POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Homework 2 - Time Series forecasting challenge

MSc in Bioinformatics for Computational Genomics - Computer Science and Engineering
Course: Artificial Neural Networks and Deep Learning course

Authors: Daniele Bottazzi, Sara Resta, Gaia Saldarini
Professor: Matteo Matteucci
Academic year: 2023-2024

---

1. Introduction

This project aims at predicting the 18 future values of uncorrelated time series, exploiting past observations. The dataset is composed of 48.000 univariate times series belonging to 6 different domains. The forecasting model needs to exhibit generalization capabilities to predict in any time context.

2. Exploratory analysis and issues

A visual inspection of the time series unveils their heterogeneous patterns: they show ascending, descending, smooth, noisy and spiked behaviours.

2.1. Categories

In addition to the uneven distribution across categories, series within each category exhibit diverse patterns, as shown in Figure 1. The initial attempt to model the data based on categories yielded unpromising results, prompting a decision to disregard categories in future modeling efforts.

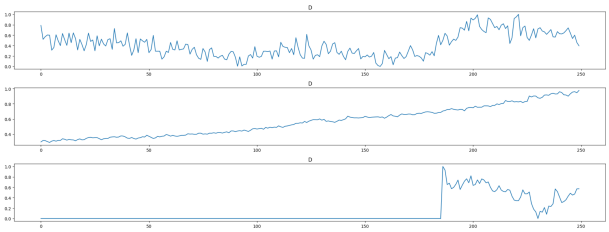

Figure 1: Example of time series from category D

The key observation is that the division in categories does not necessarily capture meaningful characteristics, as each category consists of series from diverse data sources without a guarantee of close intra-category relationships.

2.2. Normalization

The original time series were normalized in $[0, 1]$, independently from each other, before being cut in smaller sequences. Therefore `MinMaxScaler` gives slight changes, and does not reduce the impact of outliers. `StandardScaler` and `RobustScaler` were also applied, but none of these normalizations seemed to bring substantial improvements. [2]

2.3. Series length

Series' length distribution (Figure 2) is highly variable, going from a minimum of 24 to a maximum of 2776; the mode is 51 and the mean is 198.30.

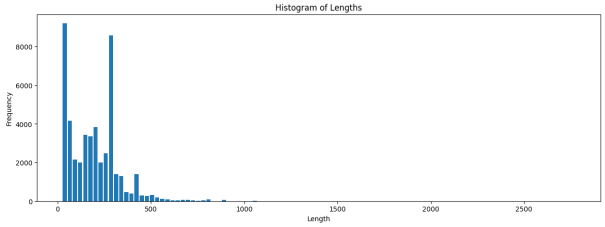

Figure 2: Length distribution

2.4. Dynamic Time Warping

From the beginning, 26 duplicates were removed. To spot pseudo-duplicate series, hardly recognizable otherwise, Dynamic Time Warping (DTW) similarity metric was employed. DTW is a dynamic programming algorithm that measures the distance between a pair of not synchronized series [8].

The sequences are "warped" non-linearly in the time dimension to determine a measure of their similarity independent of certain non-linear variations in the time dimension; the goal is to find the optimal match between similar elements of the time series, by applying a local stretch or compression. From the identified sets of series with pairwise DTW distances lower than a threshold of 0.2, all but the longest were removed to prevent overfitting. From this point onward, the analysis is performed on the `filtered_datasets.npy`, with 47048 almost-unique time series.

2.5. Trends and Seasonality

Upon examining the series, one can observe a periodic pattern, which may vary in its clarity across different series. This observation is supported by autocorrelation analysis, a statistical technique employed to assess the extent of resemblance between a particular time series and its delayed versions across consecutive time intervals (lags). Autocorrelation serves as a valuable tool for revealing trends and patterns, as it quantifies the connection between the current and the preceding values of a variable [1].

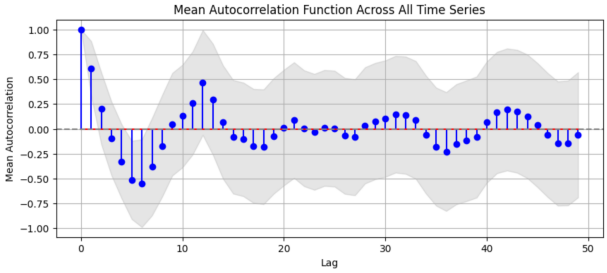

Figure 3: Autocorrelation plot

In Figure 3, the autocorrelation plot show distinct peaks every 12 values, a pattern that aligns with the plausible assumption considering the economic context of the categories (nominally, demography, finance, industry, macro-economy, micro-economy, and other).

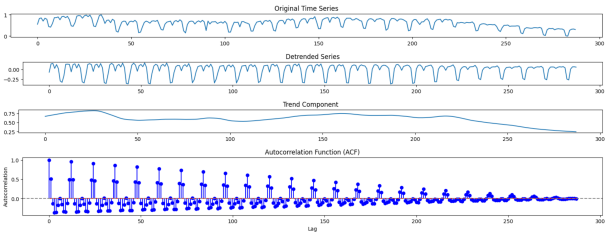

Figure 4: STL decomposition of a time series

Seasonal-Trend decomposition by LOESS minimization (`STL` from `statsmodels`, [3]) was adopted to extract from each series its constituent elements: trend and seasonality, as in Figure 4. Trend denote continuous linear shifts in values over time, while seasonality encompasses the recurring patterns or short-term cycles. Both of these components coexist within the series, often intertwined and confounded by the presence of residual noise.

2.6. Build sequences and padding

The aim is to forecast the next 18 values of a given time series, so telescope is set to 18. The parameters `window` and `stride` have been tuned to build sequences that could capture useful information from the times series. Considering the 12-periodicity , the seasonality pattern should be present from different starting points across the sequences, so that the behaviour can be learnt even if the pattern is not centered in the window. For example, to see the seasonality from 6 points of view, stride must be 22. To avoid unnecessary padding and given the high number of provided series, sequences are built starting from the most recent one going backward till the last available window. Series shorter than `window + telescope` were backward replicated to avoid padding with zeros.

2.7. Loss metrics

Traditional loss metric function for time series forecasting is `MeanSquaredError` (`MSE`); but given the individual normalization of the series, to give a more robust interpretation of the results, `MeanAbsoluteError` (`MAE`) was also considered, which is the average percentage error (scaled between 0 and 1) obtained on the predicted points compared to the ground truth.

3. Model experiments

3.1. Long-Short Term Memory RNN

Both LSTM and GRU have been employed in the first simple models; since LSTM gave better prediction scores, it has been used also for the following experiments. The LSTM model is built as Bidirectional RNN in order to traverse the sequences both left-to-right and right-to-left. It is trained with `EarlyStopping` and `ReduceLROnPlateau`.

3.2. Attention mechanism

A Multi Head Attention layer was added after the Bidirectional LSTM to allow the model to capture more complex relationships, improve representation learning and handle long-term dependencies [6]. This improved the model performances, so it has been adopted for most of the other experiments [9].

3.3. Model ensemble based on clustering

Clustering have been employed to identify groups of series with the same pattern. An autoencoder model have been trained to extract series' latent representations. Clustering based on the hidden vectors of the latent space was performed with `KMeans` which identified 3 groups, which by inspection can be considered as ascending, descending and irregular/spiked.

For each cluster, one model has been built; the three models were combined in an ensemble for testing. At test time, the latent representation of test series was extracted with the autoencoder trained before and it was assigned to the closest cluster by `KNN` to compute the prediction.

3.4. Model with trend and seasonality

The model is composed of two modules: one takes the trend-residual part of the series as input, and the other takes the last 36 observations of the seasonality. Both models performed their respective forecasts with a Bidirectional LSTM and a `MultiHeadAttention` layer. Then, the two models were combined to optimize the sum of the predictions. [5]

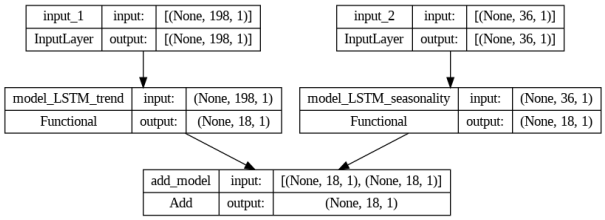

Figure 5: Model to forecast separately trend/seasonality

3.5. ResNet-style architecture

To extract richer information from the times series, two Residual blocks have been added between the input layer and the BiLSTM [4]. For each block dropout and $L1L2$-regularization were added to prevent overfitting. The BiLSTM is followed by a `MultiHeadAttention` layer.

3.6. Transformers and Time2Vec

A Temporal Fusion Transformer (TFT) was adapted to includes static components, exogenous time series, and utilizes recurrent layers, interpretable self-attention layers, and specialized components for feature selection and gating. Time2Vec embedding is used for a model-agnostic time representation.

Time2Vec embedding was used to provide a model-agnostic vector representation for time [7].

3.7. Final model and comparison

The described models exhibit similar performance levels despite their distinct approaches. ResNet-style CNN may possess superior generalization capabilities. The clustering-based method holds promise, especially with a greater variety and distinguishability of time series typologies. The Bidirectional LSTM double-input model stands out for its interpretability in forecasting results. By predicting future seasonality and trends independently, based on past observations, it provides a more explainable output, as illustrated in Figure 6.

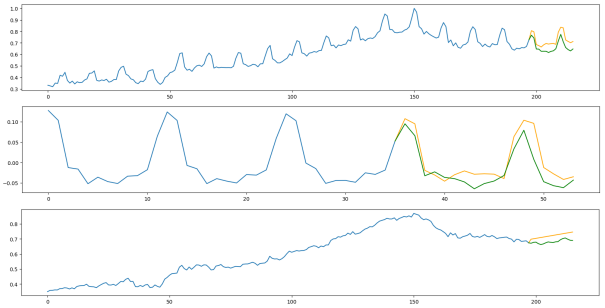

Figure 6: Predictions of a time series (top), seasonality (mid), trend (bottom); shifted for visualization purposes

The most effective model for predicting the next 18 values in a time series is the latter one, which operates on the decomposition of the original series. This model combines predictions using an `Add` layer, resulting in a robust forecasting approach.

4. Results

In the Table below, the performances of the best models in forecasting 18 values are reported.

Model	test MSE	test MAE
Cluster-based ensemble	0.01158	0.06966
ResNet-style (BiLSTM)	0.01127	0.07045
BiLSTM trend+seasonality	0.01025	0.06974

Contributions

Notebooks		Tasks	Bottazzi	Resta	Saldarini
0	Exploratory Analysis (link)	Categories inspection and data cleaning	✓	✓	✓
		Normalization using different scalers	✓	✓	✓
		Clustering on series in the time domain	✓		
		Dynamic Time Warping	✓		
		Trend, periodicity and autocorrelation	✓		✓
		Examples of trend + seasonality decomposition			✓
1	Simple Bidirectional LSTM (link)	Vanilla model	✓	✓	✓
		Ablation studies		✓	
		Hyperparameter tuning		✓	
		Experiments with different length of sequences			✓
2	Bidirectional LSTM with attention (link)	Experiments with attention layers and LSTM stacking	✓	✓	
		Hyperparameter tuning		✓	
3	Auto Encoder cluster ensemble (link)	Autoencoder modeling and training			✓
		Clustering sequences using their latent representation			✓
		Building a model for each cluster		✓	✓
		KNN for cluster assignment and model ensemble			✓
4	ResNet style BiLSTM (link)	Ablation studies		✓	
		Hyperparameter tuning	✓		
		Build sequences with backward replication	✓	✓	✓
		Experiments with different length of sequences		✓	
5	Trend and Seasonality (link)	Decompose trend and seasonalities	✓		
		Ablation studies	✓		
		Hyperparameter tuning	✓	✓	
		Experiments with different length of sequences		✓	✓
		Other Tasks and Architectures	**Bottazzi**	**Resta**	**Saldarini**
	Temporal Fusion Transformer with Time2Vec embedding	Adapting of the transformer model	✓	✓	
		Experiments with attention layers and different embeddings	✓	✓	

Data and code availability statement

The `training_dataset` for the project can be downloaded at this link. The `filtered_dataset` is made available here. The Notebooks produced to perform the analyses reported are available at links in the Table. A Cython implementation of STL function, adapted from `statsmodels`, can be downloaded at this link.

References

[1] Autocorrelation function (ACF) by statsmodels. `https://www.statsmodels.org/stable/generated/statsmodels.tsa.stattools.acf.html`. Accessed: 2023-12-22.

[2] Compare the effect of different scalers on data with outliers. `https://scikit-learn.org/stable/auto_examples/preprocessing/plot_all_scaling.html`. Accessed: 2023-12-22.

[3] Seasonal-Trend decomposition using LOESS (STL). `https://www.statsmodels.org/dev/examples/notebooks/generated/stl_decomposition.html`. Accessed: 2023-12-22.

[4] Saleh Albelwi. A robust energy consumption forecasting model using resnet- lstm with huber loss. 22:301, 07 2022.

[5] Serdar Arslan. A hybrid forecasting model using lstm and prophet for energy consumption with decomposition of time series data. *PeerJ Computer Science*, 8:e1001, 2022.

[6] Yuntong Hu and Fuyuan Xiao. Network self attention for forecasting time series. *Applied Soft Computing*, 124:109092, 2022.

[7] Seyed Mehran et al Kazemi. Time2vec: Learning a vector representation of time. *Borealis AI*, 2021.

[8] Stan Salvador and Philip Chan. Toward accurate dynamic time warping in linear time and space. *Intell. Data Anal.*, 11(5):561–580, oct 2007.

[9] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.