HERMES: Hybrid Error-corrector Model with inclusion of External Signals for nonstationary fashion time series

Etienne DAVID[a,b], Jean BELLOT[b], Sylvain LE CORFF[a]

[a] *Samovar, Télécom SudParis, Département CITI, Institut Polytechnique de Paris, France.*
[b] *Heuritech, 71 Rue Réaumur, 75002 Paris, France.*

Abstract

Developing models and algorithms to draw causal inference for time series is a long standing statistical problem. It is crucial for many applications, in particular for fashion or retail industries, to make optimal inventory decisions and avoid massive wastes. By tracking thousands of fashion trends on social media with state-of-the-art computer vision approaches, we propose a new model for fashion time series forecasting. Our contribution is twofold. We first provide publicly[1] the first fashion dataset gathering 10000 weekly fashion time series. As influence dynamics are the key of emerging trend detection, we associate with each time series an external weak signal representing behaviors of influencers. Secondly, to leverage such a complex and rich dataset, we propose a new hybrid forecasting model. Our approach combines per-time-series parametric models with seasonal components and a global recurrent neural network to include sporadic external signals. This hybrid model provides state-of-the-art results on the proposed fashion dataset, on the weekly time series of the M4 competition Makridakis et al. (2018), and illustrates the benefit of the contribution of external weak signals.

Keywords: Hybrid models, Recurrent neural networks, Time series.

[1] `http://files.heuritech.com/raw_files/f1_fashion_dataset.tar.xz`

1. Introduction

Multivariate time series forecasting is a widespread statistical problem with many applications, see for instance Särkkä (2013); Douc et al. (2014); Zucchini et al. (2017) and the numerous references therein. Parametric generative models provide explainable predictions with statistical guarantees based on a precise modeling of the predictive distributions of new data based on a record of past observations. Calibrating these models, for instance using maximum likelihood inference, often requires a fair amount of tuning to design a time series-specific model to provide accurate forecasts and sharp confidence intervals. Depending on the use case, statistical properties of the signal and the available data, many families of models have been proposed for time series. The exponential smoothing model Brown & Meyer (1961), the Trigonometric Box-Cox transform, ARMA errors, Trend, and Seasonal components model (TBATS) Livera et al. (2011), or the ARIMA with the Box-Jenkins approach Box et al. (2015) are for instance very popular parametric generative models. Hidden Markov models (HMM) are also widespread and presuppose that available observations are defined using missing data describing the dynamical system. This hidden state is assumed to be a Markov chain such that at each time step the received observation is a random function of the corresponding latent data. Although hidden states are modeled as a Markov chain, the observations arising therefrom have a complex statistical structure. In various applications where signals exhibit non-stationarities such as trends and seasonality, classical HMM are not adapted. However, Touron (2017) recently proposed seasonal HMM, assuming that transition probabilities between the states, as well as the emission distributions, are not constant in time but evolve in a periodic manner. Strong consistency results were established in Touron (2019) and Expectation Maximization based numerical experiments were proposed. Although these works provide promising results, HMM are computationally expensive to train and are not yet well studied for seasonal sequences with thousands of components.

In many fields, single or few time series have become thousands of sequences

with various statistical properties. In this new context, classical time series specific statistical models show limitations when dealing with numerous heterogeneous data. Recurrent neural networks and recent sequence to sequence deep learning architectures offer very appealing numerical alternatives thanks to their capability of leveraging any kind of heterogeneous multivariate data, see for instance Hochreiter & Schmidhuber (1997); Vaswani et al. (2017); Siami-Namini et al. (2018); Li et al. (2019); Lim et al. (2019); Salinas et al. (2020). The DeepAR model proposed in Salinas et al. (2020) provides a global model from many time series based on a multi-layer recurrent neural network with LSTM cells. More recently, applications using the Transformer model have been proposed Li et al. (2019). The Temporal Fusion Transformers (TFT) approach is a direct alternative to the DeepAR model Lim et al. (2019). Unfortunately, all these solutions suffer from two main weaknesses. Firstly, many of them are black-boxes as the final forecast usually does not come with a statistical guarantee although a few recent works focused on measuring uncertainty in recurrent neural networks, see Martin et al. (2021). Secondly, without a fine preprocessing and well chosen hyper-parameters, these methods may lead to poor results and be outperformed by traditional statistical models, see Makridakis et al. (2018).

In this paper, we consider a new time series forecasting application referred to as *fashion trends prediction*. Based on a cutting-edge image recognition technology, we built the first fashion dataset containing 10000 weekly sequences of fashion trends on social media from 01-01-2015 to 01-01-2019. This dataset has very appealing properties: all time series have the same length, no missing value and there is no sparse time series even for niche trends. The originality of our dataset comes from the fact that additional external weak signals can also be introduced. With our fashion expertise, we detected several groups of highly influential fashion users. Analyzing their specific behaviours on social media, we associate with each time series an external weak signal representing the same fashion trends on a sub-category of users. They are called weak signals because they are often alerts or events that are too sparse, or too incomplete to allow on their own an accurate estimation of their impact on the prediction of the

target signal. With this totally new application, we aim at designing a model able to deal with such a large dataset, leveraging complex external weak signals and finally providing the most accurate forecasts.

Recurrent neural networks are appealing to tackle our forecasting problem due to their capability of leveraging external data. Recently, hybrid models combining deep neural network (DNN) architectures with widespread statistical models to deal with seasonality and trends have been proposed, see for instance Zhang (2003); Jianwei et al. (2019); Bandara et al. (2020). The approach providing the most striking results was proposed in Smyl (2020) in the context of the M4 forecasting competition Makridakis et al. (2020). Given a large dataset, a per-time-series multiplicative exponential smoothing model was introduced to estimate simple but fundamental components for each time series and compute a first prediction. Then a global recurrent neural network was trained on the entire dataset to correct errors of the previous exponential smoothing models.

Following this work, we present in this paper HERMES, a new hybrid recurrent model for time series forecasting with inclusion of external signals. This new architecture is decomposed into two parts: local predictors and a global corrector. First, a per-time-series parametric statistical model is trained on each sequence. Then, a global recurrent neural network is trained to evaluate and correct the forecast weaknesses of the first collection of models. The external weak signals reveal the real potential of the hybrid approach: a global neural network, able to leverage large amounts of heterogeneous data, deal with any kind of external weak signals, learn context and finally correct weaknesses and errors of parametric models.

The paper is organized as follows. Section 2 introduces the proposed hybrid model. Then, the new fashion dataset provided with this article is presented in Section 3. Section 4 describes the HERMES results and comparisons with several benchmarks. Finally, a general conclusion and some research perspectives are given in Section 5.

2. Hybrid model with external signals

We introduce a new hybrid approach for time series forecasting composed of two parts: a collection of per-time-series parametric models, and a global error-corrector neural network train on all time series. Per-time-series parametric models are used to learn local behaviours, to normalize sequences by removing trends and seasonality, and to compute a first forecast. Then, gathering information of the first predictions and external variables, a recurrent neural network is trained to correct the predictions provided by the first collection of per-time-series models.

Consider $N \geqslant 1$ time series. For all $1 \leqslant n \leqslant N$ and $1 \leqslant t \leqslant T$, let y_t^n be the value of the n-th sequence at time t and $\mathbf{y}^n = \{y_t^n\}_{1 \leqslant t \leqslant T}$ be all the values of this sequence. The objective of this paper is to propose a model to forecast all time series in a given time frame $h \in \mathbb{N}$, i.e. we aim at sampling $\{y^n_{T+1:T+h}\}_{1 \leqslant n \leqslant N}$ based on $\{y^n_{1:T}\}_{1 \leqslant n \leqslant N}$.

2.1. Per-time-series predictors

The time-series-specific predictors compute, for each sequence, a first h-ahead prediction based on the past. For all $1 \leqslant n \leqslant N$, we note $f^n(.; \theta^n_{predictor})$ the n-th parametric model of the n-th sequence where $\theta^n_{predictor}$ are unknown parameters. Given the sequences $\{y^n_{1:T}\}_{1 \leqslant n \leqslant N}$ and the estimated parameters $\{\theta^n_{predictor}\}_{1 \leqslant n \leqslant N}$, the time-series-specific forecasts $\{\widehat{y}^{pred,n}_{T+1:T+h|T}\}_{1 \leqslant n \leqslant N}$ are, for all $n \in \{1, \dots, N\}$, for all $i \in \{1, \dots, h\}$,

$$\widehat{y}^{pred,n}_{T+i|T} = f^n(y^n_{1:T}; \theta^n_{predictor})_i \,.$$

During the M4 competition, the hybrid model of Smyl (2020) was based on a multiplicative exponential smoothing model as the time-series-specific predictor. However, on sporadic time series, this choice leads to poor results and instability. In this paper, a more general framework able to deal with any kind of per-time-series models is provided. In Section 4, two versions of our framework are proposed. The first one is based on an exponential smoothing as a reference

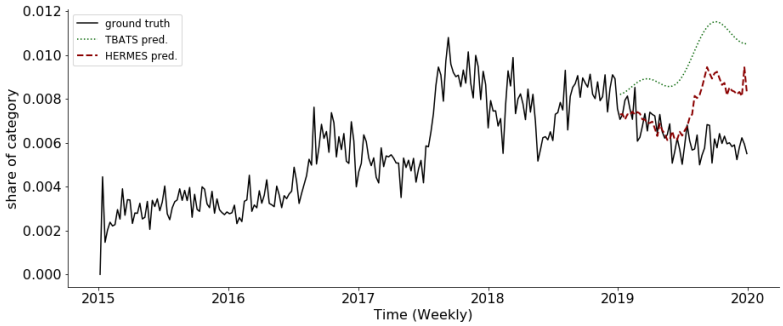

Figure 1: Hermes forecast examples. In green the prediction of the TBATS per-time-series predictors. In red the final forecast of our HERMES hybrid model. Time series representing the vertical stipes texture fashion trend for females in Brazil.

similar to the baseline Smyl (2020) and the second one uses a TBATS model Livera et al. (2011) which provides better results as this parametric model includes Fourier representations with time varying coefficients, and ARMA error correction.

In the specific application of fashion trends prediction, two main forms of standard weaknesses were detected and defined for univariate parametric forecasts. The first one is a weakness of definition, when the model is not well defined for its forecasting task. For instance, an additive model provides good predictions on time series with additive seasonality but fails at handling multiplicative seasonalities. The second form, harder to detect and correct, is the weakness of information. For non stationary time series, huge changes of behaviours are not always predictable using the past of the sequence. In somes cases, these changes depend on external variables not considered by univariate parametric models. The difficulty is that the exact influence of external variables on the main signal is mostly unknown. This motivates the introduction of a global RNN trained on all time series and able to consider and leverage external signals. First forecast example of our new hybrid model is given in Figure 1.

2.2. Error-corrector recurrent model

The second part of the model is a global RNN, trained on all the N sequences to correct the weaknesses of the first per-time-series parametric models. This task requires a thorough data pre-processing as recurrent neural networks training is highly sensitive to the scale of the data and requires well-designed inputs. Since no assumption about the scale of our time series was made, inputs require a careful normalization before being fed to the RNN.

Let $w \in \mathbb{N}$ be the window size, usually this window is proportional to the forecast horizon $w \propto h$. The RNN input is defined as the following normalized, deseasonalized and rescaled sequence $\mathbf{z}_T^n = \{z^n_{T-w+i|T}\}_{1 \leqslant i \leqslant w}$, where, for all $1 \leqslant n \leqslant N$, $1 \leqslant i \leqslant w$ and $k = i - h\lfloor i/h \rfloor$,

$$z^{n,T}_{T-w+i|T} = \frac{y^n_{T-w+i} - \widehat{y}^{pred,n}_{T+k|T}}{\bar{y}^n_T}, \quad \bar{y}^n_T = \frac{1}{w}\sum_{i=1}^{w} y^n_{T-w+i}\,.$$

Let $\text{RNN}(.;\theta_{corrector})$ be the recurrent neural network model where $\theta_{corrector}$ are unknown parameters. Given the RNN input sequences $\{\mathbf{z}_T^n\}_{1 \leqslant n \leqslant N}$ and the global RNN estimated parameters $\theta_{corrector}$, the error-corrector predictions $\{\widehat{y}^{corr,n}_{T+1:T+h|T}\}_{1 \leqslant n \leqslant N}$ are, for all $n \in \{1, \ldots, N\}$, for all $i \in \{1, \ldots, h\}$,

$$\widehat{y}^{corr,n}_{T+i|T} = \text{RNN}(\mathbf{z}_T^n; \theta_{corrector})_i \cdot \bar{y}^n_T \ .$$

Our hybrid model forecast is, for all $1 \leqslant n \leqslant N$ and all $i \in \{1, \ldots, h\}$,

$$\begin{aligned} \widehat{y}^n_{T+i|T} &= \widehat{y}^{pred,n}_{T+i|T} + \widehat{y}^{corr,n}_{T+i|T} \\ &= f^n(y^n_{1:T}; \theta^n_{predictor})_i + \text{RNN}(\mathbf{z}_T^n; \theta_{corrector})_i \cdot \bar{y}^n_T\,. \end{aligned} \tag{1}$$

2.3. Weak signal

Using well-fitted time-series-specific parametric models, the new hybrid network corrects the first form of weakness and provides very good performance on the fashion dataset, see Table 2. Then, to correct the second form of weakness, in addition to the N target time series, $K \times N$ external sequences indexed from 0 to T are considered. For all $1 \leqslant n \leqslant N$, $1 \leqslant k \leqslant K$ and $1 \leqslant t \leqslant T$, let $w_t^{n,k}$ be the value of the k-th external sequence at time t associated with the sequence

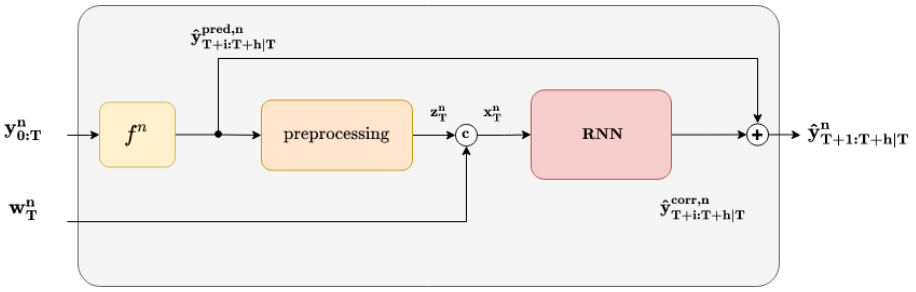

Figure 2: Architecture of the hybrid model with weak signals.

$\mathbf{y}^n$. Let $\mathbf{w}^n = \{\{w_t^{n,k}\}_{1\leqslant t\leqslant T}\}_{1\leqslant k\leqslant K}$ be all the values of the weak signals associated with the n-th sequence. In addition, let $\mathbf{w}_T^n = \{\{w_{T-w+i}^{n,k}\}_{1\leqslant i\leqslant w}\}_{1\leqslant k\leqslant K}$ be only the last w terms of the sequence. Concatenating $\mathbf{z}_T^n$ and $\mathbf{w}_T^n$, a new input for the RNN is defined:

$$\begin{aligned}\mathbf{x}_T^n &= \{x_{T-w+i|T}^n\}_{1\leqslant i\leqslant w} \\ &= \{z_{T-w+i|T}^n, w_{T-w+i}^{n,1}, ..., w_{T-w+i}^{n,K}\}_{1\leqslant i\leqslant w}\,.\end{aligned}$$

Finally, for all $1 \leqslant n \leqslant N$ and for all $i \in \{1, \ldots, h\}$ the final prediction becomes:

$$\begin{aligned}\widehat{y}_{T+i|T}^n &= \widehat{y}_{T+i|T}^{pred,n} + \widehat{y}_{T+i|T}^{corr,n} \\ &= f^n(y_{1:T}^n; \theta_{predictor}^n)_i + \text{RNN}(\mathbf{x}_T^n; \theta_{corrector})_i \cdot \bar{y}_T^n\,.\end{aligned} \tag{2}$$

An illustration of the proposed model is displayed in Figure 2.

3. Fashion dataset with external weak signals

3.1. Translate fashion to data

A collection of vision neural networks were designed and trained at detecting clothes details on pictures: type of clothing (pants, shoes, tops, etc.), form, size, color, texture, etc. Then, fashion experts designed fashion trends by aggregating these clothes details: a meaningful combination of items that represent an existing trend in the fashion sphere. To finely represent human behaviours based on social media, a group of thousands of random users, called a panel,

was created on several geolocalisations. Analyzing every day images shared on social networks by these panels with our computer vision algorithms, we can translate the history of thousands of fashion trends in thousands of time series. All sequences have 261 time steps, from 2015-01-05 to 2019-12-31 with weekly values and no missing values. Each value represents the number of users posts in a week where computer vision algorithms detected the fashion trend. As an illustration, an example of fashion time series is given in Figure 3.

3.2. Fashion dataset

Due to the increasing use of social media and behaviour changes, a normalization step is applied to the raw sequences. Each fashion trend is divided by its hierarchical parent category trend. Moreover, in order to avoid removing the seasonality of all sequences, we deseasonalized the hierarchical parent category trend before the normalization. For instance, as displayed in Figure 3, the raw Jersey Top trend for females in China is divided by the deseasonalized global Top trend for females in China. The final normalized sequence is expressed in share of category.

We therefore introduce a new dataset for fashion time series forecasting. It contains a sample of 10000 anonymized and normalized fashion trends for men and women, in 9 different categories and 5 geozones. An overview of it can be found in Table 1. This collection of 10000 fashion trends was selected in order to represent finely the issues faced by the fashion industry. For instance, some sequences show complex behaviours with sudden changes, referred to as emerging or declining trends. A central point of this work is to accurately detect and forecast such trends.

3.3. Weak signal

In theoretical fashion dynamics Rogers (1962), different categories of adopters follow a trend in succession, resulting in several adoption waves. Numerous social media influencers were selectioned by hand by fashion experts. By aggregating them, a specific “fashion-oriented“ panel is created. With the same

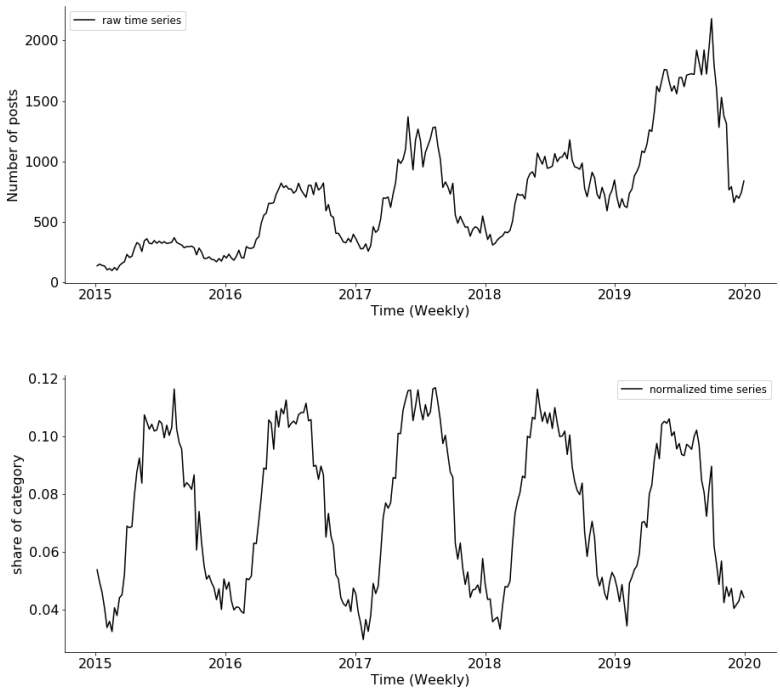

Figure 3: Example of difference between the raw sequence and the normalized one. In this example, we normalize by the deseasonalized global top fashion trend for females in China. (Top) Time series representing the raw signal of a top fashion trend for females in China. (Bottom) Time series representing the normalized signal of a top fashion trend for females in China.

Table 1: Fashion time series overview. For each couple geozone/category, the table gives the number of trends (Female/Male).

	Top	**Pants**	**Short**	**Skirt**	**Dress**	**Coat**	**Shoes**	**Color**	**Texture**
United States	411/208	149/112	47/22	29/-	20/-	208/151	293/86	38/44	85/81
Europe	409/228	134/114	48/21	28/-	20/-	211/159	303/78	41/42	87/74
Japan	403/218	136/107	49/31	28/-	23/-	185/149	311/78	46/42	92/65
China	424/202	147/114	46/29	27/-	27/-	178/161	310/78	41/47	88/77
Brazil	431/222	134/117	49/27	30/-	28/-	203/152	311/76	48/41	107/84
Total	2078/1078	700/564	239/130	142/-	118/-	985/772	1528/396	214/216	459/381

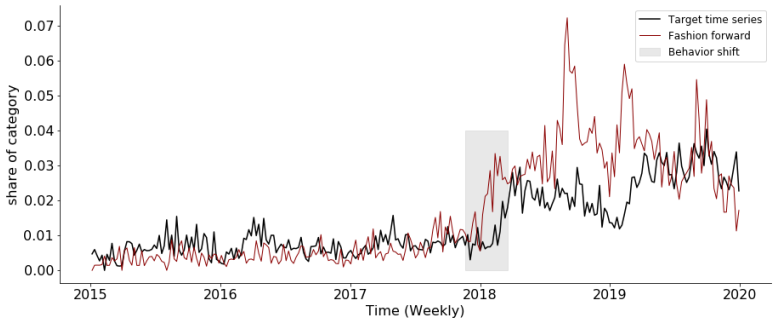

Figure 4: A shoes trend of the fashion dataset. In black the main signal and in red its associated *fashion-forward* weak signal. The shift between these two signals at the end of 2017/beginning of 2018 announces the future burst of the trend.

methodology as for the main panel described in Section 3.1 and Section 3.2, a normalized time series representing each fashion trend on this specific population is created. We named *fashion-forwards* this weak signal. For all trends $\{y_t\}_{1\leqslant t\leqslant T}$, let $y_t^{f,n}$ be the value of the n-th *fashion-forwards* sequence at time t and $\mathbf{y}^{f,n} = \{y_t^{f,n}\}_{1\leqslant t\leqslant T}$ be all the values of this sequence. As we want to detect shifts between the main signal and the fashion forward signal, the following input is computed for our hybrid model: for all $n \in \{1, \ldots, N\}$, for all $t \in \{1, \ldots, T\}$,

$$w_t^{f,n} = \frac{y_t^{f,n}}{y_t^{f,n} + y_t^n}\,.$$

Values close to 0.5 indicate a similar behaviour between the influencers panel and the general panel. For instance, an impressive emerging fashion shoes trend with its *fashion-forwards* weak signal is represented in Figure 4.

4. Experimental results

4.1. Training

The dataset is split into three blocks, *train*, *eval* and *test* sets. The 3 first years are used as the *train* set, the 4th year is kept for the *eval* set and the *test* set is made of the last year. The hybrid model is trained to compute a one-year

ahead prediction, h equal to 52, and the window size w is fixed at 104. Using the two first years of the *train* set, a first per-time-series parametric model for each time series is fitted. With the resulting collection of local models, a forecast of the third year is computed for each sequence. Corrector inputs are finally computed and the RNN is trained at correcting this first collection of third-year forecasts. For the *eval* set, per-time-series predictors are fitted a second time using the three first years and forecasts of the fourth year are computed. The *eval* set is used during training to control the learning of the RNN model and prevent overfitting. The per-time-series predictors are fitted a last time for the *test* set using the four first years. The final accuracy measures of all our models are computed on this *test* set. As an illustration, an example of our split is shown in Figure 5.

For the first parametric per-time-series models, existing Python or R libraries are used to estimate the different parameters $\theta^n_{predictor}$. Depending of the choice of local parametric models, two versions of HERMES are proposed. The first one uses as predictors an additive exponential smoothing model as a reference close to Smyl (2020). The second one uses the TBATS model of Livera et al. (2011) and achieves the best accuracy results on the fashion dataset. The neural network architecture is summarized in Figure 6. It is composed of 3 LSTM layers of shape 50 and a final Dense layer to provide the correct output dimension. A classical Adam optimizer is used with a learning rate set at 0.001 or 0.005, the batch size is fixed to 64 and the loss function is defined as follows:

$$\ell(y^n_{T+1:T+h}, \widehat{y}^n_{T+1:T+h|T}) = \frac{1}{\bar{y}^n_T} \sum_{i=1}^{h} \left| y^n_{T+i} - \widehat{y}^n_{T+i|T} \right| .$$

This choice of L_1 loss function is motivated by its robustness to outliers which accounts for some time series in the fashion industry with very specific behaviors. The loss and previous parameters are all set with a grid search (additional materials can be found in Appendix B). The code is developed in Python using the Tensorflow library. It allows the use of GPU to speed up the training process.

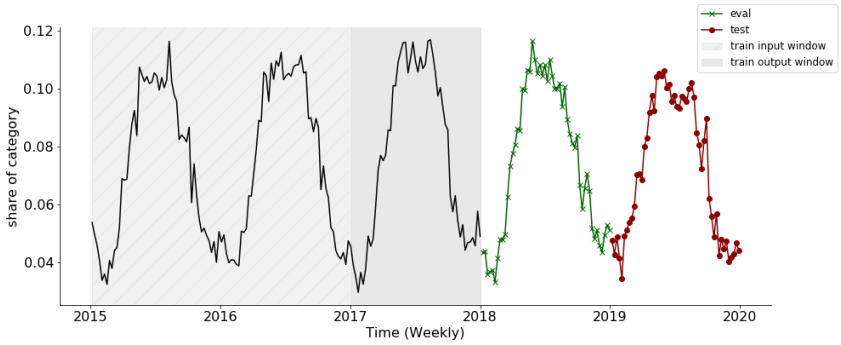

Figure 5: Temporal split for our training process. The three first years define our training set. The fourth year is used as our eval set and the final year is reserved for the test set.

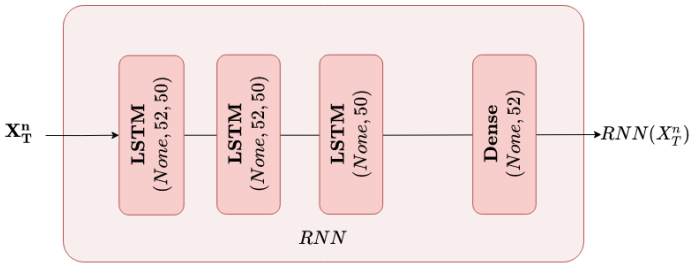

Figure 6: Architecture of the RNN corrector part of the HERMES framework. The same architecture is used in the *lstm* benchmark model.

4.2. Benchmarks, hybrid models and Metrics

As benchmarks, several widespread statistical methods and deep learning approaches were selected. Using the R package `forecast` and the Python packages `statsmodels`, `tbats`, for each time series, predictions are computed with the following methods: *snaive*, *ets*, *stlm*, *thetam*, *tbats* and *auto.arima*. The forecast of the *snaive* method is only the repetition of the last past period. The *ets* model is an additive exponential smoothing with a level component and a seasonal component. The *stlm* approach uses a multiplicative decomposition and models the seasonally adjusted time series with an exponential smoothing model. The *Thetam* model decomposes the original signal in θ-lines, predicts each one separately and recomposes them to produce the final forecast and *tbats* uses a trigonometrical seasonality. Finally, *auto.arima* is the R implementation of the ARIMA model with an automatic selection of the best parameters. A complete description and references for these models can be found in Hyndman et al. (2020). As a deep learning approach, a full LSTM (*lstm*) neural network composed of 3 LSTM layers of shape 50 and a final Dense layer of shape 52 is considered. Two versions of HERMES are proposed. They are called respectively *hermes-ets* and *hermes-tbats* according to the per-time-series model choice. Moreover, two versions with the inclusion of the weak signals (ws) are proposed. They are referred to as *hermes-ets-ws* and *hermes-tbats-ws*. In order to provide a fair comparison, a *lstm* with the weak signals named *lstm-ws* is trained.

To compare the different methods, we use the Mean Absolute Scaled Error (MASE) for seasonal time series. As our sequences have completely different scales, from 10^{-5} to 10^{-1}, this metric was chosen to compute a fair error measure, independent of the scale of the sequence and suited for our seasonal fashion time series. The MASE metric is defined as follows, with m the seasonal period:

$$\text{MASE} = \frac{T-m}{h} \frac{\sum_{j=1}^{h} |Y_{T+j} - \hat{Y}_{T+j}|}{\sum_{i=1}^{T-m} |Y_i - Y_{i-m}|}.$$

Detecting emerging and declining trends is a crucial issue for the fashion industry. A correct or incorrect prediction could lead to good returns or massive

waste due to overstock or unsold clothes. In addition to the MASE accuracy metric, the different methods are also evaluated on a classification task and especially differences between methods using weak signals or not. In a given year, an increasing trend is defined as a trend that does more than 5% of growth on average with respect to the previous year. In the same way, a decreasing trend is defined as a trend that declines by 5% on average or more. Other trends are classified as flat trends. With this threshold, the proposed fashion dataset is almost balanced on the *test* set: There are 3087 increasing trends, 3342 decreasing trends and 3571 flat trends.

4.3. Result for Heuritech Fashion dataset

10000 Heuritech Fashion time series global accuracy. For the two metrics and for each model, we compute the average on all sequences in the final year. Results are displayed in Table 2. For our model using neural networks, 10 models are trained with different seeds. The average and the standard deviation of their results are computed and displayed. For the statistical models, TBATS largely dominates the alternatives in terms of MASE. It is one of the main motivations why this model is used on the best HERMES candidate as the predictor model.

Considering the new HERMES approach, *hermes-tbats* and *hermes-tbats-ws* slightly outperform the alternatives in terms of MASE and are stable across the different trainings. Regarding *hermes-ets*, although it is very similar to the baseline Smyl (2020), its accuracy remains low in comparison to the *lstm* benchmark or HERMES using TBATS.

Models using our weak signals perform similarly as without-weak-signals models for the MASE. Interestingly, weak signals significantly improve the accuracy in detecting emerging and declining trends. Figure 7 displays some examples of *hermes-tbats* models and some weaknesses that can be corrected.

10000 Heuritech Fashion time series classification task. Classification results between the *tbats* model and the hybrid method *hermes-tbats* are given in Table 3, we note an impressive decrease of impactful errors: i.e.

Table 2: Results summary on the 10000ts Fashion dataset. For each metric, the average on all our time series is computed. For approaches using neural networks, 10 models are trained with different seeds. The mean and the standard deviation of the 10 results are displayed.

	MASE ↓		**ACCURACY** ↑	
	mean	*std*	*mean*	*std*
snaive	0.881	-	0.357	-
thetam	0.844	-	0.482	-
arima	0.826	-	0.464	-
ets	0.807	-	0.449	-
stlm	0.770	-	0.482	-
hermes-ets-ws	0.769	0.005	0.501	0.007
hermes-ets	0.758	0.001	0.490	0.006
tbats	0.745	-	0.453	-
lstm-ws	0.728	0.004	0.500	0.008
lstm	0.724	0.003	0.498	0.007
hermes-tbats	0.715	0.002	0.488	0.008
hermes-tbats-ws	**0.712**	0.004	**0.510**	0.005

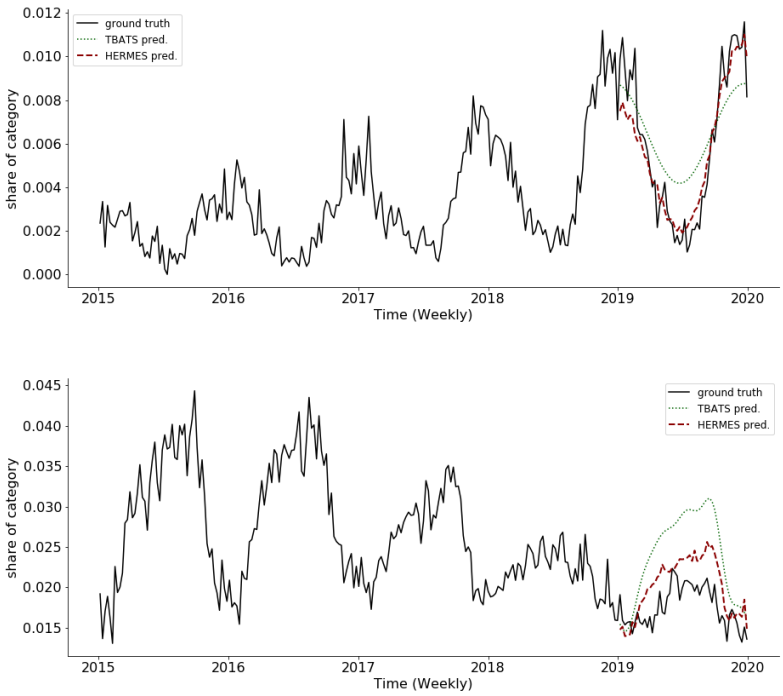

Figure 7: *hermes-tbats* forecast examples. In green the prediction of the per-time-series predictors *tbats*. In red the final forecast of our HERMES hybrid model *hermes-tbats*. (Top) Time series representing a top fashion trend for females in The United States. (Bottom) Time series representing the horizontal stipes texture fashion trend for females in China.

Table 3: *tbats*, *hermes-tbats* and *hermes-tbats-ws* models confusion matrix

	tbats		
	pred-dec	pred-flat	pred-inc
true-dec	902	2113	327
true-flat	351	2920	300
true-inc	300	2078	709

	hermes-tbats		
	pred-dec	pred-flat	pred-inc
true-dec	1261	1960	121
true-flat	549	2823	199
true-inc	214	2004	869

	hermes-tbats-ws		
	pred-dec	pred-flat	pred-inc
true-dec	1956	1245	141
true-flat	1257	2087	227
true-inc	358	1620	1109

forecasting an increase instead of a decrease and vice versa. The *hermes-tbats* model divides by 3 the error rate in comparison to *tbats* with only a slight decrease of the number of correct increase/decrease predictions. However, with our weak signals, we see that *hermes-tbats-ws* is able to catch twice as much as its relative model without weak signals while keeping a relatively low number of impactful errors.

Size of the dataset. In addition to the results on the fashion dataset gathering 10000 time series, the behaviour of the HERMES model is analyzed when it is trained on smaller datasets: two experiments were performed, HERMES models were trained on a reduced dataset of 1000 time series and on a reduced dataset of 100 time series. Results are given in Table 4.

First, the hybrid framework *hermes-tbats* achieves the best performance in terms of global accuracy on both datasets. Due to the strength of its per-time-series predictor TBATS, the hybrid model succeeds at correcting TBATS and reaches a satisfactory final accuracy. Secondly, we can note that the accuracy of the full neural network *lstm* decreases when the dataset size decreases. On the small dataset of 100 time series, a local statistical model like *tbats* or *stlm* largely outperforms its accuracy level. Providing sharp predictions from scratch

Table 4: Results summary on the 1000 time series and 100 time series Fashion dataset. The MASE average on all our time series is computed. For the two approaches using a neural network, 10 models with different seeds are trained. the mean and the standard deviation of the 10 results are displayed.

1000 time series Fashion dataset

	MASE	
	mean	*std*
snaive	0.871	-
thetam	0.849	-
arima	0.821	-
ets	0.801	-
stlm	0.765	-
lstm	0.740	0.007
tbats	0.734	-
hermes-tbats	**0.719**	0.002

100 ts Fashion dataset

	MASE	
	mean	*std*
snaive	0.876	-
thetam	0.823	-
arima	0.814	-
ets	0.785	-
lstm	0.767	0.045
stlm	0.742	-
tbats	0.745	-
hermes-tbats	**0.739**	0.003

is a complex task and high-dimensional recurrent neural networks require large amounts of data to do so. Nevertheless, with the HERMES framework, the RNN task is largely simplified. Our model needs less data to be trained and to obtain interesting performance.

4.4. Result for M4 weekly dataset

We also assessed the performance of HERMES using the M4 weekly dataset Makridakis et al. (2020). The M4 dataset gathers 359 weekly time series and has 3 main differences compared to our proposed fashion dataset. Firstly, sequences do not have the same length with sequence lengths lying between 93 and 2610 time steps. Secondly, as some of the sequences represent financial signals or some others are demographic sequences, the 359 time series have very distinct scales and dynamics. Thirdly, compared to the previous fashion application, the time horizon of the prediction is set to 13 for the weekly dataset.

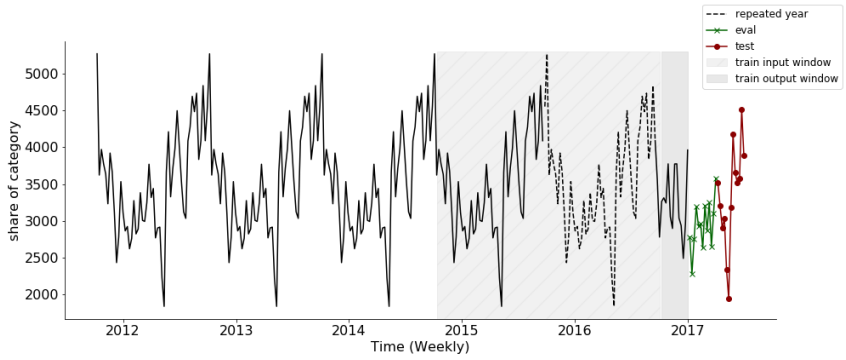

Figure 8: One of the shortest sequences of the M4 weekly dataset (93 time steps). In order to fit its predictor, the last complete year is duplicated in order to reach a total length of 300 time steps.

Training. The M4 dataset is firstly preprocessed. This preliminary step is motivated by two reasons. First, as some sequences are short (93 time steps), they limit the window size w of the RNN part and consequently the global accuracy of the HERMES approach. Second, the longer time series slow down the training of the collection of the first predictors especially for complex methods like TBATS. We kept 300 time steps for each sequence in order to train the HERMES model. For the shorter sequences, the train set is duplicated in order to reach the length of 300. Longer sequences are cropped in order to keep the last 300 time steps.

The horizon h is set to 13 and the window size w is set to 104. For the RNN part, the same architecture as the one described in Figure 6 is used. The Adam optimizer is used with a learning rate equal to 0.005 and a batch size set to 8. As the M4 weekly dataset is small, a rolling window is used on the train set in order to increase the train number of examples and improve training results. Three windows are computed for each sequence for the RNN train set. An overview of our train, eval, test set split and the resizing of the shortest sequences is given in Figure 8. The previous parameters: window size, learning rate, batch size and the number of train windows per time series are set using a grid search, see Appendix B.

Benchmarks. The M4 competition provides a rich collection of benchmarks emcompassing statistical models and neural network approaches. The same candidates are used in this part as baselines. In addition, the hybrid model named *Uber* of S.Smyl is added. For a complete description and references of the benchmark models, see Makridakis et al. (2020). As a HERMES candidate, a version using TBATS is proposed and called *hermes-tbats*. Following the M4 competition methodology, models are evaluated according to the MASE, the SMAPE and the OWA measures. A complete definition of these metrics is proposed in Makridakis et al. (2020), see also Appendix 5 for additional information about the M4 weekly dataset.

Results and discussion. The final results for the M4 weekly dataset are displayed in Table 5. The HERMES approach *hermes-tbats* outperforms all the benchmarks. This result is partially induced by the use of TBATS per-time-series predictors which achieves very good results on the test set. Regarding the hybrid model proposed by S.Smyl, its accuracy remains low in comparison to *tbats* and *hermes-tbats*. With this second application, two important conclusions can be made. Firstly, the results provided by *hermes-tbats* confirm that the HERMES approach is a general framework, well suited for a large collection of forecasting tasks. Secondly, the accuracy gap between the two hybrid candidates validates the HERMES approach and illustrates the importance of a global framework able to leverage any kind of per-time-series predictors.

5. Conclusion

The motivation of this paper was to present HERMES, a new hybird model for non stationary time series forecasting. By mixing the performance of local parametric models and a global neural network, *hermes-tbats* clearly outperforms traditional statistical methods and full neural network models on two forecasting tasks. Furthermore, this new model is totally suited to deal with external signals. With a fine pre-processing and a well-designed architecture, our hybrid framework succeeds at leveraging our complex extra data and reaches

Table 5: Results summary on the m4 weekly dataset. For each metric, the average on all our time series is computed. For approaches using a neural network, 10 models are trained with different seeds. The mean and the standard deviation of the 10 results are displayed.

	SMAPE		**MASE**		**OWA**	
	mean	*std*	*mean*	*std*	*mean*	*std*
MLP	21.349	-	13.568	-	3.608	-
RNN	15.220	-	5.132	-	1.755	-
snaive	9.161	-	2.777	-	1.000	-
SES	9.012	-	2.685	-	0.975	-
Theta	9.093	-	2.637	-	0.971	-
Holt	9.708	-	2.420	-	0.966	-
Com	8.944	-	2.432	-	0.926	-
Damped	8.866	-	2.404	-	0.917	-
S.Smyl Hyndman et al. (2020)	7.817	-	2.356	-	0.851	-
tbats	8.111	-	2.214	-	0.841	-
hermes-tbats	**7.597**	0.113	**2.205**	0.042	**0.812**	0.011

very promising accuracy levels in terms of classification. In addition to this article, a fashion dataset gathering a sample of 10000 time series and a collection of weak signals is provided. We believe that this dataset contains really fine dynamics and interactions where complex models would express their potential. By making it publicly available[1], we hope that it will enhance the diversity of datasets for time series forecasting and pave the way for further explorations. As a possible future work, designing new models for the weak signals would improve their inclusion in the HERMES architecture. Focusing on the examples with huge changes of behaviours, a fine analysis of the impact of the collection of weak signals is the topic of ongoing works. In the same way, an interesting improvement of the hybrid framework can be to introduce not a single but several neural networks trained at correcting different kinds of weaknesses. A perspective is to add a latent discrete label to select dynamically the regime shifts.

[1] `http://files.heuritech.com/raw_files/f1_fashion_dataset.tar.xz`

Table 6: M4 weekly dataset overview. For each category, the number of sequences and the average length are given.

	Nb. of sequences	**Avg. length**	**Min. length**
Demographic	24	1659	1615
Finance	164	1237	260
Industry	6	834	356
Macro	41	1264	522
Micro	112	473	**93**
Other	12	1598	470

Appendix A. M4 weekly dataset and results

The M4 weekly dataset is a collection of 359 time series with contrasting behaviours and sizes. An overview of the dataset is given in Table 6 and some examples of sequences are given in Figure 9. As sequences come from a wide diversity of sectors, the forecasting task is very challenging. However, on some examples as in Figure 10, efficient corrections of the TBATS forecasts can be obtained.

Looking at the final results of the M4 competition, some models reach a higher accuracy than the HERMES framework on the weekly dataset. All of them are ensemble frameworks: methods that mix different kinds of approaches to reach a higher final accuracy. As the M4 weekly dataset gathers really heterogeneous sequences, combining several methods to leverage their strengths appear to be a promising way. However in this paper, only individual models were evaluated in order to provide a fair comparison with the HERMES framework.

Appendix B. Training parameters and loss

5.1. Loss grid search on the Fashion Dataset

Using deep learning models in time series forecasting is an appealing way to achieve higher accuracy performance. However, it induces two main issues.

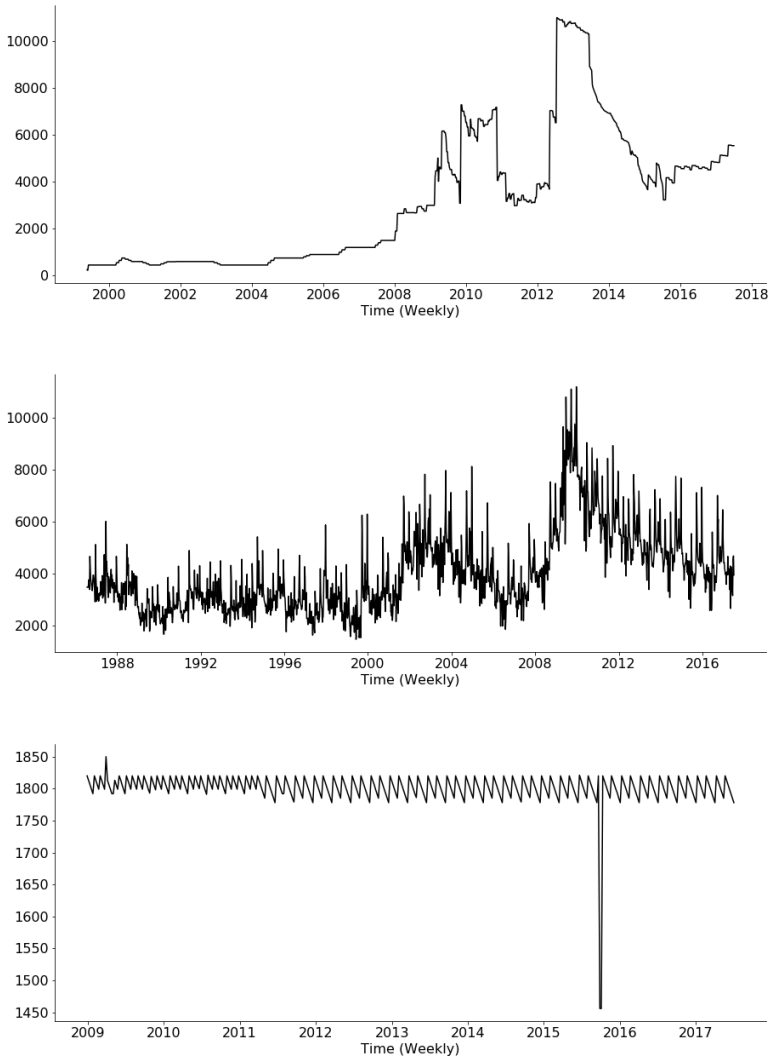

Figure 9: Examples of time series from the M4 weekly dataset. From Top to Bottom : time series called *W10* from the *Other* category, *W20* from the *Macro* category and *W220* from the *Finance* category.

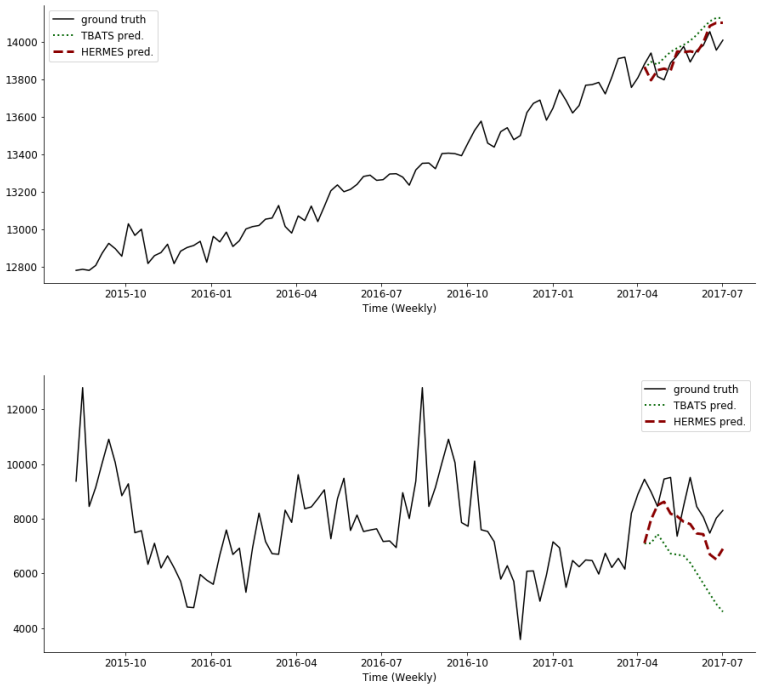

Figure 10: *hermes-tbats* forecast examples on the M4 weekly dataset. In green the prediction of the per-time-series predictors *tbats*. In red the final forecast of our HERMES hybrid model *hermes-tbats*. (Top) the *W133* time series, (Bottom) the *W314* time series.

First, it requires a large enough dataset to train the model as illustrated in Section 4. Second, a dataset can hide contrasting time series in terms of scale, noise and behaviour. These differences can impact training performance. For the HERMES architecture, some candidate losses were defined for the training: the Mean Absolute Error (MAE), the Mean Square Error (MSE), the Scaled Mean Absolute Error (SMAE) and the Scaled Mean Square Error (SMSE). The loss functions are defined as follows:

$$MAE = \frac{1}{h}\sum_{i=1}^{h} |y^n_{T+i} - \widehat{y}^n_{T+i|T}| \,,$$

$$MSE = \frac{1}{h}\sum_{i=1}^{h} (y^n_{T+i} - \widehat{y}^n_{T+i|T})^2 \,,$$

$$SMAE = \frac{1}{\bar{y}^n_T}\sum_{i=1}^{h} |y^n_{T+i} - \widehat{y}^n_{T+i|T}| \,,$$

$$SMSE = \frac{1}{\bar{y}^n_T}\sum_{i=1}^{h} (y^n_{T+i} - \widehat{y}^n_{T+i|T})^2 \,.$$

For each loss, 10 *hermes-tbats-ws* models have been trained with different seeds and the final mean and standard deviation are given in Figure 11. The final Scaled Mean Absolute Error reaches the lowest MASE and was selected to train all the HERMES model in this paper.

5.2. Parameters grid search on the M4 weekly Dataset

In addition to the loss function, the HERMES model also depends on several hyperparameters to set correctly in order to reach satisfactory performance. For instance, an overview of the learning rate, batch size and number of windows per time series grid search for the M4 weekly dataset is shown in Figure 12. For each parameter, a collection of 10 *hermes-tbats* models have been trained with a range of values and the final OWA was calculated. As in the Figure 11, the mean and the standard deviation of each group of 10 trainings is computed. For the final *hermes-tbats* model of the M4 weekly dataset, the following set of parameters was selected: 3 windows per time series were used as the train set, the batch size was set to 8 and the learning rate was fixed to 0.005.

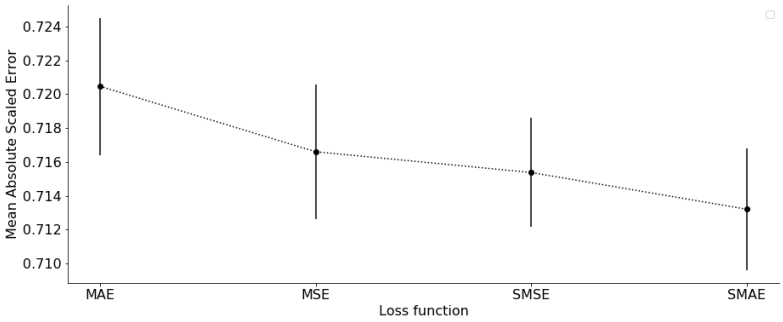

Figure 11: MASE accuray for the *hermes-tbats-ws* model depending on the loss used during the RNN training. For each loss, 10 models with different seeds have been trained. The mean and the standard deviation are represented with a point and a vertical line.

References

References

Bandara, K., Bergmeir, C., & Hewamalage, H. (2020). LSTM-MSNet: Leveraging forecasts on sets of related time series with multiple seasonal patterns. *IEEE transactions on neural networks and learning systems*, .

Box, G. E., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). *Time series analysis: forecasting and control*. John Wiley & Sons.

Brown, R. G., & Meyer, R. F. (1961). The fundamental theorem of exponential smoothing. *Operations Research*, *9*, 673–685.

Douc, R., Moulines, E., & Stoffer, D. (2014). *Nonlinear time series: Theory, methods and applications with R examples*. CRC press.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*, 1735–1780.

Hyndman, R. J., Athanasopoulos, G., Bergmeir, C., Caceres, G., Chhay, L., O'Hara-Wild, M., Petropoulos, F., Razbash, S., &

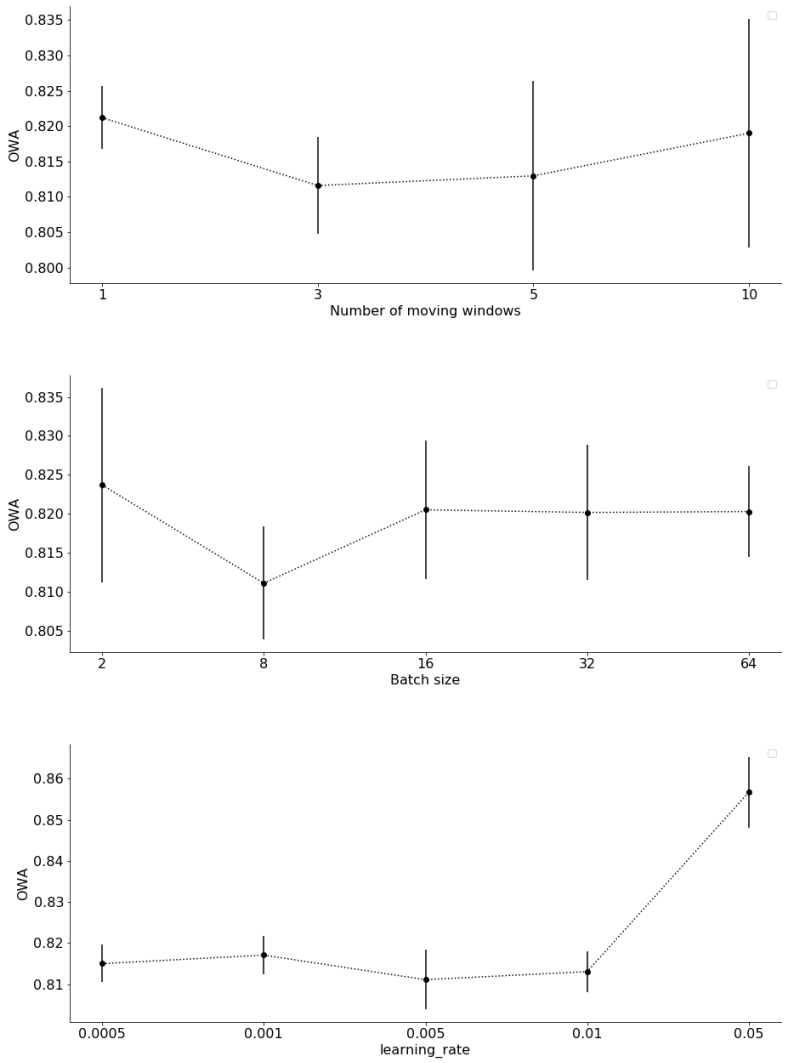

Figure 12: OWA for the *hermes-tbats* model on the M4 weekly dataset depending on 3 parameters used during the RNN training: Number of moving windows per time series, the batch size and the learning rate. For each parameter, 10 models with different seeds have been trained. The mean and the standard deviation are represented with a point and a vertical. (Top) Result of the HERMES model depending on the number of windows provided per time series to the RNN corrector. (Middle) Result of the HERMES model depending on the size of the batch size. (Bottom) Result of the HERMES model depending on the learning rate of the optimizer.

Wang, E. (2020). Package 'forecast'. *Online] https://cran. r-project. org/web/packages/forecast/forecast. pdf*, .

Jianwei, E., Ye, J., & Jin, H. (2019). A novel hybrid model on the prediction of time series and its application for the gold price analysis and forecasting. *Physica A: Statistical Mechanics and its Applications*, *527*, 121454.

Li, S., Jin, X., Xuan, Y., Zhou, X., Chen, W., Wang, Y.-X., & Yan, X. (2019). Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. *arXiv preprint arXiv:1907.00235*, .

Lim, B., Arik, S. O., Loeff, N., & Pfister, T. (2019). Temporal fusion transformers for interpretable multi-horizon time series forecasting. *arXiv preprint arXiv:1912.09363*, .

Livera, A. M. D., Hyndman, R. J., & Snyder, R. D. (2011). Forecasting time series with complex seasonal patterns using exponential smoothing. *Journal of the American Statistical Association*, *106*, 1513–1527.

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2018). The M4 competition: Results, findings, conclusion and way forward. *International Journal of Forecasting*, *34*, 802–808.

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2020). The M4 competition: 100,000 time series and 61 forecasting methods. *International Journal of Forecasting*, *36*, 54–74.

Martin, A., Ollion, C., Strub, F., Le Corff, S., & Pietquin, O. (2021). The Monte Carlo Transformer: a stochastic self-attention model for sequence prediction. *arXiv preprint arXiv:2007.08620*, .

Rogers, E. M. (1962). *Diffusion of innovations*. Simon and Schuster.

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

Särkkä, S. (2013). *Bayesian Filtering and Smoothing*. New York, NY, USA: Cambridge University Press.

Siami-Namini, S., Tavakoli, N., & Siami Namin, A. (2018). A comparison of ARIMA and LSTM in forecasting time series. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)* (pp. 1394–1401). doi:10.1109/ICMLA.2018.00227.

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

Touron, A. (2017). Modeling rainfalls using a seasonal hidden markov model. arXiv:1710.08112.

Touron, A. (2019). Consistency of the maximum likelihood estimator in seasonal hidden markov models. *Statistics and Computing*, *29*, 1055–1075.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. *31st Conference on Neural Information Processing Systems (NeurIPS 2017)*, .

Zhang, G. P. (2003). Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing*, *50*, 159–175.

Zucchini, W., MacDonald, I. L., & Langrock, R. (2017). *Hidden Markov models for time series: an introduction using R*. CRC press.