecasting method simply averages the forecasts of multiple direct and recursive multi-step models that are constructed via partial pooling from multi-level data. The competition data has the hierarchical structure, as known as *multi-level data* (Gelman & Hill, 2006), which is composed of regional levels (state and store) and product levels (category, department, and product). The goal of the competition is *multi-step forecasting* for daily product sales over the next 28 days. One can make various forecasting strategies for this multi-step forecasting of multi-level data. The presented method, named direct and recursive forecast averaging method via partial pooling (DRFAM), takes the way of integrating a few well-known strategies for the multi-step forecasting of multi-level data.

- The presented method adopts a *partial pooling* strategy for the forecasting of *multi-level data*. Partial pooling is often more suitable for multi-level structured data than two extreme pooling strategies: complete pooling and no pooling (Gelman & Hill, 2006). The former builds a single forecasting model from whole data, namely a single pool, while the latter does individual forecasting models for data groups at the bottom level. Complete pooling cannot

[*] Corresponding author.
E-mail addresses: yeonjun.in@kaist.ac.kr (Y. In), jyjung@khu.ac.kr (J.-Y. Jung).

[1] Current affiliation: Department of Industrial and Systems Engineering, KAIST, 291 Daehak-ro, Yuseong-gu, Daejeon 34141, Republic of Korea.

https://doi.org/10.1016/j.ijforecast.2021.11.007
0169-2070/© 2021 International Institute of Forecasters. Published by Elsevier B.V. All rights reserved.

reflect different properties among groups, and no pooling may have deficient information in the case of a small group. In contrast, the partial pooling approach prepares multiple data pools at a proper middle level that has been found by grouping or clustering and then constructs a forecasting model for each pool. In the presented method, we choose three levels of partial pools, namely store, store-category, and store-department levels, combining the store level with two different product levels. As a result, for every store, we prepare 11 data pools, which include 1 store pool, 3 store-category pools, and 7 store-department pools, and we totally obtain 110 data pools for 10 stores.

- Multiple base forecasting models are generated by applying both *direct and recursive multi-step forecasting* methods to each data pool. In forecasting theory, a single-step forecasting problem requires the forecast for a single time point in the future, while a multi-step forecasting problem does the forecasts for multiple time points. There are two general approaches to the multi-step forecasting; a direct forecasting method builds different forecasting models for each time point in the future, while a recursive forecasting method builds a single forecasting model where the forecasted values for the previous time points are used recursively as input of the next forecasting (Marcellino et al., 2006; Taieb & Hyndman, 2012). The presented DRFAM averages the forecasts of the multiple direct and recursive forecasting models that have been trained at different levels of partial pools by using machine learning. Particularly, we do not use different direct forecasting models for different time points, but a single direct forecasting model in order to reduce the time cost of training machine learning models. If we had built different direct forecasting models for 28 days of 110 pools, we would have trained totally 3,080 direct forecasting models that should be trained by machine learning.
- The presented method takes the *arithmetic mean* of forecasts of multiple base models that have been constructed by direct and recursive forecast methods from many data pools. The simple averaging methods have been shown to improve the forecasting performance of the individual method in many studies (Bates & Granger, 1969; Clemen & Winkler, 1986; Makridakis & Winkler, 1983), and advanced model averaging methods such as Bayesian model averaging (BMA) and frequentist model averaging (FMA) were also developed to find optimal weights of combining base models (Fletcher, 2018; Hoeting et al., 1999; Steel, 2020). Nevertheless, many powerful ensemble methods such as bagging, random forest, and extra-tree in machine learning still evenly average the results of multiple prediction models (Geurts et al., 2006; Ho, 1998), which focus on building a larger number of base models instead of finding optimal weights of combining base models. The proposed DRFAM also adopts the simple averaging of the forecasts that are predicted by the advanced machine learning models that can reflect a large number of feature variables. The base models are trained using Light Gradient Boosting Machine (LightGBM), which is a powerful and relatively fast ensemble learning algorithm (Ke et al., 2017).

The overall procedure is summarized in Fig. 1. The partial pooling strategy is first adopted for the multi-level data of the competition. In this research, three different levels of data pools (i.e., store, store-category, and store-department levels) are prepared for sales forecasting at each store. Second, single-model direct and recursive forecasting models are designed for the multi-step forecasting over the 28-day period. To that end, many features are extracted from the given time series data in every pool. Third, the base forecasting models are trained by a machine learning technique, LightGBM, and three types of averaging models are then constructed by combining the trained base models. Finally, the forecasting performances of three forecast averaging models are evaluated through time series cross-validation in favor of an out-of-sample evaluation (Bergmeir et al., 2018; Hyndman & Athanasopoulos, 2018). In addition, we also investigate significant features of the final forecasting models.

To construct base forecasting models, LightGBM was selected among advanced machine learning techniques for regression. It is an open-source gradient boosting framework that uses a tree-based ensemble learning algorithm. LightGBM was successfully invented to tackle the weaknesses of eXtreme Gradient Boosting (XGBoost), which has been one of the best effective gradient boosting decision trees, by improving its inefficient computational speed and memory consumption (Ke et al., 2017). For these reasons, LightGBM has been receiving top marks in many recent data analysis competitions. We also selected the algorithm for training base forecasting models to take such advantages such as speed and scalability.

The remainder of the paper is organized as follows. In Section 2, the studies related to the proposed method are introduced. In Section 3, the framework of the proposed DRFAM and its algorithm is presented. The experiments with the competition data are described along with the evaluation of the forecasting averaging models in Section 4. Finally, we conclude this paper in Section 5.

2. Related work

2.1. Hierarchical forecasting

The time series data in the M5 competition are organized in a hierarchy. In the hierarchical forecasting problem, the participants have to forecast unit sales of products at 12 levels. One can generally adopt two approaches to such hierarchical forecasting: top-down and bottom-up approaches (Hyndman & Athanasopoulos, 2018). A *top-down approach* estimates the forecast at the top level and then breaks it down into lower levels according to appropriate proportions. For instance, Gross and Sohl (1990) applied the average proportions to disaggregate the forecast to lower levels, and Athanasopoulos et al. (2009) developed a disaggregation model of estimating the proportions based on the historical data.

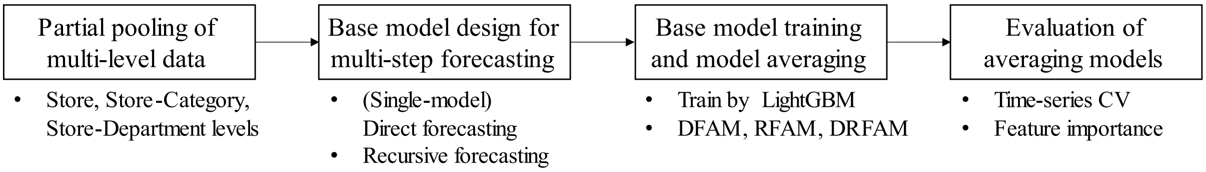

Partial pooling of multi-level data
Base model design for multi-step forecasting
Base model training and model averaging
Evaluation of averaging models
• Store, Store-Category, Store-Department levels
• (Single-model) Direct forecasting
• Recursive forecasting
• Train by LightGBM
• DFAM, RFAM, DRFAM
• Time-series CV
• Feature importance

Fig. 1. Procedure of the forecast averaging methods with partial pooling.

On the other hand, a *bottom-up approach* estimates forecasts at the bottom level and then sums them up at higher levels. To obtain the forecasts at the bottom level, the autoregressive integrated moving average (ARIMA), the exponential smoothing based on innovations state-space models (ISSM), and the regression with ARIMA errors (RegARIMA) were used (Athanasopoulos et al., 2009; Hyndman et al., 2011; Spiliotis et al., 2020). In this research, the bottom-up approach is adopted to solve the hierarchical forecasting problem. We train the base forecasting models at the bottom level (i.e., the store-product level) using the machine learning algorithm, LightGBM, and then sum the forecasts at 11 higher levels.

2.2. Multi-step time series forecasting

In contrast to one-step ahead forecasting, multi-step ahead time series forecasting is defined as the task of predicting a sequence of values in a time series (Cheng et al., 2006). Two approaches to the multi-step forecasting can be adopted: recursive forecasting and direct forecasting. A *recursive forecasting* method builds a single one-step ahead forecasting model, and the forecasted value is recursively used to predict the next forecast. It often suffers from the accumulation of errors because the forecasts are repeatedly used for the future forecasting (Chevillon, 2007). On the contrary, a *direct forecasting* method constructs multiple different forecasting models, not a single model, to obtain forecasts at each time step (Marcellino et al., 2006). Therefore, the longer the forecasting step is, the more computation and time are required. In this research, we try to combine recursive and direct forecasting methods for the multi-step forecasting, but we use only a single model for the direct forecasting in order to reduce computation time, which is elaborated in Section 3.2.2.

2.3. Forecast averaging

Forecast averaging is often used to complement multiple models in the field of forecasting. Many studies have proved that averaging the forecasts of individual models is an effective way to improve forecast accuracy. For instance, Bates and Granger (1969) introduced the simple averaging of different forecasting models to produce more accurate forecasts. Makridakis and Winkler (1983) analyzed how many forecasting methods were better to combine and which one is better to choose. Moreover, Clemen and Winkler (1986) showed a simple combination using the arithmetic mean outperformed the best individual model as well as complex combination methods such as Bayesian combination. Since that time, many advanced model averaging methods such as FMA or BMA have been developed (Fletcher, 2018; Hoeting et al., 1999; Liang et al., 2011). In the field of forecasting, Raftery et al. (2005) introduced BMA, and Liu and Kuo (2016) adopted the FMA to improve forecasting accuracy.

Some of the top solutions in the M4 competition also adopted complex model combination methods. The feature-based forecast model averaging (FFORMA) method achieved the second place (Montero-Manso et al., 2020), and the weighted combination of forecasting models achieved the third place (Pawlikowski & Chorowska, 2020). Compared to their methods, the proposed DRFAM takes a simple combination of various forecasts predicted by multiple machine learning models trained with different levels of partial pools, rather than a complex combination.

3. Proposed method

3.1. Framework

The dataset of the M5 competition includes the hierarchical unit sales data of 3,049 products in terms of region and product. At the regional levels, the unit sales of all products were available for 10 stores in three US states, California, Texas, and Wisconsin. At product levels, the product sales at each store can be aggregated for three product categories and also for seven product departments. The historical data ranged from January 29, 2011 to June 19, 2016 for 1,969 days, and the data for the last 28 days were provided as the test set of the competition. The submitted forecasts for all products' unit sales of the 10 stores for the last 28 days were evaluated based on the weighted root mean squared scaled errors (WRMSSE). See the details of the M5 Accuracy competition described by Bojer and Meldgaard (2020) and Makridakis et al. (2021), and available at the competition guide (MOFC, 2020).

To forecast the hierarchical unit sales of the M5 competition, the overall framework of the presented method is designed in Fig. 2. From the provided multi-level data, three different levels of data such as store, category, and department are first prepared. Second, we obtain three levels of partial pools: 10 store pools, 30 store-category pools, and 70 store-department pools. For each pool, we then build direct and recursive base forecasting models after extracting their corresponding forecasting features. Note that all the base forecasting models using LightGBM are constructed only at the final level (i.e., the product-store level), although they can use different levels of data pools. Third, we investigate three types of forecast averaging models according to base model selection: direct forecast averaging model (DFAM), recursive forecast averaging model (RFAM), and DRFAM. Fourth, the averaging models are evaluated using 13 time series cross-validation sets. Finally, the forecast result of the averaging model that performs well for the validation sets is submitted to

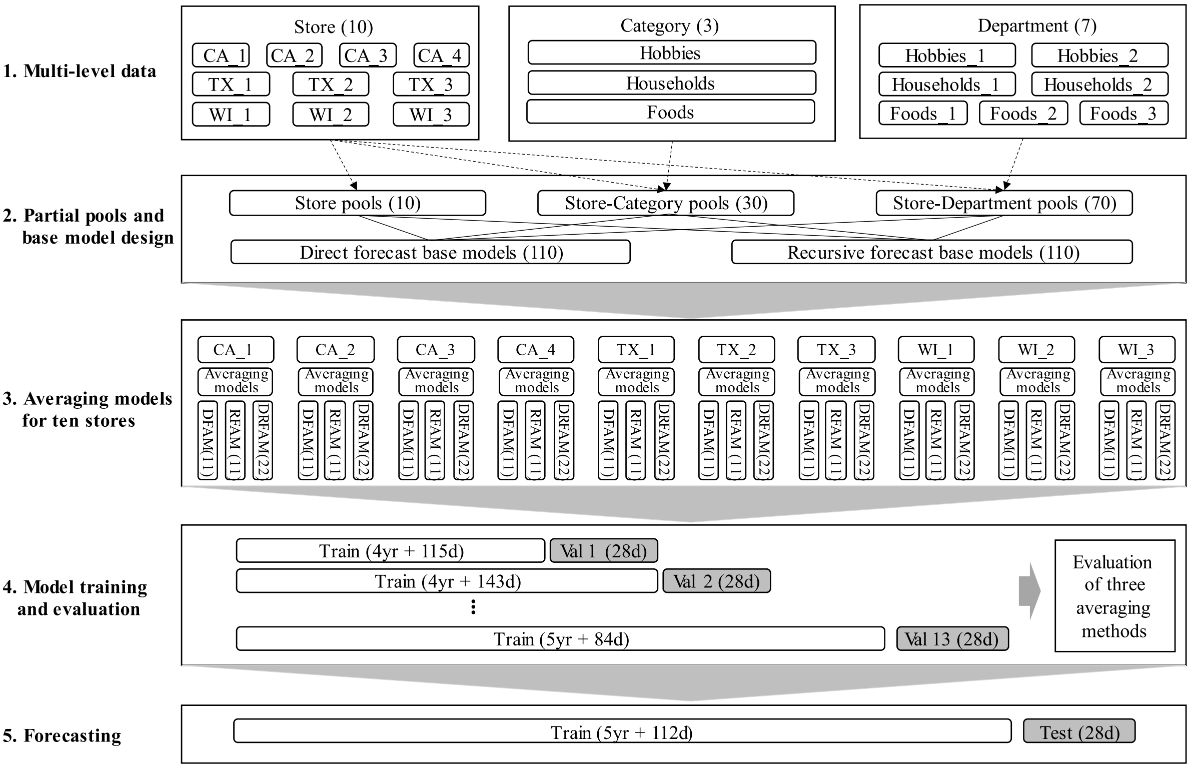

1. Multi-level data
Store (10)
CA_1 CA_2 CA_3 CA_4
TX_1 TX_2 TX_3
WI_1 WI_2 WI_3
Category (3)
Hobbies
Households
Foods
Department (7)
Hobbies_1 Hobbies_2
Households_1 Households_2
Foods_1 Foods_2 Foods_3
2. Partial pools and base model design
Store pools (10)
Store-Category pools (30)
Store-Department pools (70)
Direct forecast base models (110)
Recursive forecast base models (110)
3. Averaging models for ten stores
CA_1 CA_2 CA_3 CA_4 TX_1 TX_2 TX_3 WI_1 WI_2 WI_3
Averaging models
DFAM(11) RFAM (11) DRFAM(22)
4. Model training and evaluation
Train (4yr + 115d) Val 1 (28d)
Train (4yr + 143d) Val 2 (28d)
⋮
Train (5yr + 84d) Val 13 (28d)
Evaluation of three averaging methods
5. Forecasting
Train (5yr + 112d) Test (28d)

Fig. 2. Framework of the proposed method.

Table 1
Features for base forecasting models of product sales on date t.

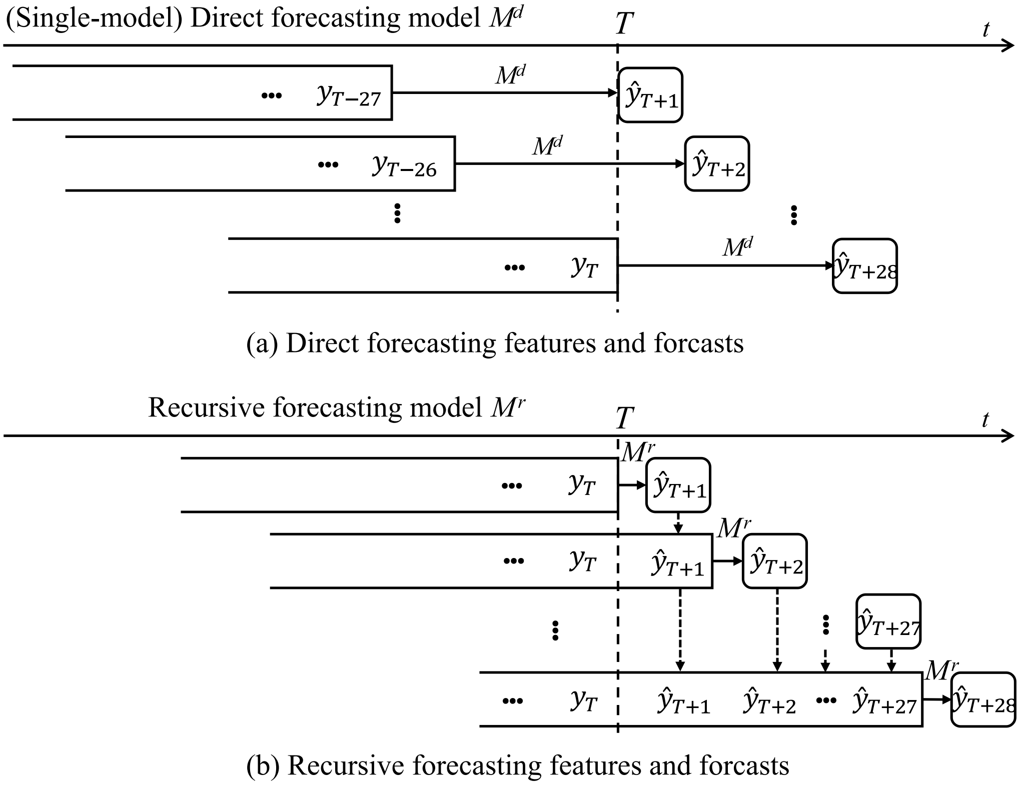
Category	Features	Comment
Identifier (5)	state_id, store_id, cat_id, dept_id, prod_id	Provided features
Day (8)	day$_t$, month$_t$, year$_t$, weekday$_t$, weeknum$_t$, month_week$_t$, is_workingday$_t$, is_weekend$_t$	
Event (5)	event_name$_t$, event_type$_t$, snap_CA$_t$, snap_TX$_t$, snap_WI$_t$	
Price (10)	price$_t$, price_norm$_t$, price_max, price_min, price_mean, price_std, price_n_changes, price$_t$/price$_{t-1}$, price$_t$/monthly_price$_t$, price$_t$/yearly_price$_t$	Derived features
Sales (28)	sales$_{t-h}$ ($h = 28, \ldots, 41$), sales_mean$_{t-h,L}$ and sales_std$_{t-h,L}$ ($h = 28$; $L = 7, 14, 30, 60, 180$), sales_mean_at_state, sales_std_at_state, sales_mean_at_store, sales_std_at_store	
Recursive sales[a] (12)	sales_mean$_{t-h,L}$ ($h = 1, 7, 14$; $L = 7, 14, 30, 60$)	

[a] Recursive sales features were used only for recursive forecasting models.

the competition after training again with all the training data.

To construct the base forecasting models for each store's daily product sales by using machine learning, the features were extracted from the given dataset, as listed in Table 1. The features are grouped into six categories: identifier, day, event, price, sales, and recursive sales features. Direct forecasting models for three levels of data pools use 56 features in the first five categories: identifier, day, event, price, and sales. Recursive forecasting models use additionally 12 features in the recursive sales category.

The *identifier* features include information regarding the product-related identifiers, such as the state, store, category, department, and product identifiers. The *day* features represent the characteristics of the date in terms of the week, month, year, and working day. Furthermore, *event* features include the date of some events which affect the amount of daily sales, such as the Super Bowl, Valentine's Day, and a supplemental nutrition assistance program. The *price* features, such as mean and standard deviation values, are extracted from the price time series of each product that was given in the competition data.



(a) Direct forecasting features and forcasts

(b) Recursive forecasting features and forcasts

Fig. 3. Two types of multi-step forecasting features and methods used for base models.

The identifier and event features were handled as categorical values in the machine learning models, whereas all other features were considered numeric values.

The *sales* features are engineered from the unit sales time series of each product. The most recent historical sales for two weeks available for any target date t are first extracted from the sales time series, that is, $[t-28, t$-$41]$. The historical sales feature starts from $t-28$ because there is no sales information for the previous 27 days when predicting the last target date of the test set. In Fig. 3(a), every forecast $\hat{y}_t$ $(T+1 \le t \le T+28)$ requires only historical sales data, y_{t-28} and earlier y's, to predict it. Also, the mean and standard deviation of the unit sales for different time window length L (e.g., one week, two weeks, one month, two months, and half a year), and the statistics of the product sales in the corresponding state and store during the entire period of the training set are derived from the historical sales data. The *recursive sales* features are calculated by additionally including the recursively forecasted sales values that replaced the unknown sales of the previous days by the forecasting model. In Fig. 3(b), forecast $\hat{y}_t$ is predicted by using historical sales data before T and previous forecasted sales data after T. The mean sales values for different time window length L from one day, one week, and two weeks before the target date t are calculated.

3.2. Algorithm – direct and recursive forecast averaging method (DRFAM)

The algorithm of DRFAM via partial pooling is described below. The inputs of the algorithm are time series data for training and for test; the former is labeled with y, but the latter is unlabeled. In the training data, p means an indicator that will be used to map the data to one of data pools in every level, and T is the forecasting time. The output of the algorithm is the forecasted values of three averaging models for test data.

The algorithm is composed of four steps: (1) partial pooling, (2) feature extraction for training data, (3) training base models, and (4) forecasting for test data. In Step 1, partial pools are first prepared. Data are partitioned in order that any p can belong to only one pool at every level. In Step 2, it extracts time series features for direct and recursive forecasting, $\tilde{X}^d_{p,t}$ and $\tilde{X}^r_{p,t}$, from the past sales time series. $\tilde{X}^d_{p,t}$ and $\tilde{X}^r_{p,t}$ denote direct and recursive forecasting features of the pth product-store at time t, respectively. In Step 3, for every partial pool, a direct forecasting model $M^d_{i,v}$ and a recursive forecasting model $M^r_{i,v}$ are trained by using the extracted features and their labels. $M^d_{i,v}$ and $M^r_{i,v}$ are direct and recursive forecasting models that are trained by the ith pool at level v, $P_{i,v}$, respectively. The last step is forecasting for test data including feature extraction of test data, forecasting of base models, and averaging of base model forecasts. For every test data, feature extraction (in line 9) and direct forecasting for test data (in lines 10 to 12) are conducted,

```
Algorithm. DRFAM (Direct and Recursive Forecast Averaging Method via Partial Pooling)
Input: time series data for training $D^{tr}=\{(X_{p,t}, y_{p,t}) \mid p$ is a product-store index, $t \le T\}$ and for test $D^{te}=\{X_{p,t} \mid t=T+1, \ldots, T+h\}$
Output: multi-step forecasts of DFAM, RFAM and DRFAM, $\hat{Y}=\{(\hat{y}^d_{p,t}, \hat{y}^r_{p,t}, \hat{y}^w_{p,t}) \mid t=T+1, \ldots, T+h\}$
# Partial pooling
1: Prepare partial pools $P=\{P_{i,v} \subset D^{tr} \mid P_{i,v}$ is the $i$-th pool at level $v\}$
# Feature extraction of training data
2: for every $(X_{p,t}, y_{p,t})$ in $D$:
3:     Extract features $\tilde{X}^d_{p,t}$ for direct forecasting from past time series $y_{p,s}$ for $s \le t-h$.
4:     Extract features $\tilde{X}^r_{p,t}$ for recursive forecasting from past time series $y_{p,s}$ for $s \le t-1$.
# Train base models
5: for every $P_{i,v}$ in $P$:
6:     Train a direct forecasting model $M^d_{i,v}(X_{p,t}, \tilde{X}^d_{p,t})$ with every $(X_{p,t}, y_{p,t})$ in $P_{i,v}$ and its $\tilde{X}^d_{p,t}$.
7:     Train a recursive forecasting model $M^r_{i,v}(X_{p,t}, \tilde{X}^d_{p,t}, \tilde{X}^r_{p,t})$ with every $(X_{p,t}, y_{p,t})$ in $P_{i,v}$ and its
       $\tilde{X}^d_{p,t}$ and $\tilde{X}^r_{p,t}$.
# Forecasting for test data
8: for every $X_{p,t}$ in $D^{te}$:
9:     Extract features $\tilde{X}^d_{p,t}$ for direct forecasting from past time series $y_{p,s}$ for $s \le t-h \le T$.
10:    for every level $v$:
11:        Choose a direct forecasting model $M^d_{i,v}$ according to $p$ of $X_{p,t}$.
12:        Obtain a forecast $\hat{y}^d_{p,t,v}= M^d_{i,v}(X_{p,t}, \tilde{X}^d_{p,t})$
13:    Extract features $\tilde{X}^r_{p,t}$ for recursive forecasting from $y_{p,s}$ for $s \le T$ and $\hat{y}^{ram}_{p,s}$ for $T < s \le t-1$.
14:    for every level $v$:
15:        Choose a recursive forecasting model $M^r_{i,v}$ according to $p$ of $X_{p,t}$.
16:        Obtain a forecast $\hat{y}^r_{p,t,v}= M^r_{i,v}(X_{p,t}, \tilde{X}^d_{p,t}, \tilde{X}^r_{p,t})$
17:    $\hat{y}^d_{p,t} = \sum_v \hat{y}^d_{p,t,v} / V$
18:    $\hat{y}^r_{p,t} = \sum_v \hat{y}^r_{p,t,v} / V$
19:    $\hat{y}^w_{p,t} = (\hat{y}^d_{p,t} + \hat{y}^r_{p,t})/2$
20:    $\hat{Y} \leftarrow (\hat{y}^d_{p,t}, \hat{y}^r_{p,t}, \hat{y}^w_{p,t})$
21: return $\hat{Y}$
```

and additional feature extraction (in line 13) and recursive forecasting for test data (in lines 14 to 16) are also done. Finally, three types of the averaged forecasts at V levels are calculated and appended to the forecasting results $\hat{Y}$ (in lines 17 to 20).

3.2.1. *Partial pooling of multi-level data*

To accurately forecast the product sales at the stores, we should understand the trend of such sales at the regional and product levels according to season, event, and product price, among other factors. In the presented method, we focus on the sales forecasting of each store in the hierarchical structure of the competition data. The product sales time series of each store can be analyzed by the store itself or by combining with product groups such as product categories and departments. In this way, we prepare 110 partial pools including 10 stores, 30 combinations of stores and categories, and 70 combinations of stores and departments.

3.2.2. *Direct and recursive multi-step forecasting*

At this competition, participants must forecast all product sales over the next 28 days. The latest data are generally more informative for the forecasting. Sales forecasting for a few days later can use the known recent sales data effectively, whereas forecasts for approximately 28 days later cannot use known sales data in the near past. For that reason, one can use a recursive forecasting method as depicted in Fig. 3(b), where the model predicts future sales after predicting near-future sales in advance (Taieb & Hyndman, 2012). However, as a forecasting horizon is longer, the recursive approach causes more forecast errors. To complement the forecast error of long-term recursive forecasting, we use together with a single-model direct forecasting method.

A single-model direct forecasting model for h-step-ahead forecasting uses only a single model M^d for forecasts in the period. To let the model use only known values, the historical sales data used for a forecast $\hat{y}_{T+f}$ starts from y_{T+f-h}. As depicted in the top of Fig. 3, to obtain the forecasts during the 28-day future period ($\hat{y}_{T+1}$ to $\hat{y}_{T+28}$), we set a forecast horizon to 28 days (i.e., h = 28). The single-model direct forecasting is represented as:

$$\hat{y}^d_{T+f} = M^d(X_{T+f}, \tilde{X}(y_{T+f-h}, \ldots, y_{T+f-h-L-1}))$$

where $\hat{y}^d_{T+f}$ is a direct forecast for time T+f, M^d is a direct forecasting model, X_{T+f} is a feature vector for time T+f, $\tilde{X}(\ldots)$ is an feature extraction from a y_t's time series, and L

Table 2
Forecast errors of three averaging models for validation and test sets.

Averaging model	13 validation sets		Test set
	Mean (WRMSSE)	S.D. (WRMSSE)	WRMSSE
DFAM	0.669	0.087	0.525
RFAM	0.572	**0.047**	0.592
DRFAM	**0.560**	0.050	**0.520**

is a window length for feature extraction (e.g., two weeks, one months, and half a year).

A recursive forecasting model is similar to a single-model direct forecasting model. But, as shown in the bottom of Fig. 3, to extract the features for a forecast $\hat{y}^r_{T+f}$, it uses the estimated recent sales series, $(\hat{y}^r_{T+f-1}, \dots, \hat{y}^r_{T+1})$, which were forecasted from its own model M^r. The recursive forecasting model for h-step-ahead forecasting can also be represented as:

$$\hat{y}^r_{T+f} = M^r(X_{T+f}, \tilde{X}(\hat{y}_{T+f-1}, \dots, \hat{y}_{T+1}, y_T, \dots, y_{T+f-h-L-1}))$$

where $\hat{y}^r_{T+f}$ is a recursive forecast for time *T+f*, M^r is a recursive forecasting model, and $\hat{y}_{T+f-1}, \dots, \hat{y}_{T+1}$ are the estimated time series for previous time points.

In the presented method, we use both types of forecasting models as base models to complement each other; direct forecasting models use only known historical data and their statistics, while recursive forecasting models use the prior forecasted values to extract the statistical features from a more recent period.

3.2.3. Model training by machine learning

LightGBM, one of the gradient boosting tree algorithms, is used to train base forecasting models in the proposed method (Ke et al., 2017). In gradient boosting algorithms, weak models are sequentially trained to compensate the weakness of their precedent models. The strong model that is composed of many weak models produces powerful prediction; however, too many weak models could lead to overfitting (Friedman, 2001). For that reason, it does not only shrink the number of leaf nodes, but also supports the early stopping. In this analysis, we only use shrinking the number of leaf nodes to avoid overfitting. Moreover, LightGBM supports a number of hyperparameters to tune which would have a great impact on the model performance. The guidance about how to tune them is available on the LightGBM's official documentation (Microsoft, 2021).

In this research, the Tweedie loss function was selected as an objective function for training LightGBM models since the function is effective when the target values are zero-inflated (Zhou et al., 2020). As a matter of fact, the competition data contained numerous zero sales values, e.g., 55.6% for CA, 60.4% for TX, and 60.5% for WI. Moreover, in the product category, the zero sales values were 50.2% for foods, 70.3% for hobbies, and 63.4% for households.

3.2.4. Evaluation of averaging models

To benefit complementary effects between two types of multi-step forecasting models at different levels, we average the forecasting results. We consider three approaches to model selection: DFAM, RFAM, and DRFAM. DFAM evenly averages three forecasts of direct forecasting models at the store level, the store-category level, and the store-department level. Specifically, consider the product "A" which belongs to store "CA_1", category "Hobbies", and department "Hobbies_1". In order to forecast the daily unit sales of "A", DFAM will average the forecasts of the three direct forecasting base models that have been trained with three data pools: the "CA_1" pool, the "CA_1-Hobbies" pool, and the "CA_1-Hobbies_1" pool. In a similar manner, RFAM does three forecasts of recursive forecasting models. Finally, DRFAM uses all of the six forecasts of direct forecasting and recursive forecasting models to forecast the sales of "A". In other words, to forecast the sales of any products, DRFAM will average the six forecasts of the above models.

To test the performance of the three averaging models, several validation sets were prepared. As depicted in Fig. 2, the last thirteen 28-day periods were held out repetitively as validation sets, which is known as an out-of-sample (Bergmeir et al., 2018), and all the sales data before each validation set were used to train base models. In this research, DRFAM showed the best performance among three averaging models, and finally, the forecasting results of DRFAM for test data were submitted to the M5 Accuracy competition.

4. Experiments

4.1. Performance comparison among averaging models

The forecasting performance of three averaging models, DFAM, RFAM, and DRFAM, is compared with WRMSSE, which was used for the evaluation measure at this competition. The WRMSSE is calculated by averaging $WRMSSE_l$ values at twelve levels, and $WRMSSE_l$ is the weighted average of root mean squared scaled errors ($RMSSE_k$) of the kth time series (Hyndman & Koehler, 2006).

$$WRMSSE = \frac{1}{12}\sum_{l=1}^{12} WRMSSE_l = \frac{1}{12}\sum_{l=1}^{12}\sum_{k=1}^{n_l}(w_k \times RMSSE_k)$$

$$RMSSE_k = \sqrt{\frac{1}{h}\sum_{t=n+1}^{n+h}(y_{k,t} - \hat{y}_{k,t})^2 \Big/ \frac{1}{n-1}\sum_{t=2}^{n}(y_{k,t} - y_{k,t-1})^2}$$

where $y_{k,t}$ and $\hat{y}_{k,t}$ are the actual and predicted sales of the kth time series at time t, respectively, n is the length of the time series, and h is the forecasting horizon (28 days in this research). Then, n_l is the number of the items that belong to the lth level. For instance, $n_3 = 10$ at the store level, $n_8 = 30$ at the store-category level, and $n_{12} = 30{,}490$ at the final level.

The WRMSSE values of three averaging models for 13 validation sets and the errors for test set are shown in Fig. 4, and the performance of the averaging models are also summarized in Table 2. The performance for the test set can be evaluated after submission to the Kaggle M5 Accuracy site (Kaggle, 2020), and the test data is currently available on the M5 GitHub (M5, 2021). Unfortunately, we could not do the fine-tuning of the hyperparameters

Table 3
Forecast errors of three averaging models at different levels for test set.

Level id	Aggregation level	Level description	$WRMSSE_l$		
			DFAM	RFAM	DRFAM
1	Total	Unit sales of all products, aggregated for all stores/states	0.214	0.381	0.199
2	State	Unit sales of all products, aggregated for each state	0.305	0.429	0.310
3	Store	Unit sales of all products, aggregated for each store	0.416	0.472	0.400
4	Category	Unit sales of all products, aggregated for each category	0.269	0.417	0.277
5	Department	Unit sales of all products, aggregated for each department	0.367	0.461	0.365
6	State-category	Unit sales of all products, aggregated for each state and category	0.376	0.488	0.390
7	State-department	Unit sales of all products, aggregated for each state and department	0.472	0.538	0.474
8	Store-category	Unit sales of all products, aggregated for each store and category	0.484	0.542	0.480
9	Store-department	Unit sales of all products, aggregated for each store and department	0.578	0.616	0.573
10	Product	Unit sales of a product, aggregated for all stores/states	0.984	0.952	0.966
11	State-product	Unit sales of a product, aggregated for each state	0.941	0.922	0.929
12	Store-product	Unit sales of a product, aggregated for each store	0.893	0.880	0.884
Total			**0.525**	**0.592**	**0.520**

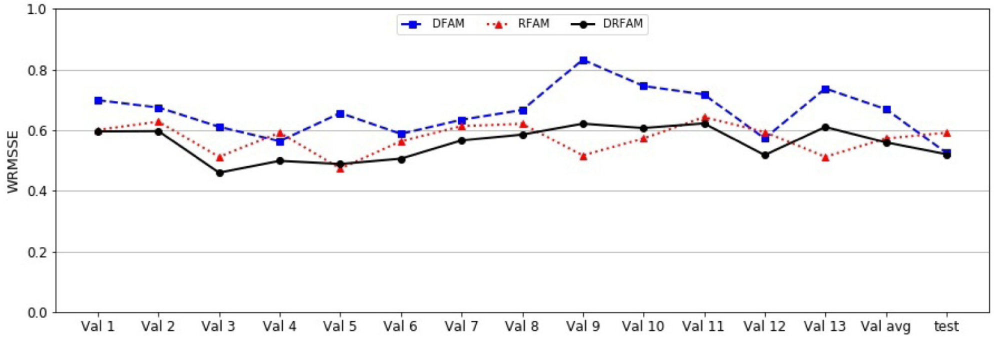


Fig. 4. Forecast errors of three averaging models for validation and test sets.

of the models because of the large number of base forecasting models (220 base models). The hyperparameter setting values of two types of base forecasting models, DFAM and RFAM, are presented in Appendix A.

The performance comparison results of the three models can be summarized as follows. First, the errors of RFAM (red line) are smaller than those of DFAM (blue line) in Fig. 4. Then, the standard deviation of errors of RFAM for the 13 validation is also smaller than that of DFAM. This means that the recursively forecasted sales features, which were used only for RFAM, could allow notably lower forecasting errors. Second, DRFAM (black line) shows the smallest WRMSSE on average, and its variation is similar to that of RFAM. Specifically, DRFAM always defeats DFAM, and it defeats RFAM 9 times among 13 validation sets. It can be said that the combination of DFAM and RFAM could complement each other by relieving their errors. Hence, the complementary effect of DRFAM is also shown in forecasting the test set (last points in the graph), and finally, the forecast result of DRFAM achieved the best performance among the participants at the M5 Accuracy competition.

4.2. Forecast errors at different levels

In this subsection, the forecast results at different levels are investigated. Fig. 5 shows the forecast errors of three averaging models at twelve levels for 13 validation sets and the test sets. Three averaging models have similar patterns at different levels, but the performances of DFAM and RFAM were reversed between the validation set and the test set. In Fig. 5(a), points are the averaged error values for validation sets at each level. For the validation set, the errors of DFAM are larger than those of RFAM and DRFAM, and the errors of DRFAM are a little smaller than those of RFAM. But, as shown in Fig. 5(b), for the test set, RFAM becomes the worst among the three, and DFAM has a similar performance to DRFAM. In conclusion, DRFAM could achieve the best and robust performance in both validation and test sets. It can be said that RFAM has good performance in many cases such as validation sets, but it sometimes results in large errors in variable cases such as the test set because RFAM utilizes recursively forecasted features. However, DRFAM could relieve the errors by complementing RFAM with DFAM, which uses only the known features, as shown in Fig. 5(b).

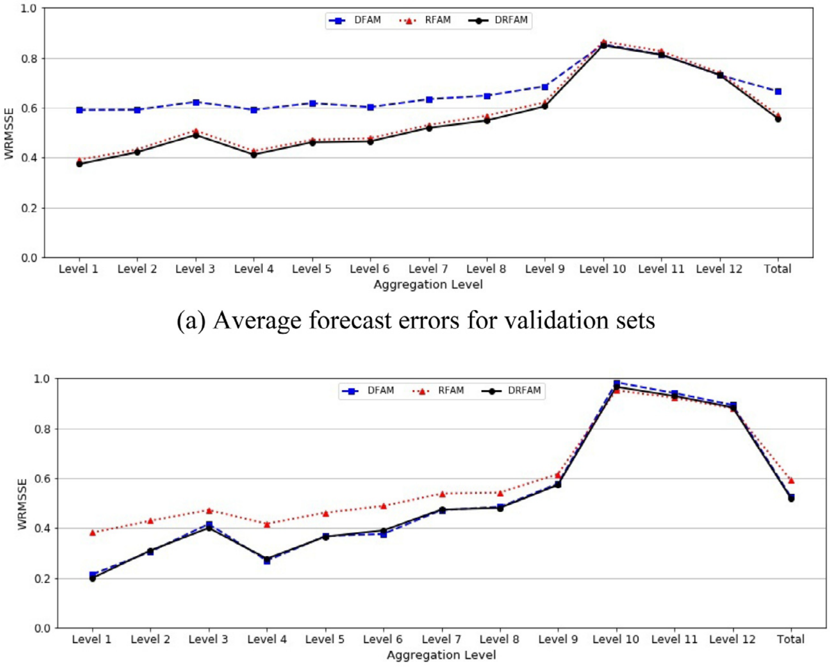
In the meantime, WRMSSE values at higher levels were roughly smaller than those at lower levels, as presented in Table 3. In particular, in case that two levels have the subdivision relationship, the error of a sub-group level is always larger than that of its super-group level. For example, three states at level 2 can be subdivided into ten stores at level 3 in this dataset (see Fig. 2), and therefore $WRMSSE_2$ is smaller than $WRMSSE_3$. Three categories at level 4 can be also subdivided into seven departments at level 5, and so $WRMSSE_4$ is smaller than $WRMSSE_5$. In the same way, the increase of WRMSSE at the subdivided level can be listed as follows:

Table 4
Forecast errors for test set at the levels of state, store, category, and department.

	State	Store	w_k	$RMSSE_k$	$WRMSSE_l$	Category	Department	w_k	$RMSSE_k$	$WRMSSE_l$
DFAM	CA		**0.441**	**0.231**		Hobbies		**0.129**	**0.491**	
		CA_1	0.111	0.324			Hobbies_1	0.122	0.469	
		CA_2	0.113	0.399			Hobbies_2	0.007	0.818	
		CA_3	0.151	0.446						
		CA_4	0.066	0.526						
	TX		**0.276**	**0.474**		Households		**0.298**	**0.268**	
		TX_1	0.079	0.470			Households_1	0.223	0.245	
		TX_2	0.096	0.489			Households_2	0.075	0.536	
		TX_3	0.101	0.560						
	WI		**0.283**	**0.256**		Foods		**0.573**	**0.219**	
		WI_1	0.087	0.316			Foods_1	0.073	0.454	
		WI_2	0.113	0.284			Foods_2	0.147	0.509	
		WI_3	0.083	0.395			Foods_3	0.354	0.286	
	State (level 2)				**0.305**	Category (level 4)				**0.269**
		Store (level 3)			**0.416**		Department (level 5)			**0.367**
	State	Store	Weight	RMSSE	WRMSSE	Category	Department	Weight	RMSSE	WRMSSE
RFAM	CA		**0.441**	**0.422**		Hobbies		**0.129**	**0.352**	
		CA_1	0.111	0.469			Hobbies_1	0.122	0.353	
		CA_2	0.113	0.308			Hobbies_2	0.007	0.702	
	C	A_3	0.151	0.614						
		CA_4	0.066	0.461						
	TX		**0.276**	**0.505**		Households		**0.298**	**0.311**	
		TX_1	0.079	0.317			Households_1	0.223	0.352	
		TX_2	0.096	0.574			Households_2	0.075	0.303	
		TX_3	0.101	0.656						
	WI		**0.283**	**0.369**		Foods		**0.573**	**0.487**	
		WI_1	0.087	0.411			Foods_1	0.073	0.469	
		WI_2	0.113	0.492			Foods_2	0.147	0.632	
		WI_3	0.083	0.297			Foods_3	0.354	0.523	
	State (level 2)				**0.429**	Category (level 4)				**0.417**
		Store (level 3)			**0.472**		Department (level 5)			**0.461**
	State	Store	Weight	RMSSE	WRMSSE	Category	Department	Weight	RMSSE	WRMSSE
DRFAM	CA		**0.441**	0.262		Hobbies		**0.129**	0.442	
		CA_1	0.111	0.336			Hobbies_1	0.122	0.426	
		CA_2	0.113	0.306			Hobbies_2	0.007	0.767	
		CA_3	0.151	0.482						
		CA_4	0.066	0.446						
	TX		**0.276**	0.452		Households		**0.298**	0.249	
		TX_1	0.079	0.380			Households_1	0.223	0.258	
		TX_2	0.096	0.521			Households_2	0.075	0.444	
		TX_3	0.101	0.539						
		WI	**0.283**	0.246		Foods		**0.573**	0.254	
		WI_1	0.087	0.345			Foods_1	0.073	0.440	
		WI_2	0.113	0.290			Foods_2	0.147	0.535	
		WI_3	0.083	0.340			Foods_3	0.354	0.301	
	State (level 2)				0.310	Category (level 4)				0.277
		Store (level 3)			0.400		Department (level 5)			0.365

- Regional levels: $WRMSSE_1$ (total) < $WRMSSE_2$ (3 states) < $WRMSSE_3$ (10 stores)
- Categorical levels: $WRMSSE_1$ (total) < $WRMSSE_4$ (3 categories) < $WRMSSE_5$ (7 departments) < $WRMSSE_{10}$ (3,049 products)
- State levels: $WRMSSE_2$ (3 states) < $WRMSSE_6$ (3 states × 3 categories) < $WRMSSE_7$ (3 states × 7 departments) < $WRMSSE_{11}$ (3 states × 3,049 products)
- Store levels: $WRMSSE_3$ (10 stores) < $WRMSSE_8$ (10 stores × 3 categories) < $WRMSSE_9$ (10 stores × 7 departments) < $WRMSSE_{12}$ (10 stores × 3,049 products)
- Product levels: $WRMSSE_{10}$ (3,049 products) < $WRMSSE_{11}$ (3,049 products × 3 states) < $WRMSSE_{12}$ (3,049 products × 10 stores).

It is expected that this trend was caused by a lack of data at lower levels. It seems similar to the relationship between dimensionality and data size. Generally, more data are needed when the dimension of the data increases. In a similar way, it is thought that in this competition the amount of given data at lower levels is not enough, the errors are slightly larger. However, the size of data pools at higher levels is larger, and the errors at higher levels were mitigated.



(a) Average forecast errors for validation sets

(b) Forecast errors for test data

Fig. 5. Forecast errors of three averaging models at different levels.

4.3. Forecast averaging results

Table 4 describes the forecast errors of three averaging models at different levels such as state (level 2), store (level 3), category (level 4), and department (level 5). The $WRMSSE_l$ is calculated with the weighted sum of $RMSSE_k$ values with weight w_k. For example, the $WRMSSE_2$ of DFAM at the state level (level 2) is the weighted sum of three $RMSSE_k$ values of DFAM in CA, TX, and WI ($k = 1$ to 3), and the $WRMSSE_3$ of DFAM at the store level (level 3) is the weighted sum of ten $RMSSE_k$ values of DFAM at 10 stores ($k = 1$ to 10). It is shown that DRFAM can reduce the forecast errors by simply averaging DFAM and RFAM at most levels such as state, store, category, and department.

The actual sales and the forecast results of DRFAM for test data are depicted in Fig. 6. The actual and forecasted sales values according to state, category, and department are compared in the graphs of Fig. 6(a), (b), and (c), respectively.

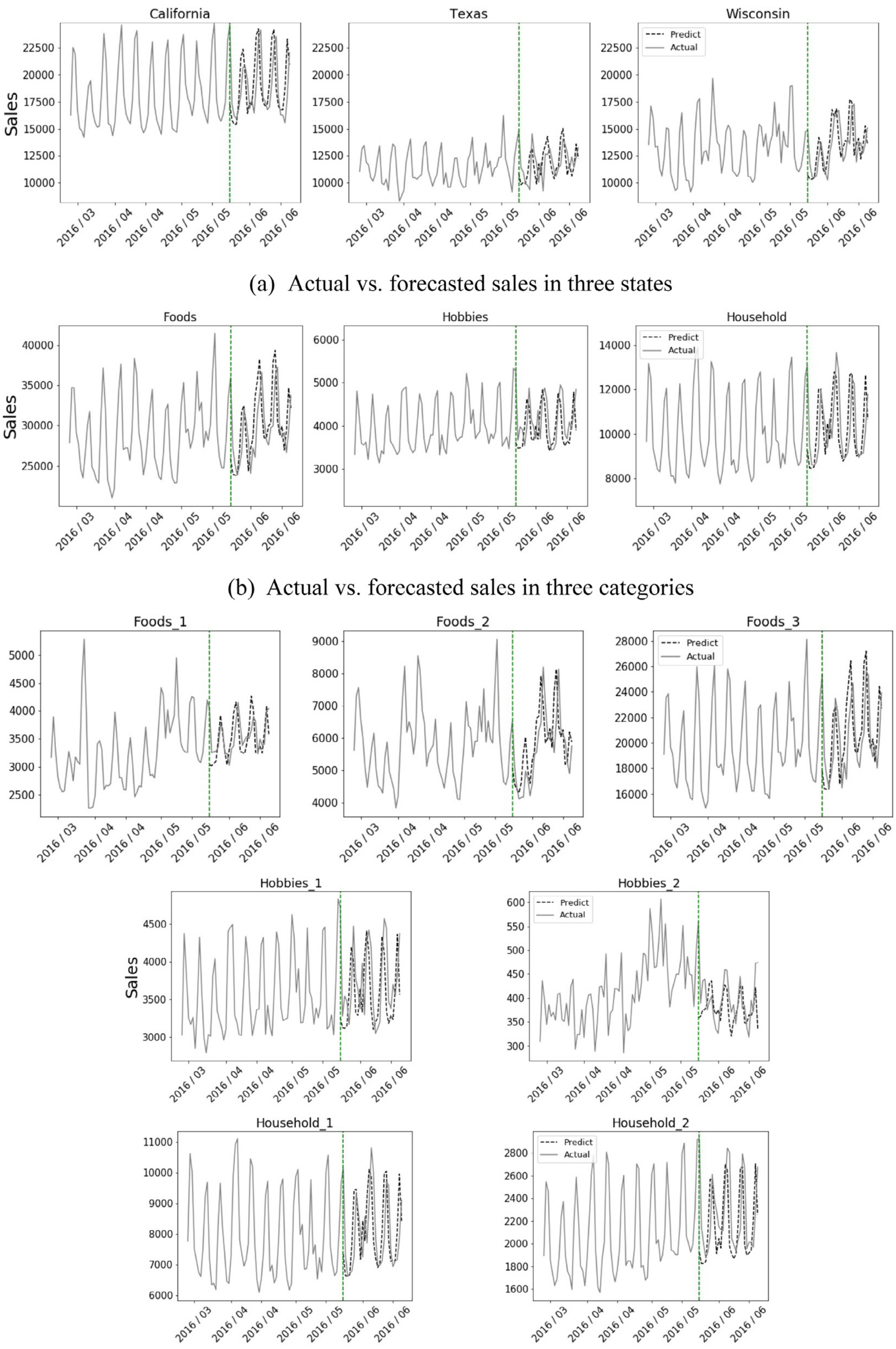
4.4. Feature importance of forecast features

To understand the resulted forecast averaging models, we investigated the feature importance (FI) of the trained DFAM, RFAM, and DRFAM models. The FI score was calculated based on the split gain of LightGBM (Ke et al., 2017); the feature with higher gain plays a more important role in the prediction. In our averaging models, since all base models are evenly contributed to the final forecast, the percentage FIs of the base LightGBM models are averaged to calculate the feature importance in the averaging models as follows:

$$pFI_k^{AM} = \frac{1}{B}\sum_{b=1}^{B} pFI_k^b = \frac{1}{B}\sum_{b=1}^{B}\frac{FI_k^b}{\sum_j FI_j^b},$$

where pFI_k^{AM} is the percentage feature importance of the kth feature in the averaging model, and pFI_k^b and FI_k^b are the percentage feature importance and the split-gain-based feature importance of the kth feature in the b-th base model, respectively. B is the number of the base models that were used in the averaging model.

The important features of top-10 features of DRFAM are listed in Table 4 along with their percentage FIs in three averaging models. prod_id is the most important feature in DRFAM, and it plays an important role in all the three averaging models. The important features except for prod_id are the mean values of the product sales in specific periods in the past; in other words, the forecast results rely largely on the past average sales than on external information. Note that sales_mean_at_store is the mean value of the product sales at the store in the whole period of the training data. The important periods of the sales mean related features are different between DFAM and RFAM; in DRFAM, the mean sales value over longer periods (i.e., 30 days and 60 days) before $t - 28$ were more important, while in RFAM, the mean sales value over shorter periods (i.e., 14 days and 7 days) before $t - 1$ were more important. Meanwhile, it can be said that DRFAM finally leverages the mean sales values of various past periods to forecast the future product sales at a store. For example, the top-5 important features of DRFAM include the average sales for past 30 days from $t - 30$, and the



(a) Actual vs. forecasted sales in three states

(b) Actual vs. forecasted sales in three categories

(c) Actual vs. forecasted sales in seven departments

Fig. 6. Actual sales and the forecasted values of DRFAM for test set.

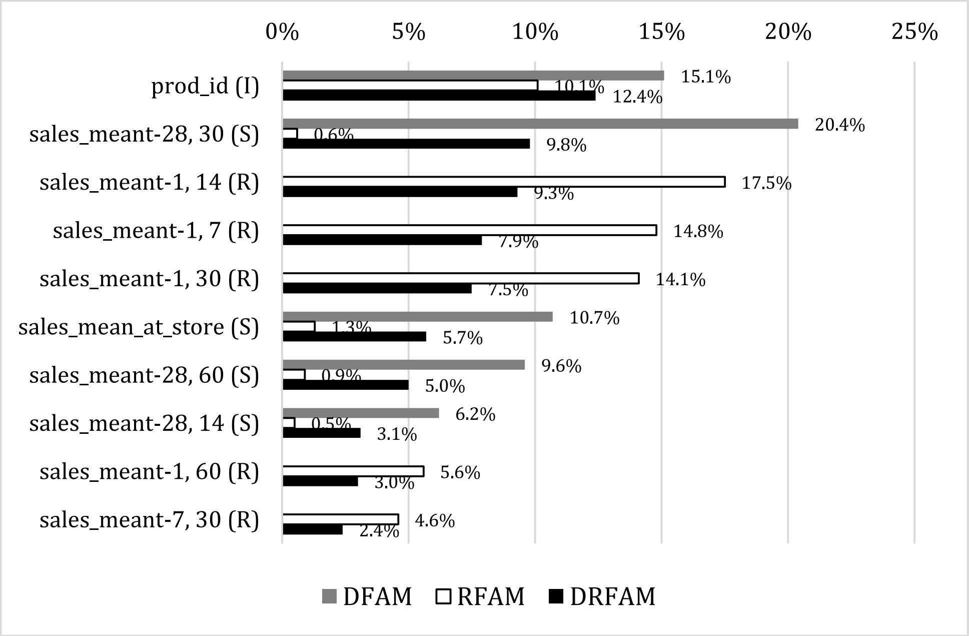


Fig. 7. Percentage feature importance of major features. A letter in the parenthesis indicates the feature category such as identifier (I), sales (S), and recursive sales (R).

average sales for past 14 days, 7 days, and 30 days from $t-1$.

5. Conclusions and future work

In this article, we introduced the winning method at the M5 Accuracy competition. To build base forecasting models for product sales at the stores, partial pools at three different levels were prepared. The direct forecasting features and recursive forecasting features were then extracted from each pool, and single-model direct forecasting and recursive forecasting models were trained by a machine learning technique, LightGBM. Finally, the whole averaging model that could obtain the complementary effects of multiple base forecasting models was submitted to the competition. In addition, the important features of the averaging models were investigated, and the use of both direct and recursive forecasting models was found to be useful.

An advanced machine learning algorithm, LightGBM, was used for the base forecasting models in this research. However, it is expected that other advanced ensemble algorithms such as XGBoost or random forest will also show similarly good accuracy. The reason for choosing the LightGBM is that it is faster to train than other advanced machine learning algorithms. We believe that feature engineering is more important in forecasting accuracy if any of the advanced machine learning algorithms are chosen in most practical data analysis problems. In other words, more data and information are crucial, and they can be obtained through good feature engineering.

In this research, two major tasks were conducted for feature extraction. One is partial pooling from the different levels of the dataset, and the other is the combination of direct forecasting features for DFAM and recursive forecasting features for RFAM. As shown in Fig. 7, both direct and recursive forecasting features play an important role in the best model, DRFAM. It is expected that the two feature extraction tasks could improve the forecasting accuracy of the proposed method, DRFAM, in this challenge of the hierarchical forecasting problem.

This study has some limitations that can be overcome. First, the possible combinations of partial pools can be exhaustively investigated to increase the performance of the proposed method. Even if we considered only the partial pools including a store level in this research, it may improve the forecasting accuracy to investigate additional partial pools such as the department level or the combinations of states and categories. Second, hyperparameters of LightGBM can be optimized to obtain better performance of base forecasting models. The 220 LightGBM models used in this research were not optimized because of time limitation. Finally, better model selection and averaging methods of base forecasting models can be found than evenly averaging. More appropriate weights could have been used for combining the direct and recursive forecasting models.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Hyperparameter setting

See Tables A.1 and A.2.

Table A.1
Hyperparameter setting of the single-model direct forecasting models.

	Store pools	Store-category pools	Store-department pools
boosting_type	gbdt	gbdt	gbdt
objective	tweedie	tweedie	tweedie
tweedie_variance_power	1.1	1.1	1.1
subsample	0.5	0.5	0.5
subsample_freq	1	1	1
learning_rate	0.015	0.015	0.015
num_leaves	2^8-1	2^8-1	2^8-1
min_data_in_leaf	2^8-1	2^8-1	2^8-1
feature_fraction	0.5	0.5	0.5
max_bin	100	100	100
n_estimators	3000	3000	3000
boost_from_average	False	False	False

Table A.2
Hyperparameter setting of the recursive forecasting models.

	Store pools	Store-category pools	Store-department pools
boosting_type	gbdt	gbdt	gbdt
objective	tweedie	tweedie	tweedie
tweedie_variance_power	1.1	1.1	1.1
subsample	0.5	0.5	0.5
subsample_freq	1	1	1
learning_rate	0.015	0.015	0.015
num_leaves	$2^{11}-1$	2^8-1	2^8-1
min_data_in_leaf	$2^{12}-1$	2^8-1	2^8-1
feature_fraction	0.5	0.5	0.5
max_bin	100	100	100
n_estimators	3000	3000	3000
boost_from_average	False	False	False

Appendix B. Forecasting errors of DRFAM

See Table B.1.

Table B.1
Forecasting errors of three averaging models for validation and test sets. The values are also depicted in Fig. 4.

Dataset	WRMSSE		
	DFAM	RFAM	DRFAM
Val 1	0.699	0.600	**0.596**
Val 2	0.674	0.628	**0.597**
Val 3	0.611	0.512	**0.460**
Val 4	0.564	0.592	**0.499**
Val 5	0.656	**0.473**	0.488
Val 6	0.588	0.563	**0.506**
Val 7	0.634	0.613	**0.566**
Val 8	0.666	0.621	**0.585**
Val 9	0.832	**0.516**	0.621
Val 10	0.745	**0.572**	0.607
Val 11	0.718	0.643	**0.622**
Val 12	0.573	0.594	**0.518**
Val 13	0.737	**0.512**	0.610
Val avg	0.669	0.572	**0.560**
Test	0.525	0.592	**0.520**

References

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

Bates, J. M., & Granger, C. W. (1969). The combination of forecasts. *Journal of the Operational Research Society*, *20*(4), 451–468.

Bergmeir, C., Hyndman, R. J., & Koo, B. (2018). A note on the validity of cross-validation for evaluating autoregressive time series prediction. *Computational Statistics & Data Analysis*, *120*, 70–83.

Bojer, C., & Meldgaard, J. P. (2020). The M5: A preview from prior competitions. *Foresight*, *58*, 17–23.

Cheng, H., Tan, P. N., Gao, J., & Scripps, J. (2006). Multistep-ahead time series prediction. In *Proceedings of Pacific-Asia Conference on Knowledge Discovery and Data Mining* (pp. 765–774). Berlin, Heidelberg: Springer.

Chevillon, G. (2007). Direct multi-step estimation and forecasting. *Journal of Economic Surveys*, *21*(4), 746–785.

Clemen, R. T., & Winkler, R. L. (1986). Combining economic forecasts. *Journal of Business & Economic Statistics*, *4*(1), 39–46.

Fletcher, D. (2018). *Model Averaging*. Berlin, Heidelberg: Springer.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *The Annals of Statistics*, *118*, 9–1232.

Gelman, A., & Hill, J. (2006). *Data Analysis using Regression and Multilevel/Hierarchical Models*. Cambridge university press.

Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. *Machine Learning*, *63*(1), 3–42.

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

Ho, T. K. (1998). The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *20*(8), 832–844.

Hoeting, J. A., Madigan, D., Raftery, A. E., & Volinsky, C. T. (1999). Bayesian model averaging: a tutorial with comments by M. Clyde, David Draper and EI George, and a rejoinder by the authors. *Statistical Science*, *14*(4), 382–417.

Hyndman, R. J. (2020). A brief history of forecasting competitions. *International Journal of Forecasting*, *36*(1), 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*(9), 2579–2589.

Hyndman, R. J., & Athanasopoulos, G. (2018). *Forecasting: Principles and Practice*. OTexts.

Hyndman, R. J., & Koehler, A. B. (2006). Another look at measures of forecast accuracy. *International Journal of Forecasting*, *22*(4), 679–688.

Kaggle (2020). Kaggle web site of M5 forecasting accuracy competition. https://www.kaggle.com/c/m5-forecasting-accuracy/.

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 *Advances in neural information processing systems 30* (pp. 3146–3154).

Liang, H., Zou, G., Wan, A. T., & Zhang, X. (2011). Optimal weight choice for frequentist model average estimators. *Journal of the American Statistical Association*, *106*(495), 1053–1066.

Liu, C. A., & Kuo, B. S. (2016). Model averaging in predictive regressions. *The Econometrics Journal*, *19*(2), 203–231.

M5 (2021). The github of the M5 competition data. https://github.com/Mcompetitions/M5-methods/tree/master/validation/.

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2021). The M5 accuracy competition: Results, findings and conclusions. *International Journal of Forecasting*, (in press).

Makridakis, S., & Winkler, R. L. (1983). Averages of forecasts: Some empirical results. *Management Science*, *29*(9), 987–996.

Marcellino, M., Stock, J. H., & Watson, M. W. (2006). A comparison of direct and iterated multistep AR methods for forecasting macroeconomic time series. *Journal of Econometrics*, *135*(1–2), 499–526.

Microsoft (2021). LightGBM documentation site. https://lightgbm.readthedocs.io/en/latest/.

MOFC (2020). M5 competition guide web site. https://mofc.unic.ac.cy/m5-guidelines/.

Montero-Manso, P., Athanasopoulos, G., Hyndman, R. J., & Talagala, T. S. (2020). FFORMA: Feature-based forecast model averaging. *International Journal of Forecasting*, *36*(1), 86–92.

Pawlikowski, M., & Chorowska, A. (2020). Weighted ensemble of statistical models. *International Journal of Forecasting*, *36*(1), 93–97.

Raftery, A. E., Gneiting, T., Balabdaoui, F., & Polakowski, M. (2005). Using Bayesian model averaging to calibrate forecast ensembles. *Monthly Weather Review*, *133*(5), 1155–1174.

Spiliotis, E., Abolghasemi, M., Hyndman, R. J., Petropoulos, F., & Assimakopoulos, V. (2020). Hierarchical forecast reconciliation with machine learning. arXiv preprint arXiv:2006.02043.

Steel, M. F. (2020). Model averaging and its use in economics. *Journal of Economic Literature*, *58*(3), 644–719.

Taieb, S. B., & Hyndman, R. J. (2012). *Recursive and direct multi-step forecasting: The best of both worlds, Vol. 19*. Department of Econometrics and Business Statistics, Monash University.

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