COMP6248 REPRODUCIBILITY CHALLENGE: MIXSEQ: CONNECTING MACROSCOPIC TIME SERIES FORECASTING WITH MICROSCOPIC TIME SERIES DATA

Amisha Singh, Lisandra Madero Lorenzo, Reshma Abraham & Oliver Schamp [*]
Department of Electronics and Computer Science
University of Southampton
{as6u21, lcml1n21, ra2n21, oras1u18}@soton.ac.uk

ABSTRACT

This report analyses and describes the attempt to reproduce the paper, *MixSeq: Connecting Macroscopic Time Series Forecasting with Microscopic Time Series Data*. Under the assumption that macroscopic time series follow a mixture distribution, they hypothesise that due to the lower variance of constituting latent mixture components, forecasting on microscopic time series could improve the estimation of macroscopic time series. We learned the challenges of re-implementing the proposed model, and as a result, we developed our own implementation based on this conjecture to prove its validity.

1 INTRODUCTION

Macroscopic time series are an aggregation of microscopic time series. While there are many resources on macroscopic time series analysis, few papers consider leveraging time series data on a microscopic level, where a macroscopic time series composed of *m* microscopic components would be given as $x_t = \sum_{i=1}^{m} x_{i,t}$ where $x_{i,t}$ ϵ R. *"MixSeq: Connecting Macroscopic Time Series Forecasting with Microscopic Time Series Data"* aims clustering the time series with an assumption that macroscopic time series are generated from probabilistic models with *'K'* components and demonstrates the benefit of this clustering in time series forecasting.

Assuming a mixture model with probability density function f(x) and corresponding components $\{f_i(x)\}_{i=1}^{K}$ with constants $\{p_i\}_{i=1}^{K}$, f(x) can be given as $\Sigma_i p_i f_i(x)$. In condition that $f(\cdot)$ and $\{f_i(\cdot)\}_{i=1}^{K}$ have first and second moments, i.e., $\mu^{(1)}$ and $\mu^{(2)}$ for f(x), and $\{\mu_i^{(1)}\}_{i=1}^{K}$ and $\{\mu_i^{(1)}\}_{i=1}^{K}$ for components $\{f_i(x)\}_{i=1}^{K}$, the variance relation between the microscopic and macroscopic time series can be written as:

$$\sum_i p_i \cdot Var(f_i) \leq Var(f) \tag{1}$$

Equivalently, the variance of aggregation of clustered data should be less than or at most equal to the variance of the macroscopic data. Resultantly, optimal clustering should lead to decreased time series variance. If predictions can be done on the less variable clusters separately, this reduces randomness and increases stability in the time series we try to forecast. Using this as the motivation, the authors attempted to predict macroscopic time series by clustering the underlying microscopic time series.

The authors proposed a mixture of Seq2Seq (MixSeq) architecture modelling macroscopic time series as K mixtures of Gaussians, where each microscopic time series present was generated from this K-component probabilistic mixture model. Additionally, they conducted experiments on synthetic data (generated by ARMA and DeepAR) to demonstrate the superiority of the approach and extended the analysis of the performance to real-world datasets.

[*] There were no existing codes available for the reviewed publication.

2 Experiment Methodology and Implementations

We made two attempts at reproducing the paper. The first attempt targeted implementing the exact model as defined by the authors, however it was rendered futile due to some missing key details of implementation. The second attempt was aimed at verifying the hypothesis of the paper using a Variational Recurrent Auto-Encoder (VRAE) as the backbone of the clustering model.

2.1 First Attempt

A Multihead Attention (Vaswani et al. (2017)) was implemented following the paper's architecture. The classic attention model was modified by adding a causal convolution layer, ConvTrans (Li et al. (2019)) to ensure that the current position never has access to the feature information. This causal convolution computes the query similarities with local context information instead of point-wise values, enabling more precise forecasting. The model produced a generative probability for each microscopic time series. As the log marginal likelihood for the case was intractable, another ConvTrans model was used to approximate the posterior inference. The latent representation generated enabled the clustering of the time series. The transformer iterated the self-attention layer L times, where the input was passed through a Softmax function to project the encoding to K cluster dimensions. Each microscopic time series x_i was then assigned to cluster $z_i = \arg\max_{z} q(z|x_i)$.

The authors evaluated the clustering capability of the MixSeq on data synthesised using DeepAR (Flunkert et al. (2017)) and ARMA (George E. P. Box (2015)). In synthetic data generation, our initialise attempts were directed towards generating data using the DeepAR approach. For the DeepAR data, we adapted the model implementation mentioned in the DeepAR paper. To generate the first cluster, DeepAR was modified with one LSTM layer and 16 hidden units and trained on a real-world dataset, Wiki (Tran et al. (2021)). Two other components were created by introducing random disturbance to the parameters of the trained model by adding a Gaussian layer to it.

2.2 Implementation Bottlenecks

The paper fails to provide the implementation details required for training the transformer and generating synthetic data with DeepAR for model evaluation. More Specifically, we found that the implementation of the paper was not reproducible due to the following uncertainties in finer details:

- **Posterior inference using ConvTrans**
  For posterior inference, the paper proposes passing a linear transform of latent representation $h_t = v(H^{(L)})$ through a Softmax function. However, it was not clear how $H^{(L)}$ was generated since a part of the implementation generated $H^{(L)}$ as samples from the feedforward function $H^{(L)} \sim g(H^{(0)})$ while at times they also generated it using direct function composition as $H^{(L)} = g(\rho(Y_{t_0}))$.
  Additionally, the paper develops two mutually exclusive ConvTrans based models to capture the generative probability of each series and the posterior probability of cluster given the series. The two models use distinct $\rho(\cdot)'s$, $g(\cdot)'s$ and $v(\cdot)'s$ with different parameters. Given that MixSeq only clusters time series and does not generate forecasts, the purpose of having an independent generative probability model was unclear. Moreover, the objective of generating sufficient statistics for generative probability of time series was not addressed in the paper either.
- **Generating data with Deep AR**
  The data generation using DeepAR was not reproducible due to a couple of bottlenecks. Firstly, the authors did not specify the epochs or other hyperparameters used to train DeepAR. Considering the size of the Wiki dataset and the DeepAR architecture, our learning did not converge and tuning the model proved infeasible given the duration of the coursework. Secondly, for each cluster the authors generated 10,000-time series from randomly initialised sequences. However, the paper failed to mention the range of these random sequences. Due to the above limitations, the data was generated using ARMA, another method utilised by the authors to synthesise data (Section 2.3).
- **Multihead Attention**
  The authors conducted several experiments to validate their hypothesis and evaluate the

performance. However, not all the parameter values used were described, and since the space complexity of self-attention grows quadratically with sequence length (Huang et al. (2018)), the model was not reproducible with the resources available.

Since the model mentioned in the paper was not reproducible, we decided to evaluate the validity of their key idea in Section 1. In essence, this implies verifying that the predictions from aggregated clusters of microscopic time series are closer to the ground truth compared to that from direct prediction on macroscopic time series data. The following section outlines the experiment methodology used to assess the plausibility of the underlying idea of MixSeq.

2.3 SECOND ATTEMPT

To test the hypothesis that the variance of the aggregated clustered data should be at most the variance of macroscopic data, we used a Variational Recurrent Auto-Encoder (VRAE) (Fabius et al. (2014)) to obtain the latent representation of microscopic time series data. We implemented the VRAE architecture with the encoder having one LSTM layer and a hidden layer of 90 hidden units, while the decoder comprised one LSTM layer and two fully connected layers.

The model was trained on synthetic data generated using ARMA (George E. P. Box (2015)). We experimented with a different number of clusters (between 2 and 5), each generated by components governed by the parameters [ϕ_1, ϕ_2]. For each experiment, the VRAE model was trained for 40 epochs with batch size 32, a train-test ratio of 0.9, a learning rate of 0.0005 and Adam as the optimiser. The model produced a latent representation of (200, 20). We then performed PCA on the latent vector to reduce it to two components for the convenience of plotting the vector after clustering. This was followed by K-means clustering yielding us clustered microscopic time series. Resultantly, we transformed *m* microscopic time series into *K* (2-5) clustered microscopic time series.

Following this, ARIMA (Kotu & Deshpande (2019)) was used to forecast the next 200 time steps of each clustered time series and the macroscopic data. The forecast obtained on the macroscopic time series and the aggregate of predictions on individual clustered time data were evaluated against the ground truth in our test dataset. The results on various cluster counts are shown in Section 3.

We build our models using Pytorch framework version 1.8.1. Python 3 was used for development, training, visualisation and evaluation. We used ECS GPU Compute Service to run DeepAR training for synthetic data generation. All the code implementations, generated synthetic datasets and the trained models along with their clustering plots are available in our Github repository.

The authors also conducted experiments using three real world datasets: Rossmann[1], Wiki (Tran et al. (2021)) and M5[2] to evaluate the performance of their architecture. Given the duration of the coursework and resources available, all of it could not be reproduced. We evaluated the model developed in our second attempt on the Wiki Dataset, the results of which are shown in Section 3

The paper emphasised the importance of choosing a proper number of clusters and suggested cluster number as a critical hyperparameter of MixSeq. To evaluate this we performed sensitivity analysis on the ARMA synthetic data. The results of our analysis are provided in Section 3.

3 RESULT AND EVALUATION

For our analysis on synthetic ARMA data, we trained the model on 1800 microscopic time series with 1000 timestamps and varied cluster numbers (between 2 and 5) for different iterations. Figure 1 shows the distortion for different number of clusters, with the ground truth being K=6. With optimal clustering in each case for synthetic dataset, we could reproduce the results indicated in the paper. Table 1 summarises the quality of the forecast results obtained from direct prediction on macroscopic time series and from aggregation of clustered predictions. We can see that superior results are obtained with the aggregation technique.

[1] https://www.kaggle.com/competitions/rossmann-store-sales/data

[2] https://www.kaggle.com/competitions/m5-forecasting-accuracy/data

The results hold for real-world dataset as well. Table 2 collates the errors in forecasting on Wiki dataset with K=5.

		corr	MAE	MAPE	minmax	MPE	RMSE
2 Clusters	**Macroscopic**	0.093	86.58	3.83	0.58	-0.809	105.43
	Microscopic	0.067	63.56	1.27	3.56	-1.027	80.47
3 Clusters	**Macroscopic**	0.083	97.869	4.858	0.636	-0.769	117.02
	Microscopic	0.0725	86.5841	3.832	0.579	-0.809	105.436
4 Clusters	**Macroscopic**	0.0925	97.03	4.785	0.634	-0.774	116.105
	Microscopic	0.079	86.584	3.832	0.5787	-0.809	105.44

Table 1: Comparison of forecasting results on synthetic dataset with different number of clusters

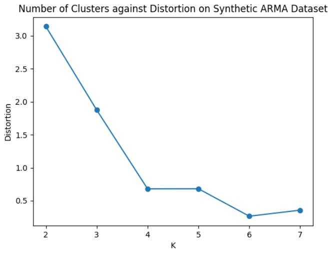

Figure 1: Sensitivity analysis on ARMA dataset generated with K=6

	MAE	MAPE	MPE	RMSE
Microscopic	67.89275	11.0578	-5.39836	89.182640
Macroscopic	71.22197	13.8463	-6.81540	93.082741

Table 2: Comparison of forecast results on Wiki dataset

4 Conclusion

We argue that the model described in the paper is not reproducible due to missing details of implementation. We could, however, successfully demonstrate the plausibility of the underlying concept on synthetic datasets and one real-world dataset. Experiments using multivariate data such as Rossmann and M5 should follow. Such experimentations have high computational and time requirements, hence the inability to reproduce them all within our constraints. Additionally, we confirmed the sensitivity of the model to the number of clusters chosen on the ARMA synthetic dataset.

We also found that the paper does not clearly indicate the reason for choosing a transformer (self-attention) as the model backbone and not some other Seq2Seq model such as LSTM or GRU. For our implementation, we used LSTM layers and were able to reproduce the results on synthetic dataset. While the theory of the paper holds water, the paper leaves a few fundamental questions unanswered.

References

Otto Fabius, Joost Amersfoort, and Diederik Kingma. Variational recurrent auto-encoders. 12 2014.

Valentin Flunkert, David Salinas, and Jan Gasthaus. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36, 04 2017. doi: 10.1016/j.ijforecast.2019.07.001.

Gregory C. Reinsel Greta M. Ljung George E. P. Box, Gwilym M. Jenkins. *Time Series Analysis: Forecasting and Control*. John Wiley Sons, 5 edition, 2015. ISBN 9781118674925.

Cheng-Zhi Huang, Ashish Vaswani, Jakob Uszkoreit, Noam Shazeer, Curtis Hawthorne, Andrew Dai, Matthew Hoffman, and Douglas Eck. An improved relative self-attention mechanism for transformer with application to music generation. 09 2018.

Vijay Kotu and Bala Deshpande. Chapter 12 - time series forecasting. In Vijay Kotu and Bala Deshpande (eds.), *Data Science (Second Edition)*, pp. 395–445. Morgan Kaufmann, second edition edition, 2019. ISBN 978-0-12-814761-0. doi: https://doi.org/10.1016/B978-0-12-814761-0.00012-5. URL https://www.sciencedirect.com/science/article/pii/B9780128147610000125.

Shiyang Li, Xiaoyong Jin, Yao Xuan, Xiyou Zhou, Wenhu Chen, Yu-Xiang Wang, and Xifeng Yan. *Enhancing the Locality and Breaking the Memory Bottleneck of Transformer on Time Series Forecasting*. 2019.

Alasdair Tran, Alexander Mathews, Cheng Soon Ong, and Lexing Xie. Radflow: A recurrent, aggregated, and decomposable model for networks of time series. In *Proceedings of the Web Conference 2021*. ACM, apr 2021. doi: 10.1145/3442381.3449945. URL https://doi.org/10.1145%2F3442381.3449945.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017. URL https://arxiv.org/abs/1706.03762.