Neural Forecasting at Scale

Philippe Chatigny[a], Shengrui Wang[a], Jean-Marc Patenaude[c], Boris N. Oreshkin[b]

[a] *University of Sherbrooke, Sherbrooke, QC, Canada*

[b] *Unity Technologies, Labs, Montreal, QC, Canada*

[c] *Laplace Insights, Sherbrooke, QC, Canada*

Abstract

We study the problem of efficiently scaling ensemble-based deep neural networks for time series (TS) forecasting on a large set of time series. Current state-of-the-art deep ensemble models have high memory and computational requirements, hampering their use to forecast millions of TS in practical scenarios. We propose N-BEATS(P), a global multivariate variant of the N-BEATS model designed to allow simultaneous training of multiple univariate TS forecasting models. Our model addresses the practical limitations of related models, reducing the training time by half and memory requirement by a factor of 5, while keeping the same level of accuracy. We have performed multiple experiments detailing the various ways to train our model and have obtained results that demonstrate its capacity to support zero-shot TS forecasting, i.e., to train a neural network on a source TS dataset and deploy it on a different target TS dataset without retraining, which provides an efficient and reliable solution to forecast at scale even in difficult forecasting conditions.

Keywords: Univariate time series forecasting, deep neural networks, N-BEATS, ensemble models

1. Introduction

In the past few years, abundant evidence has emerged suggesting that deep neural networks (DNN) constitute an effective modeling framework for solving time series (TS) forecasting problems. DNN models have been shown to produce state-of-the-art forecasts when large homogeneous datasets with multiple observations are available [1]. The success of DNN is largely accounted for by two factors: (i) the cross-training on multiple time-series and (ii) the use of over-specified large capacity ensemble models. However, the high computational requirements of such models in comparison to statistical models have raised concerns regarding their applicability in practical scenarios [2]. Indeed, the deployment of a reliable DNN with an automatic training procedure is far more challenging [3, 4, 5, 6].

On the other hand, there are plentiful examples of successful deployment of neural networks in large-scale TS forecasting. It appears that the benefits of using such models in production definitely outweigh the associated costs and difficulties [7]. First, such models are scalable: a neural TS forecasting model usually performs better as the scale of the data used to train it increases. This has been observed in various TS competitions [8, 2, 9]. Second, they can be reusable: we can reuse a model to produce forecasts over multiple TS [3, 10, 11] not observed during training. They also offer flexibility: they can be trained efficiently on various loss functions that are aligned with business/scientific objectives [12] and some architectures, such as the one in [13], are able to handle missing values with ease.

These benefits made neural TS forecasting models popular and even mainstream in various settings. In fact, the prevalent use of neural networks manifests a paradigm shift in data-driven forecasting techniques, with fully-automated models being the de-facto standard. Many examples of DNN for TS exist. Some of the largest online retail platforms are using neural networks to forecast product demand for millions of retail items [14, 15, 16, 17]. AutoML solutions with heavy use of DNNs like [18] are being used in various settings and have been demonstrated to be very competitive [2] with almost no human involvement. Some companies that need to allocate a large pool of resources in different environments are using neural networks to anticipate required resources for different periods of the day [19, 20]. Large capital markets companies are using neural networks to predict the future movement of assets [21] via a process that links the trade-generating strategies with notifications and trade automation from these forecasts. However, these approaches come with high computational cost.

The need to render these methods more efficient has been pointed out multiple times [22, 2] and is one of the core challenges that must be solved to democratize their use. Currently, they require much time, specialized hardware and energy to train and deploy. Besides their model size, which can render their use cumbersome, re-training these models at every forecast for different TS is not viable for most organizations. Only recently has some work been done to evaluate how to generalize these models to multiple types of TS with different characteristics while maintaining an acceptable level of accuracy within a zero-shot regime [3], or in a few-shot learning regime, i.e., by fine-tuning the model to the target dataset [10, 11, 23]. In the ensemble case, this issue is amplified since most of the current top-performing models rely on independent training of ensemble members. Producing forecasts with a small ensemble size without affecting accuracy is of great interest. We propose to tackle these problems within a single approach.

2. Related Work

TS forecasting models: Traditional local, univariate models for TS forecasting include the autoregressive integrated moving average (ARIMA) model [24], exponential smoothing methods (HOLT, ETS, DAMPED, SES) [25, 26] decomposition-based approaches, including the THETA model and its variants [27, 28, 29], and autoregressive (AR) models with time-varying coefficients as in [30, 31]. Global univariate TS models that rely on neural networks (DNNs) have been proposed recently as alternatives to these models [15, 14, 32, 33, 34, 35, 36]. In contrast to the traditional approaches, they allow learning from multiple independent TS simultaneously and handle non-stationary TS without preprocessing steps. Some concerns have been raised regarding machine learning (ML) publications claiming satisfactory accuracy without adequate comaprison with the well-established statistical baselines and using inappropriate criteria often leading to misleading results [37, 38, 39]. It is inspiring to see that recent ML publications have largely solved these problems by following more rigorous evaluation protocols and baseline comparisons [26, 40, 15, 14, 41].

Ensemble models: Combining multiple models is often a more straightforward strategy to produce accurate forecasts than finding the best parameterization for one particular model [42, 43]. Recently, both M4 and M5 forecasting competitions have empirically confirmed the accuracy of ensemble models [8, 2]. Notable instances of these models for univariate TS forecasting include FFORMA [40] (second entry in M4), ES-RNN [44] (first entry in

M4) and subsequently N-BEATS [1] [41]. These ensemble methods, especially N-BEATS, have high computational and memory complexities, which require specialized infrastructure to accelerate their training and store the trained models [26]. For example, the full N-BEATS model consists of 180 individual models. It takes around 11'755 hours to train on the full M4 dataset using 1 NVIDIA GTX 2080Ti GPU. Furthermore, the total size of the models in ensemble is 160 GB, which, depending on the number of training logs and saved snapshots of the model, can increase to over 450 GB.

Hence, the major issues come down to parameter size of the model, time to train the model and whether or not we can offset the operating costs. This paper seek to reduce the computational complexity gap between classical and neural time series models by proposing a more memory- and computation -efficient version of the N-BEATS architecture [41]. Our approach achieve this by re-formulating the original fully-connected N-BEATS architecture as a single kernel convolution, which allows for training multiple of models in parallel on the same GPU while sharing most of the parameters in the network. This leads to reduced ensemble training time and memory footprint as well as reduced ensemble model size, which positively affects the costs of training, querying and storing the resulting ensemble without compromising its accuracy.

Our contributions can be summarized as follows:

[1] We introduce N-BEATS(P), a multi-head parallelizable N-BEATS architecture that permits the simultaneous training of multiple global TS models. Our model twice as fast as N-BEATS, has 5 times fewer parameters, and performs at the same level of accuracy as the current state-of-the-art TS model on M4.

[2] We demonstrate that N-BEATS(P) has comparable level of accuracy than N-BEATS for zero-shot generalization ability in various settings. It can operate on various domains of applications and on target dataset that are out-of-distribution of the source dataset it was trained on.

The remainder of this paper is organized as follows. Section 3 describe the

[1]N-BEATS was not part of the M4 competition, and attained state-of-the-art results on M4 benchmark ex post facto. N-BEATS was the core part of the second-entry solution in M5 competition.[2]

univariate TS forecasting problem. Section 4 presents our modeling approach. Section 5 outlines empirical evaluation setup and our results. Finally, Section 6 presents our conclusions.

3. Problem Statement

We consider the univariate point forecasting problem in discrete time where we have a training dataset of N time series, $\mathscr{D}_{\text{train}} = \{\mathbf{X}^{(i)}_{1:T_i}\}_{i=1}^N$ and a test dataset of future values of these time series $\mathscr{D}_{\text{eval}} = \{\mathbf{Y}^{(i)}_{T_i+1:T_i+H}\}_{i=1}^N$. The task is to forecast future values of the series, $\mathbf{Y}^{(i)}_{T_i+1:H} \in \mathbb{R}^H$, given a regularly-sampled sequence of past observations, $\mathbf{X}^{(i)}_{1:T_i} \in \mathbb{R}^{T^{(i)}}$. To solve the task, we define a forecasting function $f_\theta : \mathbb{R}^T \to \mathbb{R}^H$, parameterized with a set of learnable parameters $\theta \in \Theta \subset \mathbb{R}^M$. The parameters of the forecasting function can be learned using an empirical risk minimization framework based on the appropriate samples of forecasting function inputs, $\mathbf{Z}_{\text{in}} \in \mathbb{R}^T$, and outputs, $\mathbf{Z}_{\text{out}} \in \mathbb{R}^H$, taken from the training set:

$$\widehat{\theta} = \arg\min_{\theta \in \Theta} \sum_{\mathbf{Z}_{\text{in}}, \mathbf{Z}_{\text{out}} \in \mathscr{D}_{\text{train}}} \mathcal{L}(\mathbf{Z}_{\text{out}}, f_\theta(\mathbf{Z}_{\text{in}})) \tag{1}$$

A few remarks are in order regarding the selection of the model input window size T. The optimal choice of T is highly data-dependent. In terms of general guidelines, TS with a swiftly changing generating process will favor small values of T, as historical information quickly becomes outdated. TS with long seasonality periods will favor larger T, as observing at least one and maybe a few seasonality periods may be beneficial for making a more informed forecast. Obviously, serveral conflicting factors can be at play here and finding a universally optimal solution for all TS does not seem viable. Therefore, T can be treated as a hyperparameter selected on a TS-specific validation set. A more productive and accurate solution would entail using an ensemble of several models, each trained with its own T, as in [41]. However, this solution tends to inflate the ensemble size, and that is the problem we addressed in this paper. In general, increasing the diversity of an ensemble with different forecasting models usually results in the inflation of the ensemble size and computational costs. Therefore, we focus on providing a solution to more effectively parallelize training of the N-BEATS ensemble, which is obviously applicable to situations other than using multi-T ensembles.

4. Model

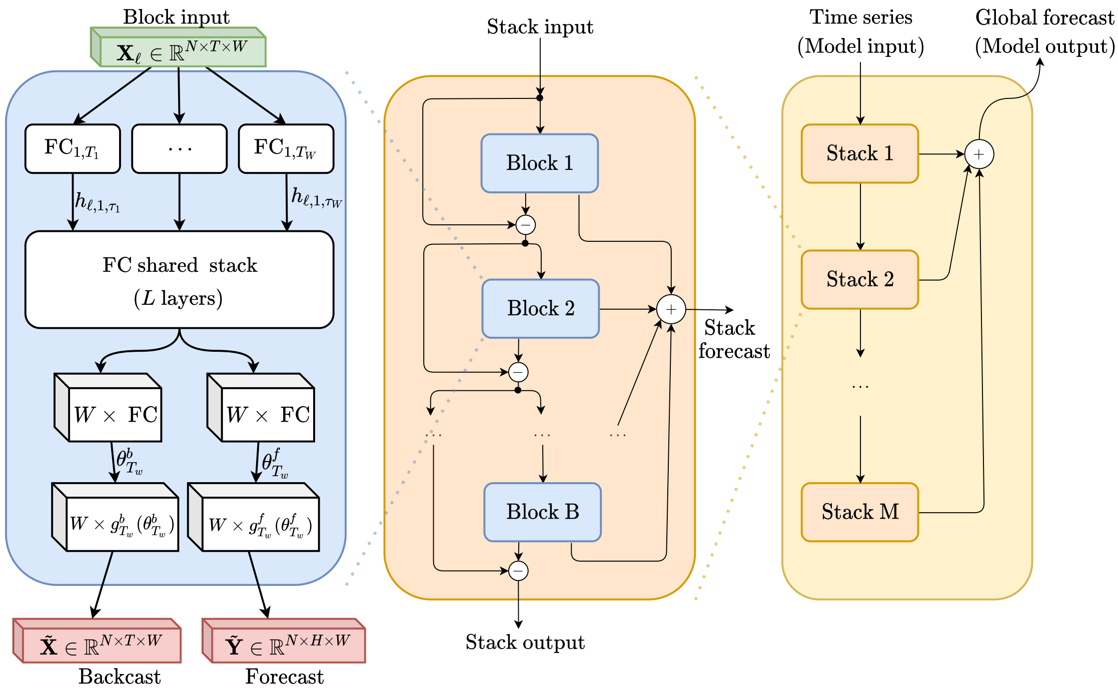

Figure 1: Illustration of the proposed model. The basic block consists of multi-head and multi-output fully connected (FC) layers with ReLU non-linear activations, where some layers are shared between the W models. Each block input $\mathbf{X}_l \in \mathbb{R}^{N\times T\times W}$ contains the same input signal at different lookback windows $T_1 \cdots T_W$, where for each of the W representations of the signal, missing values are padded with 0. The multi-output part of the block consists of W independent layers (represented by the blue cube in the figure) that predict basis expansion coefficients both forward $\theta^f_{T_w}$ (Forecast) and backward $\theta^b_{T_w}$ (Backcast) for each of the W models. A stack can have layers with shared $g^b_{T_w}$ and $g^f_{T_w}$. Forecasts are aggregated by summing over all partial forecasts of each block, enabling us to retrieve which block had the most impact in making the forecast. Parallelization is achieved by forcing head layers of each block to have the same input size, by using mask layers in the input layer to consider only the T_w first observations of input signals and reshaping the tensor to force computation in parallel instead of sequentially applying computation in a loop for each of the W models.

The basic building block of the proposed model has a multi-head architecture and is depicted in Fig. 1 (left). Each block can take as input up to W input signals $\mathbf{x}_{T_w}; w \in \{1,\ldots,W\}$ of the same TS with different lookback windows, and generates two output vectors for each of the input signals provided: the *backcast* signal $\tilde{\mathbf{x}}_{T_w}$ of length T_w and the *forecast* signal $\tilde{\mathbf{y}}_{T_w}$

of length h. We set each T_w to a multiple of the forecast horizon h ranging from $2h$ to $7h$. $\tilde{\mathbf{x}}_{T_w}$ is fed to the next block for its input and $\tilde{\mathbf{y}}_{T_w}$ is added to a stack to produce a forecast.

Internally, the basic building block is divided into four parts. The first part consists of W independent FC input layers that project the signal into a fixed higher-dimensional representation $\mathbf{z}_\ell \in R^+$ with the same dimensionality for all W models. The second part consists of a shared FC stack that takes as input the TS representation produced in the first part and outputs forward θ_ℓ^f and backward θ_ℓ^b predictors of expansion coefficients for each of W lookback periods. The third part consists of the W independent backward g_ℓ^b and forward g_ℓ^f basis layers that take as input their respective forward θ_ℓ^f and backward θ_ℓ^b expansion coefficients, projecting them over basis functions to produce the backcast $\tilde{\mathbf{x}}_{\ell,T_w} \in \mathbb{R}^T$ and the forecast $\tilde{\mathbf{y}}_{\ell,T_w} \in \mathbb{R}^H$. This approach allows us to parallelize the computation of the forecast by considering input of $\mathbf{X} \in \mathbb{R}^{N\times T\times W}$ where $T = \max_{w\in W} T_w$, and producing output $\tilde{\mathbf{Y}} \in \mathbb{R}^{N\times H\times W}$.

The computation of the forecast and backcast for a block given the signal of length T_w, is described by the following equations:

$$\mathbf{z}_{T_w,1} = \mathrm{FC}_{\ell,T_w}(\mathbf{x}_{T_w}^\ell) \tag{2}$$

$$\mathbf{z}_{T_w,l} = \mathrm{FC}_\ell(\mathbf{z}_{T_w,l-1}) \tag{3}$$

$$\theta_{T_w}^f = \mathrm{FC}_{T_w}^f(\mathbf{z}_{T_w,L}) \tag{4}$$

$$\theta_{T_w}^b = \mathrm{FC}_{T_w}^b(\mathbf{z}_{T_w,L}) \tag{5}$$

$$\tilde{\mathbf{y}}_{T_w}^\ell = \sum_{i=1}^{\dim(\theta_{\ell,T_w}^f)} \theta_{\ell,T_w}^f \mathbf{v}_{i,T_w}^f \tag{6}$$

$$\tilde{\mathbf{x}}_{T_w}^\ell = \sum_{i=1}^{\dim(\theta_{\ell,T_w}^b)} \theta_{\ell,T_w}^b \mathbf{v}_{i,T_w}^b \tag{7}$$

Here $l \in \mathbb{N}^+, 1 < l < L, l_l \in L = \{l_1, \cdots l_L\}$ is the number of layer sin the ℓ-th block and $Z_{T_w,l}$ corresponds to the embedding computed for the l-th hidden layers. FC corresponds to a fully connected layer with ReLU non-linearity activation [45], and $\mathbf{v}_{i,T_w}^f$ and $\mathbf{v}_{i,T_w}^b$ are forecast and backcast basis vectors. These vectors can either be chosen to be learnable parameters or can be set to specific functional forms that are fixed prior to training the model.

Eqs.2-7 are then repeated iteratively for ℓ blocks, following the same architecture topology as N-BEATS [41]. The individual blocks are stacked using two residual branches. The first branch, illustrated in Fig. 1 (middle), runs over the backcast signal produced by each block and iteratively decomposes the initial TS signal such that the subsequent block consider the residual of its preceding block. The second branch, illustrated in Fig. 1 (right), aggregates the partial forecast of each block and aggregate the total forecast by adding them. These operations are described by the following equations:

$$\mathbf{x}_{T_w}^{\ell+1} = \mathbf{x}_{T_w}^{\ell} - \tilde{\mathbf{x}}_{T_w}^{\ell} \tag{8}$$

$$\tilde{\mathbf{y}}_{T_w} = \sum_{i}^{\ell} \tilde{\mathbf{y}}_{T_w}^{\ell} \tag{9}$$

4.2. Generic and Interpretable Model Version

Multiple versions of this approach can be provided to parameterize each of the W models. For instance, both the generic and interpretable versions of N-BEATS proposed in [41] are compatible with our model. We will briefly these two extensions; we refer the reader to the original paper for more details [41].

The generic architecture: in this version, g_{ℓ,T_w}^{b} and g_{ℓ,T_w}^{f} are specified as a linear projection of the previous layer output such that the outputs of the ℓ-th block are described as follows:

$$\tilde{\mathbf{y}}_{T_w}^{\ell} = \mathbf{V}_{\ell,T_w}^{f}\theta_{\ell,T_w}^{f} + \mathbf{B}_{\ell,T_w}^{f} \quad \tilde{\mathbf{x}}_{T_w}^{\ell} = \mathbf{V}_{\ell,T_w}^{b}\theta_{\ell,T_w}^{b} + \mathbf{B}_{\ell,T_w}^{b} \tag{10}$$

where $\mathbf{V}_{\ell,T_w}^{f} \in \mathbb{R}^{H\times dim(\theta_{\ell,T_w}^{f})}$, $\mathbf{B}_{\ell,T_w}^{f} \in \mathbb{R}^{H}$ and $\mathbf{V}_{\ell,T_w}^{b} \in \mathbb{R}^{T\times dim(\theta_{\ell,T_w}^{b})}$, $\mathbf{B}_{\ell,T_w}^{b} \in \mathbb{R}^{b}$ are basis vectors learned by the model, which can be taught as waveforms, but we do not have inherent constraints on how these waveforms should look.

The interpretable architecture: Similar to the traditional TS decomposition into trend and seasonality found in [46, 47], trend and seasonality decomposition can be enforced in $\mathbf{V}_{\ell,T_w}^{f}$ and $\mathbf{V}_{\ell,T_w}^{b}$. [41] proposed to do this by conceptually separating the L block into two sets such that one stack of blocks is parameterized with a **trend model** and the other with a **seasonal model**. The **trend model** consists of constraining the basis function to modelize a trend signal, i.e., $g^f_{\ell,T_w}(\theta^f_{\ell,T_w}; \mathbf{V}^f_{s,\ell,T_w})$, using a function polynomial of small degree p as follows:

$$\tilde{\mathbf{y}}^{\ell}_{T_w} = \mathbf{T}\theta^f_{\ell,T_w}; \mathbf{T} = [\mathbf{1}, t, \cdots t^p] \tag{11}$$

where $\mathbf{T}$ is a matrix of powers of p. Thus the waveform extracted will follow a monotonic or a slowly varying function. The **seasonal model** constrains the basis function to modelize periodic functions, i.e, $g^f_{\ell,T_w}(\theta^f_{\ell,T_w}; \mathbf{V}^f_{s,\ell,T_w})$, using Fourier series as follows:

$$\begin{split}\tilde{\mathbf{y}}^{\ell}_{T_w} = \mathbf{S}\theta^f_{\ell,T_w}; \mathbf{S} = [\mathbf{1}, cos(2\pi\mathbf{t}, \cdots, cos(2\pi\lfloor H/2-1\rfloor\mathbf{t}),\\ sin(2\pi\lfloor H/2-1\rfloor\mathbf{t})]\end{split} \tag{12}$$

Thus, by first (1) applying the **trend model** and then (2) applying the **seasonal model** within the doubly residual stacking topology of the model, we obtain a model that applies TS component decomposition in a similar way to than traditional decomposition approaches.

In any configuration of the model, estimating the parameters of the model problem is done by maximum likelihood estimation (MLE). To simplify the notation, we consider eq. 13 as the function that establishes the forecast, where θ_{NBEATS} is the set of all parameters of each block and $\mathbf{x}^i_{T_w}$ is the n-th TS considered with input size of length T_w.

$$\tilde{\mathbf{y}}_{T_w} = \text{NBEATS}(\mathbf{x}^n_{T_w}; \theta_{\text{NBEATS}}) \tag{13}$$

Thus, optimizing the model consists of optimizing eq. 14. We use a stochastic gradient descent optimization with Adam [48] and a three-steps learning rate schedule. Here $\mathcal{L}(NBEATS(\mathbf{x}^n_{T_w}; \theta_{\text{NBEATS}}), \mathbf{y}^{(n)})$ corresponds to some metric function that measures the quality of the forecast to the ground truth $\mathbf{Y}$. Note that we combine the losses of the forecasts of all models, using the mean values to promote cooperation between the different models. Following the same training framework as [41], we used the MAPE, MASE and SMAPE losses to build the ensemble, all of which are detailed in the following section.

$$\theta^*_{\text{NBEATS}} = \underset{\theta^*_{\text{NBEATS}}}{\text{argmin}} \frac{1}{N}\sum_{n=0}^{N}\frac{1}{W}\sum_{w=1}^{W}\mathcal{L}(NBEATS(\mathbf{x}^n_{T_w}; \theta_{\text{NBEATS}}), \mathbf{y}^{(n)}) \tag{14}$$

5. Experimental setup

We conducted the experimental evaluation of the forecasting methods on 6 datasets which include a total of 105'968 unique TS when combined and over 2.5 million forecasts to produce on these TS:

(1) (*public*) **M4:** 100'000 heterogeneous TS from multiple sectors that include economic, finance, demographics and other industry used in the M4 TS competition [8, 26].

(2) (*public*) **M3:** 3003 heterogeneous TS from derived from mostly from financial and economic domains [49].

(3) (*public*) **Tourism:** 1311 TS of indicators related to tourism activities sampled monthly, quarterly and yearly [50, 51].

(4) (*public*) **Electricity:** 370 TS of the hourly electricity usage of 370 customers over three years [52, 53].

(5) (*public*) **Traffic:** 963 TS of the hourly occupancy rates on the San Francisco Bay Area freeways scaled between 0 and 1 [52, 53].

(6) (*proprietary*) **Finance:** 321 TS observed between 2005-07-01 and 2020-10-16 of the adjusted daily closing price of various U.S. mutual funds and exchange traded funds traded on U.S. financial markets, each covering different types of asset classes including stocks, bonds, commodities, currencies and market indexes, or a proxy for a market index covering a larger set than the dataset used in [54].

For the **M4, M3** and **Tourism** datasets, target TS trajectories were specified by the competition's organizers with each subpopulation of TS with the same frequency (Hourly, Quarterly, etc..) having its own horizon. For the **Electricity** and **Traffic** datasets, the test was set using rolling window operation as described in Appendix A.4. For the **Finance** dataset, the forecast was evaluated on three rolling forecast setups by sampling the TS on different frequencies, i.e.: daily, weekly and monthly. In total there are 2'602'878 individual TS that were sampled from the 321 original ones across 3 forecast horizons. Despite the dataset being collected from proprietary data sources which we cannot redistribute, we provide the necessary details to help interested readers reconstruct the datasets in Appendix A.5.

We trained our model on the **M4** dataset. We replicated the results from [41] by training the two N-BEAT model variants discussed in Sec. 4.2 using the implementation provided by the original authors. We evaluated their performance in the zero-shot regime on all other datasets by training the model using scaled TS as in [3], i.e., we divided all TS observations by the maximum values observed. The reason for this preprocessing step was to prevent catastrophic failure when the target dataset scale is significantly different from that of the source dataset.

We tested 3 different setups for zero-shot forecasting, which we denote by R_O, $R_{SH,LT}$ and R_{SH}. R_O is a setup where we use the same model to produce results on **M4** (Table 1) and apply it on the target dataset. This required us to truncate the forecast or apply the model iteratively on the basis of previous forecasts to ensure the forecast size is the same as the target dataset. The model was not trained to operate when this condition occurs. R_{SH} is a setup where the model is trained with the same number of iterations as R_O but we specified the model's forecast horizon to be the same that of the target datasets. $R_{SH,LT}$ is the same training regime as R_{SH}, but we allowed the model to consider TS samples from further in the past during training and trained the model with more iterations. To produce forecasts, we used the subset of the ensemble models trained on the same TS frequency to produce the multiple forecasts and combined them by median aggregation. Detailed explanations of this aggregation and the ensemble parameters used are given in Appendix B.

We compared the forecast accuracy of our approaches with the reported accuracy of other TS models such as DEEP-STATE [15], N-BEATS [41, 3], DEEP-AR [14], FFORMA [40], ES-RNN [44], Deep Factors [55] and many others trained on the source dataset. In reporting the accuracy of these models, we relied upon the accuracy and the pre-computed forecasts reported in their respective original paper. The statistical models were produced on R using the forecast package [56] and we measured the training time to train and produce each forecast on R. For the other models, we relied upon the reported running time of the implementation provided in [26]. Finally, all models were compared on a naive forecast, i.e., a random walk model or a seasonally adjusted random walk, that assumes all future values will be the same as the last known one(s). This was done to assess whether the forecasts of these models are accurate in the first place.

$$MAPE(\tilde{\boldsymbol{x}}, \boldsymbol{x}) = \frac{100}{h} \sum_{i=1}^{H} \frac{|\tilde{\boldsymbol{x}}_{T+i} - \boldsymbol{x}_{T+i}|}{\boldsymbol{x}_{T+i}} \tag{15}$$

$$MASE(\tilde{\boldsymbol{x}}, \boldsymbol{x}) = \frac{1}{H} \sum_{i=1}^{H} \frac{|\boldsymbol{x}_{T+i} - \tilde{\boldsymbol{x}}_{T+i}|}{\frac{1}{T+H-s} \sum_{j=s+1}^{T+H} |\boldsymbol{x}_j - \boldsymbol{x}_{j-m}|} \tag{16}$$

$$SMAPE(\tilde{\boldsymbol{x}}, \boldsymbol{x}) = \frac{200}{H} \sum_{i=1}^{H} \frac{|\boldsymbol{x}_{T+i} - \tilde{\boldsymbol{x}}_{T+i}|}{|\boldsymbol{x}_j| + |\tilde{\boldsymbol{x}}_{T+i}|} \tag{17}$$

$$OWA(\tilde{\boldsymbol{x}}, \boldsymbol{x}) = \frac{1}{2} \left[\frac{SMAPE}{SMAPE_{NAIVE2}} + \frac{MASE}{MASE_{NAIVE2}} \right] \tag{18}$$

$$ND(\tilde{\boldsymbol{x}}, \boldsymbol{x}) = \frac{\sum_{i=1}^{H} |\tilde{\boldsymbol{x}}_{T+i} - \boldsymbol{x}_{T+i}|}{\sum_{i=1}^{H} |\boldsymbol{x}_{T+i}|} \tag{19}$$

$$MDA(\tilde{\boldsymbol{x}}, \boldsymbol{x}) = \frac{1}{H} \sum_{i=0}^{T} \mathrm{sign}(\tilde{\boldsymbol{x}}_{t:t+i} - \boldsymbol{x}_{t-1}) = \mathrm{sign}(\boldsymbol{x}_{t:t+i} - \boldsymbol{x}_{t-1}) \tag{20}$$

We evaluated the forecast accuracy using 8 standard TS metrics [39, 2, 26, 50]: the mean absolute percentage error (MAPE), the mean absolute scaled error (MASE), the scaled mean absolute percentage error (SMAPE), the normalized deviation (ND) and the mean directional accuracy (MDA). Additionally for the M4 competition, we evaluated the model on the overall weighted average (OWA) between the SMAPE and the MASE such that a seasonally-adjusted naive (NAIVE2) forecasting model obtains a score of 1.0. For instance, an OWA of 0.90 means that the forecast is on average 10% better than a NAIVE2 model with respect to both the SMAPE and MASE metrics. The MDA measures the model's ability to produce forecasts where the trajectory follows the actual change of the TS relative to the last known value: the higher the MDA is, the better a model predicts the trend of a TS at any given time. For all other metrics, the lower the value, the better a model predicts the TS. Eq. 15-20 describes how these metrics are computed assuming $\tilde{x}$ is the forecast and x is the ground truth.

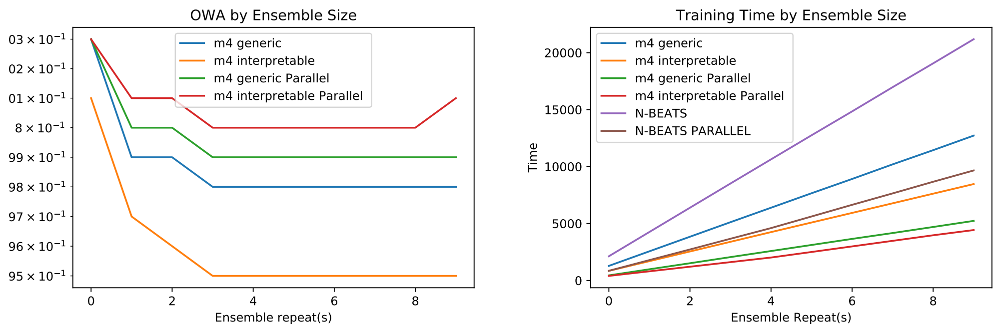

Figure 2: OWA metric (left) and Time (right) for the different N-BEATS model configuration as a function of ensemble size.

5.1. Baseline and Benchmark

We present the results of the baseline and benchmark accuracies for the M4 dataset in Table 1. The table gives the reported accuracy of N-BEATS reported in the original papers [41], the replicated results using the publicly accessible implementation provided by the original authors along with their scaled versions [3] based upon their implementation and our model NBEATS(P). Three main conclusions can be drawn from this table:

(1) Scaling TS with a scaling coefficient to permit generalization on other datasets for the N-BEATS model, as presented in [3], adds a penalty on the OWA metrics for the M4 dataset, which suggests that there is a trade-off between accuracy and generalization on other datasets for DNN-based models.

(2) Figure 2 details how the impact of the model is affected by ensemble size and training time. It can be seen that applying a bagging procedure [57] 3 to 4 times is sufficient to get an accurate ensemble for both the NBEATS and NBEATS(P) model but NBEATS(P) is more efficient the larger the ensemble size.

(3) The top-performing models do not differ significantly with respect to the coverage of the TS forecasted and the mean directional accuracy (MDA). This provides an argument that if one is mainly interested in predicting the TS variation from the forecast origin, relying on the fastest implementation of the top-performing models for a first initial prediction is a cost-effective solution.

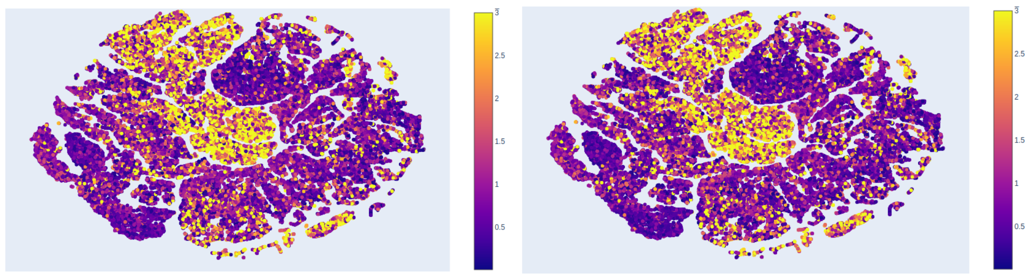

Figure 3: MASE coverage for the ES-RNN (left) and N-BEATS(i) (right) over the M4 dataset. Each point on the graphs corresponds to a single TS and the darker its color, the better the given model. Horizontal and vertical coordinates represent the value of the embedding. All TS with MASE values over 3 where assigned the same color to facilitate visualization.

Regarding **(2)**, we illustrate this phenomenon in fig. 3 by plotting the TSNE embedding of each series of M4 computed from the same set of features used in [40] and comparing the top performing model with the N-BEATS(i) model by coloring each TS with its individual MASE accuracy. Note that there are no substantial differences between the approaches, despite some subtle regions of the graph where we can observe N-BEATS(i) performing better overall than ES-RNN.

5.2. Training Time and Number of Parameters

Table 2 presents the time to train each model accuracy as well as the average pairwise absolute percentage error correlation of the forecast residuals between ensemble members, in a way to similar to the experimental evaluation of M4 submission performed in [62]. At first glance, we see that the training time of these models be as long as multiple weeks. In comparison to N-BEATS, N-BEATS(P) takes significantly less time, especially for the generic architecture. Both the original approach and ours are at the same level of correlation and, while ours is slightly less diverse, it is roughly twice as efficient and achieves the same level of accuracy. We can observe that using the scaled version has little impact in terms of diversity. In our preliminary result, we also observed that there was no significant difference in terms of the TS samplers used to train N-BEATS(P) where for instance, different TS were sampled for the W model.

The difference in improvement factor between parallelized generic and interpretable versions of N-BEATS(P) is due to the hidden layer sizes between

	Yearly	Quarterly	Monthly	Others	Average	Coverage	
	OWA					MASE % ($< T = 1.0$)	MDA(%)
NAIVE	1.000	1.066	1.095	1.335	1.058	40.299	3.2
NAIVE2	1.000	1.000	1.000	1.000	1.000	43.288	33.1
SNAIVE	1.000	1.153	1.146	0.945	1.078	36.095	42.7
ARIMA [24]	0.892	0.898	0.903	0.967	0.903	51.145	53.8
HOLT [58]	0.947	0.932	0.988	1.180	0.971	48.659	61.7
ETS [59]	0.903	0.890	0.914	0.974	0.908	50.987	48.6
THETA [27]	0.872	0.917	0.907	0.995	0.897	48.686	61.7
SES [25]	1.002	0.970	0.951	0.995	0.975	44.719	35.3
DAMPED [26]	0.890	0.893	0.924	1.005	0.907	49.838	61.1
COMB [26]	0.867	0.890	0.920	1.039	0.898	49.784	61.3
MLP' [26, 60]	1.288	1.684	1.749	3.028	1.642	26.603	60.6
RNN' [26, 60]	1.308	1.508	1.587	1.702	1.482	28.437	59.8
ProLogistica [61]	0.820	0.855	0.867	0.742	0.841	53.620	62.6
FFORMA [40]	0.799	0.847	0.858	0.914	0.838	53.418	63.7
ES-RNN [44]	0.778	0.847	0.836	0.920	0.821	53.271	63.2
N-BEATS (I) [41]	0.765	0.800	0.820	0.822	0.797	—	—
N-BEATS (G) [41]	0.758	0.807	0.824	0.849	0.798	—	—
N-BEATS (I+G) [41]	0.758	0.800	0.819	0.840	0.795	—	—
Ours:							
N-BEATS (G) [41]	0.770	0.793	0.818	0.832	0.798	55.576	64.6
N-BEATS (I) [41]	0.763	0.797	0.817	0.838	0.795	55.600	63.7
N-BEATS (I+G) [41]	0.761	0.792	0.814	0.834	0.793	55.868	64.6
N-BEATS (G) *scaled* [3]	0.784	0.810	0.827	0.836	0.809	54.960	64.5
N-BEATS (I) *scaled* [3]	0.773	0.817	0.826	0.843	0.806	54.919	63.7
N-BEATS (I+G) *scaled* [3]	0.778	0.814	0.824	0.836	0.806	55.109	64.4
N-BEATS parallel (G)	0.764	0.804	0.820	0.855	0.799	55.332	64.4
N-BEATS parallel (I)	0.759	0.817	0.824	0.850	0.801	54.966	63.7
N-BEATS parallel (I+G)	0.757	0.806	0.820	0.851	0.796	55.375	64.5
N-BEATS parallel (G) *scaled*	0.775	0.829	0.833	0.851	0.812	54.506	63.8
N-BEATS parallel (I) *scaled*	0.772	0.845	0.844	0.867	0.819	53.772	63.6
N-BEATS parallel (I+G) *scaled*	0.771	0.834	0.835	0.854	0.813	54.344	63.9

Table 1: Averaged forecasting results of the M4 competition for the evaluated models. The OWA metric is presented for each seasonal pattern observed. Forecasts from models in *italics* were pre-computed except for the N-BEATS models. We replicate the results with the implementation provided by the authors, e.g: *N-BEATS (I)* **(original)** vs N-BEATS (I) **(our)**. *MLP'* and *RNN'* models are appended with "'" to signify that these model were trained per TS using a seasonal and trend decomposition with manual pre- and post processing steps [26]. We also considered a coverage indicator which measures the number of series that a model forecasts better than an arbitrary MASE accuracy threshold of $\tau = 1.0$. We also added the MDA of the forecast.

the two versions. It turns out that having a higher number of hidden neurons reduce the computational gain of training multiple models conjointly as it saturate GPU usage. If we have a sufficiently expressive model without requiring too many hidden neurons, N-BEATS(P) is expected to produce accurate forecasts at a fraction of the cost. Otherwise the gain is not significant. Regardless, these results show that ensemble diversity and accurate forecasts could have been achieved with reduction in resources and computation time.

	Time (min.)	**Corr. (mean, std.)**		**Time (min.)**	**Corr. (mean, std.)**
ProLogistica [61]	39655	$--\pm--$			
FFORMA [40]:	46108	$--\pm--$			
ES-RNN [44]:	8056	$--\pm--$			
N-BEATS(G) [41]	11773	0.85 ± 0.02	N-BEATS(G) *scaled*	11755	0.84 ± 0.04
N-BEATS(I) [41]	7437	0.89 ± 0.02	N-BEATS(I) *scaled*	6607	0.85 ± 0.03
N-BEASTS(I+G) [41]	19211	0.84 ± 0.03	N-BEATS(I+G) *scaled*	19170	0.84 ± 0.03
N-BEATS(P, G) **(our)**	**5301**	0.87 ± 0.02	N-BEATS(P,G) *scaled* **(our)**	**6157**	0.89 ± 0.02
N-BEATS(P, I) **(our)**	**6990**	0.88 ± 0.04	N-BEATS(P,I) *scaled* **(our)**	**4785**	0.89 ± 0.08
N-BEATS(P, I+G) **(our)**	11840	0.87 ± 0.02	N-BEATS(P,I+G) *scaled* **(our)**	10943	0.88 ± 0.02

Table 2: Time required to train to train all members of the ensemble and average& standard deviation of the absolute percentage correlation between ensemble members on the test sets

Given the increasing trend of top-performing models requiring ever more training time [26], training and deploying state-of-the-art models in real-case scenarios can entail high costs for organizations — costs that are avoidable. For instance, on Google's cloud platform, the estimated cost of training N-BEATS(P) would drop to 530.11 USD$ instead of the 860.13 USD$ their price simulator gives for N-BEATS [2]. Thus, in terms of both cost and time saved, our work provides encouraging results that suggest how multiple TS ensemble models can be accelerated without any great drawback by sharing a subset of their parameterization.

5.3. Zero-Shot forecasting

Once trained, N-BEATS(P) can be applied to multiple TS in a zero-shot regime and produce competitive forecasts even in settings where TS are notoriously difficult to forecast. We tested our model on **M3**, **Tourism**, **Electricity** and **Traffic**. We also considered the **Finance** dataset to explore a set of TS that were not explored in the zero-shot setting. Table 4 describes the

[2]Prices are at the rate calculated using their cost estimator on 04-08-2021, employing their "AI Platform" configuration with a single NVIDA P100 GPU

Model name	# of parameters	Model name	# of parameters
N-BEATS(G)	28'508'265'900	N-BEATS(P, G)	**5'972'957'400**
N-BEATS(I)	42'288'737'310	N-BEATS(P, I)	**8'102'076'930**
N-BEATS(I+G)	70'797'003'210	N-BEATS(P, I+G)	**14'075'034'330**

Table 3: Number of parameters for the whole ensemble for N-BEATS and N-BEATS-CNN trained on the M4 dataset with 6 lookback windows.

zero-shot performance of N-BEATS and N-BEATS(P). Several observations can be made:

(1) N-BEATS(P) produces comparable zero-shot results to previous state-of-the-art models for all datasets. In other training regimes, where models trained with the same forecast horizon or longer ones, comparable levels of accuracy were observed.

(2) Comparing with [3], where a different training regime was used, the difference between their results and ours highlights the importance of the optimization procedure to facilitate transfer to another dataset. In certain cases, some datasets (e.g., **Tourism**, will benefit from a longer training, to the detriment of the forecast accuracy on the source dataset. In other cases, like the **Electricity** dataset, no adjustments are required between the source and the target dataset.

(3) The case of the **Tourism** dataset highlights the importance of ensuring that the forecast horizon of the source dataset used to train the model is longer than or equal of the target dataset; this is a key factor in producing reliable zero-shot forecasts.

Considering that the **M4** dataset includes a large number of heterogenous TS that contain at least some TS with similar statistical properties to those present in the target dataset, zero-shot forecasting can be easily deployed with a pre-trained DNN model and can produce initial forecasts that are on par with the level of accuracy of multiple baselines, and sometimes benchmarks, very quickly.

However, not all settings will benefit equally from this approach. The **Finance** dataset is a prime example of a setting where zero-shot forecasting produce mixed results. In this setting, the source TS dataset has very few

TS to train from in comparison to the test sets. Also, these TS are very difficult to forecast in a univariate setting since they are almost all non-ergodic, heteroscedastic, and have high noise-to-signal ratio. Despite these added difficulties, both N-BEATS and N-BEATS(P) can produce forecasts at a comparable level to a single statistical model in term of MDA. However, the zero-shot regime achieved forecasts better than a naive one only by sampling these TS at a monthly frequency, which coincidentally is the largest pool of TS in M4 (48'000.) In comparison, the daily (3594) and weekly (227) subsets contain fewer TS. Hence, even under poor conditions of application for zero-shot N-BEATS(P), we can still produce preliminary forecasts quickly. These results highlight the importance of selecting a good source dataset but even in subpar conditions, our approach can still generalize well with respect to the MDA metric.

M3, SMAPE		**Tourism**, MAPE		**Electricity**, ND		**Traffic**, ND
N-SHOT:						
Naive	16.59	*SNaive*	24.80	Naive	0.37	0.57
Comb [26]	13.52	*ETS* [59]	20.88	*MatFact* [53]	0.16	0.17
ForePro [51]	13.19	*Theta* [27]	20.88	*DeepAR* [14]	0.07	0.17
Theta [27]	13.01	*ForePro* [51]	19.84	*DeepState* [15]	0.08	0.17
DOTM [63]	12.90	*Strato* ■	19.52	*Theta* [27]	0.08	0.18
EXP [64]	12.71	*LCBaker* [65]	19.35	*ARIMA* [24]	0.07	0.15
N-BEATS[41]	12.37		18.52		0.07	0.11
*DEEP-AR**[14, 3]	12.67		19.27		0.09	0.19
ZERO-SHOT: ($R_{SH,LT}/R_{SH}/R_O$)						
M4 N-BEATS (G) *scaled** [41]	12.36/12.67/12.72		18.90/20.16/24.14		0.09/0.09/0.08	0.16/0.16/0.14
M4 N-BEATS (I) *scaled** [41]	12.43/12.63/12.66		19.43/20.58/23.26		0.10/0.09/0.08	0.16/0.16/0.14
M4 N-BEATS (g+i) *scaled** [41]	12.38/12.61/12.64		19.04/20.22/23.43		0.10/0.09/0.08	0.16/0.16/0.14
M4 N-BEATS (P+G) *scaled*	12.48/12.65/12.65		18.99/19.98/22.85		0.09/0.09/0.08	0.16/0.18/0.14
M4 N-BEATS (P+I) *scaled*	12.69/12.76/12.72		20.54/20.97/23.18		0.09/0.10/0.09	0.17/0.17/0.16
M4 N-BEATS (P+G&I) *scaled*	12.56/12.67/12.64		19.50/20.24/22.79		0.09/0.09/0.08	0.16/0.16/0.14

Table 4: Averaged forecasting for the zero-shot regime for each dataset; lower values are better. Zero-shot forecasts are compared for N-BEATS and our approach. For the models in *italic* using the following references [3, 50, 49, 14, 15]. For zero-shot results, we show the metrics for three training regimes: $R_{SH,LT}/R_{SH}/R_O$. R_O is the same model used to produce the results on **M4** (Table. 1), which required to truncation of the forecast or applying the model iteratively at the basis of previous forecasts to ensure the forecast size was the same that of the target dataset. R_{SH} is trained in the same fashion as R_O but we specified the model's forecast horizon to be the same as that of the target datasets. $R_{SH,LT}$ is the same training regime as R_{SH} except that the model is allowed to consider TS samples from further in the past while training: See Table. B.10 for more detail. Results for models with * appended to their names are replicated from the original papers and ■ signifies an anonymous submission for which we do not know the methodology.

Models	Daily ($H = 14$)$N = (1'222'866)$			Weekly ($H = 13, N = 1'091'898$)			Monthly ($H = 18, N = 288'114$)		
	OWA	MDA	Time (min.)	OWA	MDA	Time (min.)	OWA	MDA	Time (min.)
N-SHOT:									
Naive	1.000	07.1	—	1.00	02.0	—	1.000	00.5	—
ARIMA [24]	1.041	27.1	4685	1.059	28.9	3597	**0.891**	40.3	816
THETA [27]	**0.995**	49.4	241	**0.993**	53.5	262	0.913	**61.4**	49
SES [25]	1.000	09.3	174	1.001	05.2	160	1.000	02.9	25
HOLT [58]	1.081	49.6	167	1.116	**53.9**	160	0.931	60.3	42
ETS [59]	1.019	20.4	969	1.044	18.8	512	0.940	29.2	181
ZERO-SHOT:									
M4 N-BEATS (G) [41]	1.165	50.3	24	1.078	50.9	21	0.963	55.1	6
M4 N-BEATS (I) [41]	1.222	50.5	26	1.045	51.3	23	0.962	54.9	6
M4 N-BEATS (I+G) [41]	1.191	**50.7**	50	1.056	51.1	44	0.961	55.4	12
M4 N-BEATS (P+G)	1.210	48.5	25	1.098	51.6	21	0.973	54.4	6
M4 N-BEATS (P+I)	1.135	48.5	26	1.055	52.1	24	0.975	54.1	6
M4 N-BEATS (P+I&G)	1.139	49.0	51	1.044	52.0	45	0.973	54.4	12

Table 5: Comparison between statistical baselines and zero-shot application of the N-BEATS model in terms of OWA, MDA and time to produce forecast. Forecasts were made with the native zero-shot approach (R_O).

6. Conclusion

We proposed an efficient novel architecture for training multiple TS models conjointly for univariate TS forecasting. We empirically validated the flexibility of our approach on the M4 TS datasets as well as assessing its generalizability to other domains of application, using 5 other datasets which, combined, cover over 2.5 million forecasts. We provided forecasts in various TS settings at the same level of accuracy as current state-of-the-art models with a model that is twice as fast while requiring 5 times fewer parameters than the top performing model. We highlighted both stylized facts and limitations of the performance of the model studied, in an effort to provide insights to TS practitioners for operating DNN-based models at scale. Our results suggest that training global univariate models conjointly by sharing parts of their parameterizations yield competitive forecasts in a fraction of the time and does not significantly impair either forecast accuracy or ensemble diversity.

Acknowledgement

This work was supported in part by the NSERC CRD - Quebec Prompt - Laplace Insights - EAM - joint program under grants CRDPJ 537461-18 and 114-IA-Wang-DRC 2019 to Dr. S. Wang, by Mitacs Acceleration under Grant IT13429 to Dr. S. Wang and P. Chatigny and by FQRNT under Grant B2 270188 to P. Chatigny.

References

[1] S. Makridakis, R. J. Hyndman, F. Petropoulos, Forecasting in social settings: The state of the art, International Journal of Forecasting 36 (1) (2020) 15–28.

[2] S. Makridakis, E. Spiliotis, V. Assimakopoulos, The m5 accuracy competition: Results, findings and conclusions (10 2020).

[3] B. N. Oreshkin, D. Carpov, N. Chapados, Y. Bengio, Meta-learning framework with applications to zero-shot time-series forecasting, arXiv preprint arXiv:2002.02887 (2020).

[4] Z. Chen, Y. Cao, Y. Liu, H. Wang, T. Xie, X. Liu, A comprehensive study on challenges in deploying deep learning based software, in: Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, 2020, pp. 750–762.

[5] A. Paleyes, R.-G. Urma, N. D. Lawrence, Challenges in deploying machine learning: a survey of case studies, arXiv preprint arXiv:2011.09926 (2020).

[6] A. D'Amour, K. Heller, D. Moldovan, B. Adlam, B. Alipanahi, A. Beutel, C. Chen, J. Deaton, J. Eisenstein, M. D. Hoffman, et al., Underspecification presents challenges for credibility in modern machine learning, arXiv preprint arXiv:2011.03395 (2020).

[7] P. Montero-Manso, R. J. Hyndman, Principles and algorithms for forecasting groups of time series: Locality and globality, International Journal of Forecasting (2021).

[8] S. Makridakis, E. Spiliotis, V. Assimakopoulos, The m4 competition: Results, findings, conclusion and way forward, International Journal of Forecasting 34 (4) (2018) 802–808.

[9] Google, 2017. web traffic time series forecasting., https://www.kaggle.com/c/web-traffic-time-series-forecasting, accessed: 2021–06-15 (2017).

[10] A. Hooshmand, R. Sharma, Energy predictive models with limited data using transfer learning, in: Proceedings of the Tenth ACM International Conference on Future Energy Systems, 2019, pp. 12–16.

[11] M. Ribeiro, K. Grolinger, H. F. ElYamany, W. A. Higashino, M. A. Capretz, Transfer learning with seasonal and trend adjustment for cross-building energy forecasting, Energy and Buildings 165 (2018) 352–363.

[12] S. Smyl, J. Ranganathan, A. Pasqua, M4 forecasting competition: Introducing a new hybrid es-rnn model, URL: https://eng. uber. com/m4-forecasting-competition (2018).

[13] Y. Rubanova, T. Q. Chen, D. K. Duvenaud, Latent ordinary differential equations for irregularly-sampled time series, in: Advances in Neural Information Processing Systems, 2019, pp. 5321–5331.

[14] D. Salinas, V. Flunkert, J. Gasthaus, T. Januschowski, Deepar: Probabilistic forecasting with autoregressive recurrent networks, International Journal of Forecasting 36 (3) (2020) 1181–1191.

[15] S. S. Rangapuram, M. W. Seeger, J. Gasthaus, L. Stella, Y. Wang, T. Januschowski, Deep state space models for time series forecasting, in: Advances in NIPS, 2018, pp. 7785–7794.

[16] J.-H. Böse, V. Flunkert, J. Gasthaus, T. Januschowski, D. Lange, D. Salinas, S. Schelter, M. Seeger, Y. Wang, Probabilistic demand forecasting at scale, Proceedings of the VLDB Endowment 10 (12) (2017) 1694–1705.

[17] V. Flunkert, D. J. B. A. Salinas, Recurrent neural network-based artificial intelligence system for time series predictions, uS Patent 10,936,947 (Mar. 2 2021).

[18] C. Liang, Y. Lu, Using automl for time series forecasting (2020).
URL https://ai.googleblog.com/2020/12/using-automl-for-time-series-forecasting.html

[19] F. Bell, S. Smyl, Forecasting at uber: An introduction, Uber Engineering (2018).

[20] N. Laptev, Time-series modeling with neural networks at uber (2020).
URL https://forecasters.org/wp-content/uploads/gravity_forms/7-c6dd08fee7f0065037affb5b74fec20a/2017/07/Laptev_Nikolay_ISF2017.pdf

[21] B. Banushev, R. Barclay, Enhancing trading strategies through cloud services and machine learning (Jan 2021).
URL https://aws.amazon.com/fr/blogs/industries/enhancing-trading-strategies-through-cloud-services-and-machine-learning/

[22] T. Januschowski, Forecasting at amazon: Problems, methods and systems, https://forecasters.org/wp-content/uploads/gravity_forms/

7-c6dd08fee7f0065037affb5b74fec20a/2017/07/Januschowski_Tim_ISF2017.pdf, accessed: 2021–06-15 (2018).

[23] P. Gupta, P. Malhotra, J. Narwariya, L. Vig, G. Shroff, Transfer learning for clinical time series analysis using deep neural networks, Journal of Healthcare Informatics Research 4 (2) (2020) 112–137.

[24] G. E. Box, G. M. Jenkins, G. C. Reinsel, G. M. Ljung, Time series analysis: forecasting and control, John Wiley & Sons, 2015.

[25] R. Hyndman, A. B. Koehler, J. K. Ord, R. D. Snyder, Forecasting with exponential smoothing: the state space approach, Springer Science & Business Media, 2008.

[26] S. Makridakis, E. Spiliotis, V. Assimakopoulos, The m4 competition: 100,000 time series and 61 forecasting methods, International Journal of Forecasting 36 (1) (2020) 54–74.

[27] V. Assimakopoulos, K. Nikolopoulos, The theta model: a decomposition approach to forecasting, International journal of forecasting 16 (4) (2000) 521–530.

[28] J. A. Fiorucci, Time series forecasting : advances on Theta method, Ph.D. thesis, Universidade Federal de São Carlos (9 2016).

[29] R. J. Hyndman, B. Billah, Unmasking the theta method, International Journal of Forecasting 19 (2) (2003) 287–290.

[30] J. D. Hamilton, A new approach to the economic analysis of nonstationary time series and the business cycle, Econometrica: Journal of the Econometric Society (1989) 357–384.

[31] R. Prado, G. Huerta, M. West, Bayesian time-varying autoregressions: Theory, methods and applications, Resenhas do Instituto de Matemática e Estatística da Universidade de São Paulo 4 (4) (2000) 405–422.

[32] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation 9 (8) (1997) 1735–1780.

[33] I. Koprinska, D. Wu, Z. Wang, Convolutional neural networks for energy time series forecasting, in: 2018 International Joint Conference on Neural Networks (IJCNN), IEEE, 2018, pp. 1–8.

[34] A. Van Den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, K. Kavukcuoglu, Wavenet: A generative model for raw audio., SSW 125 (2016).

[35] S. Li, X. Jin, Y. Xuan, X. Zhou, W. Chen, Y.-X. Wang, X. Yan, Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting, in: Advances in Neural Information Processing Systems, 2019, pp. 5243–5253.

[36] N. Wu, B. Green, X. Ben, S. O'Banion, Deep transformer models for time series forecasting: The influenza prevalence case, arXiv preprint arXiv:2001.08317 (2020).

[37] S. Makridakis, E. Spiliotis, V. Assimakopoulos, Statistical and machine learning forecasting methods: Concerns and ways forward, PloS one 13 (3) (2018) e0194889.

[38] C. Chatfield, et al., Apples, oranges and mean square error, International Journal of Forecasting 4 (4) (1988) 515–518.

[39] R. J. Hyndman, A. B. Koehler, Another look at measures of forecast accuracy, International journal of forecasting 22 (4) (2006) 679–688.

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

[41] B. N. Oreshkin, D. Carpov, N. Chapados, Y. Bengio, N-beats: Neural basis expansion analysis for interpretable time series forecasting, in: International Conference on Learning Representations, 2019.

[42] R. T. Clemen, Combining forecasts: A review and annotated bibliography, International journal of forecasting 5 (4) (1989) 559–583.

[43] A. Timmermann, Forecast combinations, Handbook of economic forecasting 1 (2006) 135–196.

[44] S. Smyl, A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting, International Journal of Forecasting 36 (1) (2020) 75–85.

[45] V. Nair, G. E. Hinton, Rectified linear units improve restricted boltzmann machines, in: Icml, 2010.

[46] R. B. Cleveland, W. S. Cleveland, J. E. McRae, I. Terpenning, Stl: a seasonal-trend decomposition, Journal of official statistics 6 (1) (1990) 3–73.

[47] U. C. Bureau, X-13arima-seats reference manual (2016).

[48] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: Y. Bengio, Y. LeCun (Eds.), 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015.
URL http://arxiv.org/abs/1412.6980

[49] A. J. Koning, P. H. Franses, M. Hibon, H. O. Stekler, The m3 competition: Statistical tests of the results, International Journal of Forecasting 21 (3) (2005) 397–409.

[50] G. Athanasopoulos, R. J. Hyndman, H. Song, D. C. Wu, The tourism forecasting competition, International Journal of Forecasting 27 (3) (2011) 822–844.

[51] G. Athanasopoulos, R. J. Hyndman, The value of feedback in forecasting competitions, International Journal of Forecasting 27 (3) (2011) 845–849.

[52] D. Dua, C. Graff, UCI machine learning repository (2017).
URL http://archive.ics.uci.edu/ml

[53] H.-F. Yu, N. Rao, I. S. Dhillon, Temporal regularized matrix factorization for high-dimensional time series prediction, in: Advances in neural information processing systems, 2016, pp. 847–855.

[54] P. Chatigny, J.-M. Patenaude, S. Wang, Spatiotemporal adaptive neural network for long-term forecasting of financial time series, arXiv preprint arXiv:2003.12194 (2020).

[55] Y. Wang, A. Smola, D. C. Maddix, J. Gasthaus, D. Foster, T. Januschowski, Deep factors for forecasting, arXiv preprint arXiv:1905.12417 (2019).

[56] R. J. Hyndman, Y. Khandakar, et al., Automatic time series forecasting: the forecast package for r, Journal of statistical software 27 (3) (2008) 1–22.

[57] L. Breiman, Bagging predictors, Machine learning 24 (2) (1996) 123–140.

[58] C. Chatfield, M. Yar, Prediction intervals for multiplicative holt-winters, International Journal of Forecasting 7 (1) (1991) 31–37.

[59] R. J. Hyndman, A. B. Koehler, R. D. Snyder, S. Grose, A state space framework for automatic forecasting using exponential smoothing methods, International Journal of forecasting 18 (3) (2002) 439–454.

[60] I. Goodfellow, Y. Bengio, A. Courville, Y. Bengio, Deep learning, Vol. 1, MIT press Cambridge, 2016.

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

[62] P. Agathangelou, D. Trihinas, I. Katakis, Correlation analysis of forecasting methods: The case of the m4 competition, International Journal of Forecasting 36 (1) (2020) 212–216.

[63] J. A. Fiorucci, T. R. Pellegrini, F. Louzada, F. Petropoulos, A. B. Koehler, Models for optimising the theta method and their relationship to state space models, International Journal of Forecasting 32 (4) (2016) 1151–1161.

[64] E. Spiliotis, V. Assimakopoulos, K. Nikolopoulos, Forecasting with a hybrid method utilizing data smoothing, a variation of the theta method and shrinkage of seasonal factors, International Journal of Production Economics 209 (2019) 92–102.

[65] L. C. Baker, J. Howard, Winning methods for forecasting tourism time series, International Journal of Forecasting 27 (3) (2011) 850–852.

[66] L. v. d. Maaten, G. Hinton, Visualizing data using t-sne, Journal of machine learning research 9 (Nov) (2008) 2579–2605.

[67] A. Asuncion, D. Newman, Uci machine learning repository (2007).

[68] L. Vincent, N. Thome, Shape and time distortion loss for training deep time series forecasting models, in: Advances in NIPS, 2019, pp. 4191–4203.

[69] R. Kurle, S. S. Rangapuram, E. de Bézenac, S. Günnemann, J. Gasthaus, Deep rao-blackwellised particle filters for time series forecasting, Advances in Neural Information Processing Systems 33 (2020).

[70] S. Wu, X. Xiao, Q. Ding, P. Zhao, Y. Wei, J. Huang, Adversarial sparse transformer for time series forecasting, Advances in Neural Information Processing Systems 33 (2020).

Appendix A. Dataset

Fig. A.4 illustrates the difference between the statistical properties of all 6 datasets, employing the same set of TS features used in [40]. We refer the reader [40] for a detailed overview of the 42 features used and their interpretation. As an example of the observations that can be drawn from this figure: it can bee seen that both the **Electricy** and **Traffic** datasets exhibit multiple seasonal patterns, whereas datasets like **Finance** exhibit large difference from other datasets in terms of high order autocorrelation (x_acf10), autoregressive conditional heteroscedasticity (archlm, garch_r2), strength of trend (trend) and high variance of the mean of observation from non-overlapping windows (stability).

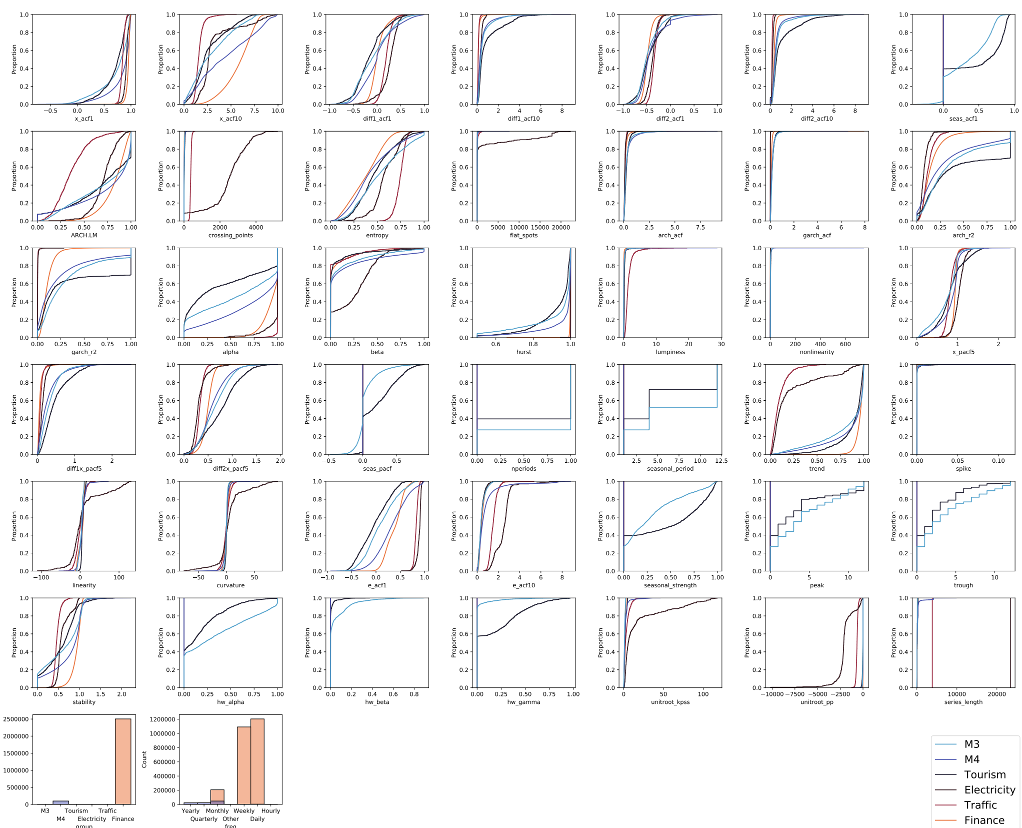

Figure A.4: Cumulative distribution function plot for TS datasets over 42 statistical TS features and TS count by dataset and frequency.

All sampled TS from all datasets are summarized in Fig. A.5 using the T-SNE algorithm [66]. Each point of this graph correspond to the 2-dimensional

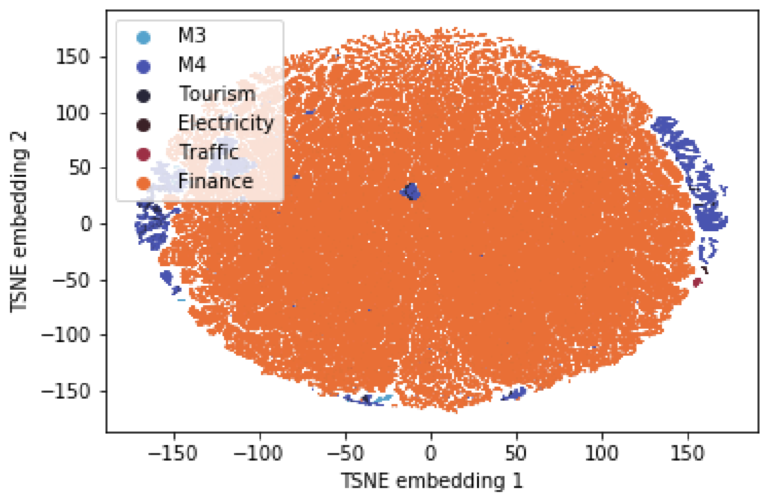

Figure A.5: TSNE embedding of all TS forecasted with their different subpopulations.

embedded space of a single TS computed from the same set of endogenous statistical features[40] . One can observe the heterogeneity of these datasets and note that the subpopulations of TS within a dataset can have high variance in their statistical properties while being similar to other subpopulations of other datasets. When considering the **Finance** dataset, we can see how the behavior of the 321 TS changes significantly over time and that the statistical behavior of these TS are more heterogenous than that of the one of the public datasets. The **Electricity** & **Traffic** TS share almost the same statistical properties as both populations of TS are concentrated in the same region of the graph.

Appendix A.1. M4 Dataset Details

	Frequency/Horizon						
Type	Yearly ($h = 6$)	Quarterly ($h = 8$)	Monthly ($h = 18$)	Weekly ($h = 13$)	Daily ($h = 14$)	Hourly ($h = 48$)	Total
Demographic	1'088	1'858	5'728	24	10	0	8'708
Finance	6'519	5'305	10'987	164	1'559	0	24'534
Industry	3'716	4'673	10'987	164	422	0	18'798
Macro	3'903	5'315	10'016	41	127	0	19'402
Micro	6'538	6'020	10'975	112	1'476	0	25,121
Other	1'236	865	277	12	633	414	3437
Total	23'000	24'000	48'000	359	4277	414	100'000

Table A.6: Composition of the **M4** TS dataset: number of time series based on their sampling frequency and type.

The **M4**[3] dataset is a publicly accessible dataset that contains a large set of 100'000 heterogenous TS sampled from the *ForeDeCk* database for the M4 competition[8]. The database is compiled at the National Technical University of Athens and is built from multiple diverse and publicly accessible sources. It includes TS frequently encountered in business domains such as industries, services, tourism, imports/exports, demographics, education, labor & wage, government, households, bonds, stocks, insurances, loans, real estate, transportation, and natural resources & environment. TS were sampled at different frequencies [Yearly, Quarterly, Monthly, Weekly, Daily and Hourly] each with different forecast horizons, i,e, [6, 8, 18, 13, 14, 48] according to the competition organizer. Table A.6 outlines the composition of the M4 dataset across domains and forecast horizons.

All TS were provided with a prepossessing scaling procedure to ensure positive observed values at all time-steps with minimum observed values greater than or equal to 10. The scaling was applied only to sampled TS whose minimum oberved value was smaller than 10 by adding a per-TS constant to all TS to ensure that the minimal values was positive. All other TS were unaltered by any preprocessing step. The dataset was subdivized into a training and a test dataset by the M4 TS competition organizers. For further details on this dataset, we refer the reader to the following: [26, 8]. We relied on the pre-computed forecasts and PI available at `https://github.com/Mcompetitions/M4-methods`.

Appendix A.2. M3 Dataset Details

	Frequency/Horizon				
Type	Yearly ($h = 6$)	Quarterly ($h = 8$)	Monthly ($h = 18$)	Other ($h = 8$)	Total
Demographic	245	57	111	0	413
Finance	58	76	145	29	308
Industry	102	83	334	0	519
Macro	83	336	312	0	731
Micro	146	204	474	4	828
Other	11	0	52	141	204
Total	645	756	1428	174	3'003

Table A.7: Composition of the **M3** TS dataset: the number of TS based on sampling frequency and type.

The **M3**[4] dataset is a publicly accessible dataset that is smaller than the M4

[3] `https://github.com/Mcompetitions/M4-methods`
[4] `https://forecasters.org/resources/time-series-data/m3-competition/`

dataset but remains relatively large and diverse. Similarly to the M4 dataset, it contains TS frequently encountered in business, financial and economic forecasting. It include yearly, quarterly, monthly, weekly, daily and hourly time series, each with different forecast horizons, i,e, [6, 8, 18, 13, 14, 48]. All series have positive observed values at all time-steps. The dataset was subdivided into a training and a test dataset by the M3 TS competition organizers. Table A.7 outlines the composition of the M3 dataset across domains and forecast horizons. For further details on this dataset, we refer the reader to [49]. This dataset was considered for zero-shot forecasting, to examine a case where the target dataset is from the same domains of application but with other TS.

Appendix A.3. Tourism Dataset Details

	Frequency/Horizon			
Type	Yearly ($h = 4$)	Quarterly ($h = 8$)	Monthly ($h = 24$)	Total
Tourism	518	427	366	1311

Table A.8: Composition of the **Tourism** TS dataset: number of time series based on sampling frequency and type.

The **Tourism**[5] dataset is a publicly accessible dataset that contains TS collected by [50] from tourism government agencies and academics who had used them in previous tourism forecasting studies. The TS of this dataset are highly variable in length. It includes yearly, quarterly and monthly TS. Table. A.8 details the proportion of TS from each frequency. For further detail on this dataset, we refer the reader to [50]. This dataset was considered for zero-shot forecasting, to examine a case where the target dataset comes from domains that are not present in the M4 dataset.

Appendix A.4. Electricity and Traffic Datasets Details

Electricity[6] and **Traffic** [67] are two publicly available datasets from the Univeristy of California Irvine Machine Learning repository. The **Electricity** dataset contains the hourly electricity usage monitoring of 370 customers over three

[5]https://robjhyndman.com/data/27-3-Athanasopoulos1.zip
[6]https://archive.ics.uci.edu/ml/datasets/PEMS-SF

years, with some clients being added during the the observation periods creating cold-start conditions for producing some forecasts. The **Traffic** dataset contains TS of the hourly occupancy rates, scaled in the (0,1) range for 963 lanes of freeways in the San Francisco Bay area over a period of slightly more than a year. Both of these dataset exhibit strong seasonal patterns due to their nature and are mostly homogeneous. These two TS datasets are used de facto to evaluate the quality of DNN-based TS models [14, 15, 68, 69, 70]. We included these two datasets as a sanity check for zero-shot forecasting, to ensure that zero-shot forecasts were accurate in a setting where it is relatively easy to produce accurate forecasts.

Appendix A.5. Finance dataset

The `Finance` dataset contains daily closing prices of U.S. MFs and ETFs observed between 2005-07-01 and 2020-10-16 and traded on U.S. financial markets, each covering different types of asset classes including stocks, bonds, commodities, currencies and market indexes, or a proxy for a market index. The dataset was obtained through three data providers: (**1**) `Fasttrack`[7], a professional-grade data provider for financial TS, (**2**) `Yahoo` Finance API and (**3**) the Federal Reserve of Saint-Louis (FRED) database. Part of this dataset is proprietary, so we do not have permission to share that part publicly. However, the list of securities is given in Table. C.11 to help interested readers reconstruct the dataset from public data sources.

We considered this dataset in our zero-shot experiments by sampling the TS at three different frequencies [dDaily, weekly and monthly] and specifying the same forecast horizon as that of the **M4** dataset. We used this dataset to present a worst-case scenario for zero-shot. First this is a case where the forecasting application is notorious for its forecasting difficulty. Moreover, the source dataset on which we train our model has at most 10K TS to train from and at worst 164 TS, which force zero-shot generalization with very few training data. Also, by sampling the TS at large scale, we emulated how zero-shot could be applied on the whole history of the TS, similar to the procedure carried out by portfolio managers and quantitative analyst to backtest the validity of their investment strategies. The TS were split into chunks of the maximum lookback period of the N-BEATS model as training sample, and h steps-ahead as testing sample.

	Frequency/Horizon					
Frequency	Yearly ($h = 6$)	Quarterly ($h = 8$)	Monthly ($h = 18$)	Weekly ($h = 13$)	Daily ($h = 14$)	Hourly ($h = 48$)
L_H	1.5	1.5	1.5	10	10	10
Iterations NBEATS P	10k	15k	15k	5k	5k	5k
Iterations NBEATS	15k	15k	15k	5k	5k	5k
Learning rate	0.001					
Losses	SMAPE,MASE,MAPE					
Lookback periods	2H,3H,4H,5H,6H,7H					
Batch size	1024					
Kernel size	1					
	N-BEATS(G)			**N-BEATS(P+G)**		
Width	512					
Blocks	1					
Blocks-layer	4					
# Stacks	30=[**Generic**, $\cdots$, **Generic**]					
	N-BEATS(I)			**N-BEATS(P+I)**		
# Stacks	2 = [**Trend, Seasonality**]					
T-width	265					
T-blocks	3					
T-blocks-layer	4					
S-width	2048					
S-blocks	3					
S-blocks-layer	4					
Sharing	Stack level					

Table B.9: Hyper parameters used to produce results on the **M4** TS dataset

	Native zero-shot (R_O)					
Frequency	Yearly	Quarterly	Monthly	Weekly	Daily	Hourly
horizon	($h = 6$)	($h = 8$)	($h = 18$)	($h = 13$)	($h = 14$)	($h = 48$)
L_H	1.5	1.5	1.5	10	10	10
Iterations NBEATS	15k	15k	15k	5k	5k	5k
Iterations NBEATS (P)	10k	15k	15k	5k	5k	5k
	Native Zero-Shot with equal forecast horizon (R_{SH})					
horizon	$(h = h^{(\mathscr{D}_{tgrt.})}_{\text{Yearly}})$	$(h = h^{(\mathscr{D}_{tgrt.})}_{\text{Quarterly}})$	$(h = h^{(\mathscr{D}_{tgrt.})}_{\text{Monthly}})$	$(h = h^{(\mathscr{D}_{tgrt.})}_{\text{Weekly}})$	$(h = h^{(\mathscr{D}_{tgrt.})}_{\text{Daily}})$	$(h = h^{(\mathscr{D}_{tgrt.})}_{\text{Hourly}})$
L_H	1.5	1.5	1.5	10	10	10
Iterations	15k	15k	15k	5k	5k	5k
	Native Zero-Shot with equal forecast horizon($R_{SH,LT}$)					
horizon	$(h = h^{(\mathscr{D}_{tgrt.})}_{\text{Yearly}})$	$(h = h^{(\mathscr{D}_{tgrt.})}_{\text{Quarterly}})$	$(h = h^{(\mathscr{D}_{tgrt.})}_{\text{Monthly}})$	$(h = h^{(\mathscr{D}_{tgrt.})}_{\text{Weekly}})$	$(h = h^{(\mathscr{D}_{tgrt.})}_{\text{Daily}})$	$(h = h^{(\mathscr{D}_{tgrt.})}_{\text{Hourly}})$
L_H	10	10	10	10	10	10
Iterations	15k	15k	15k	15k	15k	15k

Table B.10: HP differences between the different zero-shot strategies. All models were trained on the **M4** TS dataset

Appendix B. Training setup details

We used the same overall training framework as [41] including the stratified uniform sampling of TS in the source dataset to train the model. Training N-BEATS and N-BEATS(P) was done by first segmenting the training dataset into non-overlapping subsets based on the TS frequency they were observed in. Then, independent training instances were trained, one each group by specifying the forecast horizon of each instance based on the common forecast horizon of the subset. Table B.9 presents the HP settings used to train all N-BEATS and NBEATS(P) models on the different subsets of M4. Except for the number of iterations on the yearly TS, all other HPs are the same.

For zero-shot application, we relied on the scaled version of each model, i.e. where the TS is scaled based on its maximum observed value over its lookback periods. With one exception, the model trained on a given frequency split of a source dataset is used to forecast the same frequency split on the target dataset. The only exception is follows: when transferring from **M4** to **M3**, the Other subpopulation of **M3** is forecast with the model trained on the **Quarterly** subpopulation of **M4**. Table B.10 describe the different zero-shot training regimes on which the model was trained on the source dataset.

Appendix B.1. Forecasting Combination

Forecast combination with N-BEATS(P) and NBEATS was done as follows: to produce a forecast from the ensemble, all forecasts of ensemble members were considered and the median was computed for every forecast for all time t per TS forecast. When the forecast horizon of the model was shorter than the forecast horizon of the target dataset, we iteratively appended the forecast to the original TS signal and based our forecasts upon the transformed signal until the total forecast was longer than or equal to the forecast horizon of the target dataset. In cases where the forecast produced was longer than the forecast horizon, we truncated the forecast to keep only the h first observations.

Appendix C. Finance dataset: List of Securities Considered

Data Source: Yahoo

[7] https://investorsFasttrack.com

Ticker	Description	Class
DJAT	Dow Jones Asian Titan 50 Index	Regional Stock Index
DJI	Dow Jones Industrial Average	Stock Index (US)
DJT	Dow Jones Transportation Average	Stock Index (US)
DJU	Dow Jones Utility Average	Stock Index (US)
GSPC	S&P 500	Stock Index (US)
IXIC	NASDAQ Composite	Stock Index (US)
NDX	NASDAQ-100	Stock Index (US)
OEX	S&P 100	Stock Index (US)
XMI	NYSE Arca Major Market Index	Stock Index (US)
DX-Y.NYB	US Dollar/USDX - Index - Cash	Forex
FDCPX	Fidelity Select Computers	US Sector Stock Index
HSI	HANG SENG INDEX (Currency in HKD)	National Stock Index
Data Source: Fred		
GOLDPMGBD228NLBM	Gold Fixing Price 3:00 P.M. (London time) in London Bullion Market & based in U.S. Dollars	Others
WILL4500IND	Wilshire 4500 Total Market Index	Stock Index (US)
WILL4500PR	Wilshire 4500 Price Index	Stock Index (US)
WILL5000IND	Wilshire 5000 Total Market Index	Stock Index (US)
WILL5000INDFC	Wilshire 5000 Total Market Full Cap Index	Stock Index (US)
WILL5000PR	Wilshire 5000 Price Index	Stock Index (US)
Data Source: FastTrack		
FPX1	CAC 40 Ix	National Stock Index
SHCP	Shanghai Composite Ix	National Stock Index
SPXX	STOXX Europe 600 Ix	Regional Stock Index
SX5P	STOXX Europe 50 Ix	Regional Stock Index
A-CWI	MSCI ACWI DivAdj Idx	Global Stock Index
A-XUS	MSCI ACWI xUS DivAdj Idx	Global Stock Index
AUD-	US / Australia Foreign Exchange Rate	Forex
BBG-	CBOE US T-Bill 13-Week Yld Bd Ix	US Bonds - Gvmnt
BBG-9	BBG Barclay Agg Bond- US Universal TR Ix	US Bonds - Gvmnt

BBG-G	BBG Barclay Agg Bond- US Corp IG TR Ix	US Bonds - Gvmnt
BBG-H	ML US HY Bb-B Ix	US Bonds - Corp HY
BBG-I	BBG Barclay Agg Bond- US Agency Long Ix	US Bonds - Gvmnt
BBG-O	BBG Barclay Agg Bond- Yankee Ix	US Bonds - Gvmnt
BBG-S	BBG Barclay Agg Bond- US MBS Agncy TR Ix	US Bonds - Gvmnt
BBG-T	BBG Barclay Agg Bond- US MBS Agncy TR Ix	US Bonds - Gvmnt
BBG-U	BBG Muni Bond 3yr Idx	US Bonds - Gvmnt
BBG-Y	BBG Muni Bond 20yr Idx	US Bonds - Gvmnt
BBM-2	BBG Muni Bond 5yr Idx	US Bonds - Gvmnt
BBM-3	BofAML US Corp 5-7yr Total Return Ix	US Bonds - Corp Invst
BBM-5	BBG Muni Bond Composite Idx	US Bonds - Gvmnt
BBM-I	BBG Muni Bond Long Term Idx	US Bonds - Gvmnt
BBM-L	BBG Muni Bond 10yr Idx	US Bonds - Gvmnt
BBM-T	BBG Barclay Agg Bond- US Composite TR Ix	US Bonds - Gvmnt
CAD-	Canada / US Foreign Exchange Rate Ix	Forex
CDN-X	Canadian Dollar For 100 CDN Ix	Forex
CHF-	Switzerland/ US Foreign Exchange Rate Ix	Forex
CNY-	China / US Foreign Exchange Rate Ix	Forex
CR-TR	CRB Total Return Ix	Commodities
DBC	Invesco DB Commodity Index Tracking Fund	Commodities
DKK-	Denmark / US Foreign Exchange Rate Ix	Forex
DXY-Z	US Dollar Ix	Forex
EFA	iShares MSCI EAFE ETF	Regional Stock Index
EURO-	US/Euro Foreign Exchange Rate Ix	Forex
EWA	iShares ETF MSCI Australia	National Stock Index
EWC	iShares ETF MSCI Canada	National Stock Index
EWD	iShares ETF MSCI Sweden	National Stock Index

EWG	iShares ETF MSCI Germany	National Stock Index
EWH	iShares ETF MSCI Hong Kong	National Stock Index
EWI	iShares ETF MSCI Italy Capped	National Stock Index
EWJ	iShares MSCI Japan ETF	National Stock Index
EWK	iShares ETF MSCI Belgium Capped	National Stock Index
EWL	iShares ETF MSCI Switzerland Capped	National Stock Index
EWM	iShares ETF MSCI Malaysia	National Stock Index
EWN	iShares ETF MSCI Netherlands	National Stock Index
EWO	iShares ETF MSCI Austria Capped	National Stock Index
EWP	iShares ETF MSCI Spain Capped	National Stock Index
EWS	iShares ETF MSCI Singapore	National Stock Index
EWW	iShares ETF MSCI Mexico Capped	National Stock Index
EWY-X	MSCI Korea iShr Ix	National Stock Index
EWZ-X	MSCI Brazil iShr Ix	National Stock Index
FBIOX	Fidelity Select Biotechnology	US Sector Stock Index
FBMPX	Fidelity Select Communication Services Portfolio	US Sector Stock Index
FCYIX	Fidelity Select Industrials	US Sector Stock Index
FDAC-	Frankfurt Dax Ix	National Stock Index
FDFAX	Fidelity Select Consumer Staples	US Sector Stock Index
FDLSX	Fidelity Select Leisure	US Sector Stock Index
FEZ-X	Europe 50 STOXX stTr Ix	Regional Stock Index
FIDSX	Fidelity Select Financial Services	US Sector Stock Index
FNARX	Fidelity Select Natural Resources	US Sector Stock Index
FNMIX	Fidelity New Markets Income	Regional Stock Index
FRESX	Fidelity Fidelity Real Estate Investment Portfolio	Others
FSAGX	Fidelity Select Gold	US Sector Stock Index
FSAIX	Fidelity Select Air Transportation	US Sector Stock Index
FSAVX	Fidelity Select Automotive	US Sector Stock Index
FSCHX	Fidelity Select Chemicals	US Sector Stock Index
FSCPX	Fidelity Select Consumer Discretion	US Sector Stock Index
FSCSX	Fidelity Select software & Comp Service	US Sector Stock Index

FSDAX	Fidelity Select Defense & Aerospace	US Sector Stock Index
FSDCX	Fidelity Select Commun Equipment	US Sector Stock Index
FSDPX	Fidelity Select Materials	US Sector Stock Index
FSELX	Fidelity Select Semiconductors	US Sector Stock Index
FSENX	Fidelity Select Energy	US Sector Stock Index
FSESX	Fidelity Select Energy Service	US Sector Stock Index
FSHCX	Fidelity Select Health Care Service	US Sector Stock Index
FSHOX	Fidelity Select Const & Housing	US Sector Stock Index
FSLBX	Fidelity Select Brokrg & INV Mgt	US Sector Stock Index
FSLEX	Fidelity Select Environmental & Alt	US Sector Stock Index
FSNGX	Fidelity Select Natural Gas	US Sector Stock Index
FSPCX	Fidelity Select Insurance	US Sector Stock Index
FSPHX	Fidelity Select Health Care	US Sector Stock Index
FSPTX	Fidelity Select Technology	US Sector Stock Index
FSRBX	Fidelity Select Banking	US Sector Stock Index
FSRFX	Fidelity Select Transportation	US Sector Stock Index
FSRPX	Fidelity Select Retailing	US Sector Stock Index
FSTCX	Fidelity Select Telecommunications	US Sector Stock Index
FSUTX	Fidelity Select Utilities	US Sector Stock Index
FSVLX	Fidelity Select Consumer Finance	US Sector Stock Index
FTSE-	London FT-SE 100 Ix	National Stock Index
GBP-	US / UK Foreign Exchange Rate Ix	Forex
GLD	SPDR Gold Shares	Commodities
HKD-	Hong Kong / US Foreign Exchange Rate Ix	Forex
HY-	ML US HY Broadcastng Ix	US Bonds - Corp HY
HY-BC	ML US HY Build Mterl Ix	US Bonds - Corp HY
HY-BM	ML US HY Capitl Good Ix	US Bonds - Corp HY
HY-CG	ML US HY Chemicals Ix	US Bonds - Corp HY
HY-CH	ML US HY CCC & Lower Ix	US Bonds - Corp HY
HY-CL	ML US HY Containers Ix	US Bonds - Corp HY
HY-CN	ML US HY Consum Prod Ix	US Bonds - Corp HY
HY-CP	ML US HY Div Fin Svc Ix	US Bonds - Corp HY
HY-DF	ML US HY Div Media Ix	US Bonds - Corp HY

HY-DM	ML US HY Entert Film Ix	US Bonds - Corp HY
HY-EF	ML US HY Energy Ix	US Bonds - Corp HY
HY-EG	ML US HY Environmntl Ix	US Bonds - Corp HY
HY-EN	ML US HY Ex Telecom Ix	US Bonds - Corp HY
HY-ET	ML US HY Fd Bvrge Tb Ix	US Bonds - Corp HY
HY-FB	ML US HY Fd&Drg Retl Ix	US Bonds - Corp HY
HY-FD	ML US HY Homebldr Re Ix	US Bonds - Corp HY
HY-HB	ML US HY Healthcare Ix	US Bonds - Corp HY
HY-HC	ML US HY Insurance Ix	US Bonds - Corp HY
HY-IN	ML US HY Leisure Ix	US Bonds - Corp HY
HY-LE	ML US HY Metal Minng Ix	US Bonds - Corp HY
HY-MM	ML US HY Paper Ix	US Bonds - Corp HY
HY-PP	ML US HY Publsh Prnt Ix	US Bonds - Corp HY
HY-PR	ML US HY Restaurants Ix	US Bonds - Corp HY
HY-RS	ML US HY Super Retl Ix	US Bonds - Corp HY
HY-SR	ML US HY Steel Ix	US Bonds - Corp HY
HY-ST	ML US HY Services Ix	US Bonds - Corp HY
HY-SV	ML US HY Tech&Aerosp Ix	US Bonds - Corp HY
HY-TA	ML US HY Telecommnct Ix	US Bonds - Corp HY
HY-TC	ML US HY Cabl Sat Tv Ix	US Bonds - Corp HY
HY-TV	ML US HY Utilities Ix	US Bonds - Corp HY
HY-UT	ML US Indl Corps A Ix	US Bonds - Corp Invst
IC-1A	ML US Indl Corps AA Ix	US Bonds - Corp Invst
IC-2A	ML US Indl Corps AAA Ix	US Bonds - Corp Invst
IC-3A	ML US Indl Corps BBB Ix	US Bonds - Corp Invst
IEF	iShares ETF 7 10 Year Treasury Bond	US Bonds - Gvmnt
INE-X	MSCI Italy iShr Ix	National Stock Index
INH-X	MSCI Hong Kong iShr Ix	National Stock Index
INR-	India/ US Foreign Exchange Rate Ix	Forex
INR-X	MSCI Singapore iShr Ix	National Stock Index
IWC-X	Russell Microcap	Stock Index (US)
IXF-X	NASDAQ Financial-100	Stock Index (US)
JPY-	Japan/ US Foreign Exchange Rate Ix	Forex
KRW-	South Korea / US Exchange Rate Ix	Forex
LLQ-X	Russell Microcap - Dividend Adj	Stock Index (US)
LLR-X	Russell Microcap	Stock Index (US)

LQD	iShares iBoxx $ Investment Grade Corporate Bond ETF	US Bonds - Corp Invst
M-BRC	MSCI Emerging Markets BRIC DivAdj Idx	Global Stock Index
M-CN	MSCI China DivAdj Ix	National Stock Index
M-DEO	MSCI Dev Mkts Euro DivAdj Idx	Global Stock Index
M-WD	MSCI World DivAdj Idx	Global Stock Index
M16Y-	BofAML US Corporate A Effective Yield Ix	US Bonds - Corp Invst
M26Y-	BofAML US Corporate AA Effective Yield I	US Bonds - Corp Invst
M36Y-	BofAML US Corporate AAA Effective Yield	US Bonds - Corp Invst
M3EY-	BofAML US Corp AAA Option-Adj Spread Ix	US Bonds - Corp Invst
M46Y-	BofAML US Corporate BBB Effective Yield	US Bonds - Corp Invst
M56Y-	BofAML US Corporate 1-3 Year Effective Y	US Bonds - Corp Invst
M5EY-	BofAML US Corp 1-3Y Option-Adj Spread Ix	US Bonds - Corp Invst
M66Y-	BofAML US Corporate 3-5 Year Effective Y	US Bonds - Corp Invst
M6EY-	BofAML US Corp 3-5Y Option-Adj Spread Ix	US Bonds - Corp Invst
M76Y-	BofAML US Corporate 5-7 Year Effective Y	US Bonds - Corp Invst
M7TR-	BBG Muni Bond 7yr Idx	US Bonds - Gvmnt
M86Y-	BofAML US Corporate 7-10 Year Effective	US Bonds - Corp Invst
M8TR-	BBG Muni Bond 1yr Idx	US Bonds - Corp Invst
M96Y-	BofAML US Corporate 10-15 Year Effective	US Bonds - Corp Invst
M9EY-	BofAML US HY BB Option-Adj Spread Ix	US Bonds - Corp HY
MDY	StateSt ETF SPDR S&P MID-CAP 400	Stock Index (US)
MF6Y-	BofAML US Corporate 15 Year Effective Yi	US Bonds - Corp Invst
MFEY-	ML US T-Bill 0-3mo Div-Adj Ix	US Bonds - Gvmnt

ML-03	ML US T-Bill 1-10yrs Ix	US Bonds - Gvmnt
ML-10	ML US T-Bill 1-3yrs Div-Adj Ix	US Bonds - Gvmnt
ML-13	ML US T-Bill 12mo Div-Adj Ix	US Bonds - Gvmnt
ML-1Y	ML US T-Bill 3-5yrs Div-Adj Ix	US Bonds - Gvmnt
ML-35	ML US T-Bill 3-6mo Div-Adj Ix	US Bonds - Gvmnt
ML-36	ML US T-Bill 6mo Div-Adj Ix	US Bonds - Gvmnt
ML-6T	ML US T-Bill 7-10yrs Ix	US Bonds - Gvmnt
ML-70	BofAML US High Yield B Total Return Inde	US Bonds - Corp HY
ML-I0	ML US T-Bill 1-10yrs Infl-Lnk Ix	US Bonds - Gvmnt
ML-I1	BBG Barclay Agg Bond- TBill Tips TR Ix	US Bonds - Gvmnt
ML-TB	ML US Corp Non-Fd&Dru Ret Ix	US Bonds - Corp Invst
MLB-	BofAML US High Yield BB Total Return Ind	US Bonds - Corp HY
MLBB-	BofAML US High Yield CCC or Below Total	US Bonds - Corp HY
MLCC-	ML US HY Master II D-A H0A0 Ix	US Bonds - Corp HY
MLHY-	ML BBB Grade Div-Adj Muni Ix	US Bonds - Gvmnt
MLMB-	ML Municipal Master Div-Adj Ix	US Bonds - Gvmnt
MLMM-	ML US T-Bill Div-Adj Ix	US Bonds - Gvmnt
MXN-	Mexico / US Foreign Exchange Rate Ix	Forex
OSX-X	AMEX Oil Service HLDRS Ix	Commodities
PCY	Invesco Emerging Markets Sovereign Debt ETF & US Bonds - Corp HY	
RTF-X	Russell Top 50	Stock Index (US)
RU2-D	Russell 2500 - Dividend Adj	Stock Index (US)
RUA-D	Russell 3000 - Dividend Adj	Stock Index (US)
RUA-X	Russell 3000	Stock Index (US)
RUI-D	Russell 1000 - Dividend Adj	Stock Index (US)
RUI-I	Russell 1000	Stock Index (US)
RUM-D	Russell MidCap - Dividend Adj	Stock Index (US)
RUP-D	Russell Top 200 - Dividend Adj	Stock Index (US)
RUP-X	Russell Top 200	Stock Index (US)
RUS-D	Russell Small Cap - Dividend Adj	Stock Index (US)
RUT-D	Russell 2000 - Dividend Adj	Stock Index (US)
RUT-U	Russell 2000 - Unadj	Stock Index (US)

S-100	S&P Global 100 Ix	Global Stock Index
SEK-	Sweden / US Foreign Exchange Rate Ix	Forex
SGD-	Singapore / US Foreign Exchange Rate Ix	Forex
SHY	iShares 1-3 Year Treasury Bond ETF	US Bonds - Gvmnt
THB-	Thailand / US Foreign Exchange Rate Ix	Forex
TIP	iShares TIPS Bond ETF	US Bonds - Gvmnt
TLT	iShares 20+ Year Treasury Bond ETF	US Bonds - Gvmnt
TWD-	Taiwan / US Foreign Exchange Rate Ix	Forex
UC-	ML US Corp 10 Yrs Ix	US Bonds - Corp Invst
UC-10	ML US Corp 15 Yrs Ix	US Bonds - Corp Invst
UC-15	ML US Corp Gs&Elct Utl 1-10 Yrs Ix	US Bonds - Corp Invst
UC-G1	ML US Corp Gas&Elect Utl Ix	US Bonds - Corp Invst
UC-G4	ML US Corp Phones 10-15 Yrs Ix	US Bonds - Corp Invst
UC-LC	ML US T-Bill 7-10yrs Infl-Lnk Ix	US Bonds - Gvmnt
UC-P1	ML US Corp Phones 15 Yrs Ix	US Bonds - Corp Invst
UC-P2	ML US Corp Utils&Phones Ix	US Bonds - Corp Invst
UC-UP	ML US Corp Large Cap Ix	US Bonds - Corp Invst
US05-	BofAML US Corporate 7-10yr Total Return	US Bonds - Corp Invst
UUP	Invesco DB US Dollar Index Bullish Fund	Forex
VASVX	Vanguard Selected Value Fund	Stock Index (US)
VBISX	Vanguard Short Term Bond Index	US Bonds - Gvmnt
VEIEX	Vanguard Emerging Market Stock Index INV	Regional Stock Index
VEXMX	Vanguard Extended Market Index Fund	Global Stock Index
VEXPX	Vanguard Explorer Fund INV	Stock Index (US)
VFICX	Vanguard Int. Term Investment Grade Bond Fund	US Bonds - Corp Invst
VFIIX	Vanguard GNMA INV	US Bonds - Gvmnt
VFISX	Vanguard Short-Term Treasury INV	US Bonds - Gvmnt

VFITX	Vanguard Intermediate Term Treasury Fund	US Bonds - Gvmnt
VFSTX	Vanguard Short-Term INV Growth Incm INV	US Bonds - Corp Invst
VGENX	Vanguard Energy INV	National Stock Index
VGHCX	Vanguard Health Care INV	National Stock Index
VGPMX	Vanguard Global Capital Cycles Fund	Stock Index (US)
VGSIX	Vanguard REIT Index INV	Others
VINEX	Vanguard International Explorer Fund	Global Stock Index
VNQ	Vanguard Real Estate Index Fund ETF Shares	Others
VTRIX	Vanguard International Value Fund	Global Stock Index
VTSMX	Vanguard Total Stock Markets Index INV	Global Stock Index
VUSTX	Vanguard Long-Term Treasury INV	US Bonds - Gvmnt
VWEHX	Vanguard Hi-Yield Corporate INV	US Bonds - Corp HY
VWESX	Vanguard Long-Term INV Growth Income INV	US Bonds - Corp Invst
VWIGX	Vanguard International Growth INV	Others
VWINX	Vanguard Wellesley Income INV	US Bonds - Gvmnt
VWO	Vanguard FTSE Emerging Markets Index Fund ETF Shares	Global Stock Index
VWUSX	Vanguard US Growth INV	Stock Index (US)
VXF	Vanguard Extended Market Index Fund ETF Shares	Global Stock Index
WDG-X	MSCI Germany iShr Ix	National Stock Index
WPB-X	MSCI Canada iShr Ix	National Stock Index
XLB	StateSt ETF Materials Select Sector SPDR	US Sector Stock Index
XLE	StateSt ETF Energy Select Sector SPDR Fd	US Sector Stock Index
XLF	StateSt ETF Financial Select Sector SPDR	US Sector Stock Index

XLI	StateSt ETF Industrial Sel Sector SPDR	US Sector Stock Index
XLK	StateSt ETF Tech Select Sector SPDR	US Sector Stock Index
XLP	StateSt ETF Consumer Staples SelSctrSPDR	US Sector Stock Index
XLU	StateSt ETF Utilities Select Sector SPDR	US Sector Stock Index
XLV	StateSt ETF Health Care Sel Sector SPDR	US Sector Stock Index
XLY	StateSt ETF Consumer DiscretnrySlSctSPDR	US Sector Stock Index
XLC	StateSt ETF Communication Service SlSctSPDR	US Sector Stock Index
XLRE	StateSt ETF Real Estate SlSct-SPDR	US Sector Stock Index
VOX	Vanguard Communication Services Index Fund ETF Shares	US Sector Stock Index
IYR	iShares U.S. Real Estate ETF	US Sector Stock Index
XOI-I	AMEX Oil Ix	Commodities
ZAR-	South Africa/ US Exchange Rate Ix	Forex
NIKI	Tokyo Nikkei Ix	National Stock Index
BBM-1	BBG Muni Bond 1-10yr Blend Idx	US Bonds - Corp Invst
BBM-7	BBG Muni Bond 15yr Idx	US Bonds - Gvmnt
BBM-B	BBG Barclay Agg Bond- Lng Govt/Crd TR Ix	US Bonds - Corp Invst
BBM-F	US Treasury 5-Year Bd Yield Ix	US Bonds - Gvmnt
BRL-	Brazil / US Foreign Exchange Rate Ix	Forex
DIA	StateSt ETF SPDR Dow Jones IndustrilAvrg	Stock Index (US)
DJ-CO	DJ UBS Crude Oil Ix	Commodities
EWQ	iShares ETF MSCI France	National Stock Index
EWU	iShares ETF MSCI United Kingdom	National Stock Index
IC-3B	BofAML US Corporate A Semi-Annual Yield	US Bonds - Corp Invst
IEO-X	DJ US Oil & Gas iShr Ix	Commodities

M-CNA	MSCI China A DivAdj Ix	National Stock Index
M-DEA	MSCI EAFE DivAdj Idx	Global Stock Index
M-DEU	MSCI Dev Mkts EU DivAdj Idx	Regional Stock Index
M-DG7	MSCI Dev Mkts G7 Index DivAdj Idx	Global Stock Index
M-EM	MSCI Emerging Markets DivAdj Idx	Global Stock Index
M-EMA	MSCI Emerging Markets Asia DivAdj Idx	Regional Stock Index
M-EME	MSCI Emergin Martkets EMEA DivAdj Idx	Regional Stock Index
M-EMU	MSCI Emerging Markets Europe DivAdj Idx	Regional Stock Index
M1EY-	BofAML US Corporate AA Semi-Annual Yield	US Bonds - Corp Invst
M2EY-	BofAML US Corporate AAA Semi-Annual Yiel	US Bonds - Corp Invst
M3OA-	BofAML US Corporate BBB Semi-Annual Yiel	US Bonds - Corp Invst
M4EY-	BofAML US Corporate 1-3 Year Semi-Annual	US Bonds - Corp Invst
M5OA-	BofAML US Corporate 3-5 Year Semi-Annual	US Bonds - Corp Invst
M6OA-	BofAML US Corporate 5-7 Year Semi-Annual	US Bonds - Corp Invst
M7EY-	BofAML US Corporate 7-10 Year Semi-Annua	US Bonds - Corp Invst
M8EY-	BofAML US Corporate 10-15 Year Semi-Annu	US Bonds - Corp Invst
MBOA-	BofAML US Corporate 15 Year Semi-Annual	US Bonds - Corp Invst
QQQ	Nasdaq 100 ETF	Stock Index (US)
SP-GB	S&P Global BMI Idx DivAdj	Global Stock Index
SP-GL	S&P Global 1200 Idx DivAdj	Global Stock Index
SP-HB	S&P 500 High Beta Idx DivAdj	Global Stock Index
SP-IO	S&P Global 100 Idx DivAdj	Global Stock Index
SP-L4	S&P Latin America 40 Idx DivAdj	Regional Stock Index
SPY	StateSt ETF SPDR S&P 500	Stock Index (US)
ST-AG	Silver Spot	Commodities

ST-AU	Gold Spot	Commodities
ST-BC	Brent Crude Spot	Commodities
ST-CA	Cocoa Spot	Commodities
ST-CF	Coffee Bushel Spot	Commodities
ST-CO	Crude Oil Spot	Commodities
ST-CT	Cotton Bushel Spot	Commodities
ST-CU	Copper Spot	Commodities
ST-HO	Heating Oil Spot	Commodities
ST-NG	Natural Gas Spot	Commodities
ST-PD	Palladium Spot	Commodities
ST-PL	Platinum Spot	Commodities
WTI-B	Blmbrg WTI Crude Oil Sub Ix Total Return	Commodities
VIPSX	Vanguard Inflation-Protected Securities Fund Investor Shares	US Bonds - Gvmnt

Table C.11: List of US traded funds used to create the **finance** dataset. The class columns correspond to the type of securities and the source columns specify where the TS was collected. See Tab. C.12 for a brief description of the asset classes

TS type	Description
US Stock Index	Index of US stocks, such as the S&P500
US Stock	A fund (ETF or mutual funds) made up primarily of US stocks
US Sector Stock Index	US stock industry sector index
Regional Stock Index	Global region stock index, such as Europe
National Stock Index	Country stock index
Global Stock Index	Global / world stock index
US Bonds - Gvmnt	US treasury funds
US Bonds - Corp Invst	US corporate bond funds, investment grade
US Bonds - Corp HY	US bond funds, high yield
Country Funds	Country index fund
Forex	Foreign Exchange
Commodities	Commodities tracking fund
Real Estate	Real estate fund
Other	Other fund or index

Table C.12: Brief description of the typea of TS used in the **Finance** dataset.