Mini-Project (ML for Time Series) - MVA 2024/2025

Darius Dabert darius.dabert@polytechnique.edu
Elia El Khoury eliaelkhoury2@gmail.com

January 8, 2025

1 Introduction and contributions

The paper [1] introduces a new method for time series forecasting that combines a recurrent neural network with a dimension-reducing symbolic representation of the time series. This approach seeks to minimize the model's sensitivity to hyperparameters and reduce training runtime, all while maintaining the performance of the model.

The paper begins by explaining the concept of symbolic representation for time series, focusing in particular on how the ABBA representation works. It then describes the architecture of recurrent neural networks, specifically detailing the architecture of LSTM cells, leading to the adoption of a multi-layer LSTM network. Finally, the paper compares the results of forecasting using both raw and symbolic data, demonstrating significant improvements in speed and runtime, along with more visually accurate predictions that resemble the original time series.

Our study of this paper will focus on conducting various tests to explore the performance and limitations of the proposed ABBA-LSTM model. We will start by testing more in details the model's lower sensitivity to hyperparameters compared to the raw LSTM model. Additionally, we will examine the existence of a correlation between the performances of the ABBA reconstruction and the forecasting of the time series then the accuracy of the ABBA-LSTM model on the trainig set and the forecasting of the time series. Finally, we will identify the scenarios in which the proposed model performs best and the conditions that optimize its effectiveness.

For this project, we divided the work equally between us and both contributed to all aspects of the process. We began by thoroughly reading and understanding the paper, then moved on to designing new tests we believed would be interesting to explore. To carry out these tests, we studied and used the code provided in the paper to run the model, while also adding our own code to implement the new tests and analyze the results.

2 Method

As stated above, this paper introduces a model that combines a recurrent neural network with symbolic representation. In this section, we will explain how both methods work, first separately and then combined, and define the mathematical framework used throughout the paper and our project.

2.1 Multi-Layer LSTM Network

Recurrent neural networks (RNNs) are designed to process sequential data more efficiently by accounting for the sequential nature of the data. A standard recurrent unit processes the sequence elements one at a time, starting with the first element and then feeding in the next. At each time step, the unit takes two inputs: an element of the sequence $x_i \in \mathbb{R}^d$ and the output of the same unit from the previous time step, $h_{i-1} \in \mathbb{R}$.

An LSTM cell (see Appendix A fig. 6a), denoted $\mathcal{L}$, is a specific example of a recurrent unit that takes three inputs, the hidden state (h), the cell state (c) and the input vector (x), and produces two outputs:

$$(h_t, c_t) = \mathcal{L}(h_{t-1}, c_{t-1}, x_t) = (o\_g_t.tanh(c_t),\ f\_g_t.c_{t-1} + i\_g_t.c\_u_t)$$

where the forget gate (f_g) controls how much of the current cell state is forgotten, the input gate (i_g) controls how much of the cell update is added to the cell state, the output gate (o_g) controls how much of the modified cell state leaves the cell to become the next hidden state, and the cell update (c_u) is constructed as a single neuron with a tanh activation function.

In practice, multiple layers of LSTM cells are often stacked to increase the complexity of the function represented by the network. A graphical illustration is given in Appendix A figure 6b.

2.2 Symbolic Representation

The main idea of symbolic representation is to transform the numerical data of a time series $T = \{t_1, \ldots, t_n\}$ into a sequence of symbols $S = \{s_1, \ldots, s_m\}$, where all symbols lie in a finite alphabet $\mathbb{A} = \{a_1, \ldots, a_k\}$. In the paper, ABBA is the symbolic representation method used. In this method, the parameters m and k are chosen adaptively, and the representation process consists of two main steps.

The first step, compression, consists in selecting $m + 1$ indices $i_1 = 1 < i_2 < \cdots < i_{m+1} = n$ and approximating the time series T by fitting a straight line segment between each pair of consecutive indices. Each line segment can be characterized by its *length* $L_j = i_{j+1} - i_j$ and its *increment* $I_j = t_{i_{j+1}} - t_{i_j}$, for $j = 1, 2, \ldots, m$. The second step, digitization, consists in clustering the set of line segments $\{(L_1, I_1), (L_2, I_2), \ldots, (L_m, I_m)\}$ into k clusters. Each cluster is then assigned a unique symbol from $\mathbb{A}$, resulting in the symbolic representation $S = \{s_1, s_2, \ldots, s_m\}$.

Converting from symbolic to numeric representation involves reversing the process by taking the center of each cluster, redefining the length of each line, and then stitching them together.

2.3 ABBA-LSTM Model

The combination of these two methods described above enables time series forecasting by simply using the ABBA model to compute a symbolic representation of the series and then feeding this representation into the multi-layer LSTM network, which predicts the next symbol from the alphabet $\mathbb{A}$. This process is repeated a certain number of times and finally the predicted symbols are converted back into numerical data, representing the forecasted values of the time series. The hyperparameters influencing this process include the lag l, which represents the length of the input sequence, and max_k, the maximum number of elements allowed in the alphabet $\mathbb{A}$.

3 Data

For our study of this paper and the tests we conducted, we used four time series datasets. These datasets are among those used in [1], although we selected only few ones relevant to the tests we aimed to perform. The first dataset is the `'M3 Competition dataset'` [2], and the remaining three are taken from the UCR Classification Archive [3], which contains a wide variety of time series classes. Specifically, we chose the `'Earthquakes'`, `'HouseTwenty'`, and `'Lightning7'` datasets.

We chose these datasets because of the specific types of time series they contain. The `'M3 Competition dataset'` is well-known for its use in forecasting evaluation, and presents short time series on which we can run tests quickly. An example of a time series in this dataset is presented in the appendix B, figure 7a . We also selected the three other datasets from the UCR Archive because they offer unique characteristics in time series data. The `'Earthquakes'` dataset features time series with abrupt, distinct patterns that recur irregularly, marking earthquake events (see appendix B fig. 7b). The `'HouseTwenty'` dataset shows time series with significant fluctuations in temperament, alternating between low steady values and sharp rectangular oscillations (see appendix B fig. 7c). Lastly, the `'Lightning7'` dataset doesn't present clear patterns or seasonality but includes rare lightning strike events (see appendix B fig. 7d).

These datasets display different intrinsic characteristics of time series, providing a great and varied test set for the proposed ABBA-LSTM model. This will allow us to see where the model performs well and where it doesn't, as well as test its sensitivity to hyperparameters. We intentionally selected different types and classes of time series, expecting to find different optimal hyperparameters for each class but similar or close hyperparameters for time series within the same class.

4 Results

We conducted a variety of tests on the selected datasets to evaluate the performance and limitations of the proposed ABBA-LSTM model. Our experiments focused on four key aspects: assessing the model's lower sensitivity to hyperparameters, investigating the relationship between the performance of the ABBA representation and forecasting accuracy, investigating the relationship between the accuracy of the ABBA-LSTM model on the trainig set and forecasting accuracy and analyzing how the model performs across time series from different classes with diverse characteristics.

4.1 Sensitivity to Hyperparameters

In the paper, the authors examine the sensitivity of the ABBA-LSTM and raw LSTM models to initialization weights and the frequency of a periodic time series. Building on this, we designed a new experiment to demonstrate that the ABBA-LSTM model is also less sensitive to hyperparameters such as the lag l and max_k. For this experiment, we performed a grid search on the first time series of each of the four datasets to determine the optimal hyperparameter combinations. These selected combinations were then applied to test the remaining time series within each dataset.

Figure 1 presents the results of this experiment on the `'Earthquakes'` dataset. The ABBA-LSTM model demonstrates lower sensitivity to hyperparameters, as shown by DTW values remaining consistently close to 20. In contrast, the raw LSTM model shows a wide and inconsistent variation in DTW values, ranging between 20 and 80. This indicates that, for time series belonging to the same class, the ABBA-LSTM model is more generalizable and performs effectively with a fixed set of hyperparameters. More generally, and for the other datasets, we observe that ABBA-LSTM values are usually lower than raw LSTM values with a few oulier exceptions.

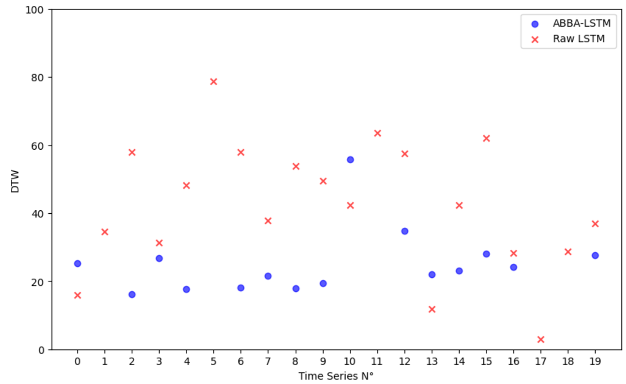


Figure 1: Experiment comparing the accuracy of raw LSTM and ABBA-LSTM on 20 time series of the `'Earthquakes'` dataset after a hyperparameter grid search on the first time series of the dataset.

Adding on these quantitative results, we can also compare qualitative outcomes by examining the forecasting performance of both the ABBA-LSTM and raw LSTM models. In the figures presented in the appendix C, we can see that the first set of figures (8a, 8c, 8e, 8g) illustrates the forecasting results on the first time series of each dataset, where the grid search was conducted. The other set of figures (8b, 8d, 8f, 8h) shows forecasting results using the same hyperparameters but applied to different series of the same class. These comparisons reveal that the ABBA-LSTM model is less sensitive, generalizes better, and performs more effectively overall.

4.2 Correlation between ABBA's performance and forecasting performance

Another experiment we conducted aimed to test whether there was a correlation between the performance of the symbolic representation of the time series and the forecasting accuracy. To do this, we measured the DTW distance between the forecasted data and the expected data, as well as the DTW distance between the ABBA reconstruction of the series and the original series, across 20 time series for each of the four datasets. We expected a positive correlation, meaning that better reconstruction by the ABBA method would lead to more accurate forecasts. However, our results showed no correlation between these two values, indicating that the performance of the symbolic representation did not significantly impact forecasting accuracy. For example, Figure 2 shows the results for both `'HouseTwenty'` and `'Earthquakes'` dataset. We can see that the DTW of the forecasts aren't really influenced by the DTW of the reconstruction. The plot appears mostly horizontal, with a few outliers, but these outliers don't seem to occur with higher DTW reconstruction values either.

This observation might be explained by noting that, in general, the ABBA symbolic representation is quite effective and able to reconstruct the series well. In fact, the average value of DTW over the energy of the signal for the reconstructed series is very small as shown in the table 1, indicating a good overall reconstruction. In other words, the DTW values are still within an acceptable range compared to the magnitude of the signals. It's possible that with much larger DTW/Energy ratio values for the reconstruction, we might begin to see a correlated decline in forecasting accuracy, but in these experiments and with this data, that is not the case.

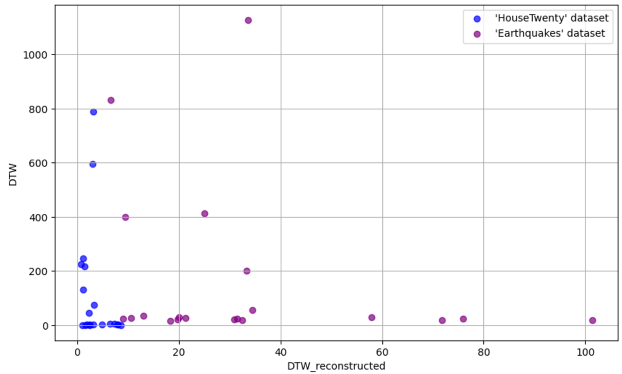


Figure 2: Experiment comparing the DTW distance of forecasts and reconstructions of 20 time series of the `'HouseTwenty'` and `'Earthquakes'` datasets.

Dataset	`'HouseTwenty'`	`'Earthquakes'`	`'Lightning7'`	`'M3 Dataset'`
Mean of DTW/Energy	$2,1.10^{-3}$	$6,4.10^{-2}$	$7,4.10^{-3}$	$1,5.10^{-2}$

Table 1: Mean of DTW Distances over energies of Reconstructed Series for Different Datasets

4.3 Performance of ABBA-LSTM on different classes of time series

The figure 3 compares the performance of the ABBA-LSTM model across the four different datasets we're using. The results show that the model doesn't perform equally on all types of datasets and time series. The best performance is observed on the `'HouseTwenty'` dataset, which has the lowest median value. The model also shows similarly strong performances on the `'Earthquakes'` and `'Lightning7'` datasets, with a smaller variance. Finally, the model performs the least effectively on the `'M3 Dataset'`.

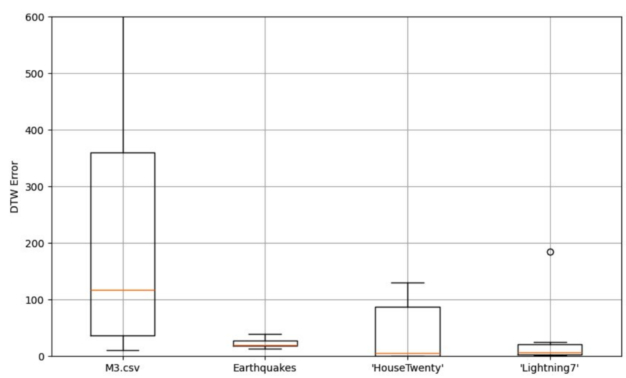
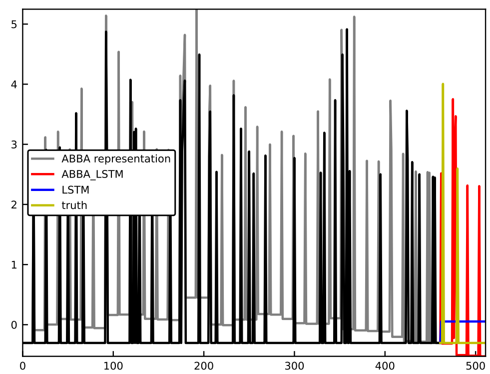


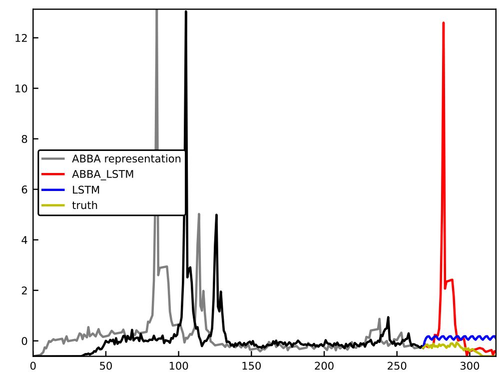
Figure 3: Comparison of the performance of the ABBA-LSTM Model on our 4 datasets.

We must now consider what factor determines the strong performance of the ABBA-LSTM model in forecasting time series. Based on the results we obtained, we believe the answer lies in the type, and more specifically the shape, of the time series we're trying to forecast.

The ABBA-LSTM model forecasts time series by representing the series as symbols from a finite alphabet $\mathbb{A}$ and predicting the next symbols within this same alphabet. This design ensures that the model always predicts a shape resembling the initial time series, following the same patterns it was trained on. As a result, the ABBA-LSTM model performs better on time series that exhibit some form of periodicity or repetitive patterns, such as those in the 'Earthquakes' or the 'HouseTwenty' dataset. For these types of time series, the model effectively predicts abrupt changes but struggles to accurately predict their frequency of occurrence as seen in figure 4a. Similarly, on the 'Lightning7' dataset, the model generally performs well but occasionally predicts rare lightning strike events by simply replicating the exact patterns it learned during training as seen in figure 4b.



(a) Forecast of time series N16 of the 'Earthquakes' dataset using the raw LSTM and ABBA-LSTM models.



(b) Forecast of time series N10 of the 'Lightning7' dataset using the raw LSTM and ABBA-LSTM models.

Figure 4: Forecasting vunlerabilities of the ABBA-LSTM Model.

Figure 5 illustrates the training set accuracy of the ABBA-LSTM model for two datasets: 'Lightning7' and 'M3 Dataset'. These results help explain why the model performs less effectively on the 'M3 Dataset' compared to the other datasets. In fact, the training set accuracy for the 'M3 Dataset' is lower, indicating that it is more challenging to predict. Additionally, the small size of the training set for the M3 dataset contributes to lower-quality predictions on the test set. This thus lead to poorer predicitions quality overall.

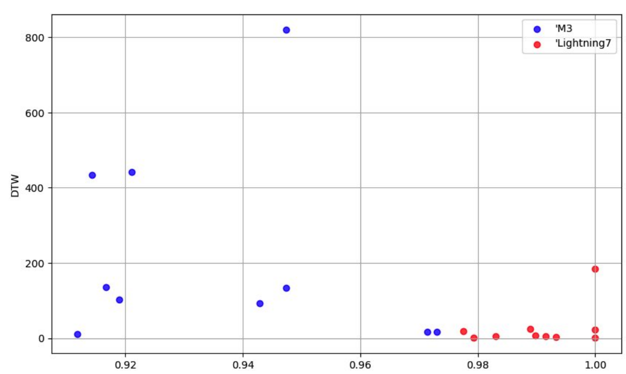


Figure 5: Experiment comparing the accuracy of the ABBA-LSTM Model on the training set for the 'Lightning7' and 'M3' datasets.

References

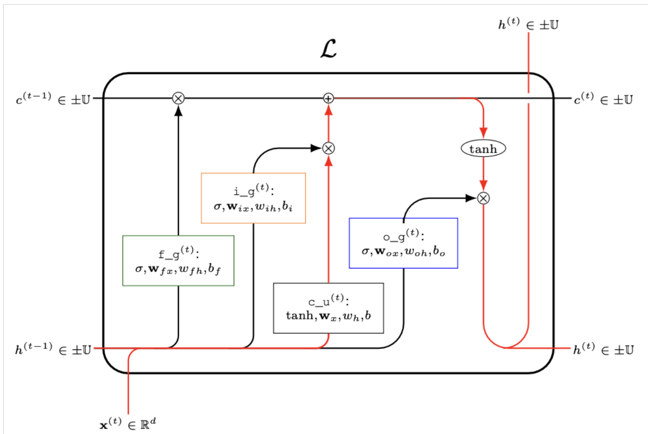
[1] S. Elsworth and S. Güttel, *Time Series Forecasting Using LSTM Networks: A Symbolic Approach*, arXiv preprint arXiv:2003.05672, 2020. [Online]. Available: https://doi.org/10.48550/arXiv.2003.05672.

[2] International Institute of Forecasters, "M3 Competition Dataset." [Online]. Available: https://forecasters.org/resources/time-series-data/m3-competition/. Accessed: Jan. 8, 2025.

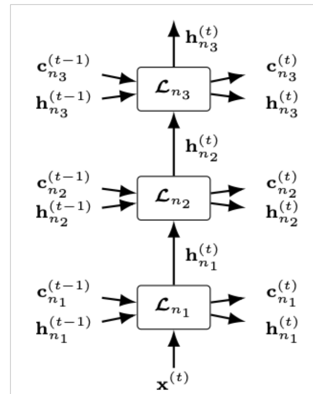
[3] H. A. Dau, E. Keogh, K. Kamgar, C.-C. M. Yeh, Y. Zhu, S. Gharghabi, C. A. Ratanamahatana, Yanping, B. Hu, N. Begum, A. Bagnall, A. Mueen, and G. Batista, "The UCR time series classification archive," Oct. 2018. [Online]. Available: https://www.cs.ucr.edu/~eamonn/time_series_data_2018/.

Appendices

A Architecture of a Multi-Layer LSTM Network



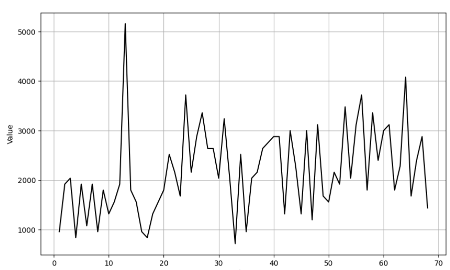
(a) The architecture of a single LSTM Cell



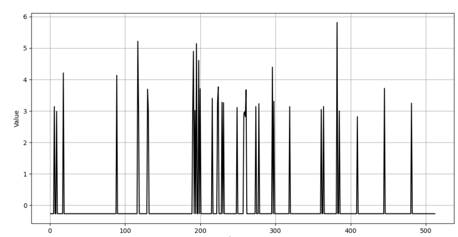
(b) Multi-Layer LSTM

Figure 6: Multi-Layer LSTM Architecture - figures are taken from the paper [1].

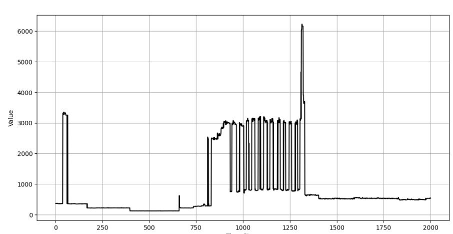
B Time series from the different datasets



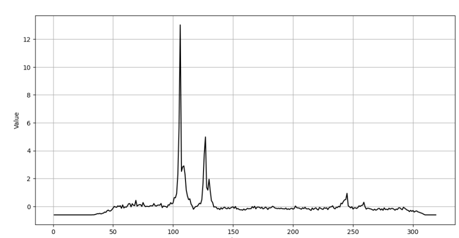
(a) Plot of a time series in the M3 Dataset



(b) Plot of a time series in the 'Earthquakes' Dataset



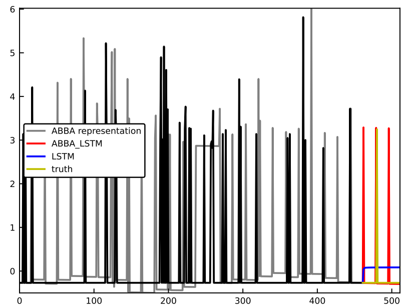
(c) Plot of a time series in the 'HouseTwenty' Dataset



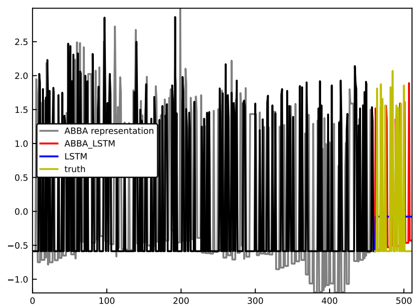
(d) Plot of a time series in the 'Lightning7' Dataset

Figure 7: Plots of time series from various datasets highlighting the characteristics of each class.

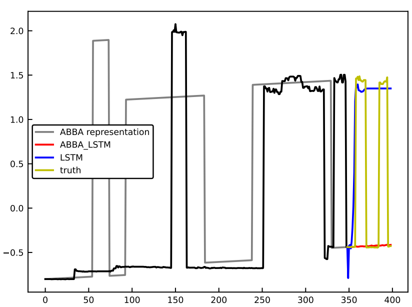
C Forecasting results



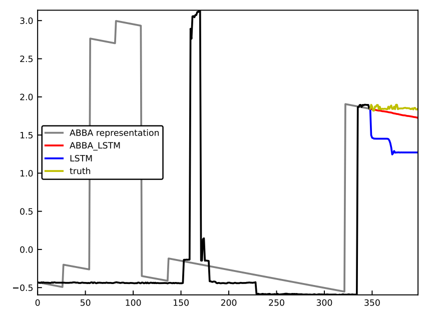
(a) Time series N0 from `'Earthquakes'`.



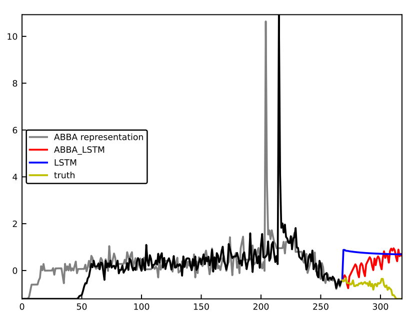
(b) The time series N6 from `'Earthquakes'`.



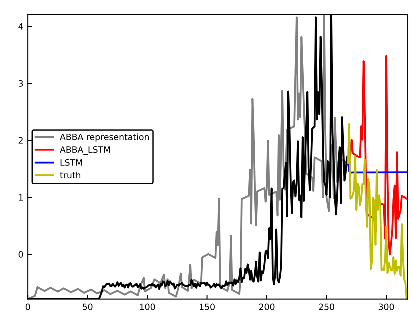
(c) Time series N0 from `'HouseTwenty'`.



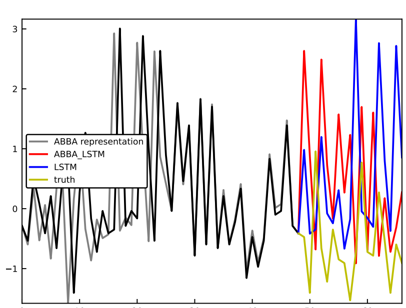
(d) Time series N4 from `'HouseTwenty'`.



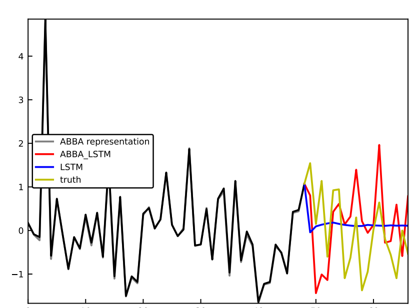
(e) Time series N0 from `'Lightning7'`.



(f) Time series N17 from `'Lightning7'`.



(g) Time series N0 from `'M3 Dataset'`.



(h) Time series N14 from `'M3 Dataset'`.

Figure 8: Forecasts of different time series using the raw LSTM and ABBA-LSTM models.