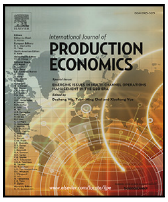
Int. J. Production Economics 279 (2025) 109449

Contents lists available at ScienceDirect

International Journal of Production Economics

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



Scalable probabilistic forecasting in retail with gradient boosted trees: A practitioner's approach

Xueying Long [a], Quang Bui [a], Grady Oktavian [b], Daniel F. Schmidt [a], Christoph Bergmeir [a,c,*], Rakshitha Godahewa [a], Seong Per Lee [d], Kaifeng Zhao [d], Paul Condylis [d]

[a] *Department of Data Science and Artificial Intelligence, Monash University, Australia*
[b] *Data Science, Tokopedia, Indonesia*
[c] *Department of Computer Science and Artificial Intelligence, University of Granada, Spain*
[d] *Data Science, Tokopedia, Singapore*

ARTICLE INFO

Keywords:
Probabilistic forecasting
Gradient boosted trees
Global models
Disaggregation

ABSTRACT

The recent M5 competition has advanced the state-of-the-art in retail forecasting. However, there are important differences between the competition challenge and the challenges we face in a large e-commerce company. The datasets in our scenario are larger (hundreds of thousands of time series), and e-commerce can afford to have a larger stock assortment than brick-and-mortar retailers, leading to more intermittent data. To scale to larger dataset sizes with feasible computational effort, we investigate a two-layer hierarchy, namely the decision level with product unit sales and an aggregated level, e.g., through warehouse-product aggregation, reducing the number of series and degree of intermittency. We propose a top-down approach to forecasting at the aggregated level, and then disaggregate to obtain decision-level forecasts. Probabilistic forecasts are generated under distributional assumptions. The proposed scalable method is evaluated on both a large proprietary dataset, as well as the publicly available Corporación Favorita and M5 datasets. We are able to show the differences in characteristics of the e-commerce and brick-and-mortar retail datasets. Notably, our top-down forecasting framework enters the top 50 of the original M5 competition, even with models trained at a higher level under a much simpler setting.

1. Introduction

Forecasting plays an important role in decision-making processes. In the retail industry, accurate sales forecasting is crucial for different phases such as supply chain management (Fildes et al., 2022a,b) and inventory control (Kourentzes et al., 2020). Probabilistic forecasts, which quantify uncertainty about the future, are often essential in these cases, e.g., for determining the stock level and reorder points (do Rego and De Mesquita, 2015). However, effective uncertainty estimation is a challenging problem due to the fact that the series are often intermittent, i.e., a large percentage of entries are zero.

The recent M5 competition (Makridakis et al., 2021, 2022b) established the state of the art of retail forecasting, through both an accuracy track, which focused on point forecasting, and an uncertainty track, which focused on probabilistic forecasting. Many of the M5 findings are applicable to our situation; however, we observe that our use cases, drawn from a large Indonesian e-commerce retail company, exhibit some important difference from the challenges posed in the M5 competition. The two biggest differences we have identified are that the datasets in our application are often significantly larger and more intermittent than the datasets provided by the M5 competition. While the M5 has less than 50,000 time series, over half a million different types of products are purchased on the e-platform each day. Furthermore, the M5 data is derived from traditional brick-and-mortar retail situations, which have some important differences to the e-commerce setting; most notably, e-commerce platforms can typically afford to have a larger assortment of products available, and that many of these products may have slow sales. This leads to a higher proportion of intermittent series, and thus a high level of overall intermittency in the data. In addition to handling the challenges presented by these differences, our aim is to develop an approach that is ready for production use, and as such involves additional constraints regarding robustness and execution time that were not an element of the M5 competition. It is important to mention that while promotions are often key drivers in retail forecasting, they are not a main consideration in the M5, as the data in this competition was taken from Walmart, which utilises an everyday low price strategy. They are also not relevant in our

[*] Correspondence to: D3, UGR AI, Av. del Conocimiento, 37, 18016 Granada, Spain.
E-mail address: bergmeir@ugr.es (C. Bergmeir).

https://doi.org/10.1016/j.ijpe.2024.109449
Received 27 February 2024; Received in revised form 24 October 2024; Accepted 25 October 2024
Available online 5 November 2024

business use cases, so we do not consider them when developing the methodology in this paper.

Consequently, the main aim of our work is to adapt the best-performing M5 methodologies to the problem of forecasting in an e-commerce setting. The M5 is dominated by global models (Januschowski et al., 2020), which are learned across series. This has important consequences for scalability, as global models cannot be fitted in parallel as trivially as local models, which are embarrassingly parallelisable along the series dimension. As such we require our methods to scale to datasets that are at least an order of magnitude larger than the M5 data. There are three immediate strategies to handle this:

Model simplification An obvious option is to try and use simpler models. However, by itself this does not guarantee the ability to train models with a feasible computational effort, and the resulting forecast accuracy may be poor due to the model simplification.

Data partitioning Data partitioning is an intuitive way of scaling global models. The global models are trained not in a truly "global" way, i.e., across all available series, but several such models are trained on subsets of the data. This is a popular processing step, and most competitors in the M5 subdivide the data in one way or another. One of the earliest papers proposing this procedure that we are aware of is (Bandara et al., 2020), and later this idea is studied more systematically in Godahewa et al. (2021). However, subdividing the data is mainly done with the aim of improving accuracy, and cannot be seen as a step with the primary purpose of achieving scalability.

Training with less data Another option is to train with less data. One may simply omit part of the historical data and fit a model to a subsample. Additionally, if data has a suitable hierarchical structure, we can train models at a higher level of the hierarchy using substantially less series (with consequent reduction in intermittency), and then apply a top–down disaggregation strategy to obtain forecasts at lower levels.

Regardless of the strategy chosen, the forecasting must be done in a probabilistic manner. This usually involves modelling either via parametric distribution assumption, or some more flexible non-parametric approach. Using the quantile loss function (Koenker and Bassett, 1978), probabilistic forecasts can be generated without assumptions. However, a drawback of this approach is that separate models must be trained for each quantile of interest, which can make the process expensive when handling large datasets. Additionally, quantile crossing (Bassett and Koenker, 1982; He, 1997) can happen as a consequence of training quantiles separately, adding another layer of complexity. Compromises may also need to be made to ensure a feasible implementation; for example, having to train with reduced sample sizes. In contrast, parametric methods based on distributional assumptions (Snyder et al., 2012) are relatively straightforward to implement and apply in practice. They are faster and scale more readily to large datasets in comparison with non-parametric methods. More importantly, classical choices such as a Poisson or negative binomial distribution have useful mathematical properties (Steutel and Van Harn, 2003) that can be leveraged when scaling to large datasets.

In the M5 competition tree-based methods were very successful, and most top competitors based their solutions on LightGBM (Ke et al., 2017), a highly efficient gradient boosted tree (GBT) algorithm. For example, the winning method in the accuracy track leveraged LightGBM by training on grouped data from multiple categories and combining the forecasts with equal weights (Makridakis et al., 2022b). Tree-based implementations such as LightGBM and XGBoost (Chen and Guestrin, 2016) are open source and highly flexible tools. As LightGBM offers fast training while maintaining predictive accuracy, it is generally considered a superior solution to other implementations of GBTs that yield lower accuracy with longer training times.

In this paper, we propose an efficient way of generating accurate and scalable forecasting systems. We make the most of a two-layer hierarchy of raw and aggregated data, and develop a top–down forecasting framework that is able to scalably predict with small computational effort while maintaining competitive accuracy. Instead of directly dealing with data on the decision level, we forecast with the aggregated series and disaggregate back in a top–down fashion according to historical proportions. Our forecasting framework is capable of generating accurate probabilistic forecasts with simple assumptions of distributions. The proposed approach is analysed on a proprietary e-commerce dataset, as well as the public Corporación Favorita dataset and the M5 competition dataset. As a notable side-product of this research, we have implemented a negative binomial loss function for LightGBM (Ke et al., 2017), for which the details are given in Appendix.

The rest of this paper is organised as follows. Section 2 reviews the related work. Section 3 provides a comprehensive description of the proposed top–down forecasting framework. Section 4 explains the experimental setup. Section 5 reports the results and provides a further discussion. Section 6 concludes our work.

2. Related work

In this section, we cover relevant prior work; specifically, global, hierarchical, probabilistic modelling strategies and intermittent forecasting.

2.1. Modelling across series with global models

Global modelling (Januschowski et al., 2020) has received substantial recent attention in the forecasting community. All top contenders in the M5 were global models, and even before this, global models have shown strong performance in various Kaggle competitions (Bojer and Meldgaard, 2021). Under the global modelling paradigm, available time series are pooled together and a single model is built across them, with shared parameters. As a global model is trained with more data, it can afford to be more complex, compared with traditional local per-series models in which each time series is viewed as a distinct dataset, and models are built for each series separately. Montero-Manso and Hyndman (2021) present some theoretical explanations for the superiority of global models over local models, and argue that no similarity or relatedness between series is necessary for global models to work well. Hewamalage et al. (2022) confirm these findings empirically and make them more nuanced in a simulation study. They argue that minimal assumptions on relation between time series are necessary as global models have the capacity of learning complex patterns and perform well even when the series are heterogeneous. One of the earliest and most prominent global models in the literature is DeepAR (Salinas et al., 2020), which is a global forecasting method based on autoregressive neural networks. It has demonstrated high forecasting accuracy for Amazon sales data, and can be considered a standard benchmark in retail forecasting. Other modelling choices for implementation can involve classical linear models, standard machine learning models such as LightGBM (Januschowski et al., 2021), and neural networks (Kunz et al., 2023). Consequently, we focus in our work on global models, as prior research has established their general superiority over local models in retail settings similar to the one under consideration in this work.

2.2. Hierarchical forecasting

Retail sales data is naturally organised in a hierarchal fashion, i.e., per-store product sales data at the bottom level can be combined according to product categories and regions. Typically, hierarchical forecasting is concerned with producing coherent forecasts across different levels of the hierarchy (for different decisions to be made, such as strategical, tactical, or operational decisions). Additionally, hierarchical forecasting methods have been used in the past to transport information between series, such as bringing seasonal patterns only emerging at higher levels of the hierarchy into the noisy bottom-level series forecasting. Classical approaches of hierarchical forecasting in the literature are top–down, bottom–up and middle-out methods (Hyndman et al., 2011), in which forecasts are produced on only a single level of the hierarchy and then aggregated up or disaggregated down, using historical (or through other ways obtained) proportions. More sophisticated alternatives include optimal reconciliation approaches (Hyndman et al., 2011), in which all series in the hierarchy are forecasted, and then a subsequent step a reconciliation (optimisation) is performed to adjust the forecasts and make them coherent. The most recent methods combine forecasting and reconciliation into a single step, building global models that are able to produce reconciled forecasts directly. The most prominent methods in this space are HierE2E (Rangapuram et al., 2021), SHARQ (Han et al., 2021), HIRED (Paria et al., 2021), and PROFHIT (Kamarthi et al., 2022).

On the other hand, probabilistic hierarchical forecasting is a much more challenging problem as it requires, in theory, the distribution of the forecasts of the aggregated series being the same as the distribution of sum of the forecasts of its children series. This is difficult to achieve, for example, quantile forecasts produced at a certain level cannot be simply added together, or divided up, to derive forecasts on other levels. In contrast, point forecasts can be straightforwardly generated based on the summation constraint of the hierarchy. In the literature, different definitions on the coherence of probabilistic hierarchical forecasting have been provided. Taieb et al. (2017, 2020) define probabilistic coherence from the perspective of convolution of marginal predictive distributions of the children series. Panagiotelis et al. (2022) propose a more intuitive definition where densities of children series should lie on a coherent subspace, and a similar notation can be found in Rangapuram et al. (2021). Han et al. (2021) explore the coherence of quantiles with a regularised quantile loss function. Kamarthi et al. (2022) propose a distributional coherency regularisation to ensure the distributional consistency of the entire hierarchy.

Our motivation for using a hierarchy differs from the usual use cases. We do not use the hierarchical structure from the perspective of reconciliation, and are not particularly interested in coherent forecasts for the entire hierarchy. Instead, we leverage the hierarchy as a way to scale the forecasts from more aggregated levels in the hierarchy, where fewer time series exist, to lower levels where the amount of series and their intermittency hinder traditional forecasting techniques. Thus, the sophisticated methods from the literature are not directly applicable to our use case. We are interested in generating probabilistic forecasts in our application; however, as noted previously, quantile forecasts cannot be directly used to produce forecasts at other levels. This motivates us to explore distributional assumptions and properties that could potentially make the problem tractable. These are discussed in the next section.

2.3. Probabilistic forecasting for intermittent data

We categorise the existing probabilistic forecasting approaches into two main parts: non-parametric methods such as quantile regression and bootstrapping, and parametric methods under some distributional assumptions. A particularly flexible non-parametric technique is quantile regression. By utilising the pinball loss, quantile forecasts can be directly generated, and implementations are available in most open-source GBT frameworks. In this case, the modelling and training process needs to be repeated for each quantile of interest. For intermittent data, Lainder and Wolfinger (2022) propose a quantile forecasting method using LightGBM and data augmentation techniques; this technique achieved first place in the M5 uncertainty track. Bootstrapping has been utilised to solve intermittent forecasting problems (Willemain et al., 2004; Viswanathan and Zhou, 2008; Zhou and Viswanathan, 2011; Hasni et al., 2019) with some highlights in forecast accuracy, but it requires an access to a large amount of historical data and potentially huge computational costs, both of which pose questions regarding plausibility in real-life problem settings (Syntetos et al., 2015). Using empirical in-sample quantiles is an especially simple way to generate probabilistic forecasts and has been found to work well in retail forecasting (Kolassa, 2016; Spiliotis et al., 2021; Kolassa, 2022). We employ this established method as a strong benchmark. Another method to turn point forecasts into probabilistic forecasts is through level set forecasting (Hasson et al., 2021). Level-set forecasting first partitions the training set according to the predicted values obtained from a certain point forecaster. Then, when forecasting, the algorithm picks the closest set based on its point forecast and takes the corresponding true values of that set as distributional forecasts. However, level-set forecasting is a general algorithm that is not specifically designed to deal with the challenges caused by intermittent series.

On the other hand, parametric methods involve understanding, or making assumptions regarding, the characteristics of historical data and the nature of the data generating process. Classical distributional choices for fitting retail data in the literature include the Poisson distribution (Heinen, 2003; Snyder et al., 2012), or the negative binomial distribution (Agrawal and Smith, 1996; Snyder et al., 2012), potentially mixed with zero-inflation (Lambert, 1992) and hurdle models (Cragg, 1971) to accommodate the excess zeros typical in this domain. Based on distributional assumptions, relevant model parameters are learned empirically. Snyder et al. (2012) proposed a hurdle shifted Poisson model and introduced a dynamic state-space structure for both damped and undamped versions. de Rezende et al. (2021) extended this structure to the negative binomial distribution, and this technique achieved sixth place in the M5 uncertainty competition. Parameter estimation of such state-space models is often performed via maximum likelihood, frequently in conjunction with the expectation maximisation algorithm; these procedures can be computationally intensive. Kolassa (2016) studied a set of parametric methods with Poisson and negative binomial assumptions and applied these methods in a later paper to the M5 data (Kolassa, 2022). They emphasised the consideration of overdispersion in retail forecasting, which is in line with the parametric methods studied in Spiliotis et al. (2021). However, these works only focus on local methods, and did not consider ways of scaling up the forecasting process.

Unlike many machine learning algorithms which are only capable of producing a single output, the generalised additive model (location, shape, scale) (GAMLSS, Stasinopoulos and Rigby, 2007) approach can produce estimates for all relevant parameters of the assumed distribution. Ziel (2021) applied this approach to the M5 dataset with different distribution assumptions, including a zero-inflated Poisson distribution. A major pitfall of GAMLSS is the huge computational cost; to deal with this, models are trained only based on subsamples in that work. DeepAR generates probabilistic forecasts based on distributional assumptions. For example, a negative binomial distribution can be chosen for count data, with both mean and shape parameters produced as the outputs of the neural network. Following the literature, we consider the Poisson distribution and negative binomial distribution, as mixed distributions require extra parameters which can bring with them additional complexity during the modelling process. Moreover, these two distributions are characterised as being infinitely divisible (Steutel and Van Harn, 2003); for example, a Poisson random variable can be expressed as the sum of an arbitrary number of independent Poisson

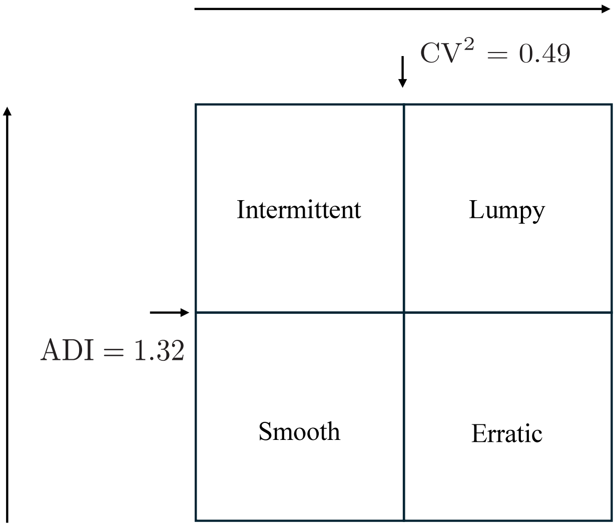


Fig. 1. The demand classification scheme and cutoff values used in this work (Syntetos et al., 2005).

random variables. In this case, we can decompose the aggregated level forecasts and generate probabilistic forecasts based on the distributions for both layers. Olivares et al. (2021) tested Poisson mixtures from a perspective of hierarchical reconciliation while modelling with a deep neural network. In this paper, we also examine negative binomial mixtures, as the addition of a dispersion parameter can introduce more modelling flexibility.

3. Methodology

As outlined earlier, our methodology consists of improvements to the state of the art in retail forecasting, specifically to address issues regarding large amounts of data and intermittency in the training series. In particular, we propose a methodology consisting of the following two components: (1) a data partitioning step commonly used in retail settings (see Section 3.1); and (2) a hierarchical top–down approach to forecasting, in which we forecast the top level series and disaggregate the forecasts to the lower level series (see Section 3.2).

3.1. Demand classification

Following the scheme proposed by Syntetos et al. (2005), we classify the time series into one of four groups: smooth, erratic, lumpy, and intermittent. This is done according to the average demand interval (ADI) and coefficient of variation squared (CV^2) of the series:

$$\mathrm{ADI} = \frac{\text{Days available since first sale}}{\text{Days with sale}}, \tag{1}$$

$$\mathrm{CV}^2 = \left(\frac{\text{Standard deviation of daily sales}}{\text{Mean of daily sales}}\right)^2. \tag{2}$$

Specifically, we dichotomise the ADI and CV^2 values for the series using thresholds of 1.32 and 0.49 respectively, yielding four distinct categories (see Fig. 1). Even though these threshold values are originally proposed as an optimal method for choosing between simple exponential smoothing and a modified Croston's method (Syntetos and Boylan, 2005), methods which we are not using in our work, we employ these threshold values as they are well-established in the literature.

Naturally, there are certain limitations of such a classification scheme. The use of hard cutoffs means that series which are inherently similar, but have ADI and CV^2 values close to the thresholds, may fall into different categories. Furthermore, the classification is usually performed in a one-off manner and thus may not be accurate if there is a shift of characteristics in the series with time. However, these are common problems affecting any type of hard classification, and we argue this type of partitioning is suitable for our work as it is the most established method used in the literature. We also argue that such a partitioning is in fact necessary; this is because the series in the different classes described above have characteristics which make them behave quite differently in terms of forecasting. Smooth and erratic series tend to have larger values on average by definition. If we evaluate all series together, they are likely to dominate the error measure when using scaled metrics. Likewise for a scale-free measure, the intermittent and lumpy series will typically contribute a very large part of the overall error, as their values are generally smaller which makes them more difficult to forecast in relative terms, due to the integer nature of the series. Thus, we perform a demand classification and evaluate using scaled metrics for each group separately.

3.2. Top–down distributional forecasting framework

We can form a two-layer hierarchy by aggregating the series at the decision level, denoted as level L, based on product hierarchy to an aggregated level, denoted as level A. The constructed two-layer hierarchy is illustrated in Fig. 2. At each time point t, a series j at level A, denoted as $A_{t,j}$, can be constructed from the sum of the corresponding n_j series at level L. $L_{t,j,i}$ is used to denote a series i at level L at time t, where j matches the jth series at level A in the hierarchy. Thus the relation

$$A_{t,j} = \sum_{i=1}^{n_j} L_{t,j,i}$$

is always satisfied.

We are interested in producing forecasts at the decision level L. Based on the two-layer hierarchy introduced above we first train global models at level A, a higher level in which the data are less intermittent and the number of series to forecast is feasible. We then disaggregate and produce forecasts recursively for the entire horizon. Any off-the-shelf global forecasting model can be used in this framework to generate point forecasts at level A; that is, the top–down distributional framework is model-agnostic. In this work, we use LightGBM models and linear models. For time point t in the horizon h, we denote the point forecast (conditional mean) at the aggregated level for series j by $\hat{A}_{t,j}$.

The proposed forecasting framework consists of four steps. At each time point in the forecast horizon, we (1) point-forecast the values at the aggregated level A using the predicted conditional means; (2) estimate the parameter(s) of the distributions at the aggregated level; (3) obtain the historical proportion of lower-to-higher level sales, and disaggregate to obtain the lower level L point forecast; and (4) estimate the parameter(s) of the distributions at the lower level. In this section, we start by introducing the distribution properties and then discuss each step in detail.

3.2.1. Distribution properties and forecasting

Poisson forecasts. We assume that sales are realisations of either Poisson or negative binomial random variables. For Poisson distributed sales, $X \sim \mathrm{Poisson}(\lambda)$ with rate parameter λ. Once we have the point forecast (i.e., the estimated conditional mean), the parameter λ can be estimated using the point forecast, as the maximum likelihood estimate of λ is simply the sample mean, i.e., $\hat{\lambda}_{A_{t,j}} = \hat{A}_{t,j}$. We can then produce distributional forecasts according to the probability model

$$A_{t,j} \sim \mathrm{Poisson}\left(\hat{\lambda}_{A_{t,j}}\right).$$

in the usual fashion.

Negative-binomial forecasts. Consider a random variable $X \mid \lambda \sim \mathrm{Poisson}(\lambda)$ that conditionally follows a Poisson distribution, and let λ be a Gamma distributed random variable, i.e.,

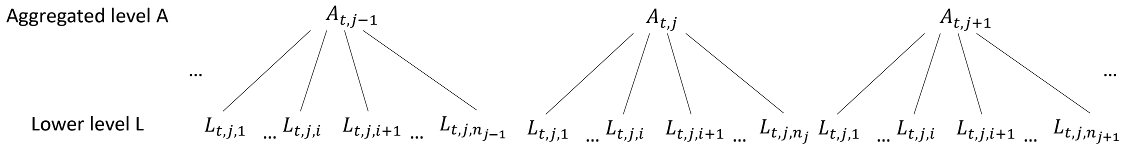

Aggregated level A
$A_{t,j-1}$
$A_{t,j}$
$A_{t,j+1}$
...
...
Lower level L
$L_{t,j,1}$ ... $L_{t,j,i}$ $L_{t,j,i+1}$ ... $L_{t,j,n_{j-1}}$ $L_{t,j,1}$ ... $L_{t,j,i}$ $L_{t,j,i+1}$ ... L_{t,j,n_j} $L_{t,j,1}$ ... $L_{t,j,i}$ $L_{t,j,i+1}$ ... $L_{t,j,n_{j+1}}$

Fig. 2. An illustration of the two-layer hierarchical structure.

$$\lambda \sim \text{Gamma}\left(r, \frac{1-p}{p}\right),$$

where $\text{Gamma}(\alpha, \beta)$ denotes a Gamma distribution with scale α and shape β. Then, the random variable X is marginally distributed as per a negative binomial distribution $X \sim \text{NB}(r, p)$ (Hilbe, 2011), with the probability mass function given by

$$P(x \mid r, p) = \frac{\Gamma(r+x)}{\Gamma(r)\Gamma(x+1)} p^r (1-p)^x.$$

We can view the negative binomial distribution as an extension of the Poisson distribution. The relationship between the mean and variance of the negative binomial random variable, and the parameters r and p, is

$$p = \frac{\mathbb{E}[X]}{\mathbb{V}[X]} \quad \text{and} \quad r = \mathbb{E}[X]\left(\frac{p}{1-p}\right). \tag{3}$$

Since $0 \le p \le 1$, the variance of the negative binomial distribution is greater than its mean; this is known as over-dispersion. To produce a distributional forecast for observation t in series j at the aggregate level (i.e., $A_{t,j}$), we substitute the sample variance of sales over the series A_j (i.e., $\hat{\mathbb{V}}\left[A_j\right]$), and mean forecast for observation $A_{t,j}$ (i.e., $\hat{A}_{t,j}$) for the population variance and mean in (3), respectively, i.e., we use the method-of-moments estimator to obtain parameter estimates:

$$\hat{p}_{A_{t,j}} = \frac{\hat{A}_{t,j}}{\hat{\mathbb{V}}\left[A_j\right]} \quad \text{and} \quad \hat{r}_{A_{t,j}} = \hat{A}_{t,j}\left(\frac{\hat{p}_{A_{t,j}}}{1-\hat{p}_{A_{t,j}}}\right). \tag{4}$$

We can then produce distributional forecasts for $A_{t,j}$ using the estimated negative binomial distribution

$$A_{t,j} \sim \text{NB}\left(\hat{p}_{A_{t,j}}, \hat{r}_{A_{t,j}}\right)$$

in the usual fashion.

3.2.2. Disaggregation

The disaggregation process is performed by weighting the sales forecasts by the historical proportion-of-contribution to the aggregate series A_j. This proportion $\rho_{j,i}$ is calculated by

$$\rho_{j,i} = \frac{\sum_{t=1}^{T} L_{t,j,i}}{\sum_{t=1}^{T} A_{t,j}}, \quad i = 1, \ldots, n_j, \tag{5}$$

where T is the timestamp of the last observation in the training set for aggregate series A_j. The point forecasts at the lower levels, $\hat{L}_{t,j,i}$, are then given by

$$\hat{L}_{t,j,i} = \rho_{j,i}\hat{A}_{t,j}, \tag{6}$$

i.e., the proportion of the aggregate point-forecast attributed to series i.

3.2.3. Parameter estimation for lower level series

Poisson forecasts. Poisson random variables are infinitely divisible, that is, they can be decomposed into a sum of arbitrary many independent Poisson random variables (Steutel and Van Harn, 2003). We use this assumption to obtain the probabilistic forecasts for level L, as they are assumed to come from the same distributional family as the corresponding aggregated level series. Despite the fact that the lower level series could potentially be cross-related in reality, we decompose the aggregated forecasts under a simplifying independence assumption. Under the Poisson assumption, the lower-level observation $L_{t,j,i}$ follows

$$L_{t,j,i} \sim \text{Poisson}(\hat{\lambda}_{L_{t,j,i}}),$$

where $\hat{\lambda}_{L_{t,j,i}} = \hat{L}_{t,j,i}$, and $\hat{L}_{t,j,i}$ is the conditional mean for observation $L_{t,j,i}$, given by (6).

Negative-binomial forecasts. Negative binomial random variables also possess the same property of infinite divisibility; however, for this to be the case it is required that the parameter p must be the same across all series in the hierarchy. That is, $p_{A_{t,j}} = p_{L_{t,j,i}}$ for all $i = 1, \ldots, n_j$. One could adhere to this restriction and use the estimated $\hat{p}$ from the aggregated level, $\hat{p}_{A_{t,j}}$, as an estimate of $p_{L_{t,j,i}}$ for the lower level series. However, in our preliminary experiments (not reported), this procedure did not yield satisfactory results, and we do not pursue this approach further. Instead, we estimate $p_{L_{t,j,i}}$ individually for each of the lower level series. We estimate the variance of lower level series $L_{j,i}$ by the sample variance, denoted as $\hat{\mathbb{V}}\left[L_{j,i}\right]$. Then, the estimation of the parameters of negative binomial distribution at the lower level can be performed using the method-of-moments technique in a similar fashion to Section 3.2.1, i.e.,

$$\hat{p}_{L_{t,j,i}} = \frac{\hat{L}_{t,j}}{\hat{\mathbb{V}}\left[L_{j,i}\right]} \quad \text{and} \quad \hat{r}_{L_{t,j,i}} = \hat{L}_{t,j}\left(\frac{\hat{p}_{L_{t,j,i}}}{1-\hat{p}_{L_{t,j,i}}}\right). \tag{7}$$

Once we have estimated the relevant parameters we can produce distributional forecasts for the lower-level observation $L_{t,j,i}$ based on

$$L_{t,j,i} \sim \text{NB}\left(\hat{p}_{L_{t,j,i}}, \hat{r}_{L_{t,j,i}}\right).$$

It is worth noting that when series are highly intermittent, the large number of zero entries in the series could potentially lead to a sample variance smaller than the mean, resulting in an under-dispersed model, i.e., $\hat{p}_{L_{t,j,i}} \ge 1$. In principle, a Conway–Maxwell–Poisson distribution could be used in these situations; however, in practice, as the negative binomial distribution reduces to the Poisson distribution when $r \to \infty$ (Hilbe, 2011), we use probabilistic forecasts based on the Poisson model in these cases.

Fig. 3 provides a visual example to illustrate our proposed top–down forecasting framework. We consider a randomly chosen series A_j at level A with a hierarchy that consists of three series at level L. We first produce point forecasts for series A_j with an off-the-shelf global forecasting model, in this case a LightGBM model. We may then choose an appropriate distributional model (i.e., Poisson or negative binomial) and estimate the relevant distributional parameters for the forecast observations using the procedures in Section 3.2.1. The historical proportion-of-contribution of each of the series $L_{j,i}$ at level L to the aggregate A_j is calculated using (5) (shown in Fig. 3). These are then used to disaggregate the point forecasts from level A to level L. Parameter estimation for the distributional models is performed at level L following the procedures in Section 3.2.3. Finally, using these estimated distributional models, a probabilistic forecast, for example a 90% prediction interval, is produced for each series $L_{j,i}$ at level L.

4. Experimental framework

This section describes the datasets, benchmarks, and error measurements used in our experimental study.

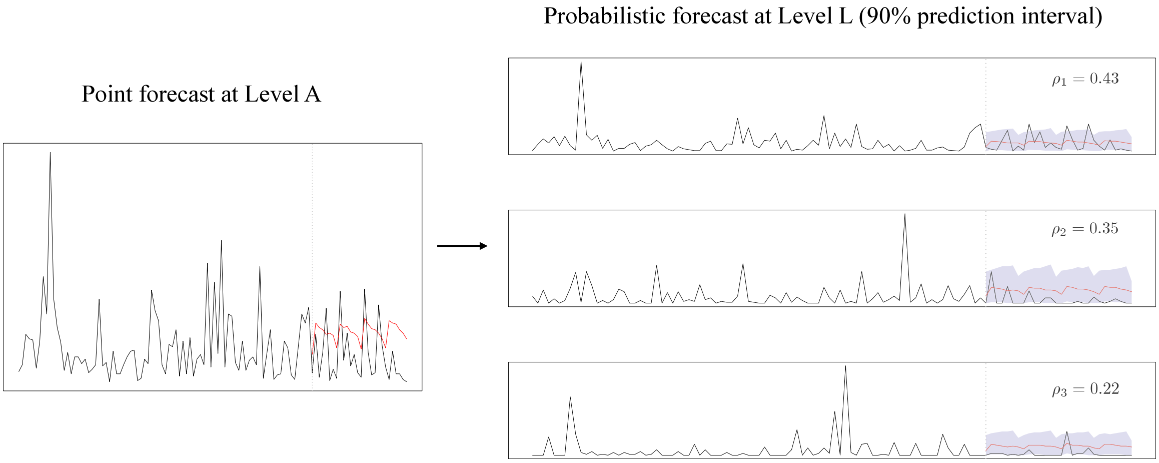


Fig. 3. An illustration of the proposed top–down forecasting framework with a toy example.

Table 1
Summary of the percentage of series and percentage of zeros out of the days since the first sale across all series, in each category on the lower level of the three datasets analysed in this paper (in percent).

	Dataset	Smooth	Erratic	Lumpy	Intermittent
Percentage of series	E-commerce	0.10	0.53	42.00	57.37
	Corporación Favorita	20.51	19.58	33.62	26.29
	M5	6.23	2.83	18.38	72.56
Percentage of zeros	E-commerce	14.06	16.04	83.00	91.57
	Corporación Favorita	9.50	12.80	52.65	59.48
	M5	14.13	16.77	54.03	67.13

4.1. Datasets

We are aware of two openly available large retail datasets, namely the M5 dataset (Makridakis et al., 2022a) and the Corporación Favorita dataset (Kaggle, 2018). Both of these represent traditional brick-and-mortar sales datasets. We use these datasets in addition to a proprietary e-commerce dataset. Based on demand classification (see Section 3.1), we can categorise the lower level series into four classes, and the percentage of series that fall into each class is summarised in Table 1. We find that in the examined e-commerce dataset, lumpy and intermittent series are the biggest subgroups. The Corporación Favorita dataset contains series which are more evenly distributed over the four categories, while the intermittent series form a large part of the M5 dataset as well. We further calculate the percentage of zeros out of the days since the first sale in each category of the three datasets. From Table 1, the proprietary e-commerce series are more intermittent compared with the brick-and-mortar datasets we also use in the experiments. We describe the datasets in more details in the following.

4.1.1. The examined proprietary e-commerce dataset

This dataset consists of 211,765 series of daily unit sales across all regions of Indonesia from May 7th of 2019 to May 8th of 2021 from one particular department of the company. In the dataset, similar products are grouped and regarded as a ‘Catalogue’, and products in a catalogue have a high level of similarity in price. For example, an iPhone 11 could be one item of the catalogue, which contains different specific models such as green iPhone 11. We use the catalogue level as level A, 101,944 series, and the specific models level as level L in the experiments. Around half of the categories have only 1 or 2 products. We are able to scale the methods to this large dataset by training them on a much smaller dataset and then adapting their forecasts to the original dataset.

Forecasts at different quantile levels are often required to determine the optimal inventory level. While businesses generally strive for a high service level, such as 90%, constraints like limited warehouse capacity and working capital may necessitate a lower optimal service level. Consequently, forecasts at various quantile levels are needed for service level optimisation, which is beyond the scope of this study. For demonstration purposes, we use the 10th percentile to illustrate the performance of the proposed model at a lower quantile level. Thus, to evaluate the top–down approach, we forecast 28 days ahead with the catalogue level series and evaluate the 10th and 90th percentile forecasts at level L.

4.1.2. The Corporación Favorita dataset

The Corporación Favorita dataset (Kaggle, 2018) provides daily unit sales data in brick-and-mortar grocery stores from January 1st of 2013 to August 15th of 2017. The original data contains negative values which denote the number of returns for a certain product, and these negative values are set to zero in our experiments as we are only interested in sales forecasting. A natural way of constructing a two-layer hierarchy is to use the original data as the lower level, and sum up unit sales by item as an aggregated level, i.e., add up the volumes in different stores for each item. In this way, level A contains 3998 series, whereas level L consists of 172,906 series. The tasks performed are similar: we evaluate the 10th and 90th percentiles of the future 28 days ahead at level L with models trained with the item-level series.

4.1.3. The M5 dataset

With data available for over 5 years in the M5 dataset (Makridakis et al., 2022a), participants were required in the original competition to submit 9 quantile forecasts for each series. The provided sales data is hierarchically structured and can be aggregated to 12 different levels. To provide further insights of the proposed methods, we evaluate the performance of the proposed top–down probabilistic forecasting framework in line with the competition settings, i.e., we evaluate the 0.005, 0.025, 0.165, 0.250, 0.500, 0.750, 0.835, 0.975, and 0.995 quantiles. We utilise the hierarchy between level 10 (product unit sales aggregated by stores, 3049 series) and level 12 (product unit sales, 30,490 series, the lowest level). Models are trained with data from level 10 and forecasts are disaggregated proportionally to level 12, and quantile forecasts are then generated according to distributional assumptions.

4.2. Compared settings

The proposed top–down forecasting framework is implemented with LightGBM model variants and linear model variants. Models are trained with 100 lags as input features to capture possible weekly, monthly, and quarterly seasonality while being not too computationally expensive and complex. Fourier terms are also introduced to model yearly and weekly seasonality. The LightGBM models are named by the corresponding loss functions and parameter settings, and linear models are named by specific regression settings. In the following, we list the techniques used in this work. The models below are trained on level A and a top–down disaggregation is then applied to obtain forecasts on level L.

LightGBM LightGBM models are trained in a top–down fashion under different loss functions and parameter settings. The LightGBM package provides L1, L2, Poisson, Huber, and Tweedie loss functions for regression problems (Shi et al., 2022). Following the literature, the negative binomial loss is the most adequate loss function to use as it takes over-dispersion into consideration (Kolassa, 2016), however, no off-the-shelf implementation of the negative binomial loss function is available. We implement it with the custom loss and evaluation function in Python (refer to Appendix). It is not straightforward to implement such a loss function where two parameters are considered, in a common machine learning framework which only supports a single output. Thus the implementation is integrated with an iterative optimisation step for updating the r parameter. We are exploring three different sets of parameters. We consider default regression parameters, and a preset parameter setting (Bandara et al., 2021) that has shown to perform well for the M5 competition, but on the decision level (Level 12), which is not the level on which we forecast. They are named as `default` and `preset` in the models, respectively. Instead of modelling with a constant, piecewise linear trees use linear functions to produce the outcomes, and have demonstrated accurate performance in forecasting (Godahewa et al., 2022). So we also include the piecewise linear GBTs, which can be selected with the `linear_tree` parameter in LightGBM.

Linear models Linear models, or Pooled Regression (PR, Gelman and Hill, 2006) models linear relationships between predictors and target values fitted via ordinary least squares. Penalised linear regression, specifically Lasso regression models (Tibshirani, 1996) are also trained in the experiments. We implement pooled regression with ordinary least squares and penalised models with the R `glmnet` package (Simon et al., 2011) under default settings with cross-validation. Moreover, apart from using the 100 lags and Fourier terms as stated previously, it is intuitive to consider quadratic terms in the regression models. We trained models with Lasso penalty and extra 100 quadratic lag terms, but they did not show improvements in accuracy so results are not reported here.

In terms of benchmarks, we consider the following baselines of forecasts directly performed on level L, namely direct quantile modelling with LightGBM models, DeepAR, traditional univariate forecasting models, and some relatively simple methods tailored to count data as used by Kolassa (2022). An input window of 100 lags and Fourier terms is used for the former two approaches, similarly to the proposed methods. The details are as follows.

Direct LightGBM Direct quantile models are trained on the lower level L to get the lower level prediction. This approach requires training a model for each quantile of interest. We use LightGBM with the preset parameters from Bandara et al. (2021) as those authors report promising accuracy of this parameterisation on the M5 decision level (level 12). Quantile forecasts are generated with the quantile loss function.

DeepAR The autoregressive neural network forecasting framework developed by Salinas et al. (2020) is another competitive standard benchmark nowadays. We trained DeepAR models globally with the Python `GluonTS` package (Alexandrov et al., 2020) on the lower level L with default parameters and a negative binomial output. Considering the massive computational costs, we use DeepAR as a prototype for other deep-learning methods.

Local statistical methods Five classic statistical methods, namely Autoregressive Integrated Moving Average model (ARIMA, Box et al., 2015), ExponenTial Smoothing model (ETS, Hyndman et al., 2008), Mean, Naïve, Drift, and Seasonal Naïve (SNaïve, with weekly seasonality) are considered in the experiments. Models are fitted using the R `fable` package (O'Hara-Wild et al., 2021) under their default configurations, and probabilistic forecasts are produced by specifying the `level` parameter.

The following five per-series methods analysed by Kolassa (2022) are considered in this work as strong benchmarks for count data.

In-sample quantiles If we take the distribution of the in-sample data as an estimate of the true marginal distribution, quantile forecasts in the future horizon can be then obtained according to this distribution, denoted as in-sample quantiles. The in-sample quantile forecasts on the lower level can be thought of as the probabilistic variant of a mean forecast for point forecasts.

Empirical weekday (Emp-Wd) In-sample quantiles are calculated for each day of the week separately.

Empirical Poisson (Pois) A Poisson distribution is fitted to each series with moment matching using the R `fitdistrplus` package (Delignette-Muller and Dutang, 2015). Quantiles are then generated from the empirical Poisson distribution.

Empirical negative binomial/Conway–Maxwell–Poisson (NB-CMP) Either a negative binomial distribution, when the series is over-dispersed, or a Conway–Maxwell–Poisson distribution, when the series is equi- or under-dispersed, is fitted to the series where quantiles are generated from. A negative binomial distribution with moment matching using the R `fitdistrplus` package and a Conway–Maxwell–Poisson distribution is fitted through the `glm.cmp()` function provided in the R `COMPoissonReg` package (Sellers et al., 2023).

Zero-inflated Poisson (ZIP) Quantiles are generated through a Zero-Inflated Poisson distribution fitted to each series. The `zeroinfl()` function is used from the R `pscl` package (Jackman, 2024; Zeileis et al., 2008).

Zero-inflated negative binomial (ZINB) Quantiles are generated through a Zero-Inflated negative binomial distribution fitted to each series. The `zeroinfl()` function is used from the R `pscl` package (Jackman, 2024; Zeileis et al., 2008). A ZIP model is fitted if a numerical singularity error occurs when fitting a ZINB model.

4.3. Evaluation metrics

Following the setup of the M5 competition, we evaluate the probabilistic forecasts using the Weighted Scaled Pinball Loss (WSPL, Makridakis et al., 2021). We denote $q_t^{[u]}$ as the predicted value for quantile u at time t, and y_t as the corresponding ground truth. Then, for a series i, the Scaled Pinball Loss (SPL) is calculated for each quantile as follows,

$\mathrm{SPL}_i[u]$

$$= \frac{1}{h} \frac{\sum_{t=T+1}^{T+h} (u(y_t - q_t^{[u]})\mathbf{1}\{q_t^{[u]} \leq y_t\} + (1-u)(q_t^{[u]} - y_t)\mathbf{1}\{q_t^{[u]} > y_t\})}{\frac{1}{n-1}\sum_{t=2}^{T} |y_t - y_{t-1}|}, \tag{8}$$

where the pinball loss (Gneiting, 2011) over the forecast horizon h is scaled by the average absolute error of the one-step-ahead in-sample naïve forecast within the period between the first non-zero sales to time T. $\mathbf{1}$ is the indicator function. For example, for the 10th and 90th percentile forecast evaluation, $u \in \{0.1, 0.9\}$, and $q = 2$ corresponds to the number of quantiles of interest. The WSPL is computed by the weighted average of the average SPL for all the quantiles per series with weights w_i,

$$\text{WSPL} = \sum_{i=1}^{n} w_i \times \frac{1}{q} \sum_{j=1}^{q} \text{SPL}_i[u_j].$$

When evaluating the proposed methods on the M5 dataset, we follow the M5 competition setup and use the same weighting for a direct comparison with other participants, where dollar sales in the last 28 days are calculated as weights. In the examined proprietary dataset and in the Corporación Favorita dataset, such information on dollar sales is not available. While one can still possibly propose a weighting process with certain assumptions, we opt for weighting series equally during evaluation. A lower WSPL indicates a better estimate of the forecast intervals. The SPL uses the in-sample naïve forecast as the denominator, a procedure that was first proposed by Hyndman and Koehler (2006) for the MASE and is nowadays standard practice in forecasting. However, this process has the problem that a division by zero can occur if the series is constant. Due to the procedure of trimming leading zeros, series can be very short and this situation can happen in our experiments. However, such cases are rare, for example, only 8 series with such property are present in the Corporación Favorita dataset, so that we omit such series during the evaluation process.

5. Results and discussion

In the following, we present an evaluation on the three different datasets separately. The proposed top–down forecasting framework is first evaluated on the e-commerce dataset. Based on the results, we aim at transferring the findings to the brick-and-mortar datasets. Therefore, we use the most competitive models for further experiments on the Corporación Favorita dataset and the M5 dataset. For the M5 dataset, we are able to directly compare the performance of the proposed top–down forecasting framework with the results of the original competition participants.

5.1. Evaluation with the e-commerce dataset

In this section, we present detailed performance evaluations on the proprietary e-commerce dataset. Models are globally trained on level A and a top–down approach is then applied to get forecasts for level L.

Table 2 presents the WSPL results on level L, based on the demand classification category of the respective level L series. The benchmarks are placed at the top of the table, and models trained in a top–down fashion are arranged by distribution assumptions. Noticeably, the direct LightGBM model outperforms all other models in all categories except being in third place for lumpy data. DeepAR models beat other methods for lumpy data, and have consistently accurate performance in other categories. It is somewhat surprising to find that simply using the in-sample quantiles can lead to a competitive forecasting accuracy, especially for the intermittent series. This is in line with findings in the literature that empirical models can be able to outperform sophisticated ones, as shown by Kolassa (2016) and Spiliotis et al. (2021). In addition, the Emp-Wd model, which treats each day of the week separately yields accurate forecasts. The zero-inflated models are competitive on this dataset. No consistently good performance can be found for the local statistical methods.

For the proposed top–down method, the LightGBM models have achieved competitive accuracy especially under a negative binomial assumption. More sophisticated hyperparameter settings such as the preset parameters do not show an advantage over the default parameters, which can even lead to better accuracy. Interestingly, linear models fitted via least squares have demonstrated even more competitive accuracy as PR models and Lasso models present satisfactory results across all data categories. The PR even beats the Direct LightGBM on the lumpy series, and is slightly better than DeepAR on the intermittent series. With regard to different distribution assumptions, we can find that models with negative binomial assumptions outperform those with Poisson assumptions, indicating that the data is over-dispersed.

Table 3 compares the total training time of the forecasting models. Models were trained on a server machine (16 vCPUs, 64 GB RAM) using R 4.1. The proposed top–down methods are much faster than the direct LightGBM models. Specifically, the top–down LightGBM methods under default parameterisation can be trained within 10 min, whereas the direct LightGBM approach takes around 5 h. The training process of the top–down PR model is efficient, and the Lasso model is relatively slower as it fits additional regularisation parameters. Among the LightGBM model variants, those using user-defined negative binomial loss take the longest time. This is due to the iterative search of parameter r of the negative binomial distribution (see Appendix). Such a loss function does not demonstrate the promised accuracy within a practical timeframe. With competitive accuracy discussed previously, the in-sample quantile is also superior in terms of computational efficiency. Other local benchmarks such as zero-inflated models, ETS and ARIMA can take a long time to train. Finally, the DeepAR model appears to be fast and computationally efficient.

5.2. Evaluation with the Corporación Favorita dataset

Based on the previous experiments, we limit our experiments on this dataset on a selection of the best-performing methods from the previous experiments, from the different categories of methods, to run with the Corporación Favorita dataset, namely LightGBM with Poisson loss, Tweedie loss and negative binomial loss functions, pooled regression, and Lasso. Again, we use 100 lags and Fourier terms as input, and LightGBM models are trained under default parameter settings. In the top–down probabilistic experiments, we assume sales data to follow a Poisson distribution or a negative binomial distribution across the hierarchy. From the results on the e-commerce dataset, we utilise Direct LightGBM models, DeepAR and the five local count data models as the comparison methods on level L, with the same parameter setting discussed in Section 4.2. In the case of direct training, the lag matrix is over 230 GB, which hinders the implementation on our available computing resources. In addition, the series are much less intermittent compared to the e-commerce dataset which leads to a much denser input matrix. The limit on the size of input sparse matrices restricts the amount of series and the number of lags that can be trained at the same time. Therefore, we need to make compromises and the direct LightGBM model is trained as follows. As the partitioning technique introduced in Section 3.1 can also be used as a pre-processing step to render the methods more scalable when a single global model cannot fit into memory. We first partition the lower level series into the smooth, erratic, lumpy and intermittent categories and train four LighGBM models separately. Due to the restriction of the size on the input matrix, we intend to use as many lags as possible for a fair comparison. With the Fourier terms to capture seasonality, we use 20 lags for the lumpy category and 30 lags for the other three categories. Another option is to remove the Fourier terms and give more importance to the lags as input. This approach leads to another Direct LightGBM (max lags) model where we use 50 lags for the smooth and erratic series, and 35 and 45 for lumpy and intermittent series, respectively.

Table 4 reports the WSPL errors that are calculated on the lower level. From the second column, we can compare the top–down approach against the strong direct methods. We observe that our methods

Table 2
The WSPL on level *L* of the examined proprietary dataset, categorised based on the demand class. The WSPL for all series on level *L* are provided in the last column. The amount of series in each category is provided in parenthesis. The top–down forecasting methods are sorted by distribution assumptions.

Model	Smooth (209)	Erratic (1,126)	Lumpy (88,933)	Intermittent (121,497)	All
ARIMA	0.2187	0.2336	0.3088	0.1853	0.2374
Drift	0.3002	0.4161	0.8530	0.7287	0.7788
ETS	0.2165	0.2348	0.3081	0.1813	0.2350
Mean	0.2313	0.2507	0.2947	0.1836	0.2307
Naïve	0.2964	0.4114	0.8434	0.7187	0.7690
SNaïve	0.2460	0.3107	0.5033	0.3573	0.4183
In-sample quantiles	0.2229	0.2545	0.2204	0.1712	0.1923
Emp-Wd	0.2254	0.2556	0.2208	0.1718	0.1929
Pois	0.3322	0.3359	0.2344	0.1727	0.1997
NB-CMP	0.2558	0.2671	0.2215	0.1711	0.1929
ZIP	0.2212	0.2504	0.2235	0.1712	0.1937
ZINB	0.2229	0.2549	0.2211	0.1712	0.1926
DeepAR	0.1976	0.2200	**0.2093**	0.1660	**0.1845**
Direct LightGBM	**0.1931**	**0.2078**	0.2139	**0.1634**	0.1849
Negative binomial distribution assumption					
Lasso	0.2095	0.2446	0.2240	0.1736	0.1952
Pooled Regression	0.2015	0.2327	0.2132	0.1659	0.1861
LightGBM Huber loss default	0.2689	0.2607	0.2160	0.1690	0.1893
LightGBM Huber loss linear leaf	0.2974	0.2769	0.2171	0.1696	0.1902
LightGBM Huber preset	0.2005	0.2269	0.2145	0.1681	0.1879
LightGBM L1 loss default	0.3208	0.2973	0.2235	0.1733	0.1952
LightGBM L1 loss linear leaf	0.2207	0.2483	0.2200	0.1711	0.1921
LightGBM L1 loss preset	0.2318	0.2463	0.2200	0.1718	0.1925
LightGBM L2 loss default	0.2076	0.2323	0.2171	0.1734	0.1921
LightGBM L2 loss linear leaf	0.2042	0.2329	0.2188	0.1748	0.1936
LightGBM L2 loss preset	0.2173	0.2413	0.2262	0.1810	0.2003
LightGBM Neg. Bin. loss default	0.2144	0.2466	0.2362	0.1946	0.2124
LightGBM Poisson loss default	0.2057	0.2348	0.2192	0.1748	0.1938
LightGBM Poisson loss linear leaf	0.2284	0.2568	0.2255	0.1786	0.1988
LightGBM Poisson loss preset	0.2175	0.2466	0.7572	0.2231	0.4476
LightGBM Tweedie loss default	0.2108	0.2359	0.2185	0.1747	0.1935
LightGBM Tweedie loss linear leaf	0.2145	0.2440	0.2225	0.1782	0.1972
LightGBM Tweedie preset	0.2192	0.2477	0.2323	0.1879	0.2070
Poisson distribution assumption					
Lasso	0.2522	0.3225	0.2436	0.1763	0.2055
Pooled Regression	0.2343	0.2870	0.2214	0.1662	0.1901
LightGBM Huber loss default	0.3080	0.3371	0.2223	0.1689	0.1923
LightGBM Huber loss linear leaf	0.3369	0.3595	0.2246	0.1696	0.1939
LightGBM Huber preset	0.2329	0.2790	0.2178	0.1676	0.1894
LightGBM L1 loss default	0.3630	0.3868	0.2263	0.1727	0.1965
LightGBM L1 loss linear leaf	0.2675	0.3299	0.2283	0.1716	0.1964
LightGBM L1 loss preset	0.2655	0.3108	0.2218	0.1711	0.1932
LightGBM L2 loss default	0.2417	0.2947	0.2369	0.1756	0.2021
LightGBM L2 loss linear leaf	0.2390	0.2937	0.2389	0.1770	0.2037
LightGBM L2 loss preset	0.2522	0.3105	0.2480	0.1831	0.2111
LightGBM Neg. Bin. loss default	0.2463	0.3161	0.2527	0.1959	0.2205
LightGBM Poisson loss default	0.2394	0.3053	0.2414	0.1774	0.2051
LightGBM Poisson loss linear leaf	0.2618	0.3353	0.2494	0.1812	0.2108
LightGBM Poisson loss preset	0.2504	0.3232	0.7813	0.2254	0.4595
LightGBM Tweedie loss default	0.2435	0.3032	0.2388	0.1769	0.2037
LightGBM Tweedie loss linear leaf	0.2471	0.3112	0.2439	0.1803	0.2078
LightGBM Tweedie preset	0.2532	0.3218	0.2546	0.1900	0.2179

are competitive, and the linear models again have remarkably outperformed the LightGBM variants. The negative binomial distribution still seems to be more appropriate on this dataset compared with a Poisson assumption. The DeepAR model has the best accuracy on this dataset. As we have to make compromises when training the direct LightGBM models, we observe that the model that uses more lags seems to perform better than the one with Fourier terms. The simple in-sample quantile is still a strong benchmark, as well as the Emp-Wd method. At the same time, we can find more top–down methods that are competitive against them, with a wider gap compared to the results found on the e-commerce dataset. Table 5 reports the training time for each method. Overall, the top–down approach incorporated with linear models and LightGBM models can be trained very efficiently, even compared to the in-sample quantile method. They are much faster than the direct LightGBM models and DeepAR. As the hierarchy on the Corporación Favorita dataset contains more bottom level series, we can find a more significant scalability of the proposed top–down methods, without losing much of the accuracy. The Emp-Wd method takes more time as each day in the week needs to be considered separately. Also, the LightGBM model with negative binomial loss requires more training effort, as is to be expected.

5.3. *The M5 competition revisit*

We conduct experiments on the M5 dataset similar to Section 5.2 with selected models and parameter settings, and probabilistic forecasts are generated based on Poisson distribution or negative binomial distribution. We use the same type of result plot as the competition summary in Makridakis et al. (2021). Instead of separating by distribution assumptions as in the previous tables, we put the name of the specific distribution, i.e., Poisson, Neg. Bin., at the beginning of the names of the proposed methods. Fig. 4 compares the WSPL values on

Table 3
Training time on the examined proprietary dataset for the benchmark methods (top part) on level L, and LightGBM models and linear models on level A.

Model	Training time (minutes)
ARIMA	741.60
Drift	58.29
ETS	875.75
Mean	45.39
Naïve (fable)	54.96
Snaïve (fable)	85.99
In-sample quantile	0.99
Emp-Wd	13.78
Pois	1.86
NB-CMP	25.04
ZIP	153.52
ZINB	519.42
DeepAR	68.75
Direct LightGBM	284.36
Lasso	56.15
Pooled Regression	28.48
LightGBM Huber loss default	6.32
LightGBM Huber loss linear leaf	9.64
LightGBM Huber loss preset	30.30
LightGBM L1 loss default	11.04
LightGBM L1 loss linear leaf	16.19
LightGBM L1 loss preset	51.95
LightGBM L2 loss default	9.07
LightGBM L2 loss linear leaf	14.46
LightGBM L2 loss preset	24.13
LightGBM Neg. Bin. loss default	1540.08
LightGBM Poisson loss default	6.99
LightGBM Poisson loss linear leaf	8.37
LightGBM Poisson loss preset	22.85
LightGBM Tweedie loss default	5.53
LightGBM Tweedie loss linear leaf	7.12
LightGBM Tweedie loss preset	26.8

level 12 with the top 50 participants in the uncertainty track of the original competition. Remarkably, the proposed top–down forecasting approaches all enter the top 50 when compared with the original 892 participating teams, w.r.t. WSPL, except for the ones with a negative binomial loss function. We also notice that methods which assume future sales to follow a negative binomial distribution perform better, which is in line with the previous experiments. Benchmarks on level 12 are trained in the same fashion as in Section 4.2. Due to the computational limitations, we are able to train on the whole dataset from level 12 with a single direct LightGBM model with 70 lags. We also include a max lags version where we intend to include more lags and train a direct LightGBM model with 100 lags without Fourier terms. From Fig. 4, we see that the DeepAR model is very competitive on the M5 level 12. The direct LightGBM models are also accurate, where the one with more lags instead of Fourier terms performs better, ranking 10th against other competitors. The in-sample quantile and the Emp-Wd are strong benchmarks on this brick-and-mortar retailer dataset, but rank lower than the proposed top–down methods. The detailed WSPL results of each category on series from level 12 are also provided in Table 6 for consistency. The proposed models are competitive against the strong benchmarks on each category, especially the linear models. Table 7 presents the training time of models on level 10 and directly on level 12. Using the top–down approach, the GBTs can be trained with modest computational effort, as well as the linear models. The simple in-sample quantile benchmark is still very efficient. The LightGBM model with negative binomial loss function takes much longer time because of the iterative numerical optimisation process. In the M5 dataset, it seems there is a poorer estimate of the parameter r through the numerical search under practical time constraints. DeepAR can be trained at a fast speed as the M5 dataset is relatively smaller compared to the other two datasets in our experiments. Although it may provide certain accuracy gains, training directly on level 12 can take much more computational time and compromises may have to be made to make the approach feasible.

5.4. Further discussion

In this section, we provide a discussion on the automatic selection of the aggregated level to apply the proposed top–down forecasting framework, and a suggested workflow for forecasting the e-commerce datasets.

5.4.1. Selecting the aggregated level

We have explored two types of aggregation in our analysis, namely the category–product hierarchy on our proprietary dataset, and store–product hierarchy on the Corporación Favorita dataset and the M5 dataset. The top–down forecasting framework works well in both situations. Ultimately, the way to form a hierarchy is application-dependent, but there are some heuristics we can follow. To make the most of the proposed framework, on the one hand, practical considerations should come first. Data should be aggregated to a level where models can run without concerning the limitations of memory and computing power. On the other hand, since the probabilistic forecasts are generated based on assumptions, the aggregation levels should be chosen in a way that we would not expect too large changes of data characteristics after aggregation.

In practice, we can explore the distributions of series on the aggregated level and on the decision level and compare the similarity. For example, a negative binomial distribution can be fitted for each series of the decision level and the possible levels to aggregate to, and goodness-of-fit results can then be evaluated. We perform such an example, where we use the `glm.nb` function from the R `MASS` package (Venables and Ripley, 2002) to fit a negative binomial distribution, and examine the fitting with the p-value reported from the `poisgof` from the R `epiDisplay` package (Chongsuvivatwong, 2022). Table 8 reports the results at a significance level $\alpha = 0.05$. As the series are aggregated to higher levels, they are less and less similar to the decision level. We see that levels 10 and 11 offer similar trade-offs between the number of series to forecast and the similarity between the decision level and the aggregated level. In our experiments, we decide to aim for higher scalability and therefore choose level 10 as the aggregated level.

5.4.2. A suggested workflow for forecasting on e-commerce datasets

It is an interesting finding that in the intermittent series of the e-commerce data, the largest proportion of the dataset, simple methods such as the in-sample quantile are competitive against LightGBM variants and linear models trained under the top–down forecasting framework. In particular, the in-sample quantile method achieves 2nd place after the direct LightGBM model which achieves the best accuracy (see Table 2). If we also take training time into account, the in-sample quantile is unbeatable compared with other methods. In contrast, such an advantage in accuracy cannot be seen in the Corporación Favorita dataset (Table 4) and the M5 dataset (Fig. 4), where brick-and-mortar sales data are considered.

Recall the percentage of zeros calculated in Table 1 on the lower level of the three datasets experimented in this research. Noticeably, over 91% of entries in the intermittent series of the e-commerce data are zero, implying a high degree of intermittency. This explains why these series are relatively unpredictable and no method leads any benefits over the most simple benchmarks. One may observe that the lumpy series also present high proportion of zeros. However, from the empirical results, the proposed methods have shown a better performance on the examined e-commerce dataset. This may be due to the lumpy series by definition having larger variance compared to the intermittent series. Taking all the findings into account, we can suggest as a generic workflow in our e-commerce forecasting use case the following. For intermittent series, one can simply use in-sample quantiles to produce accurate forecasts. The proposed top–down forecasting framework, for example, integrated with linear models and LightGBM models (e.g., with Tweedie loss and Poisson loss functions), is used to generate probabilistic forecasts for other categories.

Table 4
The WSPL for lower level series from the Corporación Favorita dataset on each category. The WSPL for all lower level series are provided in the last column. The number of series in each category is provided in parenthesis.

Model	Smooth (35,458)	Erratic (33,851)	Lumpy (58,131)	Intermittent (45,466)	All
In-sample quantiles	0.2618	0.2747	0.3177	0.3259	0.2999
Emp-Wd	0.2559	0.2706	0.3154	0.3233	0.2965
Pois	0.3237	0.3287	0.3343	0.3393	0.3323
NB-CMP	0.2705	0.2853	0.3224	0.3267	0.3056
ZIP	0.2716	0.2787	0.3140	0.3252	0.3013
ZINB	0.2695	0.2749	0.3180	0.3259	0.3017
Direct LightGBM (max lags)	0.1985	0.2192	0.2584	0.2580	0.2383
Direct LightGBM (Fourier terms)	0.2048	0.2239	0.2589	0.2588	0.2409
DeepAR	**0.1811**	**0.2147**	**0.2539**	**0.2553**	**0.2317**
Negative binomial distribution assumption					
Lasso	0.2262	0.2498	0.2832	0.2783	0.2637
Pooled Regression	0.2196	0.2449	0.2768	0.2709	0.2572
LightGBM Neg. Bin. loss default	0.2339	0.2693	0.2871	0.2763	0.2699
LightGBM Poisson loss default	0.2303	0.2639	0.2851	0.2765	0.2674
LightGBM Tweedie loss default	0.2282	0.2632	0.2829	0.2742	0.2656
Poisson distribution assumption					
Lasso	0.2571	0.2863	0.3032	0.2920	0.2875
Pooled Regression	0.2416	0.2740	0.2901	0.2796	0.2744
LightGBM Neg. Bin. loss default	0.2590	0.3000	0.3040	0.2874	0.2896
LightGBM Poisson loss default	0.2553	0.2947	0.3025	0.2878	0.2874
LightGBM Tweedie loss default	0.2520	0.2928	0.2991	0.2845	0.2844

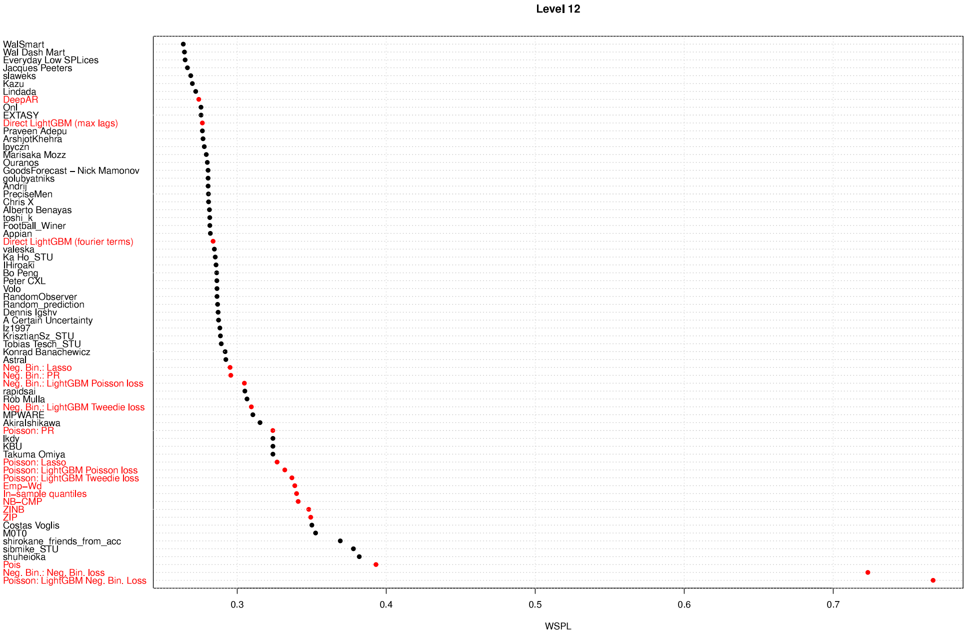


Fig. 4. The performance of the proposed methods and benchmarks on level 12 compared with the top 50 submissions of the M5 uncertainty competition.

6. Conclusion

In this paper, we have proposed a scalable top–down forecasting framework which is capable of generating reliable probabilistic forecasts at a fast speed. Direct modelling on the lower level and producing quantile forecasts is accurate, but it can be computationally expensive while no corresponding large gains of accuracy are observed. Compromises may also have to be made when training direct quantile models. In our use cases, and presumably many others in the industry, the additional computational effort is thus not justified. Our forecasting approach is feasible to implement in production. The top–down forecasting framework has also been evaluated with two public datasets and has shown good results.

As evaluated in the experiments, we have found that the accuracy depends largely on the estimation of distributional parameters. In accordance with the literature, in the three datasets in our experiments the negative binomial assumption tends to be more adequate than the Poisson assumption. However, this does not translate into higher accuracies when using a negative binomial loss function. We have shown that in practice, implementation of this loss function requires additional numerical search to fit in a common machine learning framework, which prevents it from beating other built-in loss functions under practical computational constraints. Somewhat surprisingly, linear models are competitive with the state-of-the-art LightGBM algorithm in situations where no external covariates are used (as in our research; external variables could regard pricing, promotions, and others). Here, linear models offer a simple alternative to GBTs that is fast, robust, and more interpretable.

We observe that the e-commerce dataset can be much more intermittent compared to brick-and-mortar retail datasets. In particular, the intermittent series make up the largest proportion of the dataset and they are also more intermittent, i.e., they contain proportionally more zeros. Simply using in-sample quantiles on this category can be very competitive against other sophisticated methods, with superior

Table 5
Training time of the top–down methods on aggregated level of the Corporación Favorita dataset, and the benchmarks executed on level L.

Model	Training time (minutes)
In-sample quantile	0.76
Emp-Wd	13.17
Pois	3.64
NB-CMP	5.81
ZIP	53.29
ZINB	226.76
DeepAR	68.75
Direct LightGBM (max lags)	357.67
Direct LightGBM (Fourier terms)	388.84
Lasso	5.67
Pooled Regression	2.70
LightGBM Neg. Bin. loss default	95.89
LightGBM Poisson loss default	1.14
LightGBM Tweedie loss default	1.19

computational efficiency. In addition, the proposed top–down method depends on the hierarchical structure of the series and distributional assumptions to some extent. We have investigated the distributions of the series on the lower level and on the possible aggregated levels of the M5 dataset. Based on the given hierarchy of the business, it is a trade-off between the number of series on the aggregated level to model on and the similarity between the two levels when applying the proposed top–down forecasting framework.

A limitation of the proposed framework that could be addressed as future work is the static top–down approach where total historical proportions are used during disaggregation. We assume that using a disaggregation method which accounts for future changes may improve forecasting accuracy. Additionally, the proposed top–down forecasting framework depends on the selection of the aggregated level. We have provided some preliminary results on how to perform an automatic level selection, but a more systematical procedure could be further investigated. Finally, the examined e-commerce data spreads out before and after the global pandemic lockdown periods. However, the potential structural breaks of the shopping patterns are not modelled in this study.

CRediT authorship contribution statement

Xueying Long: Writing – review & editing, Writing – original draft, Visualization, Software, Investigation, Formal analysis. **Quang Bui:** Writing – original draft, Software, Investigation, Formal analysis, Data curation. **Grady Oktavian:** Validation, Software, Formal analysis, Data curation. **Daniel F. Schmidt:** Writing – review & editing, Validation, Supervision, Methodology. **Christoph Bergmeir:** Writing – review & editing, Validation, Supervision, Resources, Project administration, Methodology, Investigation, Funding acquisition, Conceptualization. **Rakshitha Godahewa:** Writing – review & editing, Software. **Seong Per Lee:** Validation, Software, Formal analysis, Data curation. **Kaifeng Zhao:** Writing – review & editing, Validation, Software, Data curation. **Paul Condylis:** Writing – review & editing, Supervision, Funding acquisition, Data curation.

Acknowledgements

Christoph Bergmeir is supported by a María Zambrano (Senior) Fellowship that is funded by the Spanish Ministry of Universities and Next Generation funds from the European Union.

Appendix. Implementation of negative binomial loss function with LightGBM

As sales data is usually over-dispersed, i.e., the variance is greater than its mean, when we use machine learning algorithms to predict the future mean values, it is a natural choice to consider the negative binomial loss function for model training. However, the LightGBM package (Ke et al., 2017) does not provide a built-in negative binomial loss function, but it provides functionality which supports user-defined loss functions.

In order to implement any customised loss, there are two functions we need to specify: an objective function and an evaluation function. The objective function is defined according to the log likelihood of a certain distribution, and the evaluation function returns the first and second derivatives w.r.t. model predictions.

For the negative binomial distribution, the probability mass function is given by

$$P(x \mid r, p) = \frac{\Gamma(r+x)}{\Gamma(r)\Gamma(x+1)} p^r (1-p)^x.$$

with a mean value μ that equals to $(1-p)r/p$. So if we substitute p w.r.t. μ, that is, $p = r/(\mu + r)$, we can get the following,

$$P(x \mid r, \mu) = \frac{\Gamma(r+x)}{\Gamma(r)\Gamma(x+1)} \left(\frac{r}{\mu+r}\right)^r \left(\frac{\mu}{\mu+r}\right)^x.$$

So, the negative log likelihood is given by

$$\begin{aligned} L(x \mid \mu, r) = & -\log \Gamma(r+x) + \log \Gamma(r) + \log \Gamma(x+1) \\ & - r \log r + r \log(\mu+r) - x \log \mu + x \log(\mu+r). \end{aligned}$$

And we denote the predicted mean value from the LightGBM model as f. As the support of the negative binomial distribution is the set of non-negative integers, we apply a log transformation so that f is allowed to take any real value and e^f is always non-negative. For data point x_i, treating x_i as the true value and plugging in the predicted mean value after transformation, i.e., e^{f_i}, then the negative log likelihood is given by,

$$\begin{aligned} L(x_i \mid f_i, r) = & -\log \Gamma(r+x_i) + \log \Gamma(r) + \log \Gamma(x_i+1) \\ & - r \log r + r \log(e^{f_i}+r) - x_i f_i + x_i \log(e^{f_i}+r). \end{aligned}$$

Consider $\mathbf{x} = (x_1, \dots, x_n)$ and $\mathbf{f} = (f_1, \dots, f_n)$; then our objective function is defined as

$$L(\mathbf{x} \mid \mathbf{f}, r) = \sum_{i=1}^{n} L(x_i; f_i, r).$$

And we calculate the gradient and Hessian w.r.t. f,

$$g(\mathbf{x} \mid \mathbf{f}, r) = \sum_{i=1}^{n} \left(\frac{e^{f_i}(r+x_i)}{e^{f_i}+r} - x_i \right),$$

$$h(\mathbf{x} \mid \mathbf{f}, r) = \sum_{i=1}^{n} \frac{e^{f_i} r(r+x_i)}{(e^{f_i}+r)^2}.$$

With this we have defined all the required functions for implementation, except that the value of r has to be obtained for completing the calculation. Intuitively, we can treat r as a model parameter and optimise it alongside the training process, but the LightGBM package does not provide an option for defining custom parameters. A possible solution, which is the solution we are using, is the coordinate-wise optimisation, that is, updating the model and r iteratively until convergence. We initialise the value of r by the method of moments from the historical data. The optimisation process of each iteration takes three steps: (1) train a LightGBM model with the custom loss function and the current value of r; (2) predict the training set with the model obtained and then get the predicted mean values; and (3) get an updated estimate of r by minimising the negative log likelihood, which is also the function L defined above. In this case, the LightGBM models are retrained iteratively through coordinate-wise optimisation and the optimisation procedure takes longer as the length of series grows, which in return leads to an overall longer training process.

Data availability

The data that has been used is confidential.

Table 6
The WSPL for the M5 dataset level 12 series on each category. The WSPL for all level 12 series are provided in the last column. The number of series in each category is provided in parenthesis.

Model	Smooth (1,900)	Erratic (863)	Lumpy (5,604)	Intermittent (22,123)	All
In-sample quantiles	0.2797	0.3982	0.4248	0.3314	0.3398
Emp-Wd	0.2734	0.3977	0.4263	0.3314	0.3387
Pois	0.3331	0.5251	0.5006	0.3696	0.3931
NB-CMP	0.2761	0.3981	0.4263	0.3343	0.3409
ZIP	0.2884	0.4752	0.4408	0.3317	0.3494
ZINB	0.3241	0.3965	0.4245	0.3314	0.3492
Direct LightGBM (max lags)	**0.2434**	0.3055	0.3333	0.2695	0.2766
Direct LightGBM (Fourier terms)	0.2510	0.3162	0.3368	0.2773	0.2839
DeepAR	0.2474	**0.3040**	**0.3298**	**0.2649**	**0.2742**
Negative binomial distribution assumption					
Lasso	0.2585	0.3320	0.3744	0.2820	0.2952
Pooled Regression	0.2628	0.3282	0.3725	0.2822	0.2957
LightGBM Neg. Bin. loss default	0.7434	0.8479	0.6625	0.7189	0.7232
LightGBM Poisson loss default	0.2719	0.3389	0.3843	0.2904	0.3048
LightGBM Tweedie loss default	0.2746	0.3609	0.3879	0.2941	0.3095
Poisson distribution assumption					
Lasso	0.3009	0.4049	0.4298	0.2977	0.3268
Pooled Regression	0.3035	0.3871	0.4225	0.2959	0.3239
LightGBM Neg. Bin. loss default	0.8211	0.9336	0.7121	0.7433	0.7670
LightGBM Poisson loss default	0.3126	0.3978	0.4318	0.3028	0.3319
LightGBM Tweedie loss default	0.3153	0.4248	0.4343	0.3065	0.3367

Table 7
Training time on the M5 dataset of the proposed model variants on level 10 and benchmarks directly modelling on level 12.

Model	Training time (minutes)
In-sample quantiles	0.53
Emp-Wd	11.99
Pois	0.61
NB-CMP	0.93
ZIP	18.20
ZINB	91.60
DeepAR	17.07
Direct LightGBM (max lags)	483.72
Direct LightGBM (Fourier terms)	489.47
Lasso	4.56
Pooled Regression	2.05
LightGBM Neg. Bin. loss default	68.90
LightGBM Tweedie loss default	1.09
LightGBM Poisson loss default	1.09

Table 8
Number of series on Level 9 to Level 12, and the percentage of series on the corresponding level that follow a negative binomial distribution (in percent) of the M5 dataset, as a trade-off to choose an aggregation level.

	Level 12	Level 11	Level 10	Level 9
Number of series	30,490	9,147	3,049	70
Series following neg. bin. dist. (%)	83.75	51.11	25.25	17.14

References

Agrawal, N., Smith, S.A., 1996. Estimating negative binomial demand for retail inventory management with unobservable lost sales. Naval Res. Logist. 43, 839–861.

Alexandrov, A., Benidis, K., Bohlke-Schneider, M., Flunkert, V., Gasthaus, J., Januschowski, T., Maddix, D.C., Rangapuram, S., Salinas, D., Schulz, J., Stella, L., Türkmen, A.C., Wang, Y., 2020. GluonTS: Probabilistic and neural time series modeling in Python. J. Mach. Learn. Res. 21, 1–6, URL: http://jmlr.org/papers/v21/19-820.html.

Bandara, K., Bergmeir, C., Smyl, S., 2020. Forecasting across time series databases using recurrent neural networks on groups of similar series: A clustering approach. Expert Syst. Appl. 140, 112896. http://dx.doi.org/10.1016/j.eswa.2019.112896.

Bandara, K., Hewamalage, H., Godahewa, R., Gamakumara, P., 2021. A fast and scalable ensemble of global models with long memory and data partitioning for the M5 forecasting competition. Int. J. Forecast. http://dx.doi.org/10.1016/j.ijforecast.2021.11.004.

Bassett, G., Koenker, R., 1982. An empirical quantile function for linear models with IID errors. J. Amer. Statist. Assoc. 77, 407–415.

Bojer, C.S., Meldgaard, J.P., 2021. Kaggle forecasting competitions: An overlooked learning opportunity. Int. J. Forecast. 37, 587–603. http://dx.doi.org/10.1016/j.ijforecast.2020.07.007.

Box, G.E.P., Jenkins, G.M., Reinsel, G.C., Ljung, G.M., 2015. Time Series Analysis: Forecasting and Control. John Wiley & Sons.

Chen, T., Guestrin, C., 2016. XGBoost: A scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, http://dx.doi.org/10.1145/2939672.2939785.

Chongsuvivatwong, V., 2022. epiDisplay: Epidemiological data display package. URL: https://CRAN.R-project.org/package=epiDisplay. R package version 3.5.0.2.

Cragg, J.G., 1971. Some statistical models for limited dependent variables with application to the demand for durable goods. Econometrica 39, 829. http://dx.doi.org/10.2307/1909582.

de Rezende, R., Egert, K., Marin, I., Thompson, G., 2021. A white-boxed ISSM approach to estimate uncertainty distributions of walmart sales. Int. J. Forecast. http://

dx.doi.org/10.1016/j.ijforecast.2021.11.006, URL: https://www.sciencedirect.com/science/article/pii/S0169207021001801.

Delignette-Muller, M.L., Dutang, C., 2015. fitdistrplus: An R package for fitting distributions. J. Stat. Softw. 64, 1–34. http://dx.doi.org/10.18637/jss.v064.i04.

Fildes, R., Kolassa, S., Ma, S., 2022a. Post-script—Retail forecasting: Research and practice. Int. J. Forecast. 38, 1319–1324. http://dx.doi.org/10.1016/j.ijforecast.2021.09.012.

Fildes, R., Ma, S., Kolassa, S., 2022b. Retail forecasting: Research and practice. Int. J. Forecast. 38, 1283–1318. http://dx.doi.org/10.1016/j.ijforecast.2019.06.004.

Gelman, A., Hill, J., 2006. Data Analysis Using Regression and Multilevel/Hierarchical Models. Cambridge University Press, http://dx.doi.org/10.1017/cbo9780511790942.

Gneiting, T., 2011. Quantiles as optimal point forecasts. Int. J. Forecast. 27, 197–207. http://dx.doi.org/10.1016/j.ijforecast.2009.12.015.

Godahewa, R., Bandara, K., Webb, G.I., Smyl, S., Bergmeir, C., 2021. Ensembles of localised models for time series forecasting. Knowl.-Based Syst. 233, 107518. http://dx.doi.org/10.1016/j.knosys.2021.107518.

Godahewa, R., Webb, G.I., Schmidt, D., Bergmeir, C., 2022. SETAR-tree: A novel and accurate tree algorithm for global time series forecasting. http://dx.doi.org/10.48550/ARXIV.2211.08661.

Han, X., Dasgupta, S., Ghosh, J., 2021. Simultaneously reconciled quantile forecasting of hierarchically related time series. In: International Conference on Artificial Intelligence and Statistics. PMLR, pp. 190–198.

Hasni, M., Aguir, M.S., Babai, M.Z., Jemai, Z., 2019. On the performance of adjusted bootstrapping methods for intermittent demand forecasting. Int. J. Prod. Econ. 216, 145–153.

Hasson, H., Wang, B., Januschowski, T., Gasthaus, J., 2021. Probabilistic forecasting: A level-set approach. Adv. Neural Inf. Process. Syst. 34, URL: https://github.com/awslabs/gluon-ts/blob/master/src/.

He, X., 1997. Quantile curves without crossing. Amer. Statist. 51, 186–192.

Heinen, A., 2003. Modelling time series count data: An autoregressive conditional Poisson model. Available at SSRN 1117187.

Hewamalage, H., Bergmeir, C., Bandara, K., 2022. Global models for time series forecasting: A simulation study. Pattern Recognit. 124, 108441. http://dx.doi.org/10.1016/j.patcog.2021.108441.

Hilbe, J.M., 2011. Negative Binomial Regression. Cambridge University Press.

Hyndman, R.J., Ahmed, R.A., Athanasopoulos, G., Shang, H.L., 2011. Optimal combination forecasts for hierarchical time series. Comput. Statist. Data Anal. 55, 2579–2589. http://dx.doi.org/10.1016/J.CSDA.2011.03.006.

Hyndman, R.J., Koehler, A.B., 2006. Another look at measures of forecast accuracy. Int. J. Forecast. 22, 679–688.

Hyndman, R., Koehler, A.B., Ord, J.K., Snyder, R.D., 2008. Forecasting with Exponential Smoothing: The State Space Approach. Springer Science & Business Media.

Jackman, S., 2024. pscl: Classes and Methods for R Developed in the Political Science Computational Laboratory. University of Sydney, Sydney, Australia, URL: https://github.com/atahk/pscl/. R package version 1.5.9.

Januschowski, T., Gasthaus, J., Wang, Y., Salinas, D., Flunkert, V., Bohlke-Schneider, M., Callot, L., 2020. Criteria for classifying forecasting methods. Int. J. Forecast. 36, 167–177. http://dx.doi.org/10.1016/j.ijforecast.2019.05.008.

Januschowski, T., Wang, Y., Torkkola, K., Erkkilä, T., Hasson, H., Gasthaus, J., 2021. Forecasting with trees. Int. J. Forecast. http://dx.doi.org/10.1016/j.ijforecast.2021.10.004.

Kaggle, 2018. Corporación favorita grocery sales forecasting. URL: https://www.kaggle.com/c/favorita-grocery-sales-forecasting.

Kamarthi, H., Kong, L., Rodríguez, A., Zhang, C., Prakash, B.A., 2022. PROFHIT: Probabilistic robust forecasting for hierarchical time-series. arXiv preprint arXiv:2206.07940.

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: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (Eds.), Advances in Neural Information Processing Systems, vol. 30. Curran Associates, Inc..

Koenker, R., Bassett, G., 1978. Regression quantiles. Econometrica 33–50.

Kolassa, S., 2016. Evaluating predictive count data distributions in retail sales forecasting. Int. J. Forecast. 32, 788–803. http://dx.doi.org/10.1016/j.ijforecast.2015.12.004.

Kolassa, S., 2022. Commentary on the M5 forecasting competition. Int. J. Forecast. 38, 1562–1568.

Kourentzes, N., Trapero, J.R., Barrow, D.K., 2020. Optimising forecasting models for inventory planning. Int. J. Prod. Econ. 225, 107597. http://dx.doi.org/10.1016/j.ijpe.2019.107597.

Kunz, M., Birr, S., Raslan, M., Ma, L., Januschowski, T., 2023. Deep learning based forecasting: A case study from the online fashion industry. In: Forecasting with Artificial Intelligence: Theory and Applications. Springer, pp. 279–311.

Lainder, A.D., Wolfinger, R.D., 2022. Forecasting with gradient boosted trees: Augmentation, tuning, and cross-validation strategies: Winning solution to the M5 uncertainty competition. Int. J. Forecast. http://dx.doi.org/10.1016/j.ijforecast.2021.12.003.

Lambert, D., 1992. Zero-inflated Poisson regression, with an application to defects in manufacturing. Technometrics 34, 1–14.

Makridakis, S., Petropoulos, F., Spiliotis, E., 2022a. Special Issue: M5 Competition, vol. 38. Int. J. Forecast..

Makridakis, S., Spiliotis, E., Assimakopoulos, V., 2022b. M5 accuracy competition: Results, findings, and conclusions. Int. J. Forecast. http://dx.doi.org/10.1016/j.ijforecast.2021.11.013.

Makridakis, S., Spiliotis, E., Assimakopoulos, V., Chen, Z., Gaba, A., Tsetlin, I., Winkler, R.L., 2021. The M5 uncertainty competition: Results, findings and conclusions. Int. J. Forecast. http://dx.doi.org/10.1016/j.ijforecast.2021.10.009.

Montero-Manso, P., Hyndman, R.J., 2021. Principles and algorithms for forecasting groups of time series: Locality and globality. Int. J. Forecast. 37, 1632–1653. http://dx.doi.org/10.1016/j.ijforecast.2021.03.004.

O'Hara-Wild, M., Hyndman, R., Wang, E., 2021. fable: Forecasting models for tidy time series. URL: https://CRAN.R-project.org/package=fable. R package version 0.3.1.

Olivares, K.G., Meetei, O.N., Ma, R., Reddy, R., Cao, M., Dicker, L., 2021. Probabilistic hierarchical forecasting with deep poisson mixtures. In: NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications. URL: https://www.amazon.science/publications/probabilstic-hierarchical-forecasting-with-deep-poisson-mixtures.

Panagiotelis, A., Gamakumara, P., Athanasopoulos, G., Hyndman, R.J., 2022. Probabilistic forecast reconciliation: Properties, evaluation and score optimisation. European J. Oper. Res. http://dx.doi.org/10.1016/j.ejor.2022.07.040.

Paria, B., Sen, R., Ahmed, A., Das, A., 2021. Hierarchically regularized deep forecasting. arXiv preprint arXiv:2106.07630.

Rangapuram, S.S., Werner, L.D., Benidis, K., Mercado, P., Gasthaus, J., Januschowski, T., 2021. End-to-end learning of coherent probabilistic forecasts for hierarchical time series. In: ICML 2021. URL: https://www.amazon.science/publications/end-to-end-learning-of-coherent-probabilistic-forecasts-for-hierarchical-time-series.

do Rego, J.R., De Mesquita, M.A., 2015. Demand forecasting and inventory control: A simulation study on automotive spare parts. Int. J. Prod. Econ. 161, 1–16.

Salinas, D., Flunkert, V., Gasthaus, J., Januschowski, T., 2020. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. Int. J. Forecast. 36, 1181–1191. http://dx.doi.org/10.1016/j.ijforecast.2019.07.001, URL: http://creativecommons.org/licenses/by/4.0/.

Sellers, K., Lotze, T., Raim, A., 2023. COMPoissonReg: Conway-Maxwell Poisson (COM-Poisson) regression. URL: https://CRAN.R-project.org/package=COMPoissonReg. R package version 0.8.1.

Shi, Y., Ke, G., Soukhavong, D., Lamb, J., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T.-Y., Titov, N., 2022. LightGBM: Light gradient boosting machine. URL: https://CRAN.R-project.org/package=lightgbm. R package version 3.3.2.

Simon, N., Friedman, J., Hastie, T., Tibshirani, R., 2011. Regularization paths for Cox's proportional hazards model via coordinate descent. J. Stat. Softw. 39, 1–13, URL: https://www.jstatsoft.org/v39/i05/.

Snyder, R.D., Ord, J.K., Beaumont, A., 2012. Forecasting the intermittent demand for slow-moving inventories: A modelling approach. Int. J. Forecast. 28, 485–496. http://dx.doi.org/10.1016/j.ijforecast.2011.03.009.

Spiliotis, E., Makridakis, S., Kaltsounis, A., Assimakopoulos, V., 2021. Product sales probabilistic forecasting: An empirical evaluation using the M5 competition data. Int. J. Prod. Econ. 240, 108237. http://dx.doi.org/10.1016/j.ijpe.2021.108237.

Stasinopoulos, D.M., Rigby, R.A., 2007. Generalized additive models for location scale and shape (GAMLSS) in R. J. Stat. Softw. 23, http://dx.doi.org/10.18637/jss.v023.i07.

Steutel, F.W., Van Harn, K., 2003. Infinite Divisibility of Probability Distributions on the Real Line. CRC Press.

Syntetos, A.A., Babai, M.Z., Gardner, E.S., 2015. Forecasting intermittent inventory demands: Simple parametric methods vs. bootstrapping. J. Bus. Res. 68, 1746–1752. http://dx.doi.org/10.1016/j.jbusres.2015.03.034.

Syntetos, A.A., Boylan, J.E., 2005. The accuracy of intermittent demand estimates. Int. J. Forecast. 21, 303–314. http://dx.doi.org/10.1016/j.ijforecast.2004.10.001.

Syntetos, M., Boylan, J., Croston, J.D., 2005. On the categorization of demand patterns. J. Oper. Res. Soc. 56, http://dx.doi.org/10.1057/palgrave.jors.2601841.

Taieb, S.B., Taylor, J.W., Hyndman, R.J., 2017. Coherent probabilistic forecasts for hierarchical time series. In: Precup, D., Teh, Y.W. (Eds.), Proceedings of the 34th International Conference on Machine Learning. In: Proceedings of Machine Learning Research, vol. 70, PMLR, pp. 3348–3357, URL: https://proceedings.mlr.press/v70/taieb17a.html.

Taieb, S.B., Taylor, J.W., Hyndman, R.J., 2020. Hierarchical probabilistic forecasting of electricity demand with smart meter data. J. Amer. Statist. Assoc. 116, 27–43. http://dx.doi.org/10.1080/01621459.2020.1736081.

Tibshirani, R., 1996. Regression shrinkage and selection via the Lasso. J. R. Stat. Soc. Ser. B Stat. Methodol. 58, 267–288, URL: http://www.jstor.org/stable/2346178.

Venables, W.N., Ripley, B.D., 2002. Modern Applied Statistics with S, fourth ed. Springer, New York, URL: https://www.stats.ox.ac.uk/pub/MASS4/. ISBN 0-387-95457-0.

Viswanathan, S., Zhou, C.X., 2008. A New Bootstrapping Based Method for Forecasting and Safety Stock Determination for Intermittent Demand Items. Working Paper, Nanyang Business School, Nanyang Technological University Singapore.

Willemain, T.R., Smart, C.N., Schwarz, H.F., 2004. A new approach to forecasting intermittent demand for service parts inventories. Int. J. Forecast. 20, 375–387. http://dx.doi.org/10.1016/S0169-2070(03)00013-X.

Zeileis, A., Kleiber, C., Jackman, S., 2008. Regression models for count data in R. J. Stat. Softw. 27, URL: https://www.jstatsoft.org/v27/i08/.

Zhou, C., Viswanathan, S., 2011. Comparison of a new bootstrapping method with parametric approaches for safety stock determination in service parts inventory systems. Int. J. Prod. Econ. 133, 481–485.

Ziel, F., 2021. M5 competition uncertainty: Overdispersion, distributional forecasting, GAMLSS, and beyond. Int. J. Forecast. http://dx.doi.org/10.1016/j.ijforecast.2021.09.008.