N-BEATS: Neural basis expansion analysis for interpretable time series forecasting
ML for Time Series Project report

Sami Amrani sami-faycal.amrani@polytechnique.org
Hamdi Bel Hadj Hassine hamdi.belhadjhassine@ensae.fr

March 30, 2022

1 Introduction and contributions

Time Series forecasting is a hard problem with tremendous applications. Until recently, it is a task that Deep Learning hasn't yet managed to tackle efficiently, as statistical tools outperform most Deep Learning models.

It is in this context that Y. Bengio and his colleagues (Oreshkin et al. [1]) introduced N-BEATS in 2019. It is a deep neural architecture based on backward and forward residual links and a stack of fully-connected layers. According to its authors, it has a number of desirable properties, being interpretable and applicable without modification to a wide array of domains.

In their paper, Oreshkin et al. test N-BEATS on the M4 dataset[2], the M3 dataset[3], and the TOURISM dataset[4] and display state-of-the-art performance on all datasets. We reproduce their results on the M3 data set, and compare different configurations to better evaluate the model's performance and find the best-performing configurations.

Sami worked mainly on the first part of the project: analysing the model's performance on the M3 dataset, and Hamdi worked on creating a new Python class that enables us to configure the model and make new experiments (second half of the notebook). We used the authors' implementation code as a start and built our notebook upon it, with about 90% of the notebook code written by us. We started by reproducing the paper's experiments on the M3 dataset, then we provided a finer analysis of the performance w.r.t. the parameters, and finally tested the model on new series that we generated. This allowed us to provide practical results about the performance of the model, its sensitivity to the parameters, and how to improve it.

2 Method

The model is composed of different stacks, which are composed of different blocks.

The basic block has one input x_l and two outputs $\hat{x}_l$ and $\hat{y}_l$. It consists of two fully connected networks that output coefficients θ_b and θ_f, and two linear projection layers that output a linear combination of basis functions $\hat{y}_l = \sum_{i=1}^{dim(\theta_l^f)} \theta_{l,i}^f v_i^f$ and $\hat{x}_l = \sum_{i=1}^{dim(\theta_l^b)} \theta_{l,i}^b v_i^b$. $\hat{y}_l$ is the block forecast, and $\hat{x}_l$ is a reconstruction of the input signal, called the "backcast". The role of the basis functions is to constrain the structure of the output. The authors propose two architectures of the model: The **interpretable architecture**, which we will use in this project, consists of a trend stack and a seasonality stack. It restricts the basis functions to be respectively low-degree polynomials and Fourier series (i.e. sinusoids), which are used to account for trend and seasonality.

Another variant is to "learn" the basis functions, and we can set $\hat{y}_l = V_l^f \theta_l^f + b_l^f$ and $\hat{x}_l = V_l^b \theta_l^b + b_l^b$, with V_l^f and V_l^b projection matrices to be learned. This variant is called the **generic architecture**.

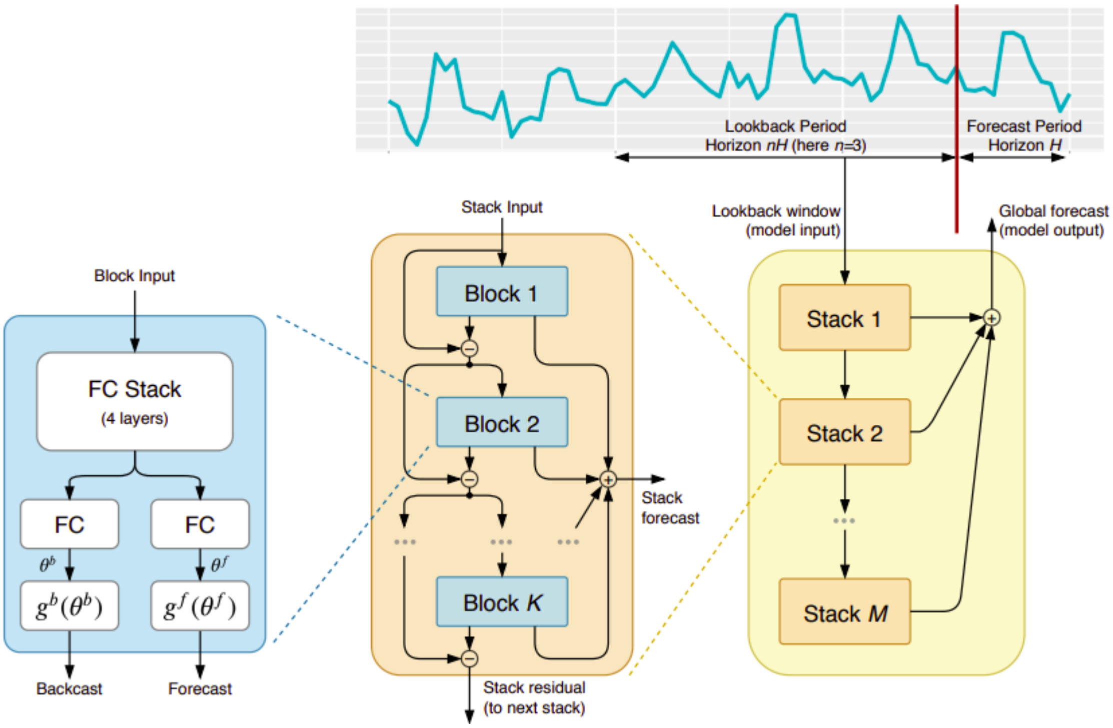


Figure 1: N-BEATS architecture from [1]

Each block has two residual branches, one running over backcast prediction of each layer and the other one is running over the forecast branch of each layer, such as $x_l = x_{l-1} - \hat{x}_{l-1}$ and $y = \sum_l \hat{y}_l$. Each block removes the portion of the signal $\hat{x}_{l-1}$ that it can approximate well, making the forecast job of the downstream blocks easier.

The model has a defined forecast horizon H and lookback period nH, with $n \in \mathbb{N}^*$ (that we'll just denote as n) representing the number of points the model takes as input. To regularize the forecasts, different models are trained using different loss functions and lookback periods, then their forecasts are ensembled using median aggregation. The three central parameters of the model are therefore the training loss, the lookback, and the repeats (i.e. number of models trained for each lookback and loss). The authors use 3 losses (MASE, sMAPE and MASE), 6 lookbacks (2 to 7) and 10 repeats, amounting to an ensemble of 180 models.

3 Data

We ran our experiments on the M3 dataset. It consists in 3003 time series selected to be representative of frequently encountered series in business, financial and economic forecasting. They are distributed as presented in 1.

This data set is often used as a benchmark for Time Series prediction. It was used in the N-BEATS paper[1], and we try here to analyse the model's performance on this dataset. Table 3 of the paper already presents a statistical summary of the dataset[1]. We note here that it is comprised of 4 subsets: monthly, quarterly, yearly and "other" series. It is also split in training and test sets with different forecasting horizons varying from 6 datapoints for yearly series to 18 datapoints for monthly series. The length of the training time series varies between 20 and 144.

We will use three precision metrics, the main one being sMAPE which was officially used in the M3 competition. For a time series y, a prediction horizon H, and a prediction $\hat{y}$, they are defined as follows [5]:

- sMAPE = $\frac{200}{H}\sum_{i=1}^{H}\frac{|y_{T+i}-\hat{y}_{T+i}|}{|y_{T+i}|+|\hat{y}_{T+i}|}$
- MAPE = $\frac{100}{H}\sum_{i=1}^{H}\frac{|y_{T+i}-\hat{y}_{T+i}|}{|y_{T+i}|}$
- MdAPE = median$(\frac{|y_{t+1}-\hat{y}_{t+1}|}{|y_{t+1}|},\ldots,\frac{|y_{t+H}-\hat{y}_{t+H}|}{|y_{t+H}|})$

4 Results

We conduct 8 experiments aiming to analyse how the performance of N-BEATS depends on its parameters and which parameters work best. The first 5 experiments are based on models trained on the whole M3 dataset using the parameters presented in the Method section (3 losses, 6 lookbacks and 10 repeats, thus 180 models). First thing we note is that the training time is rather long but still reasonable, at 8.5 minutes per model on an entry-level GPU (RTX 3060). We also note that using the 180-model ensemble we obtain an average sMAPE of 12.51 on the M3 dataset, therefore confirming the reproducibility of the paper's results (12.43 for the interpretable model).

4.1 Impact of the number of repeats on the model's performance

In Fig. 2 we plot the sMAPE decrease per number of repeats (1 repeat is an ensemble of 18 models, 10 repeats is 180 models). We notice that using bigger ensembles improves the forecasting performance and makes the model's performance more consistent, but the difference in performance is small. Using an ensemble of 180 models decreases the error rate by less than 1% compared to an ensemble of 18 models. We also note that the performance gain depends on the model's and the series' parameters; in particular models which were trained on shorter series (Yearly and Other) profited less from increasing the ensemble size. We conclude from this experiment that repeats are actually not needed and training 18 models is enough to get similar performance.

4.2 Impact of the lookback period

When using an ensemble of 30 models with all repeats, we see that lower lookbacks yield the best performance (Fig. 3). We also obtain very similar results using ensembles of 3 models (1 repeat). We also examine which combination of lookbacks is the best. As shown on figure 4, using lookbacks (2 to 5) yields the best performance, therefore an ensemble of 12 models is enough to achieve good performance using N-BEATS.

4.3 Impact of the loss function

We compare on figure 5 the performance of N-BEATS trained using each of the three error metrics MAPE, sMAPE and MASE. We observe that:

- As expected, optimizing over a loss yields a model that tends to achieve lower errors with respect to the given loss.
- Median percentage errors (MdAPE) are much lower than mean percentage errors (MAPE), meaning that errors' distribution has a thick tail and a minority of hard-to-forecast target values account for most of the MAPE error.

In figure 6 we assume that we want to minimize the sMAPE loss, and we compare models using the sMAPE loss along with different combinations of the other losses. For a fairer comparison we adjust the number of repeats so we always use ensembles of 60 models. In the figure we denote MAPE as P, SMAPE as SP and MASE as S.

We conclude that for fixed ensemble size, the best results are obtained when using either only the sMAPE loss or mixing models with sMAPE and MAPE losses. However, the difference between them is very small, and using all losses would also improve the model's performance when evaluated with other losses, so we see no clear winner here.

4.4 Comparison with other forecasting models

We compare the performance of N-BEATS with other forecasting models : ARIMA, exponential smoothing, and a naive model forecasting the last value. The results are plotted on figure 7 for different error metrics and seasonalities.

We see that on average for the whole M3 dataset, M-BEATS always outperforms the other methods. We note also that although all of the 3 other methods managed to outperform N-BEATS for (exactly) one seasonal pattern / error metric combination, they all also have errors 25% to 100% higher than N-BEATS, while N-BEATS manages to achieve consistent performance and is only slightly and rarely outperformed.

Overall, we can conclude that N-BEATS displays outstanding forecasting performance and manages to outperform popular models like ARIMA and Exponential Smoothing. This confirms the paper's results where N-BEATS also outperforms all the other tested methods.

4.5 Impact of the forecast horizon

In figure 8 we use predictions from the 180-model ensemble (for more significant results) and evaluate its performance for single-point forecasts of different horizons.

The forecasting errors tend to grow with the horizon in a linear fashion, where the slope is higher for series with higher seasonality, as we can expect (predicting a variable in 2 years horizon is harder than 2 months). Surprisingly though, monthly series don't seem to be harder to predict for higher horizons.

For N-BEATS, ARIMA and Naive forecasts, the error grows at a similar rate with respect to the forecasting horizon, unlike exponential smoothing whose errors explode for higher horizons.

4.6 Impact of the training iterations

We wanted to evaluate whether we need to train N-BEATS for many iterations to get good results or not. However the official implementation of the paper is just designed to reproduce the paper's experiments (in console mode) and doesn't allow us to easily control the parameters of the model and the training and prediction process, neither specify other training data. Therefore we built a new N-BEATS class with a sklearn-like API (define-fit-predict) to make our experiments easier. This implementation can also be useful for other practitioners who want to use N-BEATS on their own series, and provides better performance (in our tests) than the Darts implementation, and an easier interface than the official implementation.

For N-Beats, a training iteration is a prediction and backpropagation for 1 time series. We denote by epoch a number of iterations equal to the number of series in the dataset (i.e. one pass over the data). We trained 18 models for 10 epochs on the M3 monthly dataset and after each epoch we saved the models' forecasts, then we plotted the resulting error against the number of training epochs (Fig. 9). Surprisingly, we see that the lowest error is achieved after the first epoch. We conclude that one pass over the data is enough to train the model.

4.7 Impact of the size of the training dataset

An interesting question to answer is whether N-BEATS needs to be trained on a large dataset to achieve good results. To this end, we select a random subset of 100 series of M3 monthly dataset, train an 18-model ensemble for 1 epoch on it, test the ensemble on the same subset, and compare the error on that same subset with that of the previous ensemble trained on the full 1428-series M3 monthly set (for 1 epoch). We obtain a sMAPE of 14.49 for the model trained on the large set, and 15.79 for the model trained on the subset, indicating that N-BEATS works better when trained on a large dataset. We also tried increasing the number of iterations to 1428 when training on the subset, but the error increased.

4.8 Qualitative evaluation of the forecasts on synthetic series

In this experiment, we generate a selection of time series (constant, noisy, linear, noisy linear, sine with high or low frequency and with or without outliers, square and triangular signals, exponential), and we visually assess N-BEATS' performance on them. As shown in the notebook, we observe that N-BEATS only works when the series to predict have the same range of values as the training dataset. After scaling the generated series and further training the previous models from experiment 6 on them, we visualize the forecasts in Fig. 10. Although not perfect, the obtained forecasts are rather satisfying. We note that for the constant series, the model seems to confuse it with the exponential series and makes an exponential forecast, but the error is only 1% which is fine. For the exponential series, the model extrapolates decently for a short horizon but then doesn't manage to predict values much higher than its training range. In such cases, the training set needs to be adapted to such series.

Overall, we can conclude that N-BEATS can efficiently capture and learn the patterns in the data, while also showing robustness towards outliers and noise. Moreover, it doesn't need as many models as used in the paper to perform well. It also proves that neural networks can outperform classical methods in time series forecasting, and lays the way for more research in this direction.

References

[1] B. N. Oreshkin, D. Carpov, N. Chapados, and Y. Bengio, "N-BEATS: neural basis expansion analysis for interpretable time series forecasting," *CoRR*, vol. abs/1905.10437, 2019. [Online]. Available: http://arxiv.org/abs/1905.10437

[2] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "The m4 competition: 100,000 time series and 61 forecasting methods," *International Journal of Forecasting*, vol. 36, no. 1, pp. 54–74, 2020, m4 Competition. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0169207019301128

[3] S. Makridakis and M. Hibon, "The m3-competition: results, conclusions and implications," *International Journal of Forecasting*, vol. 16, no. 4, pp. 451–476, 00 2000. [Online]. Available: http://www.sciencedirect.com/science/article/B6V92-41J6944-3/1/74f19d7fbfdec216ba87bc525091f6e4

[4] G. Athanasopoulos, R. Hyndman, H. Song, and D. C. Wu, "The tourism forecasting competition," *International Journal of Forecasting*, vol. 27, no. 3, pp. 822–844, 2011. [Online]. Available: https://EconPapers.repec.org/RePEc:eee:intfor:v:27:y::i:3:p:822-844

[5] R. J. Hyndman and A. B. Koehler, "Another look at measures of forecast accuracy," *International Journal of Forecasting*, vol. 22, no. 4, pp. 679–688, 2006. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0169207006000239

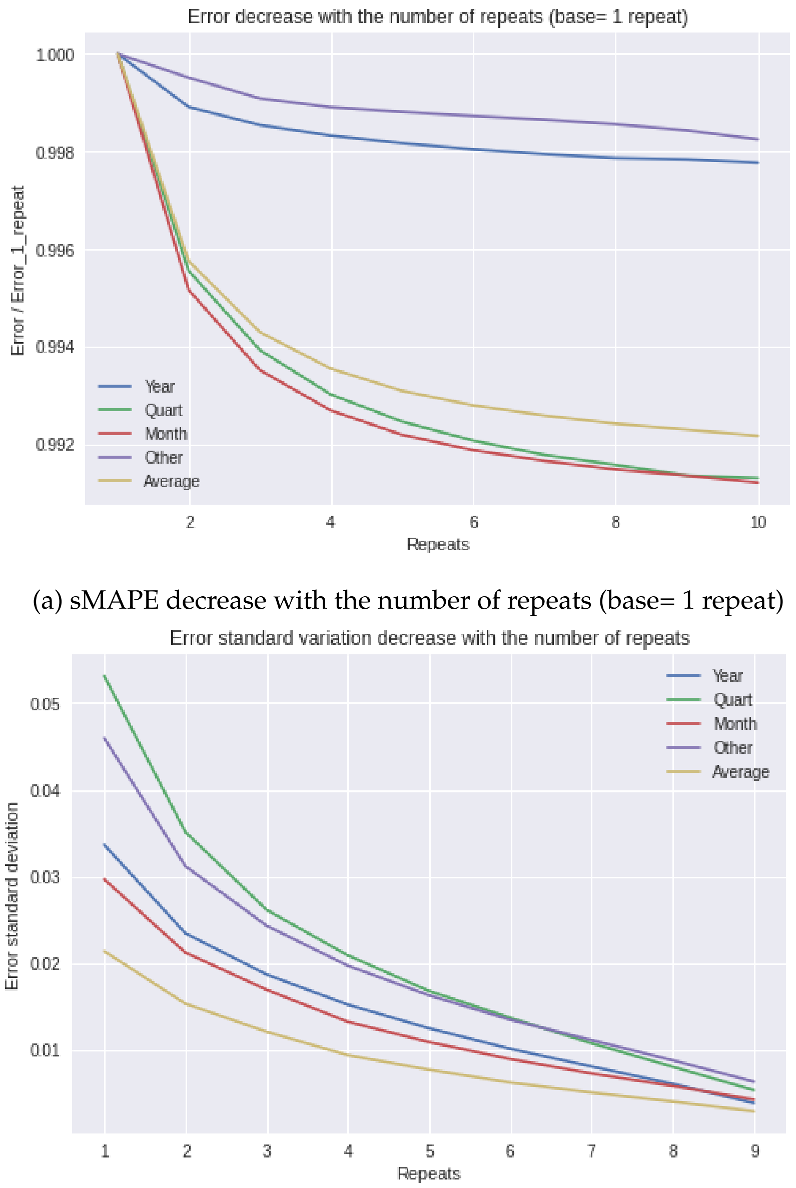
Appendices

	Type of time series data						
Interval	**Micro**	**Industry**	**Macro**	**Finance**	**Demog**	**Other**	**Total**
Yearly	146	102	83	58	245	11	645
Quarterly	204	83	336	76	57	0	756
Monthly	474	334	312	145	111	52	1428
Other	4	0	0	29	0	141	174
Total	828	519	731	308	413	204	3003

Table 1: M3 dataset distribution

	Year	Quart	Month	Other	Average
N-BEATS Interpretable (ours)	15.79	8.98	13.30	4.24	12.51
N-BEATS Interpretable (article)	15.84	9.03	13.15	4.30	12.43

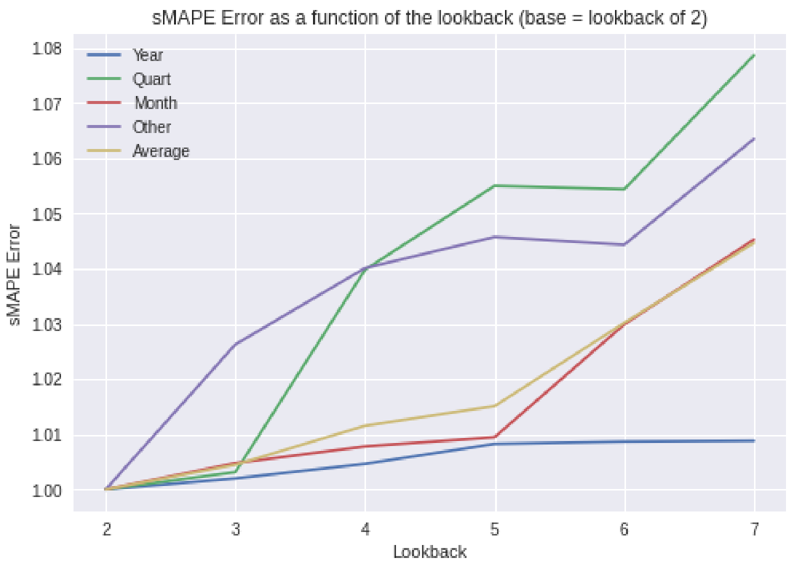
Table 2: Our results on the M3 dataset vs. those of the authors in [1]



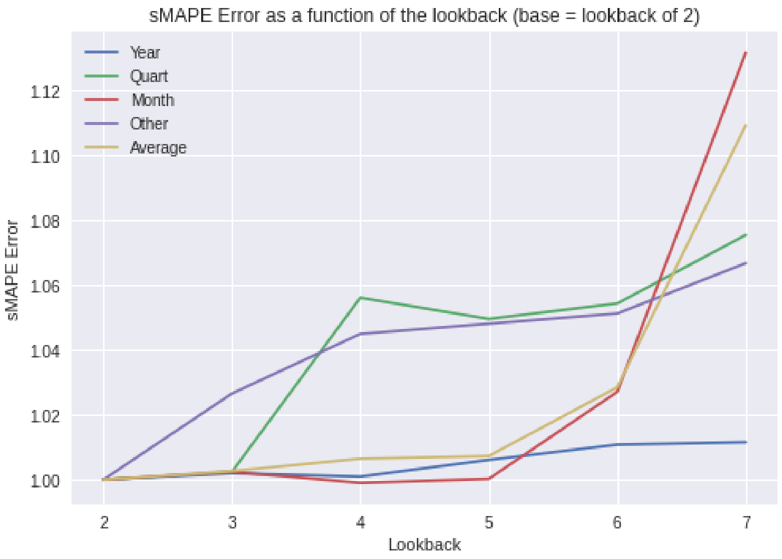
(a) sMAPE decrease with the number of repeats (base= 1 repeat)

(b) sMAPE standard deviation decrease with the number of repeats

Figure 2: sMAPE and its standard deviation for different numbers of repeats



(a) sMAPE as a function of the lookback (base = lookback of 2) for an ensemble of 30 models using all repeats



(b) sMAPE as a function of the lookback (base = lookback of 2) on average for 3 models using 1 repeat

Figure 3: sMAPE as a function of the lookback

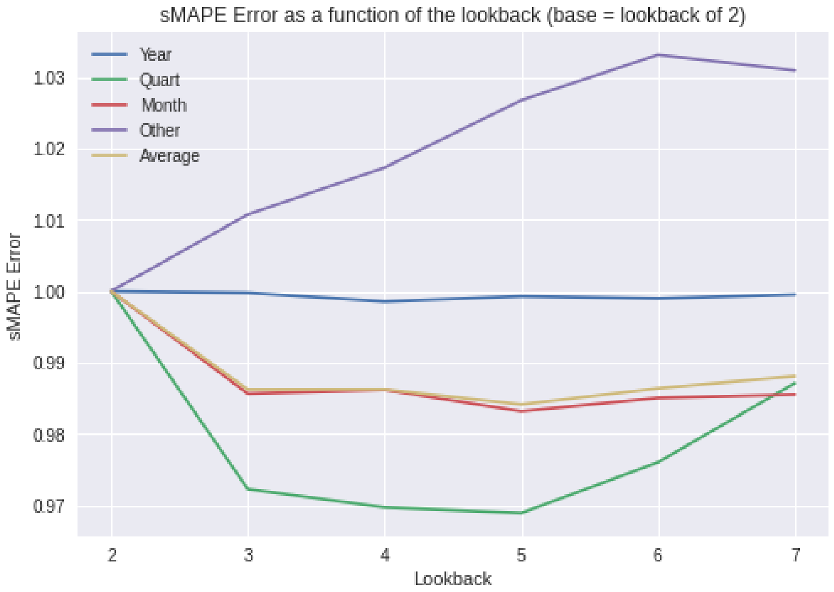
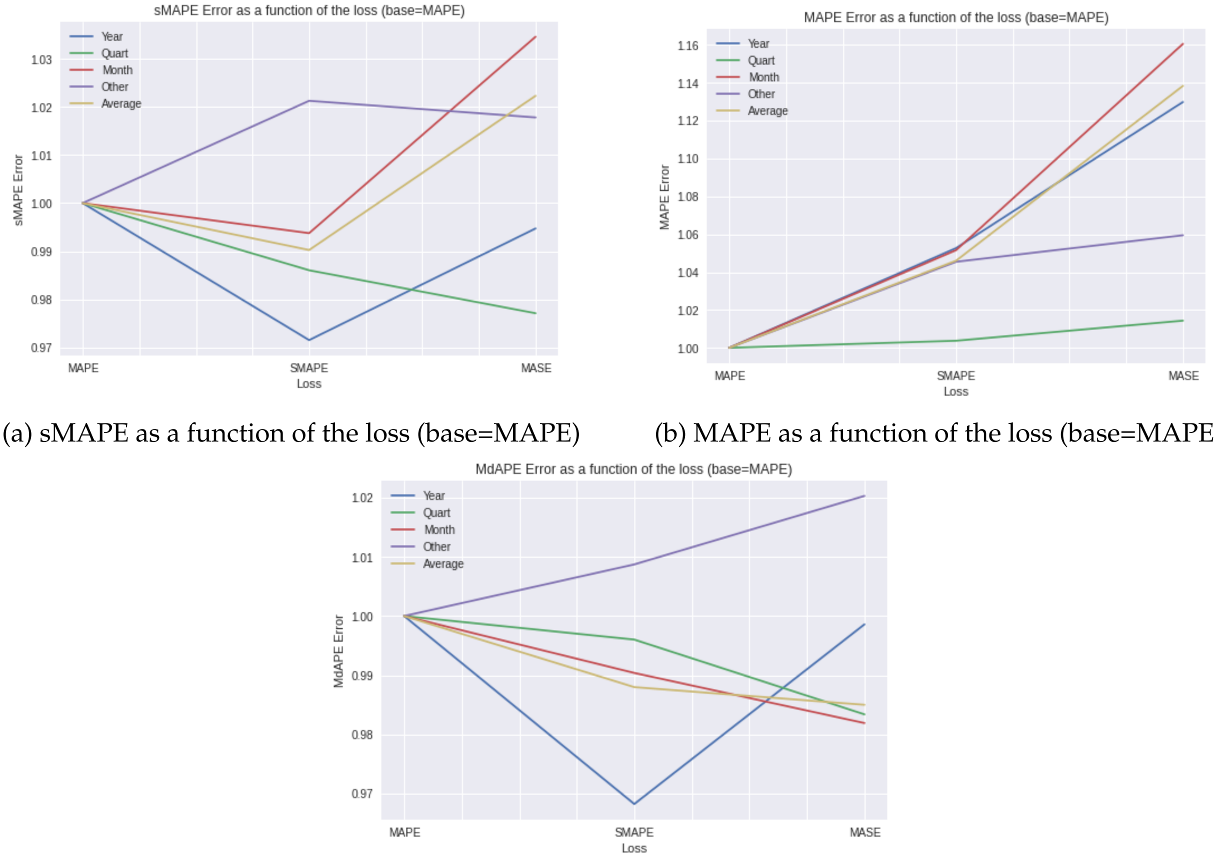


Figure 4: sMAPE as a function of the lookbacks combination (2 to k) (base = lookback of 2)



(a) sMAPE as a function of the loss (base=MAPE)

(b) MAPE as a function of the loss (base=MAPE)

(c) MdAPE as a function of the loss (base=MAPE))

Figure 5: Different error metrics as a function of the loss

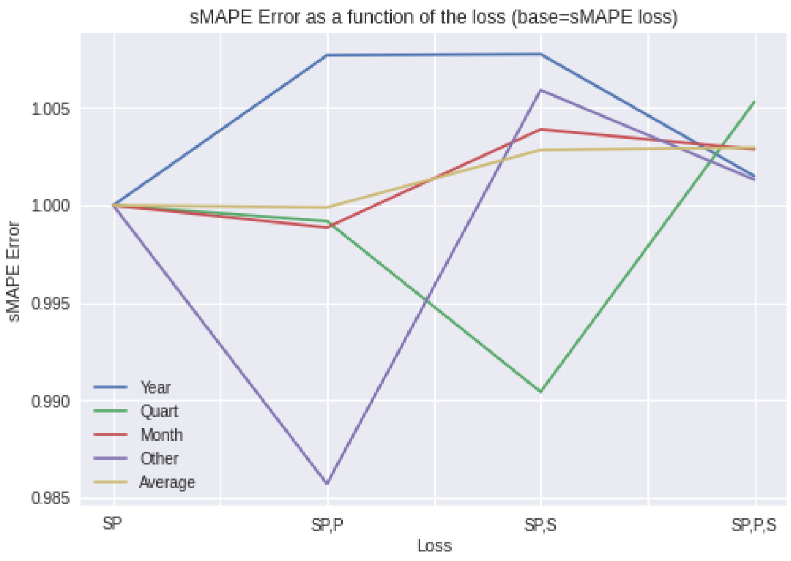
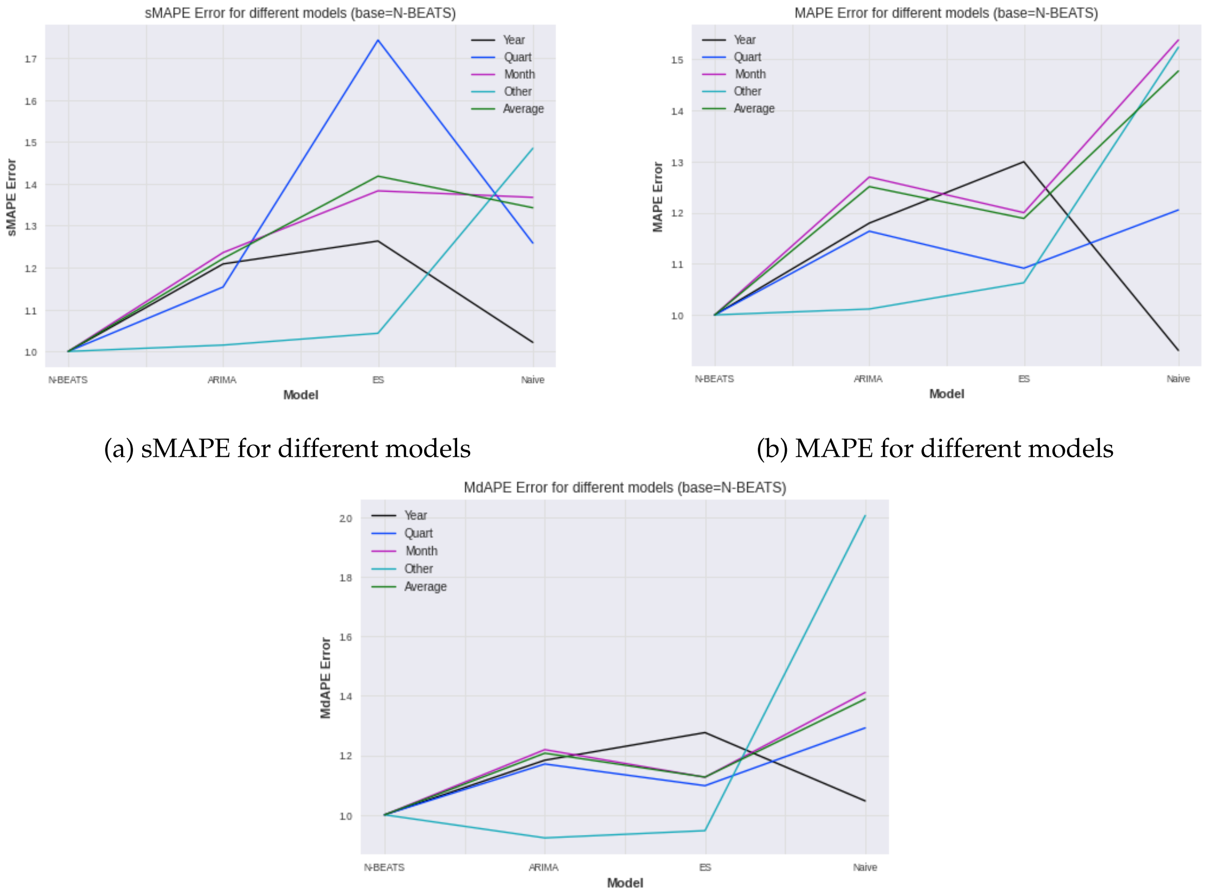


Figure 6: sMAPE as a function of the loss combination (base=sMAPE loss)

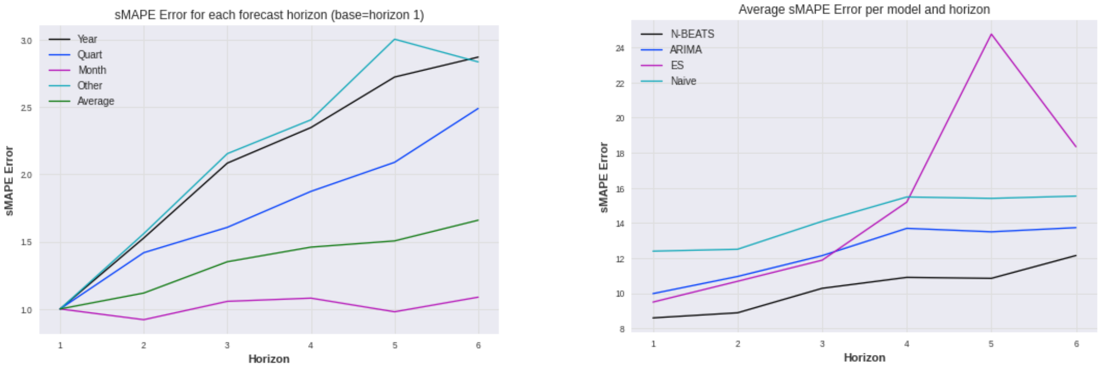


(a) sMAPE for different models

(b) MAPE for different models

(c) MdAPE for different models

Figure 7: Different error metrics for different models' forecasts (base N-BEATS)



(a) sMAPE per horizon for N-BEATS (base 1)

(b) Average sMAPE per horizon for different models

Figure 8: sMAPE variation for different horizons

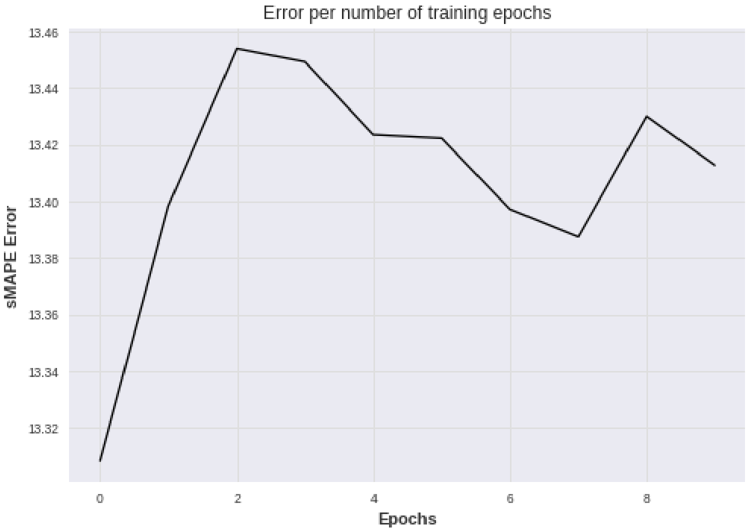


Figure 9: sMAPE as a function of the training epochs

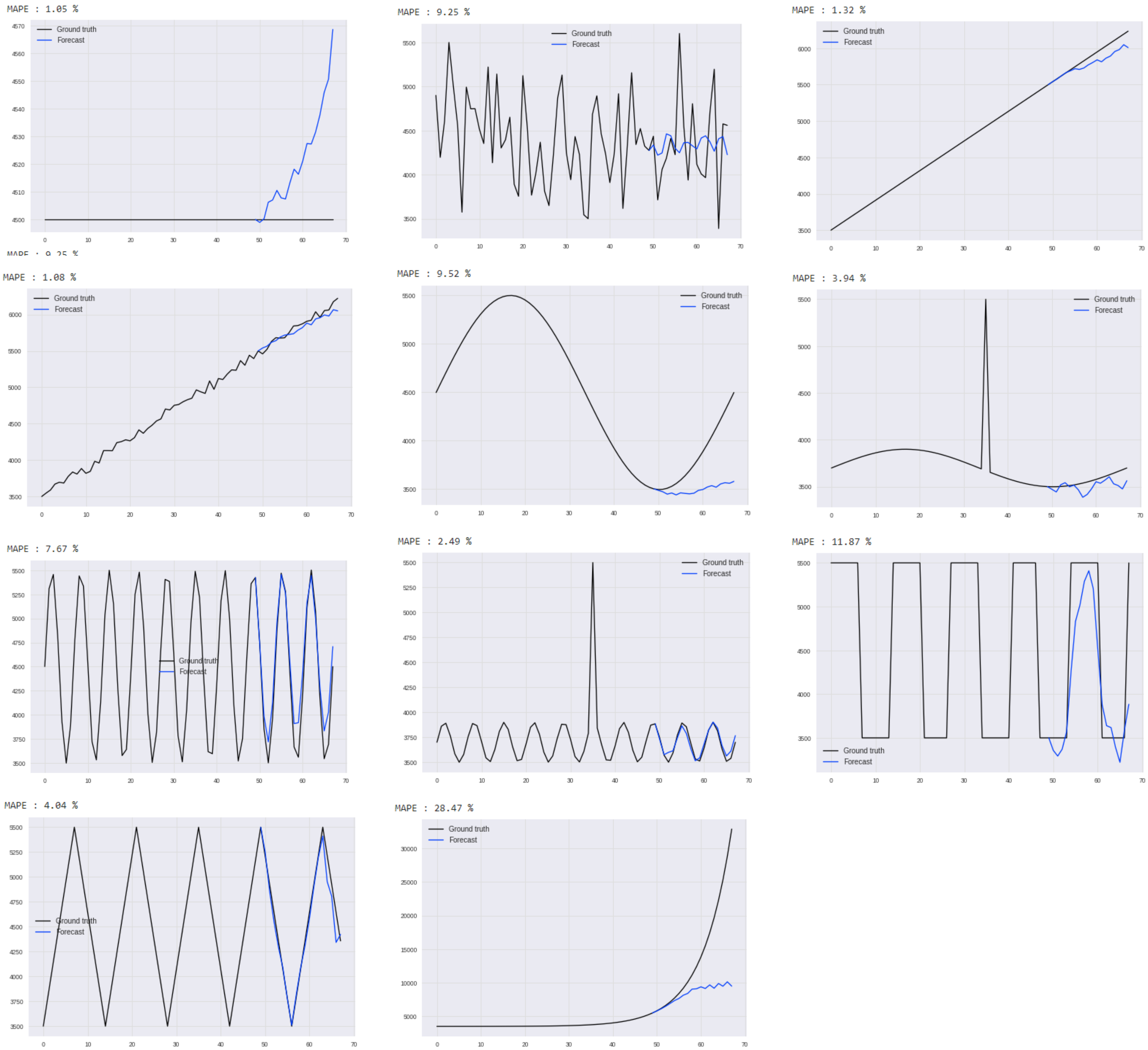


Figure 10: N-BEATS forecasts on generated time series