This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Probabilistic Forecasting With Modified N-BEATS Networks

Jente Van Belle, Ruben Crevits, Daan Caljon, and Wouter Verbeke, *Member, IEEE*

***Abstract*— In this article, we present a modification to the state-of-the-art N-BEATS deep learning architecture for the univariate time series point forecasting problem for generating parametric probabilistic forecasts. Next, we propose an extension to this probabilistic N-BEATS architecture to allow optimizing probabilistic forecasts from both a traditional forecast accuracy perspective as well as a forecast stability perspective, where the latter is defined in terms of a change in the forecast distribution for a specific time period caused by updating the probabilistic forecast for this time period when new observations become available (i.e., as time passes). We empirically show that this extension leads to more stable forecast distributions without causing considerable losses in forecast accuracy for the M4 monthly dataset. Finally, we present a second extension to the probabilistic N-BEATS network which makes it possible to jointly optimize single-period marginal and multiperiod cumulative (i.e., aggregated over multiple time periods) probabilistic forecasts. Empirical results are reported for the M4 monthly dataset and indicate that improvements in accuracy can be obtained over basic but well-established methods to produce probabilistic cumulative forecasts. The proposed probabilistic N-BEATS network and the extensions are all useful in a supply chain planning context.**

***Index Terms*— Deep learning, forecast instability, global models, supply chain planning, temporal aggregates.**

I. Introduction

THE center of attention in the time series forecasting literature is gradually shifting from point forecast to probabilistic forecast generation [1]. This is prompted by the fact that: 1) it is increasingly being recognized that forecasts ought to be probabilistic in nature to account for the uncertainty that is inherent to forecasting (in a world without uncertainty, there would simply be no need for forecasting) and that 2) one is typically also interested in the ability to correctly quantify the forecast uncertainty since it frequently is crucial to optimal decision-making (e.g., for safety stock calculations in inventory management [2], value-at-risk calculations in financial risk management [3], and managing power systems [4]). To quantify forecast uncertainty, a prediction interval can be estimated to indicate in which range the future observation will fall with a certain probability; however, to fully quantify this prediction uncertainty, one must produce entire predictive distributions, which is often referred to as probabilistic forecasting [5]. The M4 competition [6] was the first M competition which focused not only on point forecasts but also included 95% prediction intervals in the evaluation process. The M5 competition [7], [8] went a step further as it consisted of two parallel tracks using the same dataset, where the first track required 28 days ahead point forecasts and the second track required 28 days ahead forecasts of nine different quantiles (0.005, 0.025, 0.165, 0.250, 0.500, 0.750, 0.835, 0.975, and 0.995). By focusing on multiple quantiles in the evaluation process, covering the middle part of the forecast distribution as well as the tails, the objective in the second track of the M5 competition thus leans toward full probabilistic forecasting.

In this article, we present a modification to the state-of-the-art N-BEATS deep learning architecture proposed by Oreshkin et al. [9] for the univariate time series point forecasting problem to make it a parametric probabilistic forecasting method. In this regard, we build on the relatively recent trend in time series forecasting of using *global* forecasting models [10]. Following the global modeling paradigm (which is not limited to deep learning models only), a single forecasting model is built with a (typically very large) set of global model parameters that are estimated jointly across time series. Global models are particularly well-suited for operational forecasting problems with a short to medium-term forecast horizon when a large number of time series is available to train them [11]. Januschowski et al. [10] discussed the pros and cons of global forecasting models in detail (in comparison to traditional local forecasting where a model with a relatively small number of free parameters is built for each time series separately). Among other things, they explain the advantages that arise from the separation of the training (i.e., parameter estimation) and inference (i.e., generating forecasts from a trained model) steps. This separation allows the cost of training to be amortized over multiple forecast cycles, whereas these steps are typically conflated in popular implementations of local methods, where the model parameters need to be reestimated at each forecasting origin. Therefore, although training a global model typically takes significantly longer, the overall computational cost is not necessarily higher for operational forecasting problems compared to local methods.

Recently, global deep learning architectures have been proposed for both point and probabilistic forecasting.

Manuscript received 22 April 2023; revised 28 February 2024 and 13 July 2024; accepted 15 August 2024. This work was supported in part by the Research Foundation Flanders (FWO) under Project 12AZX24N and in part by Flanders Innovation and Entrepreneurship (VLAIO) and OMP under Project HBC.2016.0589. *(Corresponding author: Jente Van Belle.)*

Jente Van Belle was with the Faculty of Social Sciences and Solvay Business School, Department of Business Technology and Operations, Data Analytics Laboratory, Vrije Universiteit Brussel, 1050 Brussels, Belgium. He is now with the Faculty of Economics and Business, KU Leuven, 3000 Leuven, Belgium (e-mail: jente.vanbelle@kuleuven.be).

Ruben Crevits is with OMP, 2160 Wommelgem, Belgium.

Daan Caljon and Wouter Verbeke are with the Faculty of Economics and Business, KU Leuven, 3000 Leuven, Belgium.

Digital Object Identifier 10.1109/TNNLS.2024.3450832

2162-237X © 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on September 11,2024 at 08:33:10 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

For the point forecasting problem, the hybrid method combining exponential smoothing with advanced long short-term memory recurrent neural networks (RNNs) that won the M4 competition, is considered an important demonstration of the effectiveness of global deep learning models for time series forecasting [12], [13]. Oreshkin et al. [9] later showed that the generic configuration of their N-BEATS architecture, which can be considered a *pure* deep learning method since no time-series-specific ingredients are included, outperforms the hybrid approach of [12]. This finding led them to assert that "deep learning primitives are by themselves sufficient to solve a wide range of forecasting problems" (p. 1). For the probabilistic forecasting problem, various solutions have been put forward. The RNN-based parametric probabilistic forecasting method DeepAR, proposed by Salinas et al. [14], generates probabilistic forecasts by relying on maximum likelihood estimation to predict the parameters of a distribution, such as the normal or negative binomial distribution. Rangapuram et al. [15] extend this approach by combining state space models with deep RNNs. This hybrid approach allows for the incorporation of structural assumptions into the model and is shown to outperform DeepAR when there is limited data available. Other works adopt a nonparametric distribution-free approach by predicting a fixed preselected set of quantiles [16], [17] or the entire conditional quantile function [18], [19]. This approach can also be combined with conformal prediction [20] or used to obtain probabilistic forecasts from a point forecasting model by building quantile regressions on the residuals [21]. Finally, there is also recent work that presents a general approach to transform a global point forecast estimator (including deep learning models) into a probabilistic one, by grouping training data whose predictions are sufficiently close, and subsequently using the resulting bins of actual values in the training set as the predicted distributions [22].

In this work, we focus on parametric probabilistic forecasting to model (positive) real-valued time series. Several probability distributions can be used to model predictive distributions for such time series, including the normal (or Gaussian), student's T, and lognormal distributions. While the latter is the most appropriate option from a theoretical perspective to model real-valued time series with positive observations only (such as demand time series), we specifically opt to use the Gaussian distribution (even though its left tail extends to negative infinity) because of its useful properties in the context of two extensions to the probabilistic N-BEATS architecture presented in this work. More specifically, we will rely on the fact that closed-form solutions are available for the Kullback-Leibler (KL) divergence and Wasserstein-2 (W_2) distance between two Gaussians and that the Gaussian distribution is closed under addition. In terms of output distribution, the proposed probabilistic N-BEATS modification is similar to the Gaussian version of DeepAR [14]; however, DeepAR is a generative method based on autoregressive RNNs and therefore requires sequential sampling to obtain multistep ahead probabilistic forecasts, whereas the probabilistic N-BEATS modification is a discriminative sequence-to-sequence architecture which can be set up to directly produce multiple multistep ahead probabilistic forecasts.

Recently, Van Belle et al. [23] proposed an extension to the (standard) N-BEATS deep learning architecture to allow "optimizing forecasts from both a traditional forecast accuracy perspective as well as a forecast stability perspective," where the latter concerns "variability in forecasts for a specific time period induced by updating the forecast for this time period when new observations become available (i.e., as time passes)" (p. 1333). It was shown that the proposed methodology, which relies on the addition of a forecast instability component to the loss function, allows for improving both stability and accuracy and as such can be considered an alternative and effective regularization mechanism for generating point forecasts with global time series models. In a supply chain planning context, where demand forecasts are used as inputs to devise supply plans, using more stable forecasts leads to fewer and smaller revisions to these plans. Since these revisions induce certain supply chain costs, using more stable demand forecasts results in reduced costs. In this article, we investigate how this forecast (in)stability idea can be generalized to a probabilistic forecasting setting and propose a similar extension to the probabilistic N-BEATS modification.

Finally, we present a second extension to the probabilistic N-BEATS architecture to also produce probabilistic forecasts of temporal aggregates. In an inventory management context, specific quantiles of such probabilistic cumulative demand forecasts (aggregates along the time dimension) are of special interest as they provide valuable information for determining safety stock levels [24].

The presented research aligns with the topic of this special issue in that we leverage the flexibility offered by deep learning methods due to their native support for concurrently predicting multiple outputs which may be diverse in nature. More specifically, this allows incorporating domain-specific objectives (such as forecast stability and additional temporal aggregates) in the model optimization phase as these diverse outputs can directly be used in the (custom) loss function.

The remainder of this article is organized as follows. Section II presents how to modify the N-BEATS architecture in order to obtain parametric probabilistic forecasts. The forecast stability and temporal aggregation extensions to the probabilistic N-BEATS architecture are presented in Sections III and IV, respectively. Section V presents the results of an experimental study covering the different probabilistic N-BEATS variants. Finally, Section VI concludes this article and presents directions for future research.

II. N-N-BEATS: Modifying N-BEATS to Obtain Gaussian Probabilistic Forecasts

In this section, we present a modification to the N-BEATS deep learning architecture proposed in [9] for the univariate time series point forecasting problem to make it a parametric probabilistic forecasting method. We build on the generic configuration of the N-BEATS architecture. As discussed above, we focus on modeling real-valued time series in this work and use a Gaussian setup to this end, that is, forecasts are modeled as **n**ormal distributions. Hence,

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on September 11,2024 at 08:33:10 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

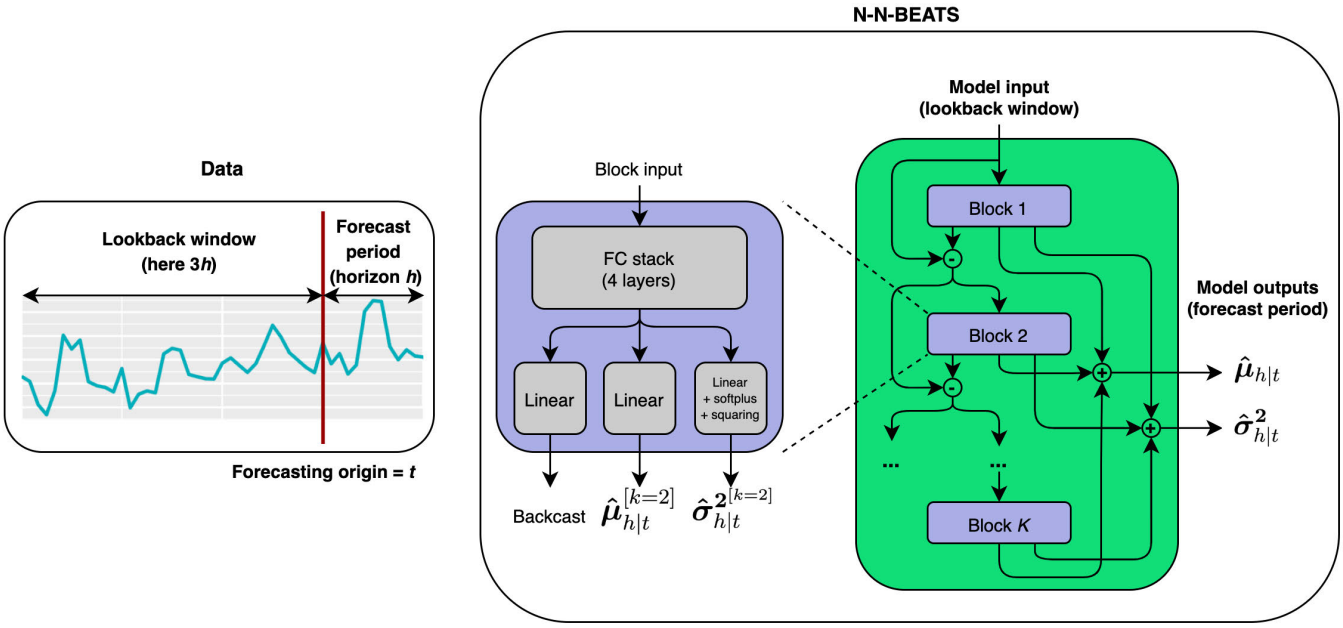

Fig. 1. N-N-BEATS architecture.

hereafter, we refer to this probabilistic N-BEATS modification as **N**-N-BEATS.

Like the standard N-BEATS network, N-N-BEATS (see Fig. 1) takes a vector of T historical observations as input, $\mathbf{x}_{T|t} \in \mathbb{R}^T = [y_{t-T+1}, \ldots, y_t]$ with length $T \leq t$ and forecasting origin t (i.e., the time period from which the forecasts are generated). We refer to this vector as the lookback window. Instead of producing point forecasts only, however, N-N-BEATS outputs both a vector of predicted means $\hat{\boldsymbol{\mu}}_{h|t} \in \mathbb{R}^h = [\hat{\mu}_{t+1|t}, \ldots, \hat{\mu}_{t+h|t}]$ and variances $\hat{\boldsymbol{\sigma}}^2_{h|t} \in \mathbb{R}^h = [\hat{\sigma}^2_{t+1|t}, \ldots, \hat{\sigma}^2_{t+h|t}]$, $\hat{\boldsymbol{\theta}}_{h|t} = (\hat{\boldsymbol{\mu}}_{h|t}, \hat{\boldsymbol{\sigma}}^2_{h|t})$, for the vector $\mathbf{y}_{h|t} \in \mathbb{R}^h = [y_{t+1}, \ldots, y_{t+h}]$ which comprises the next h observations of which each is thus modeled as a univariate normal distribution. A multiple of the forecast horizon h is used to determine the length T of the lookback window $\mathbf{x}_{T|t}$. In [9], lengths range from $2h$ to $7h$. The N-N-BEATS network, just like the standard N-BEATS network, comprises multiple sequentially connected processing blocks $k = 1, \ldots, K$ (with K being a hyperparameter) and also relies on the *doubly residual stacking* principle as proposed in [9]. In each block k, the input $\mathbf{x}^{[k]}_{T|t}$ is used to produce partial forecasts for both the mean $\hat{\boldsymbol{\mu}}^{[k]}_{h|t}$ and the variance $\hat{\boldsymbol{\sigma}}^{2^{[k]}}_{h|t}$, and a backcast $\hat{\mathbf{x}}^{[k]}_{T|t}$. These partial forecasts and the backcast are produced by going through a stack of four fully connected layers and a single task-specific layer for each block output (all using rectified linear unit (ReLU) activations), namely the backcast and the (partial) mean and variance forecasts. To obtain the latter, the linear task-specific layer outputs a vector containing predictions for the standard deviations, which is then transformed by going through a softplus activation $f(x) = \ln(1 + e^x)$ in order to ensure that $\hat{\sigma}^{[k]}_{t+i|t} > 0$ for $i = 1, \ldots, h$, after which these values are squared to serve as partial variance forecasts. The difference between the block input $\mathbf{x}^{[k]}_{T|t}$ and the backcast $\hat{\mathbf{x}}^{[k]}_{T|t}$ is used as input for the next block in the network (with $\mathbf{x}^{[1]}_{T|t} = \mathbf{x}_{T|t}$), and the final mean and variance forecasts are obtained by summing their block forecasts

$$\mathbf{x}^{[k]}_{T|t} = \mathbf{x}^{[k-1]}_{T|t} - \hat{\mathbf{x}}^{[k-1]}_{T|t} \tag{1}$$

$$\hat{\boldsymbol{\mu}}_{h|t} = \sum_{k=1}^{K} \hat{\boldsymbol{\mu}}^{[k]}_{h|t} \tag{2}$$

$$\hat{\boldsymbol{\sigma}}^2_{h|t} = \sum_{k=1}^{K} \hat{\boldsymbol{\sigma}}^{2^{[k]}}_{h|t}. \tag{3}$$

For the sake of clarity, block-level indexing is omitted in the remainder of this work.

Given a training set $\mathcal{D} = \{\mathbf{x}^j_{T|t}, \mathbf{y}^j_{h|t}\}$, obtained by sampling input–output windows across *scaled* time series and forecasting origins,[1] the parameters $\mathbf{W}$ of an N-N-BEATS network $f(\mathbf{x}_{T|t}; \mathbf{W})$ are optimized by maximizing the mean of the Gaussian log-likelihoods of the h univariate predictive distributions for the demands per period in $\mathbf{y}_{h|t}$. Given that the log-likelihood function of a normal distribution with (predicted) mean $\hat{\mu}_{t+i|t}$ and variance $\hat{\sigma}^2_{t+i|t}$, and with $\hat{\theta}_{t+i|t} = (\hat{\mu}_{t+i|t}, \hat{\sigma}^2_{t+i|t})$, is

$$\ln \mathcal{L}\left(\hat{\theta}_{t+i|t} \middle| y_{t+i}\right) = \ln p\left(y_{t+i} \middle| \hat{\theta}_{t+i|t}\right) \tag{4}$$

$$= -\frac{1}{2}\left(\ln\left(2\pi\hat{\sigma}^2_{t+i|t}\right) + \frac{\left(y_{t+i} - \hat{\mu}_{t+i|t}\right)^2}{\hat{\sigma}^2_{t+i|t}}\right) \tag{5}$$

with $\mathcal{L}(\cdot)$ and $p(\cdot)$ denoting the likelihood and density function of a normal distribution, respectively, the optimization problem

[1]Note that the index t is used to refer to the forecasting origin, irrespective of the training sample's absolute time period.

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on September 11,2024 at 08:33:10 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

can be formulated as follows:

$$\mathbf{W}^* = \arg\max_{\mathbf{W}} \sum_j \left(\frac{1}{h} \sum_{i=1}^{h} \ln \mathcal{L}\left(\hat{\theta}^j_{t+i|t} \middle| y^j_{t+i} \right) \right)$$

with

$$\hat{\boldsymbol{\theta}}^j_{h|t} = f\left(\mathbf{x}^j_{T|t}; \mathbf{W} \right). \tag{6}$$

The model parameters $\mathbf{W}$ are optimized numerically by minimizing the negative of (6) via stochastic gradient descent. This negative-log-likelihood-based optimization is similar to how the parameters of DeepAR [14] are optimized.

With such likelihood-based optimization, in order for the time series to have an equally large effect on the optimization of the network parameters, we rescale the time series in a preprocessing step. In our experiments, we divide each time series by its average value (computed on the training and validation datasets).

III. Improving the Stability of N-N-BEATS Forecasts

Demand forecasts are typically produced and updated each time a new observation becomes available, that is, demand forecasting is a rolling forecasting exercise. In this setting, forecasts can be evaluated from both a forecast accuracy and (in)stability perspective, where the latter concerns "variability in forecasts for a specific time period induced by updating the forecast for this time period when new observations become available" [23, p. 1333]. In a supply chain planning context, updating forecasts results in both benefits and costs. An increase in forecast accuracy due to a shorter forecast horizon causes benefits, while induced forecast instability potentially causes costs since adjustments to existing supply plans are possibly needed [25], [26]. In addition, forecast instability can also cause (nontechnical) forecast consumers to lose trust in a forecasting system, potentially leading to the use of unwarranted judgmental adjustments [27]. If these adjustments lead to a decrease in forecast accuracy, this may be a worse outcome than the use of a trusted system that generates more stable forecasts, even if accuracy is lower than an alternative system that generates less stable forecasts. Hence, this may justify more stable but less accurate algorithmic forecasts.

Van Belle et al. [23] proposed an extension to the N-BEATS deep learning architecture for the univariate time series point forecasting problem [9] to improve forecast stability by adding a forecast instability component to the loss function. The forecast instability component presented aims to minimize the differences between the point forecasts made at time t and $t-1$ for the same (overlapping) time periods $t+i$ with $i = 1, \ldots, h-1$ and as such relies on the use of a (second) lagged input–output pair for each input–output sample j. Their experimental results show that the methodology allows for improving forecast stability as well as forecast accuracy. For global time series point forecasting methods, the forecast instability component can thus also be used as a regularization mechanism.

The above forecast stability mechanism can be translated to a probabilistic forecasting setting as presented in Section II by minimizing the dissimilarities between the probabilistic forecasts made at time t and $t-1$ for the same (overlapping) time periods $t+i$ with $i = 1, \ldots, h-1$. There are many ways to define dissimilarities between two probability distributions P_{t+i} and Q_{t+i}, with densities $p(\hat{y}_{t+i})$ and $q(\hat{y}_{t+i})$, and cumulative distribution functions (cdfs) $P(\hat{y}_{t+i})$ and $Q(\hat{y}_{t+i})$. Two measures that can be used are the KL divergence [28] and the W_2 distance (see [29]). KL builds on the notion of entropy from information theory and is defined as follows for continuous distributions:

$$\mathrm{KL}(P_{t+i} || Q_{t+i}) = -\int_{-\infty}^{\infty} p(\hat{y}_{t+i}) \ln\left(\frac{q(\hat{y}_{t+i})}{p(\hat{y}_{t+i})} \right) d\hat{y}_{t+i}. \tag{7}$$

Wasserstein metrics find their origin in optimal transport theory. The squared W_2 distance for 1-D continuous distributions is defined as follows:

$$W_2^2(P_{t+i}, Q_{t+i}) = \int_0^1 \left| P^{-1}(\alpha) - Q^{-1}(\alpha) \right|^2 d\alpha \tag{8}$$

where $P^{-1}(\alpha)$ and $Q^{-1}(\alpha)$ are the quantile functions. KL and (squared) W_2 are two appealing options for our purpose, i.e., to quantify the dissimilarity between two univariate Gaussians, $P_{t+i} \sim \mathcal{N}(\hat{\mu}_{t+i|t}, \hat{\sigma}_{t+i|t})$ and $Q_{t+i} \sim \mathcal{N}(\hat{\mu}_{t+i|t-1}, \hat{\sigma}_{t+i|t-1})$, since both have a closed-form solution in this setting

$$\mathrm{KL}(P_{t+i} || Q_{t+i}) = -\frac{1}{2} + \ln\left(\frac{\hat{\sigma}_{t+i|t-1}}{\hat{\sigma}_{t+i|t}} \right) + \frac{\hat{\sigma}^2_{t+i|t} + \left(\hat{\mu}_{t+i|t} - \hat{\mu}_{t+i|t-1} \right)^2}{2\hat{\sigma}^2_{t+i|t-1}} \tag{9}$$

$$W_2^2(P_{t+i}, Q_{t+i}) = \left(\hat{\mu}_{t+i|t} - \hat{\mu}_{t+i|t-1} \right)^2 + \left(\hat{\sigma}_{t+i|t} - \hat{\sigma}_{t+i|t-1} \right)^2. \tag{10}$$

Both $\mathrm{KL}(P_{t+i} || Q_{t+i})$ and $W_2(P_{t+i}, Q_{t+i}) \geq 0$ with equality if, and only if, $p(\hat{y}_{t+i}) = q(\hat{y}_{t+i})$. Note, however, that KL is not a symmetrical quantity, that is to say $\mathrm{KL}(P_{t+i} || Q_{t+i}) \neq \mathrm{KL}(Q_{t+i} || P_{t+i})$ and therefore does not qualify as a distance *metric*, in contrast to the W_2 distance. To deal with the asymmetric behavior of KL, we take the mean of $\mathrm{KL}(P_{t+i} || Q_{t+i})$ and $\mathrm{KL}(Q_{t+i} || P_{t+i})$ to characterize the dissimilarity between P_{t+i} and Q_{t+i}. A second difference between the two dissimilarity measures is that, ceteris paribus, KL induces a bias toward larger variance estimates as compared to W_2. In the limiting case of $\hat{\sigma}_{t+i|t-1}$ approaching zero, $\mathrm{KL}(P_{t+i} || Q_{t+i})$ diverges to infinity implying that singular Gaussians are infinitely far from all other Gaussians. In Table I and Fig. 2, we illustrate the impact of this second difference on KL and W_2 ranks for a toy example.

In a probabilistic forecasting setting, an alternative to considering the stability of entire distributional forecasts is to only focus on the (in)stability of the mean forecasts. To this end, similar to what is proposed in [23], we can use squared changes in mean forecasts $(\hat{\mu}_{t+i|t} - \hat{\mu}_{t+i|t-1})^2$ to quantify the dissimilarity between P_{t+i} and Q_{t+i}.

By defining a forecast instability component Ω in terms of one of the dissimilarity measures above

$$\Omega_{\mathrm{KL}}\left(\hat{\boldsymbol{\theta}}_{h|t-1}, \hat{\boldsymbol{\theta}}_{h|t} \right) = \frac{1}{(h-1)} \sum_{i=1}^{h-1} \frac{1}{2} \left(\mathrm{KL}(P_{t+i} || Q_{t+i}) + \mathrm{KL}(Q_{t+i} || P_{t+i}) \right) \tag{11}$$

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on September 11,2024 at 08:33:10 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE I

TOY EXAMPLE: DISSIMILARITY COMPARED TO $\mathcal{N}(\mu = 10, \sigma = 2)$ IN TERMS OF KL DIVERGENCE AND W_2 DISTANCE. THE DISTRIBUTIONS FOR THE COLORED PARAMETER VALUES ARE VISUALIZED IN FIG. 2. THE KL AND W_2 RANKS (LOWER IS BETTER) SHOW THAT KL INDUCES A BIAS TOWARD LARGER VARIANCE ESTIMATES AS COMPARED TO W_2

μ	σ	KL	W_2	KL rank	W_2 rank
10	**2.0**	0.00	0.00	1	1
10	0.1	99.50	1.90	9	5
10	1.5	0.09	0.50	3	2
10	2.5	0.05	0.50	2	2
10	3.0	0.17	1.00	4	4
12	0.1	199.75	2.76	10	10
12	**1.5**	0.78	2.06	8	7
12	**2.0**	0.50	2.00	6	6
12	**2.5**	0.46	2.06	5	7
12	**3.0**	0.53	2.24	7	9

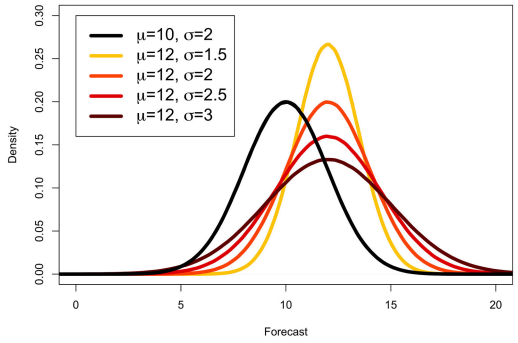

Fig. 2. Visualizations of the Gaussian distributions for the colored parameter values used in the toy example presented in Table I. While the **yellow** and **dark red** distributions are equally different from $\mathcal{N}(\mu = 10, \sigma = 2)$ (represented by the **black** line) in terms of W_2, in terms of KL the latter is more similar.

$$\Omega_{W_2}\left(\hat{\boldsymbol{\theta}}_{h|t-1}, \hat{\boldsymbol{\theta}}_{h|t}\right) = \sqrt{\frac{1}{(h-1)} \sum_{i=1}^{h-1} W_2^2(P_{t+i}, Q_{t+i})} \tag{12}$$

$$\Omega_{\text{RMSC}}\left(\hat{\boldsymbol{\theta}}_{h|t-1}, \hat{\boldsymbol{\theta}}_{h|t}\right) = \sqrt{\frac{1}{(h-1)} \sum_{i=1}^{h-1} \left(\hat{\mu}_{t+i|t} - \hat{\mu}_{t+i|t-1}\right)^2} \tag{13}$$

we can augment the optimization problem from (6) as follows:

$$\mathbf{W}^* = \arg\max_{\mathbf{W}} \sum_j \Bigg(\frac{1}{2h} \sum_{i=1}^{h} \Big(\ln \mathcal{L}\left(\hat{\theta}^j_{t+i|t} \middle| y^j_{t+i}\right) + \ln \mathcal{L}\left(\hat{\theta}^j_{t-1+i|t-1} \middle| y^j_{t-1+i}\right) \Big) - \lambda \Omega\left(\hat{\boldsymbol{\theta}}^j_{h|t-1}, \hat{\boldsymbol{\theta}}^j_{h|t}\right) \Bigg)$$

with

$$\hat{\boldsymbol{\theta}}^j_{h|t} = f\left(\mathbf{x}^j_{T|t}; \mathbf{W}\right), \quad \hat{\boldsymbol{\theta}}^j_{h|t-1} = f\left(\mathbf{x}^j_{T|t-1}; \mathbf{W}\right) \tag{14}$$

with $\lambda \geq 0$ controlling the impact of the forecast instability objective on the probabilistic forecasts produced by the network. An N-N-BEATS network that takes into account a forecast instability component in the optimization is referred to as N-N-BEATS-S-KL, N-N-BEATS-S-W_2, or N-N-BEATS-S-RMSC hereafter.

IV. EXTENDING THE N-N-BEATS ARCHITECTURE FOR INVENTORY MANAGEMENT

In this section, we present an extension to the N-N-BEATS architecture presented in Section II to also produce probabilistic forecasts of temporal aggregates. A visualization of this extended architecture, to which we refer as N-N-BEATS-C hereafter, is provided in Fig. 3. This extension is useful in an operational demand forecasting context in which forecasts are needed as input to supply chain planning decisions, both at the level of stock-keeping units (SKUs) as well as for aggregates across product and/or spatial dimensions. In that regard, most supply chain planning systems still solely rely on weekly or monthly point forecasts (of expected demand) as inputs since they can readily be used as base forecasts in the construction of different aggregates over time, product, and/or spatial dimensions. Prediction intervals around these weekly or monthly point forecasts are useful, for instance, for managing inventory of newsvendor-type SKUs, and more generally, to point out to planners that forecast uncertainty should be taken into account; however, for inventory management at SKU-level, one is often interested in a specific quantile (dependent on the target service level) of the predictive distribution of cumulative demand $\sum_{i=1}^{L} y_{t+i}$ during a lead time of length $L \geq 1$ (see [30]). The latter explicitly requires that also the dependence structure between the probabilistic forecasts for periods $t+i$ with $i = 1, \ldots, L$ is modeled, which is exactly how N-N-BEATS-C extends N-N-BEATS.

Let us denote with $\mathbf{y}_{L|t}$ the vector of cumulative demands with $L = 1, \ldots, h$: $\mathbf{y}_{L|t} \in \mathbb{R}^h = [y_{t+1}, \ldots, \sum_{i=1}^{h} y_{t+i}]$. Further, recall that in N-N-BEATS, the predictive (marginal) distributions for the demands per period in $\mathbf{y}_{h|t}$ are modeled as Gaussians. If we assume that the predictive distributions for the demands per period are also jointly normal (i.e., we assume that dependence between the Gaussian marginals can be modeled using the Gaussian copula), it follows that the predictive distributions for the cumulative demands in $\mathbf{y}_{L|t}$ are normal distributions as well for which $\hat{\boldsymbol{\mu}}_{L|t}$ can simply be obtained by summing the relevant mean forecasts in $\hat{\boldsymbol{\mu}}_{h|t}$: $\hat{\boldsymbol{\mu}}_{L|t} \in \mathbb{R}^h = [\hat{\mu}_{t+1|t}, \ldots, \sum_{i=1}^{h} \hat{\mu}_{t+i|t}]$, while the variances for the cumulative demands $\hat{\boldsymbol{\sigma}}^2_{L|t} \in \mathbb{R}^h = [\hat{\sigma}^2_{t+1|t}, \ldots, \hat{\sigma}^2_{t+1,\ldots,t+h|t}]$, are added to the network outputs

$$\hat{\boldsymbol{\sigma}}^2_{L|t} = \sum_{k=1}^{K} \hat{\boldsymbol{\sigma}}^{2^{[k]}}_{L|t} \tag{15}$$

since the variance forecasts in $\hat{\boldsymbol{\sigma}}^2_{h|t}$ are not additive due to potential correlations.

The parameters $\mathbf{W}$ of an N-N-BEATS-C network are optimized by jointly maximizing: 1) the mean of the Gaussian log-likelihoods of the predictive distributions for the demands per period and 2) the mean of the Gaussian log-likelihoods of the cumulative predictive distributions which are rescaled to ensure that the different lead times ($L = 1, \ldots, h$) have an equally large effect on the optimization of the network parameters. Given that the Gaussian log-likelihood function,

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on September 11,2024 at 08:33:10 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

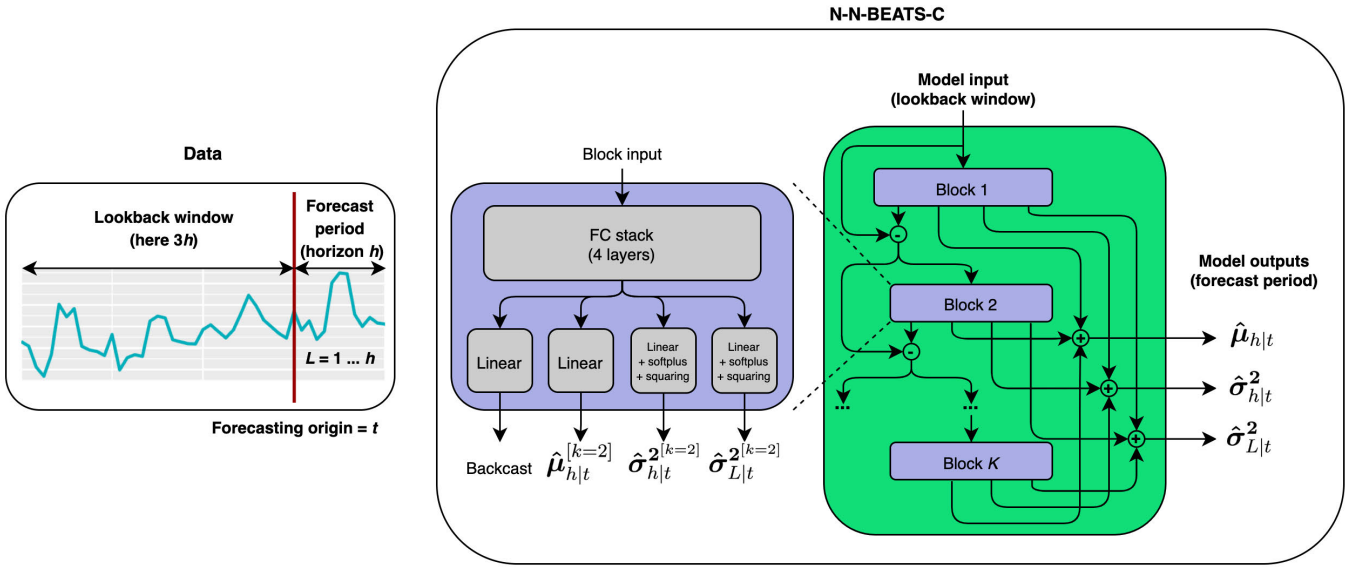

Fig. 3. N-N-BEATS-C architecture.

as defined in (5), is also applicable to the cumulative predictive distributions by replacing $\hat{\theta}_{t+i|t}$ and y_{t+i} by their respective cumulative counterparts $\hat{\theta}^{C}_{t+L|t} = (\sum_{i=1}^{L} \hat{\mu}_{t+i|t}, \hat{\sigma}^{2}_{t+1,\ldots,t+L|t})$ and $\sum_{i=1}^{L} y_{t+i}$, the optimization problem can be formulated as follows:

$$\mathbf{W}^{*} = \arg\max_{\mathbf{W}} \sum_{j} \left(\frac{1}{h} \sum_{i=1}^{h} \ln \mathcal{L}\left(\hat{\theta}^{j}_{t+i|t} \middle| y^{j}_{t+i} \right) + \frac{1}{h} \sum_{L=1}^{h} \frac{1}{L} \ln \mathcal{L}\left(\hat{\theta}^{C^{j}}_{t+L|t} \middle| \sum_{i=1}^{L} y^{j}_{t+i} \right) \right)$$

with

$$\left(\hat{\boldsymbol{\theta}}^{j}_{h|t}, \hat{\boldsymbol{\sigma}}^{2j}_{L|t} \right) = f\left(\mathbf{x}^{j}_{T|t}; \mathbf{W} \right). \tag{16}$$

V. Experimental Study

In this section, we report on the results of two different experiments to assess the performance of the different N-N-BEATS variants presented in this work. In the first experiment, we focus on the performance of the probabilistic forecasts for individual time periods produced by the different N-N-BEATS variants both in terms of forecast accuracy and forecast stability, while in the second experiment, we focus on evaluating the accuracy of the cumulative forecast distributions as output by N-N-BEATS-C. First, a detailed description is provided of the experimental design(s), including the dataset that is used, the evaluation schemes that are adopted, and an overview of the forecasting methods and training methodology that are used. Subsequently, we report and discuss the experimental results for both experiments.

A. Experimental Design

1) Dataset: In a supply chain context where forecast instability directly affects operations and therefore implies costs, monthly forecasts are (still) widely used to manage operations. Therefore, we use the monthly M4 [6] dataset in our experiments. Summary statistics are provided in Table II. All series have positive observed values at all time steps.

TABLE II
Summary Statistics of the Monthly M4 Dataset

No. of series	48,000
Min. length	60
Max. length	2812
Mean length	234.3
Std. dev. length	137.4

2) Evaluation Schemes: For the monthly M4 dataset, the evaluation scheme defined for use in the original competition required a single trace of 1–18-step ahead forecasts for out-of-sample evaluation. That is, the test set comprises the last 18 data points of each time series for which forecasts need to be generated from a single forecasting origin. However, we need forecasts from at least two different forecasting origins since we want to evaluate forecast (in)stability next to forecast accuracy. Therefore, we adopt the same evaluation scheme as used in [23]. More specifically, for the first experiment, we use a rolling forecasting origin evaluation [31] to evaluate one- to six-month ahead forecasts on the same test set that is used in the original competition (resulting in 13 traces of one- to six-month ahead forecasts). For the second experiment, a similar evaluation scheme is adopted, with the only difference being that the one- to six-month ahead monthly forecasts are replaced by one- to six-month ahead cumulative forecasts.

Below, we list the performance measures used to evaluate forecast accuracy in both experiments and forecast stability in the first experiment. All formulas are expressed in terms of probabilistic forecast evaluation per period, however, all measures can also be used to evaluate cumulative forecast distributions.

1) *Forecast Accuracy Measures:*

a) *RMSSE:* The root mean squared scaled error (RMSSE) for a specific input–output sample j with

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on September 11,2024 at 08:33:10 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

forecasting origin t is calculated as follows [32]:

$$\text{RMSSE} = \sqrt{\frac{\frac{1}{h}\sum_{i=1}^{h}\left(y_{t+i} - \hat{\mu}_{t+i|t}\right)^2}{\frac{1}{(t-1)}\sum_{i=1}^{t-1}(y_{t-i+1} - y_{t-i})^2}}. \quad (17)$$

This measure only takes into account the accuracy of the mean forecasts.

b) *sPL90:* Let $\Lambda_\alpha(\hat{q}_{t+i|t}, y_{t+i})$ denote the α-quantile loss function, also known as the pinball loss, which is defined as [33]

$$\Lambda_\alpha\left(\hat{q}_{t+i|t}, y_{t+i}\right) = \left(\alpha - \mathcal{I}_{[y_{t+i} < \hat{q}_{t+i|t}]}\right)\left(y_{t+i} - \hat{q}_{t+i|t}\right) \quad (18)$$

with $\mathcal{I}_{[y_{t+i} < \hat{q}_{t+i|t}]}$ the indicator function taking the value 1 if $y_{t+i} < \hat{q}_{t+i|t}$ and 0 otherwise, $0 < \alpha < 1$ the quantile level of interest, $\hat{q}_{t+i|t}$ the predicted αth quantile and y_{t+i} the value that materializes. sPL90 for a specific input–output sample j with forecasting origin t is then computed as follows:

$$\text{sPL90} = \frac{\frac{1}{2}\left(\frac{1}{h}\sum_{i=1}^{h}\Lambda_{0.05} + \frac{1}{h}\sum_{i=1}^{h}\Lambda_{0.95}\right)}{\frac{1}{(t-1)}\sum_{i=1}^{t-1}|y_{t-i+1} - y_{t-i}|}. \quad (19)$$

sPL90 provides a scale-free evaluation of a combination of the 0.05 and 0.95 quantile forecasts.

c) *sCRPS:* The continuous ranked probability score (CRPS) can be defined as follows[2] [34]:

$$\text{CRPS}(F, y_{t+i}) = \int_{-\infty}^{+\infty}\left[F\left(\hat{y}_{t+i}\right) - \mathcal{I}_{[y_{t+i} < \hat{y}_{t+i}]}\right]^2 d\hat{y}_{t+i} \quad (20)$$

with F a predictive cdf and where $\mathcal{I}_{[y_{t+i} < \hat{y}_{t+i}]}$ takes the value 1 if $y_{t+i} < \hat{y}_{t+i}$ and 0 otherwise. When F is the cdf of a normal distribution, a closed-form solution to the integral is available (see [34]). Intuitively, one can think of the CRPS as the quantile loss integrated over all quantile levels $\alpha \in [0, 1]$ [18], [35]. Since the CRPS reduces to the absolute error for deterministic predictions, the average CRPS for an input–output sample j with forecasting origin t is scaled in the same way as in the sPL90 measure to arrive at the sCRPS.

2) *Forecast Stability Measures:*

a) *RMSSC:* Similar to RMSSE defined above, we compute the root mean squared scaled change (RMSSC) for a specific input–output sample j and forecasting origins $t-1$ and t as in the right-hand side of (13), but with the square root taken only after the mean squared change is divided by $(1/(t-1))\sum_{i=1}^{t-1}(y_{t-i+1} - y_{t-i})^2$.

b) *KL:* For a specific input–output sample j and forecasting origins $t-1$ and t we compute KL as in the right-hand side of (11).

c) *sW$_2$:* A scaled version of the W_2 distance is computed analogous to the RMSSC described above [and based on the right-hand side of (12)].

We first compute for each time series separately the performance measures by averaging their values across forecasting origins. In the second step, we average across time series to obtain dataset-level performance measures. For robustness, results are also reported in terms of median performance across time series [36] in the Appendix.

To check whether the reported performance differences are statistically significant, we visualize the results of multiple comparisons with the best (MCB) tests [37]. Essentially, MCB tests are used to determine whether the average rank (across time series) of each method is significantly different from those of other methods. In the MCB plots, if the intervals of two methods do not overlap, this indicates statistically different performances.

3) Forecasting Methods: In addition to reporting the results of the different N-N-BEATS variants presented in this work, we also report the results of Gaussian probabilistic forecasts obtained by using the `ets` function from the `forecast` R package [38], [39]. This function implements an automated model selection procedure for local exponential smoothing models by relying on a state-space representation [40], [41]. We refer to these forecasts as ETS. In addition, as a global method for comparison, we report the results for a Gaussian DeepAR model [14]. This method, like N-N-BEATS, generates probabilistic forecasts by predicting the parameters of Gaussian distributions. However, due to its autoregressive nature, to generate multistep ahead forecasts, we need to sample multiple times from the estimated distribution to generate a series of sample paths. The means and variances of these sample paths are then used as the mean and variance forecasts.

For the second experiment, the performance of the cumulative probabilistic forecasts output by N-N-BEATS-C is also compared with the performance of Gaussian ETS and DeepAR forecasts. To obtain cumulative probabilistic ETS forecasts, three different variance estimation approaches are used (mean forecasts are simply aggregated over the lead time).

1) *ETS-C1:* $\hat{\sigma}^2_{t+1,\ldots,t+L|t} = L\hat{\sigma}^2_{\text{in-sample}}$ with $\hat{\sigma}^2_{\text{in-sample}}$ being the mse of the one-step ahead in-sample forecast errors. This approach is built on the assumption that all the forecast errors in the lead time are independent and identically distributed (i.i.d.) and follow a zero-mean Gaussian distribution. This variance estimation approach was criticized in, e.g., [42] as it ignores, by construction: 1) potential correlations between the errors of the different forecasts in the lead time and 2) the usual observation of increasing forecast errors for longer forecast horizons. Barrow and Kourentzes [43] empirically show that out-of-sample forecast uncertainty is poorly approximated when relying on this estimation procedure.
2) *ETS-C2:* $\hat{\sigma}^2_{t+1,\ldots,t+L|t} = \sum_{i=1}^{L}\hat{\sigma}^2_{t+i|t}$. This approach still assumes that all the forecast errors in the lead time are independent but is able to capture that forecast errors usually increase for longer forecast horizons.
3) *ETS-C3:* $\hat{\sigma}^2_{t+1,\ldots,t+L|t}$ is estimated from 1000 simulated future sample paths (from which we can easily compute

[2]We define the CRPS as a loss function in negative orientation (i.e., lower is better) but positively oriented definitions can be found in the literature as well (see [5]).

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on September 11,2024 at 08:33:10 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE III
HYPERPARAMETER SETTINGS FOR N-N-BEATS MODELS

No. of blocks K	20
Hidden layer width	256
Batch size	512
Lookback window length T	$4h$
Forecasting origin range	$10h$
Iterations	14,415
Learning rate	5e-4

temporal aggregates) of the automatically specified ETS model. Future sample paths are simulated in an iterative manner by sampling from a normal distribution to obtain values for the error terms. Next to the feature of increasing forecast errors for longer forecast horizons, this approach also allows capturing correlations between the forecast errors in the lead time. Note that for this ETS variant, also the cumulative means are estimated from the simulated sample paths.

To obtain the cumulative DeepAR forecasts, samples of sums are obtained from a set of generated sample paths by summing each forecast in the lead time. Means and variances of these samples of sums are used as the mean and variance forecasts.

Other approaches that allow capturing correlations between forecast errors in the lead time are proposed by Trapero et al. [44] as alternatives to sampling-based methods such as ETS-C3 and DeepAR. More specifically, their proposed data-driven approaches rely on a collection of out-of-sample forecast errors for a specific lead time to directly estimate cumulative forecast uncertainty. This requires that sufficient out-of-sample lead time forecast errors are available for each time series to enable robust variance estimation. In practice, however, this is not always evident or even possible due to, e.g., a relatively short product life cycle or even the use of monthly forecasting buckets (in combination with limited time series lengths). Since the latter is the case in this study, these empirical approaches are not included in our experimental results.

4) Training Methodology: The dataset is split into a train, validation, and test set. Both the validation and test sets contain 18 observations of each time series, where the observations of the validation set are observed just before the test set data points which are the last 18 observations for each time series. The rolling origin evaluation procedure is used for performance evaluation on both the validation and test sets.

Table III provides an overview of the values for the hyperparameters used for training the N-N-BEATS networks. These values are based on the hyperparameter values reported in [23] used to train a standard N-BEATS network on the monthly M4 dataset. Validation sCRPS results were used to validate these hyperparameter values (by comparing the results of alternative hyperparameter specifications) and only led to fine-tuning of the learning rate. For DeepAR, we used the PyTorch implementation in GluonTS [45]. We tuned the following hyperparameters: context length (32), number of RNN layers (3), number of RNN cells for each layer (40), epochs (1000), learning rate (1e-3), batch size (32), and number of batches in each training epoch (100). For all other hyperparameters, we used the default values. After specifying the hyperparameters, models are trained on the full train set (including the validation set), and results are reported on the test set.

Only the additional hyperparameters are tuned for N-N-BEATS extensions (i.e., the values in Table III are used for the common hyperparameters). For N-N-BEATS-S variants, the additional hyperparameter is λ. It is crucial since it controls the impact of the forecast instability objective on the probabilistic forecasts produced by the network. For each N-N-BEATS-S network, test set results are reported for both a conservatively selected λ and a more aggressively selected λ-value. The former is obtained by minimizing the (smoothed) validation sCRPS, while the latter is the largest λ-value for which no clear deterioration in (smoothed) validation sCRPS is observed compared to N-N-BEATS. Validation results for the different N-N-BEATS-S variants and varying λ-values are discussed in more detail in Section V-B2. For N-N-BEATS-C, there are no additional hyperparameters.

Training batches of input–output samples are created to train the N-N-BEATS and DeepAR networks. To randomize training batches, time series are first randomly sampled from the dataset after which also a time step is randomly sampled for each selected time series to serve as forecasting origin. For N-N-BEATS networks, the range from which time steps can be sampled contains the $10h$ most recent observations for which no missing values in the input–output samples can occur, taking into account the lengths of the lookback window T and the forecast period h. Finally, note that for all N-N-BEATS variants, we consider for all input–output samples also their lagged versions when training the networks (yet setting $\lambda = 0$ when appropriate). In this manner, the different N-N-BEATS variants can be compared in a fair way.

All networks are implemented in PyTorch [46] and trained using the Adam optimizer [47]. Our code is available online at https://github.com/VerbekeLab/N-N-BEATS. To increase the robustness of the evaluation, all reported results are obtained by averaging the mean and variance forecasts of ten different runs with different random seeds (resulting in different training batches and different initializations of network parameters).

B. Experiment 1: N-N-BEATS and Stability Variants

1) Results: Table IV summarizes the test set results for the first experiment. N-N-BEATS considerably outperforms the benchmark ETS forecasts in terms of both forecast accuracy and stability. It also outperforms DeepAR in terms of accuracy, but there is no clear winner in terms of stability. Comparing the results of the different N-N-BEATS-S setups to those of N-N-BEATS clearly shows that certain λ-values give rise to improvements in forecast stability while forecast accuracy is maintained. For both N-N-BEATS-S-KL variants and N-N-BEATS-S-W_2 and N-N-BEATS-S-RMSC with the larger λ-values, a limited decrease in accuracy is observed for all three performance measures; however, these settings also lead to the largest improvements in forecast stability. As a sanity check, we also evaluate the probabilistic forecasts per period output by N-N-BEATS-C. The results show that this extended

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on September 11,2024 at 08:33:10 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE IV

FORECAST ACCURACY AND STABILITY AGGREGATED (MEAN) OVER THE MONTHLY M4 TEST SET. LOWER VALUES ARE BETTER. THE MINIMUM VALUE PER COLUMN IS HIGHLIGHTED IN BOLD. *KL VALUES ARE CLIPPED BETWEEN 0 AND 30

	sCRPS	sPL90	RMSSE	KL*	sW_2	RMSSC
N-N-BEATS	1.233	0.280	1.267	0.165	0.660	0.537
N-N-BEATS-S-KL $\lambda = 0.176$	1.241	0.281	1.271	0.107	0.569	0.454
N-N-BEATS-S-KL $\lambda = 0.429$	1.256	0.286	1.278	**0.077**	**0.505**	**0.403**
N-N-BEATS-S-W_2 $\lambda = 1.000$	1.233	0.280	1.265	0.131	0.582	0.466
N-N-BEATS-S-W_2 $\lambda = 1.667$	1.241	0.281	1.270	0.116	0.549	0.437
N-N-BEATS-S-RMSC $\lambda = 0.429$	**1.227**	**0.278**	**1.262**	0.158	0.631	0.500
N-N-BEATS-S-RMSC $\lambda = 1.000$	1.242	0.281	1.272	0.138	0.593	0.448
N-N-BEATS-C	1.238	0.282	1.267	0.163	0.679	0.546
DeepAR	1.325	0.300	1.368	0.182	0.602	0.505
ETS	1.303	0.297	1.322	0.279	0.692	0.595

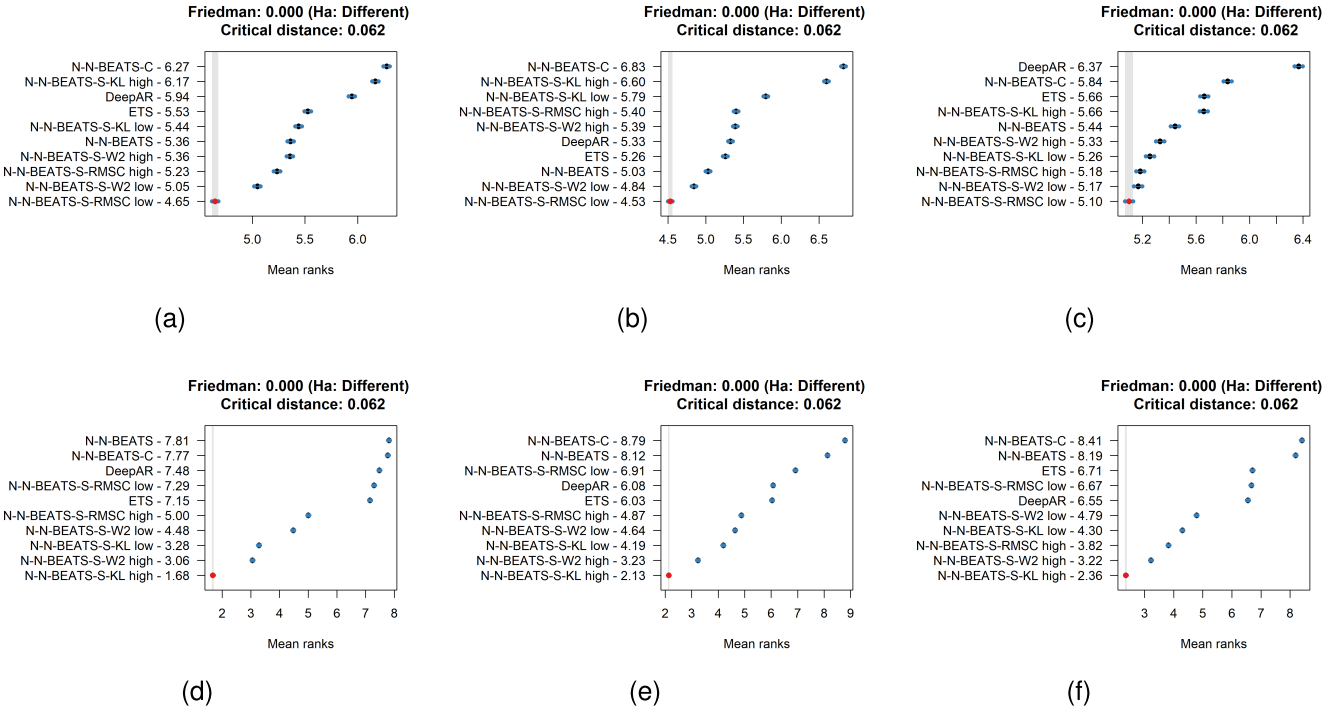

Fig. 4. MCB results for experiment 1. (a) sCRPS ranking. (b) sPL90 ranking. (c) RMSSE ranking. (d) KL ranking. (e) sW_2 ranking. (f) RMSSC ranking.

model results in similar forecast accuracy and stability as N-N-BEATS. Jointly optimizing the predictive distributions per period as well as the cumulative predictive distributions therefore does not seem to negatively impact the performance of the former, which is desirable. For robustness, results are also reported in terms of median performance in Appendix A. These results lead to (largely) the same analysis as above. Finally, the MCB results visualized in Fig. 4 confirm the above findings. However, there is one notable difference: N-N-BEATS-C consistently ranks lower than N-N-BEATS for all accuracy measures, indicating that there are in fact (very) small but consistent differences in accuracy.

2) Discussion: The test results presented in Table IV allow us to conclude that there certainly exist λ-values that result in improvements in forecast stability without causing a considerable loss in forecast accuracy. For the three different forecast instability components as proposed in Section III, Fig. 5(a)–(c), respectively, visualize validation accuracy (in terms of sCRPS) and stability (in terms of the corresponding stability metric) for different λ-values. To enhance the reliability of the results, each model is built twice with a different random seed, which is represented by the different colors. Note that $\lambda = 0$ represents an N-N-BEATS network.

Finding a suitable value for λ, the crucial (additional) hyperparameter for N-N-BEATS-S variants that controls the impact of the instability objective on the model outputs, is important; however, in all plots of Fig. 5, multiple λ-values can be considered to give rise to an eligible combination of forecast accuracy and stability (i.e., all λ-values resulting in stability improvements without causing considerable losses in accuracy). Only when λ exceeds a certain value, we observe a tradeoff between stability and accuracy, where further improvements in stability come at the cost of decreased accuracy.

The regularization effect of the instability component that was observed in [23], in which a similar problem is studied in a point forecasting context, however, is not particularly present in Fig. 5(a) and (b) for the N-N-BEATS-S networks. This

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on September 11,2024 at 08:33:10 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

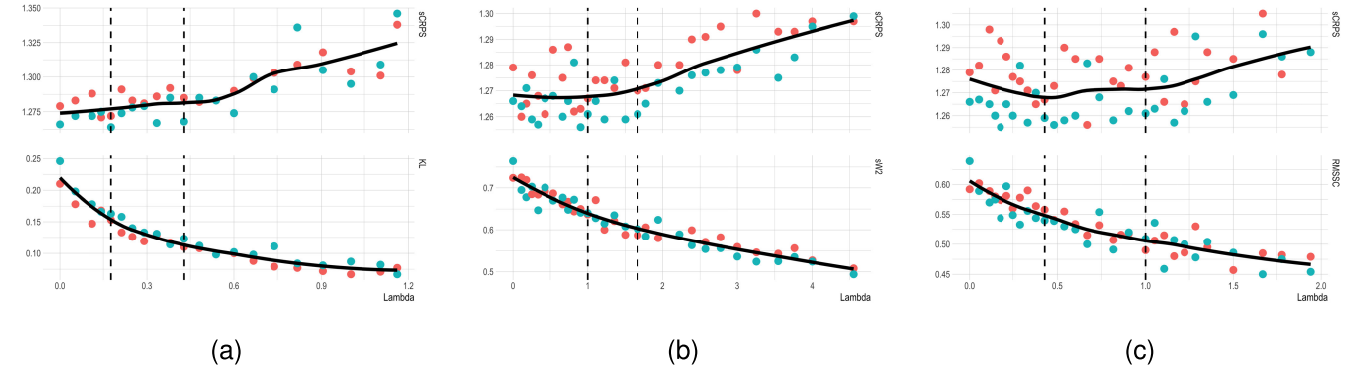

Fig. 5. Effect of λ for (a) KL, (b) W_2, and (c) RMSC penalties on validation sCRPS (top) and stability in terms of the corresponding stability metric (bottom) across all time series. Different colors represent different random seeds. $\lambda = 0$ represents a standard N-N-BEATS network. The dashed vertical lines represent the selected λ-values used in Table IV.

observation is possibly due to the fact that inducing stability in variance forecasts might affect forecast accuracy differently than inducing stability in mean forecasts. It is indeed often reasonable that variance forecasts increase with the forecast horizon to capture rising uncertainty, which stems from a greater potential for unexpected changes in underlying patterns and an increasingly significant impact of certain undetected underlying patterns, such as trends. However, penalizing for instabilities of entire distributional forecasts partly counteracts this. Fig. 5(c), which represents N-N-BEATS-S-RMSC networks in which stability is only induced in mean forecasts, indeed tends most toward exhibiting a regularization effect; however, these networks also result in more limited improvements in forecast stability. In this regard, a more flexible way of quantifying forecast instability that would allow us to directly vary the importance placed on different parts of the forecast distributions seems to be an interesting avenue for future research.

C. Experiment 2: N-N-BEATS-C

1) Results: Table V summarizes the test set results for the second experiment. For RMSSE, ETS-C1, ETS-C2, and ETS-C3 naturally coincide (when we ignore the minor differences for ETS-C3 caused by relying on only 1000 simulated future paths) as the variance estimations are not included in the RMSSE evaluation. Inspecting the evolution of RMSSE in lead time L, we observe that N-N-BEATS-C only starts to outperform ETS for $L > 1$, while for the one-month ahead probabilistic predictions ETS performs slightly better. We further observe that the gap in RMSSE between N-N-BEATS-C and ETS widens with increasing L. Combining these RMSSE results with those of the first experiment, where reported results represent averages across one- to six-month ahead probabilistic forecasts for individual time periods, we can conclude that N-N-BEATS(-C) gives rise to better mean forecasts (for individual time periods) than ETS for $h > 1$. On the other hand, in terms of sCRPS and sPL90, N-N-BEATS-C outperforms all ETS variants for all reported lead times L. The performance gaps between ETS-C1 and ETS-C2, ETS-C2 and ETS-C3, and ETS-C3 and N-N-BEATS-C, all widen with increasing L, where the first

TABLE V

FORECAST ACCURACY FOR TEMPORAL AGGREGATES (AVERAGE PER PERIOD), AGGREGATED (MEAN) OVER THE MONTHLY M4 TEST SET. LOWER VALUES ARE BETTER. THE MINIMUM VALUE PER (ACCURACY MEASURE - LEAD TIME) COMBINATION IS HIGHLIGHTED IN BOLD

	Lead time L					
	1	**2**	**3**	**4**	**5**	**6**
sCRPS						
ETS-C1	0.757	0.824	0.936	1.056	1.173	1.284
ETS-C2	0.745	0.798	0.885	0.980	1.074	1.164
ETS-C3	0.746	0.800	0.878	0.961	1.042	1.120
DeepAR	0.758	0.818	0.898	0.981	1.062	1.141
N-N-BEATS-C	**0.736**	**0.779**	**0.845**	**0.916**	**0.987**	**1.055**
sPL90						
ETS-C1	0.183	0.222	0.285	0.352	0.418	0.480
ETS-C2	0.170	0.188	0.220	0.256	0.292	0.327
ETS-C3	0.170	0.181	0.198	0.217	0.236	0.255
DeepAR	**0.166**	0.180	0.200	0.221	0.242	0.263
N-N-BEATS-C	**0.166**	**0.174**	**0.189**	**0.206**	**0.223**	**0.240**
RMSSE						
ETS-C1	**0.651**	0.698	0.766	0.838	0.907	0.973
ETS-C2	**0.651**	0.698	0.766	0.838	0.907	0.973
ETS-C3	**0.651**	0.699	0.766	0.838	0.908	0.974
DeepAR	0.686	0.738	0.806	0.876	0.943	1.009
N-N-BEATS-C	0.656	**0.696**	**0.751**	**0.812**	**0.871**	**0.927**

gap highlights the benefit of being able to capture that forecast errors (for individual time periods) usually increase for longer forecast horizons, the second gap highlights the benefit of additionally being able to capture dependencies between these different forecast errors (for individual time periods), and the third gap highlights the benefit of using the deep-learning-based N-N-BEATS-C method instead of an ETS-based forecasting method. Therefore, by and large, we can conclude that N-N-BEATS-C outperforms the ETS-based methods, with the benefit of using N-N-BEATS-C becoming larger for longer lead times L. N-N-BEATS-C also outperforms DeepAR, and in this case, for all lead times and accuracy measures. However, the results in terms of median performance reported in Appendix B, indicate that the error distributions are slightly skewed since N-N-BEATS-C only outperforms all ETS variants in terms of all accuracy measures for $L \geq 3$, and DeepAR outperforms N-N-BEATS-C in terms of sPL90 and, for $L \geq 4$, also in terms of sCRPS. MCB results for $L = 3$ and $L = 6$ depicted in Fig. 6 confirm

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on September 11,2024 at 08:33:10 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

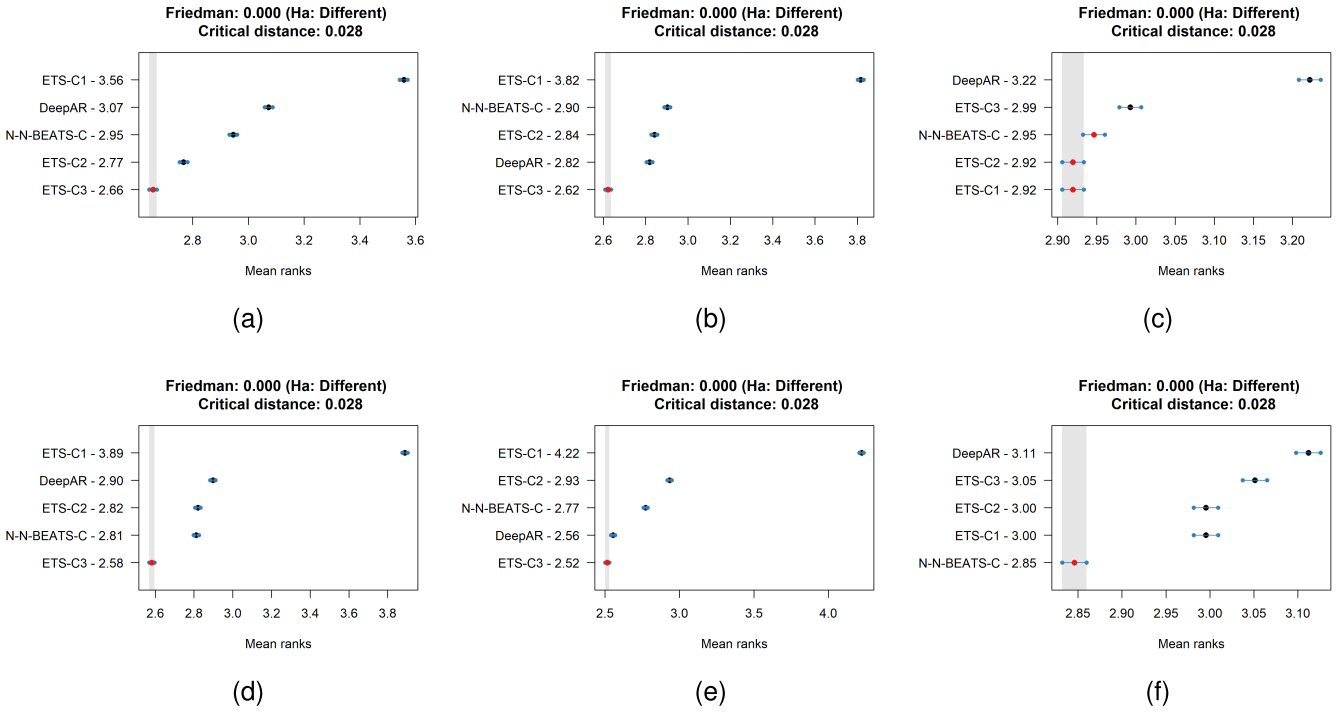

Fig. 6. MCB results for experiment 2 for lead times $L = 3$ and $L = 6$. (a) sCRPS ranking for $L = 3$. (b) sPL90 ranking for $L = 3$. (c) RMSSE ranking for $L = 3$. (d) sCRPS ranking for $L = 6$. (e) sPL90 ranking for $L = 6$. (f) RMSSE ranking for $L = 6$.

these mixed findings. However, it is important to emphasize that the rankings used in MCB tests do not take into account the magnitude of improvements. Therefore, while ETS and/or DeepAR may marginally outperform N-N-BEATS-C for a small majority of the time series in the experiment, when N-N-BEATS-C performs better, it does so by a considerable margin.

2) Discussion: A global, probabilistic, and direct approach to modeling temporal aggregates as in N-N-BEATS-C has several advantages in practice. First, N-N-BEATS-C, like DeepAR, does not require an upfront specification of a specific quantile of interest and a lead time for each SKU. This is an attractive property as lead times and target service levels are typically not equal for all products in a portfolio and may change over time. Second, N-N-BEATS-C provides an attractive alternative to sampling-based approaches such as ETS-C3 and DeepAR, and empirical variance estimation approaches as presented in [44], since N-N-BEATS-C allows capturing temporal dependencies—like the sampling-based and empirical variance estimation approaches—without having to rely on computationally demanding simulations or requiring that a relatively large collection of out-of-sample lead time forecast errors is available for each SKU, which is not evident in the experimental setting with monthly forecasting buckets considered (see also Section V-A3).

VI. Conclusion and Future Research

The center of attention in the time series forecasting literature is gradually shifting from point forecast to probabilistic forecast generation. This is prompted by the fact that one is typically also interested in the ability to correctly quantify the prediction uncertainty. In this context, we present N-N-BEATS, a parametric probabilistic forecasting version of the state-of-the-art N-BEATS deep learning architecture initially designed for tackling point forecasting problems. The focus in this work is on modeling real-valued time series and we specifically opt to model predictive distributions as Gaussians because of its useful properties in the context of the presented extensions to this probabilistic N-BEATS architecture (closed-form solutions for KL divergence and W_2 distance between two Gaussians are available and the Gaussian distribution is closed under addition). First, we generalize the forecast stability mechanism introduced in [23], and second, we present N-N-BEATS-C, an extension that allows for the generation of cumulative probabilistic forecasts (i.e., forecasts of temporal aggregates) next to probabilistic forecasts for individual time periods. Both extensions are particularly useful in a supply chain planning context: the first extension leads to more stable demand forecasts which, when used as inputs to devise supply plans, will lead to fewer and smaller revisions to these plans and hence a reduction in supply chain costs; the second extension is useful for inventory management at SKU-level where one is often interested in a specific quantile of the predictive distribution of cumulative demand. Experimental results are reported for all probabilistic N-BEATS variants presented in this work and show that: 1) N-N-BEATS results in significantly more accurate probabilistic forecasts as compared to ETS on average across one- to six-month ahead predictions; 2) the stability mechanisms presented in this work all result in more stable forecasts without causing considerable losses in forecast accuracy; and that 3) the second extension, N-N-BEATS-C, on average, results in improvements in the forecast accuracy of cumulative predictive distributions over basic but well-established methods to produce probabilistic cumulative demand forecasts. Note that combining both

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on September 11,2024 at 08:33:10 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE VI

FORECAST ACCURACY AND STABILITY AGGREGATED (MEDIAN) OVER THE MONTHLY M4 TEST SET. LOWER VALUES ARE BETTER. THE MINIMUM VALUE PER COLUMN IS HIGHLIGHTED IN BOLD. *KL VALUES ARE CLIPPED BETWEEN 0 AND 30

	sCRPS	**sPL90**	**RMSSE**	**KL***	**sW_2**	**RMSSC**
N-N-BEATS	0.931	0.190	1.035	0.130	0.542	0.444
N-N-BEATS-S-KL $\lambda = 0.176$	0.932	0.192	1.032	0.087	0.462	0.368
N-N-BEATS-S-KL $\lambda = 0.429$	0.942	0.196	1.035	**0.062**	**0.410**	**0.325**
N-N-BEATS-S-W_2 $\lambda = 1.000$	0.930	0.191	1.032	0.102	0.474	0.378
N-N-BEATS-S-W_2 $\lambda = 1.667$	0.933	0.192	1.034	0.091	0.446	0.353
N-N-BEATS-S-RMSC $\lambda = 0.429$	**0.924**	0.189	**1.030**	0.124	0.516	0.413
N-N-BEATS-S-RMSC $\lambda = 1.000$	0.930	0.190	1.032	0.106	0.476	0.363
N-N-BEATS-C	0.940	0.195	1.037	0.129	0.558	0.448
DeepAR	0.942	**0.182**	1.075	0.147	0.500	0.420
ETS	0.979	0.197	1.080	0.161	0.540	0.464

extensions to N-N-BEATS is possible, however, in the setup as considered in this work, the stability mechanisms can only be applied to the probabilistic forecasts for the demands per period since the cumulative predictive distributions always cover the first period right after the forecasting origin.

A limitation of this work is that we evaluate the proposed N-N-BEATS variants only on the M4 monthly dataset. Therefore, an important direction for future research is testing all proposed N-N-BEATS variants on a wider range of datasets to assess their effectiveness across various contexts and verify the results, as well as conducting a more detailed analysis to determine if the general conclusions are consistent across time series with different characteristics. For datasets containing demand time series and information on inventory parameters and policies, an investigation into the impact of the reported improvements in forecast accuracy (and stability) of the predictive distributions on actual inventory decisions is another interesting topic for future research.

In this work, we focused on a full Gaussian setup in that the marginal distributions used in all N-N-BEATS networks and the copula used in N-N-BEATS-C are all Gaussian. We believe that translating these probabilistic methods to other distributional settings (and dependence structures) and empirically evaluating combining the Gaussian N-N-BEATS variants with data transformations to deal with non-Gaussian data, such as the Box-Cox [48] transformations or normalizing flows [49], [50], [51], are interesting topics for future research. Considering the importance of intermittent time series [52] in industrial settings, as exemplified by, e.g., the M5 competition dataset, probabilistic methods for forecasting positive count data relying on, e.g., the Poisson or negative binomial distribution (see [14]), deserve special attention. Also the use of flexible mixture densities, which can be used to model problems with multiple possible states which should not be mixed together or averaged, and the incorporation of ideas from the literature on mixture density networks [53] seem interesting directions for future research when combined with extending the proposed probabilistic N-BEATS variants to allow including explanatory variables as model inputs. Finally, also a (direct) multivariate modeling approach, such as directly learning a multivariate normal distribution for the next h observations, seems an interesting topic for future research which also opens up new opportunities to include stability mechanisms that take into account the full covariance structure.

TABLE VII

FORECAST ACCURACY FOR TEMPORAL AGGREGATES (AVERAGE PER PERIOD), AGGREGATED (MEDIAN) OVER THE MONTHLY M4 TEST SET. LOWER VALUES ARE BETTER. THE MINIMUM VALUE PER (ACCURACY MEASURE - LEAD TIME) COMBINATION IS HIGHLIGHTED IN BOLD

	Lead time L					
	1	**2**	**3**	**4**	**5**	**6**
sCRPS						
ETS-C1	**0.597**	0.630	0.697	0.765	0.830	0.891
ETS-C2	**0.597**	**0.626**	0.676	0.728	0.776	0.821
ETS-C3	0.598	0.631	0.681	0.729	0.772	0.810
DeepAR	0.617	0.635	0.667	**0.700**	**0.729**	**0.756**
N-N-BEATS-C	0.609	0.629	**0.665**	0.703	0.737	0.770
sPL90						
ETS-C1	**0.119**	0.133	0.169	0.209	0.246	0.283
ETS-C2	**0.119**	0.125	0.138	0.155	0.171	0.186
ETS-C3	**0.119**	0.125	0.135	0.145	0.155	0.163
DeepAR	0.120	**0.122**	**0.127**	**0.132**	**0.138**	**0.142**
N-N-BEATS-C	0.125	0.127	0.135	0.143	0.151	0.159
RMSSE						
ETS-C1	**0.574**	**0.608**	0.648	0.689	0.727	0.762
ETS-C2	**0.574**	**0.608**	0.648	0.689	0.727	0.762
ETS-C3	0.575	**0.608**	0.649	0.690	0.728	0.763
DeepAR	0.607	0.629	0.658	0.689	0.717	0.742
N-N-BEATS-C	0.586	0.610	**0.640**	**0.670**	**0.697**	**0.722**

More generally, introducing similar extensions of other deep learning architectures for probabilistic forecasting seems a worthwhile endeavor. For extensions aimed at achieving more stable forecast distributions, this might require exploring other dissimilarity measures to quantify forecast instability. In this regard, flexible approaches that allow directly varying the importance placed on the stability of different parts of forecast distributions (see Section V-B2) might be of particular interest. Finally, for stability extensions, using adaptive weighting schemes [54] to dynamically adjust the weight of the forecast instability component during model optimization, instead of using a static weight (i.e., the hyperparameter λ), is yet another interesting topic for future research.

APPENDIX A
ADDITIONAL RESULTS EXPERIMENT 1

Table VI summarizes the test set results for the first experiment by using the median as aggregation operator.

APPENDIX B
ADDITIONAL RESULTS EXPERIMENT 2

Table VII summarizes the test set results for the second experiment by using the median as aggregation operator.

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on September 11,2024 at 08:33:10 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their insightful comments that helped them substantially improve this article.

REFERENCES

[1] T. Gneiting and M. Katzfuss, "Probabilistic forecasting," *Annu. Rev. Statist. Appl.*, vol. 1, pp. 125–151, Jan. 2014.

[2] J. R. Trapero, M. Cardós, and N. Kourentzes, "Quantile forecast optimal combination to enhance safety stock estimation," *Int. J. Forecasting*, vol. 35, no. 1, pp. 239–250, Jan. 2019.

[3] A. J. McNeil, R. Frey, and P. Embrechts, *Quantitative Risk Management: Concepts, Techniques and Tools-revised Edition*. Princeton, NJ, USA: Princeton Univ. Press, 2015.

[4] H. Quan, A. Khosravi, D. Yang, and D. Srinivasan, "A survey of computational intelligence techniques for wind power uncertainty quantification in smart grids," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 11, pp. 4582–4599, Nov. 2020.

[5] T. Gneiting and A. E. Raftery, "Strictly proper scoring rules, prediction, and estimation," *J. Amer. Stat. Assoc.*, vol. 102, no. 477, pp. 359–378, Mar. 2007.

[6] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "The M4 competition: 100,000 time series and 61 forecasting methods," *Int. J. Forecasting*, vol. 36, no. 1, pp. 54–74, Jan. 2020.

[7] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "The M5 competition: Background, organization, and implementation," *Int. J. Forecasting*, vol. 38, no. 4, pp. 1325–1336, Oct. 2022.

[8] S. Makridakis et al., "The M5 uncertainty competition: Results, findings and conclusions," *Int. J. Forecasting*, vol. 38, no. 4, pp. 1365–1385, Oct. 2022.

[9] B. N. Oreshkin, D. Carpov, N. Chapados, and Y. Bengio, "N-BEATS: Neural basis expansion analysis for interpretable time series forecasting," in *Proc. Int. Conf. Learn. Representat.*, 2020, pp. 1–31.

[10] T. Januschowski et al., "Criteria for classifying forecasting methods," *Int. J. Forecasting*, vol. 36, no. 1, pp. 167–177, Jan. 2020.

[11] T. Januschowski and S. Kolassa, "A classification of business forecasting problems," *Foresight, Int. J. Appl. Forecasting*, vol. 52, pp. 36–43, Jan. 2019.

[12] S. Smyl, "A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting," *Int. J. Forecasting*, vol. 36, no. 1, pp. 75–85, Jan. 2020.

[13] G. Dudek, P. Pelka, and S. Smyl, "A hybrid residual dilated LSTM and exponential smoothing model for midterm electric load forecasting," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 7, pp. 2879–2891, Jul. 2022.

[14] D. Salinas, V. Flunkert, J. Gasthaus, and T. Januschowski, "DeepAR: Probabilistic forecasting with autoregressive recurrent networks," *Int. J. Forecasting*, vol. 36, no. 3, pp. 1181–1191, Jul. 2020.

[15] S. S. Rangapuram, M. W. Seeger, J. Gasthaus, L. Stella, Y. Wang, and T. Januschowski, "Deep state space models for time series forecasting," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 1–10.

[16] R. Wen, K. Torkkola, B. Narayanaswamy, and D. Madeka, "A multi-horizon quantile recurrent forecaster," 2017, *arXiv:1711.11053*.

[17] B. Lim, S. Ö. Arik, N. Loeff, and T. Pfister, "Temporal fusion transformers for interpretable multi-horizon time series forecasting," *Int. J. Forecasting*, vol. 37, no. 4, pp. 1748–1764, Oct. 2021.

[18] J. Gasthaus et al., "Probabilistic forecasting with spline quantile function RNNs," in *Proc. 22nd Int. Conf. Artif. Intell. Statist.*, 2019, pp. 1901–1910.

[19] P. Grecov et al., "Probabilistic causal effect estimation with global neural network forecasting models," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 4, pp. 4999–5013, Apr. 2024.

[20] V. Jensen, F. M. Bianchi, and S. N. Anfinsen, "Ensemble conformalized quantile regression for probabilistic time series forecasting," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 7, pp. 9014–9025, Jul. 2024.

[21] J. de Vilmarest, J. Browell, M. Fasiolo, Y. Goude, and O. Wintenberger, "Adaptive probabilistic forecasting of electricity (Net-)Load," *IEEE Trans. Power Syst.*, vol. 39, no. 2, pp. 4154–4163, Mar. 2024.

[22] H. Hasson, B. Wang, T. Januschowski, and J. Gasthaus, "Probabilistic forecasting: A level-set approach," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 6404–6416.

[23] J. Van Belle, R. Crevits, and W. Verbeke, "Improving forecast stability using deep learning," *Int. J. Forecasting*, vol. 39, no. 3, pp. 1333–1350, Jul. 2023.

[24] A. A. Syntetos, Z. Babai, J. E. Boylan, S. Kolassa, and K. Nikolopoulos, "Supply chain forecasting: Theory, practice, their gap and the future," *Eur. J. Oper. Res.*, vol. 252, no. 1, pp. 1–26, Jul. 2016.

[25] H. Tunc, O. A. Kilic, S. A. Tarim, and B. Eksioglu, "A simple approach for assessing the cost of system nervousness," *Int. J. Prod. Econ.*, vol. 141, no. 2, pp. 619–625, Feb. 2013.

[26] Q. Li and S. M. Disney, "Revisiting rescheduling: MRP nervousness and the bullwhip effect," *Int. J. Prod. Res.*, vol. 55, no. 7, pp. 1992–2012, Apr. 2017.

[27] F. Petropoulos et al., "Forecasting: Theory and practice," *Int. J. Forecasting*, vol. 38, no. 3, pp. 705–871, 2022.

[28] S. Kullback and R. A. Leibler, "On information and sufficiency," *Ann. Math. Statist.*, vol. 22, no. 1, pp. 79–86, 1951.

[29] G. Peyré and M. Cuturi, "Computational optimal transport," *Found. Trends Mach. Learn.*, vol. 11, nos. 5–6, pp. 355–607, 2019.

[30] M. Z. Babai and Y. Dallery, "Dynamic versus static control policies in single stage production-inventory systems," *Int. J. Prod. Res.*, vol. 47, no. 2, pp. 415–433, Jan. 2009.

[31] L. J. Tashman, "Out-of-sample tests of forecasting accuracy: An analysis and review," *Int. J. Forecasting*, vol. 16, no. 4, pp. 437–450, Oct. 2000.

[32] R. J. Hyndman and A. B. Koehler, "Another look at measures of forecast accuracy," *Int. J. Forecasting*, vol. 22, no. 4, pp. 679–688, Oct. 2006.

[33] R. Koenker and G. Bassett, "Regression quantiles," *Econometrica*, vol. 46, no. 1, pp. 33–50, Jan. 1978.

[34] T. Gneiting, A. E. Raftery, A. H. Westveld, and T. Goldman, "Calibrated probabilistic forecasting using ensemble model output statistics and minimum CRPS estimation," *Monthly Weather Rev.*, vol. 133, no. 5, pp. 1098–1118, May 2005.

[35] F. Laio and S. Tamea, "Verification tools for probabilistic forecasts of continuous hydrological variables," *Hydrol. Earth Syst. Sci.*, vol. 11, no. 4, pp. 1267–1277, May 2007.

[36] E. Egrioglu and R. Fildes, "A note on the robustness of performance of methods and rankings for M4 competition," *Turkish J. Forecasting*, vol. 4, no. 2, pp. 26–32, 2020.

[37] A. J. Koning, P. H. Franses, M. Hibon, and H. O. Stekler, "The M3 competition: Statistical tests of the results," *Int. J. Forecasting*, vol. 21, no. 3, pp. 397–409, Jul. 2005.

[38] R. J. Hyndman and Y. Khandakar, "Automatic time series forecasting: The forecast package for R," *J. Stat. Softw.*, vol. 26, no. 3, pp. 1–22, 2008.

[39] R. Hyndman et al. (2020). *Forecast: Forecasting Functions for Time Series and Linear Models R Package Version 8.13*. [Online]. Available: https://pkg.robjhyndman.com/forecast/

[40] R. J. Hyndman, A. B. Koehler, R. D. Snyder, and S. Grose, "A state space framework for automatic forecasting using exponential smoothing methods," *Int. J. Forecasting*, vol. 18, no. 3, pp. 439–454, Jul. 2002.

[41] R. J. Hyndman, A. B. Koehler, J. K. Ord, and R. D. Snyder, *Forecasting With Exponential Smoothing: The State Space Approach*. Berlin, Germany: Springer-Verlag, 2008.

[42] C. Chatfield, *Time-Series Forecasting*. Boca Raton, FL, USA: CRC Press, 2000.

[43] D. K. Barrow and N. Kourentzes, "Distributions of forecasting errors of forecast combinations: Implications for inventory management," *Int. J. Prod. Econ.*, vol. 177, pp. 24–33, Jul. 2016.

[44] J. R. Trapero, M. Cardós, and N. Kourentzes, "Empirical safety stock estimation based on kernel and GARCH models," *Omega*, vol. 84, pp. 199–211, Apr. 2019.

[45] A. Alexandrov et al., "GluonTS: Probabilistic and neural time series modeling in Python," *J. Mach. Learn. Res.*, vol. 21, no. 1, pp. 4629–4634, 2020.

[46] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–12.

[47] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[48] G. E. Box and D. R. Cox, "An analysis of transformations," *J. Roy. Stat. Soc. Ser. B, Stat. Methodol.*, vol. 26, no. 2, pp. 211–243, 1964.

[49] I. Kobyzev, S. J. D. Prince, and M. A. Brubaker, "Normalizing flows: An introduction and review of current methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 11, pp. 3964–3979, Nov. 2021.

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on September 11,2024 at 08:33:10 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

[50] G. Papamakarios, E. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan, "Normalizing flows for probabilistic modeling and inference," *J. Mach. Learn. Res.*, vol. 22, no. 1, pp. 2617–2680, 2021.

[51] H. Wen, P. Pinson, J. Ma, J. Gu, and Z. Jin, "Continuous and distribution-free probabilistic wind power forecasting: A conditional normalizing flow approach," *IEEE Trans. Sustain. Energy*, vol. 13, no. 4, pp. 2250–2263, Oct. 2022.

[52] A. A. Syntetos and J. E. Boylan, *Intermittent Demand Forecasting: Context, Methods and Applications*. Hoboken, NJ, USA: Wiley, 2021.

[53] C. Bishop, "Mixture density networks," Dept. Comput. Sci. Appl. Math., Neural Comput. Res. Group, Aston Univ., Birmingham, U.K., Neural Comput. Res. Group Rep. NCRG/94/004, 1994.

[54] B. Lin, F. Ye, Y. Zhang, and I. Tsang, "Reasonable effectiveness of random weighting: A Litmus test for multi-task learning," *Trans. Mach. Learn. Res.*, Oct. 2022. [Online]. Available: https://openreview.net/forum?id=jjtFD8A1Wx



Jente Van Belle received the Ph.D. degree in applied economics (business engineering) from Vrije Universiteit Brussel, Brussels, Belgium, in 2021, which was funded by Flanders Innovation and Entrepreneurship (VLAIO) and OMP through a personal research grant.

He is currently an FWO Junior Post-Doctoral Fellow with the Faculty of Economics and Business, KU Leuven, Leuven, Belgium. His work has been published in established international scientific journals, such as *European Journal of Operational Research* (EJOR), IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING (TKDE), and *International Journal of Forecasting* (IJF). His research focuses on time series forecasting and prescriptive analytics.



Ruben Crevits received the Ph.D. degree in applied economics from KU Leuven, Leuven, Belgium, in 2018.

He is currently a Data Science Lead at OMP, Wommelgem, Belgium. He mainly focuses on demand forecasting. His research was about deep learning for forecasting and robust estimation of time series models.



Daan Caljon received the M.Sc. degree in business and information systems engineering from KU Leuven, Leuven, Belgium, in 2023, where he is currently pursuing the Ph.D. degree with the Faculty of Economics and Business.

His research interests include causal inference, network analytics, and time series forecasting.



Wouter Verbeke (Member, IEEE) received the Ph.D. degree in applied economics from KU Leuven, Leuven, Belgium, in 2012.

He is currently a Professor of data analytics with the Faculty of Economics and Business, KU Leuven. His work has been published in established international scientific journals, such as *European Journal of Operational Research* (EJOR), IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING (TKDE), and *Information Sciences*. He has also authored two books. His research focuses on prescriptive and profit-driven analytics and is driven by real-life business applications in operations, fraud, customer relationship, credit risk, supply chain, and human resources management.

Authorized licensed use limited to: KU Leuven Libraries. Downloaded on September 11,2024 at 08:33:10 UTC from IEEE Xplore. Restrictions apply.