Project (ML for Time Series) - MVA 2022/2023
Time series forecasting using symbolic approach

Paul CHAUVIN paulchauvin97@gmail.com
Julien POURCEL j.pourcel31830@gmail.com

March 27, 2023

1 Introduction and contributions

Time series forecasting is a critical task in many applications such as finance, supply and demand prediction, and health monitoring. Traditional statistical methods and machine learning models are commonly used for this task. Recurrent neural networks (RNNs) are a type of machine learning model that is often used for time series forecasting. However, RNNs can have limitations, including long training times as they are not really parallelizable and they have high sensitivity to hyperparameters.

In this context, a hybrid algorithm called ABBA-LSTM [5, 6] has been proposed to address some of these limitations. ABBA-LSTM combines the ABBA symbolic representation for time series with the LSTM neural network architecture [7] to reduce the training time and sensitivity to hyperparameters. The ABBA symbolic representation is a dimension-reducing technique that allows for faster processing of time series data. The LSTM architecture is a type of RNN that can capture long-term dependencies in time series data.

In this project, we aim to reproduce and extend the results of the ABBA-LSTM paper. Specifically, we will implement the ABBA-LSTM model and evaluate its performance on a variety of time series datasets. Additionally, we will explore ways to further improve the ABBA-LSTM model, such as modifying the LSTM architecture.

The contributions of this project will be two-fold. First, we will demonstrate the effectiveness of ABBA-LSTM as a method for time series forecasting, with our own reimplementation from scratch of the ABBA-LSTM, however as is the case in the original ABBA-LSTM paper we are using the official ABBA implementation to transform a times series into a symbolic representation. Second, we will identify potential ways to improve ABBA-LSTM and we will also conduct new experiments, such as applying the method to novel datasets.

2 Method

2.1 Adaptive Brownian Bridge-based Aggregation (ABBA) representation

Symbolic representation is a popular approach for analyzing time series data in the data mining community. It involves converting numerical time series data into a sequence of symbols, which are elements of a finite alphabet. This symbolic representation has been shown to be useful in a range of applications, including classification, clustering, motif discovery, and anomaly detection.

One approach to symbolic representation is the ABBA method, which stands for adaptive Brownian bridge-based aggregation. This method involves two stages: compression and digitization.

In the compression stage, an adaptive piecewise linear approximation of the time series is constructed. The algorithm selects m+1 indices $i_0 = 1 < i_1 < ... < i_m = N$ such that the time series is partitioned into m pieces$P_j = [t_{i_{j-1}}, t_{i_{j-1}+1}, ..., t_{i_j}]$, j=1,2,...,m where m depend on a tolerance parameters. On each piece P_j, the time series is approximated by a straight line through the endpoint values, represented by the tuple $(len_j, inc_j) \in R^2$, where $len_j = i_j - i_{j-1} \geq 1$ and $inc_j = t_{i_j} - t_{i_{j-1}}$. The sequence of tuples $(len_1, inc_1), ..., (len_m, inc_m)$ and the first value t1 represent a polygonal chain going through the points (ij, t_{ij}) for $j = 0, 1, ..., m$. This compression stage results in a polygonal chain that represents the time series with fewer points.

In the digitization stage, the tuples (len_j, inc_j) are grouped into k clusters using a mean-based clustering algorithm, with each cluster assigned a symbol from the alphabet. Converting back to a numerical representation requires three stages: inverse-digitization, quantization, and inverse-compression. The inverse-digitization stage represents each symbol by the center of the corresponding cluster, resulting in a sequence of tuples. The quantization realigns the accumulated lengths of the tuples with an integer grid. Finally, the inverse-compression stage stitches the linear pieces represented by each tuple to obtain raw time series values.

2.2 RNN

LSTM networks are an interesting and powerful type of recurrent neural network that can effectively capture long-term dependencies in time series data. However, their training time can be slow and they are highly sensitive to hyperparameters, which can make them difficult to optimize. The ABBA symbolic representation can be useful in reducing the dimensionality of time series data, which can in turn speed up training time and reduce sensitivity to hyperparameters. By combining the ABBA symbolic representation with the LSTM architecture, we can potentially create a more efficient and effective time series forecasting model.

While LSTM networks are a popular choice for time series forecasting, recent research has shown that gated recurrent unit (GRU) [2] networks can also perform well on this task [3, 1]. In addition, GRU networks have fewer parameters than LSTM networks and are therefore easier to train and less prone to overfitting. Therefore, it may be worth exploring the use of GRU networks in combination with the ABBA symbolic representation for time series forecasting. By comparing the performance of LSTM and GRU networks in combination with ABBA, we may be able to identify the best architecture for this specific application.

2.3 Data

In this project, we used a subpart of the UCR dataset to perform time series forecasting using only one signal at a time. The first n values were used to train our models (where $n = 800$ or $n = 1200$), while we used the 200 following values for testing. To ensure that the algorithm worked correctly, we normalized the time series data to have zero mean and unit variance, as it is an assumption of ABBA algorithm. We could have performed more preprocessing, for example detrending, however, we wanted to show the advantage of the ABBA representation.

2.4 Experiments

We did several experiments and tried 4 models for each experience. The first two were ordinary LSTM and GRU with a context size of 40 and they predicted the following values t_{41} =RNN$(t_1, ...t_{40})$. We used a stateless context which means that we can extract some random patch for the training as the initial states of the RNNs are initialized to zero and it was trained using a mean square error loss. Then, we tested the ABBA-LSTM and the ABBA-GRU with a context size of 5 in the same fashion as the original paper and it was trained using a cross-entropy loss. We used cross entropy as the raw data are transformed into a sequence of letters (with the ABBA transform) then each letter is converted with a one-hot encoding transform. We choose for the ABBA transform a tolerance of 0.05 and a dictionary of 10 letters. For all those architectures we used a batch size of 32, a hidden size of 32, and 2 layers. We trained all architectures for 150 epochs with the AdamW [8] optimizer with a learning rate of $5e-3$ and a weight decay of 0.01 to limit overfitting. The first experiment is on 10 datasets referred to as "medium" datasets. Where we trained all models on the first 1200 values and we tested our models on the last 200 values. Then we did another experiment on 15 datasets referred to as "small" datasets, we trained all models on the first 800 values and we tested our models on the last 200 values.

model name	ABBA-LSTM	ABBA-GRU	LSTM	GRU
ABBA-LSTM	15	7	10	10
ABBA-GRU	8	15	10	10
LSTM	5	5	15	9
GRU	5	5	6	15

Table 1: Shows the number of times each model outperformed the others based on the MSE metric on "small" dataset (for example LSTM outperforms 10 times ABBA-LSTM), see Table 2 for more details.

3 Results

In this section, we present the results of our experiments with the ABBA-LSTM and ABBA-GRU models, as well as traditional LSTM and GRU models, on a variety of UCR datasets [4]. The results are summarized in Table 2 and Table 3, where the performance of each model is evaluated based on its MSE, sMAPE, and DWT.

	MSE				sMAPE				DTW			
Dataset	ABBA-LSTM	ABBA-GRU	LSTM	GRU	ABBA-LSTM	ABBA-GRU	LSTM	GRU	ABBA-LSTM	ABBA-GRU	LSTM	GRU
ACSF1	0.564	0.564	0.045	0.044	39.108	38.500	13.159	12.173	2.036	2.012	1.499	1.393
CinCECGTorso	259.7	262.0	307.5	293.3	119.699	116.216	129.457	134.753	198.275	201.907	218.609	215.421
EOGHorizontalSignal	0.54	17.77	2.852	2.670	32.835	68.000	39.354	38.245	2.830	41.128	12.570	12.074
EOGVerticalSignal	0.424	0.416	1.959	1.201	38.192	38.048	57.538	41.767	2.133	1.874	10.764	9.093
EthanolLevel	1.629	0.725	0.354	0.399	87.328	78.240	73.127	79.283	13.942	8.103	4.978	7.150
Mallat	0.621	1.502	0.114	0.143	92.394	108.078	74.857	85.934	7.710	13.919	3.595	4.023
MixedShapesRegularTrain	0.688	0.999	1.349	1.677	56.509	54.746	80.195	75.237	5.048	4.153	10.093	8.422
MixedShapesSmallTrain	1.444	1.666	1.116	1.176	147.371	147.440	155.349	144.955	11.445	12.810	11.041	11.772
Phoneme	24.91	11.77	15.23	16.88	80.879	73.265	46.680	95.685	46.651	29.115	34.809	39.862
PigAirwayPressure	0.590	0.625	0.106	0.082	96.070	99.793	30.408	31.230	5.618	4.284	1.948	1.434
PigArtPressure	1.467	2.227	0.923	0.328	99.153	121.581	83.882	60.672	11.863	13.131	4.786	4.049
PigCVP	0.574	0.292	0.706	0.646	39.335	34.219	46.485	49.734	4.765	4.358	7.964	7.123
Rock	41.03	43.54	0.381	0.398	166.366	174.342	148.617	148.835	56.338	73.436	8.000	8.334
SemgHandGenderCh2	3.952	3.039	0.406	0.386	148.024	148.280	147.289	135.886	19.694	15.002	8.392	8.139
SemgHandMovementCh2	5.095	2.561	0.621	0.597	147.211	146.626	146.698	139.979	19.978	13.809	10.789	10.744

Table 2: Errors on "small" dataset with 800 training values and testing on the last 200 values

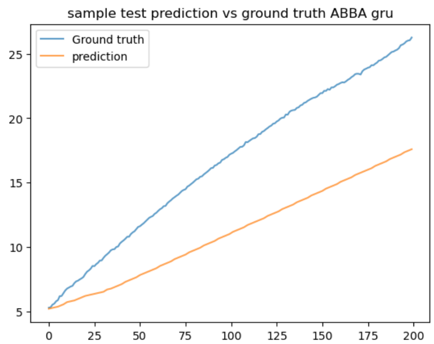
Our experiments presented in Table 2 on the "small" dataset setting, show that the traditional LSTM and GRU model outperforms the ABBA-LSTM and ABBA-GRU models on several datasets, including ACSF1, EOGVerticalSignal, and EthanolLevel. Interestingly, the ABBA-GRU and ABBA-LSTM models show competitive performances on some datasets, such as EOGVerticalSignal, MixedShapesRegularTrain and PigCVP.

	MSE				sMAPE				DTW			
Dataset	ABBA-LSTM	ABBA-GRU	LSTM	GRU	ABBA-LSTM	ABBA-GRU	LSTM	GRU	ABBA-LSTM	ABBA-GRU	LSTM	GRU
ACSF1	0.990	0.985	0.050	0.048	37.69	36.37	14.82	6.314	5.764	5.667	1.244	1.077
CinCECGTorso	0.175	0.218	0.105	0.092	77.89	78.23	56.01	61.55	3.436	4.287	2.213	1.687
EOGHorizontalSignal	0.590	0.442	0.060	0.068	31.96	29.65	11.76	11.99	8.427	7.514	2.817	2.779
EOGVerticalSignal	0.692	0.687	0.517	0.452	74.15	68.87	81.59	65.01	8.579	8.335	7.319	6.431
EthanolLevel	0.840	0.905	0.050	0.175	108.94	109.64	27.98	39.76	6.878	6.287	0.788	1.666
Mallat	2.030	1.568	0.340	0.406	110.59	123.2	75.71	92.55	12.94	11.90	4.701	5.588
MixedShapesRegularTrain	1.224	0.804	0.699	0.593	55.94	57.18	48.44	42.96	8.179	6.832	6.347	5.929
MixedShapesSmallTrain	16.49	12.34	0.259	0.259	150.67	144.0	76.88	75.97	40.01	33.53	6.452	6.378
Phoneme	2.979	6.246	0.305	0.310	134.37	138.2	95.68	99.28	16.48	24.55	7.128	7.144
PigAirwayPressure	18.30	17.91	0.339	0.343	149.55	151.7	95.12	100.61	42.14	41.03	7.508	7.481

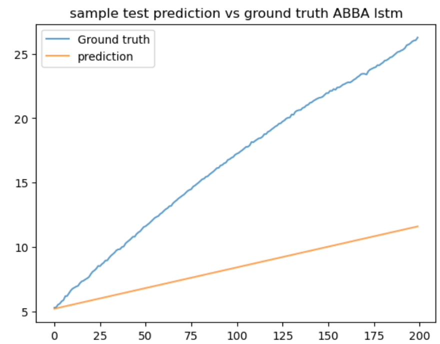
Table 3: Errors on "medium" dataset with 1200 training values and testing on the last 200 values

It is worth noting that the standard LSTM and the GRU model consistently outperform ABBA models on all datasets in the "medium" dataset setting where they have more data to be trained on, as shown in Table 3. This means that those original models without the ABBA transform have better scaling in terms of data than ABBA models.

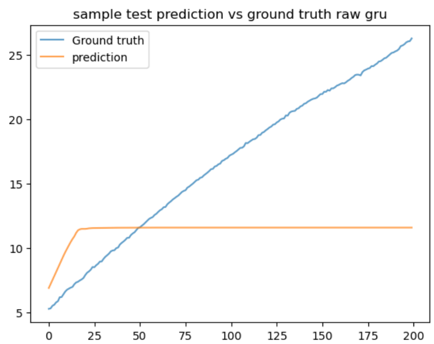
Overall, our experiments show that original RNNs models outperform ABBA symbolic representation technique with the RNNs architecture. However, The ABBA-LSTM model also demonstrates the potential to achieve higher accuracy on certain types of time series data compared to traditional LSTM and GRU models especially when we have a limited number of data.



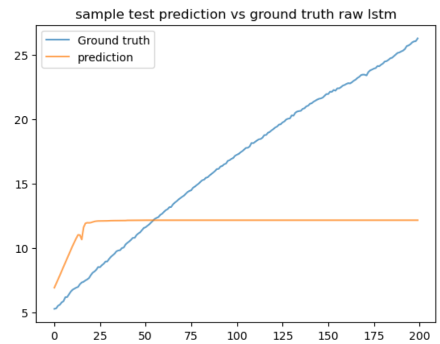
(a) ABBA-GRU



(b) ABBA-LSTM



(c) GRU



(d) LSTM

Figure 1: Prediction of our model on PigAirwayPressure dataset

Moreover, when analyzing a dataset exhibiting a trend pattern such as the one shown in Figure 1 (PigAirwayPressure dataset), it appears that the use of ABBA-enhanced RNN produces superior qualitative results in comparison to the RNN model without ABBA. Other similar results are exposed in Figure 2 in the Appendix section, for the Mallat dataset.

4 Conclusion

In this project, we aimed to reproduce and extend the results of the ABBA-LSTM paper. Specifically, we implemented the ABBA-LSTM model and evaluated its performance on a variety of time series datasets. Additionally, we explored ways to further improve the ABBA-LSTM model, such as modifying the LSTM architecture.

Our contributions are two-fold. First, we demonstrated the effectiveness of ABBA-LSTM as a method for time series forecasting, with our own reimplementation from scratch of the ABBA-LSTM, however as is the case in the original ABBA-LSTM paper we used the official ABBA implementation to transform a times series into a symbolic representation. Second, we identified potential ways to improve ABBA-LSTM and also conducted new experiments, such as applying the method to novel datasets.

Overall, our findings suggest that ABBA-LSTM can be a promising approach to time series forecasting, with potential benefits in terms of reduced training time and improved performance. Further research is needed to explore the full potential of this hybrid algorithm and to identify ways to further improve its effectiveness.

References

[1] Roberto Cahuantzi, Xinye Chen, and Stefan Güttel. A comparison of lstm and gru networks for learning symbolic sequences. *arXiv preprint arXiv:2107.02248*, 2021.

[2] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[3] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[4] Hoang Anh Dau, Anthony Bagnall, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh Gharghabi, Chotirat Ann Ratanamahatana, and Eamonn Keogh. The ucr time series archive. *IEEE/CAA Journal of Automatica Sinica*, 6(6):1293–1305, 2019.

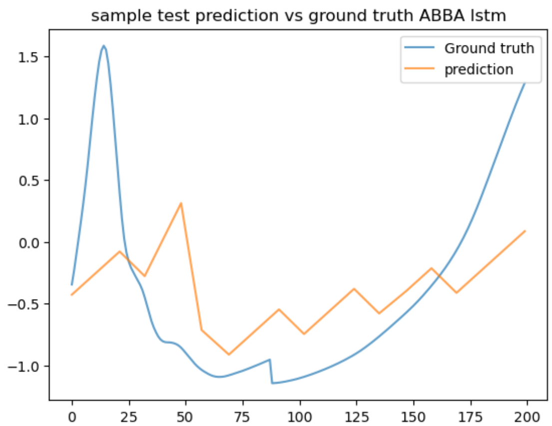
[5] Steven Elsworth and Stefan Güttel. Abba: Adaptive brownian bridge-based symbolic aggregation of time series. *Data Mining and Knowledge Discovery*, 34(4):1175–1200, 2020.

[6] Steven Elsworth and Stefan Güttel. Time series forecasting using lstm networks: A symbolic approach. *arXiv preprint arXiv:2003.05672*, 2020.

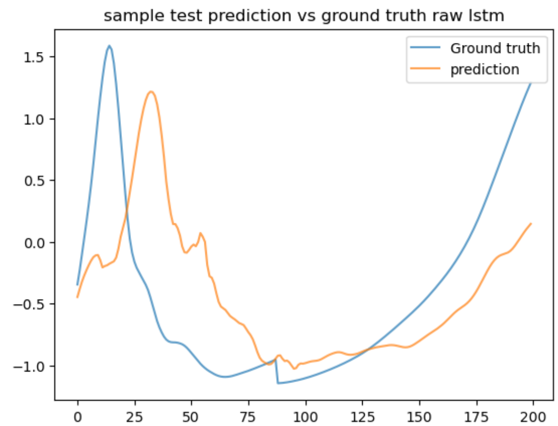
[7] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[8] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

A Appendix



(a) ABBA-GRU



(b) ABBA-LSTM

Figure 2: Prediction of our model on Mallat