Stock Market Prediction Using Deep Learning

Filipe Barnabé, João Ferreira

Abstract—The present document aims to predict the stock market using deep learning. This is a problem for a *Recurrent Neural Network (RNN)*, so *Bidirectional Long Short-Term Memory (BiLSTM)* was one of the approaches to solve the current problem based on the articles [1], [2] . However, since transformers are becoming more and more popular, and some times even said to be the solution for most deep learning problems, the second approach was based on *Transformers* with a *Attention is all you need* [3].

Keywords: Long Short-Term Memory (LSTM), Bidirectional Long Short-Term Memory (BiLSTM), Recurrent Neural Network (RNN), Artificial Neural Network (ANN), Standards and Poor's 500 (S&P 500), Machine Learning, Time Series Analysis, Transformer Model, Stock Price Prediction.

1 Introduction

The stock market has high volatility and high degree of unpredictability which by itself represents a big challenge for predicting its future. The data presented is collected from the yahoo finance API (yfinance) and the project revolved around the *S&P 500* index fund (*GSPC*). The Fig 1 shows the time series data of the closing price and the volume.

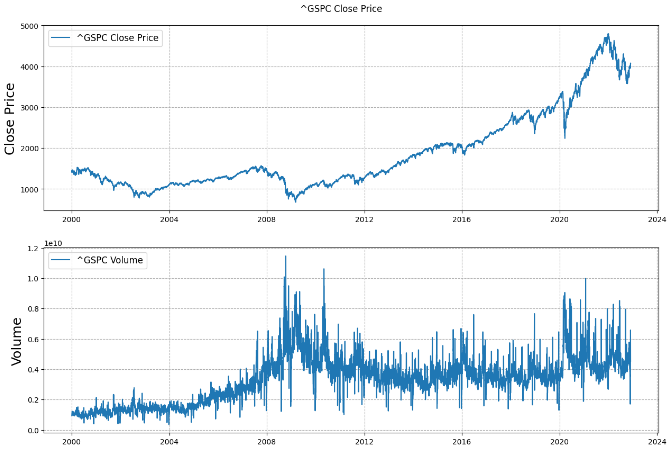


Fig. 1: S&P 500 closing price and volume dataset

This report approaches the problems of volatility and unpredictability through deep learning. There are two different approaches on this paper to solve this problem, Bidirectional Long Short-Term Memory (BiLSTM) and Transformers.

2 State of the Art

The Pipeline follows the Fig 2, however we will not talk about the deployment.

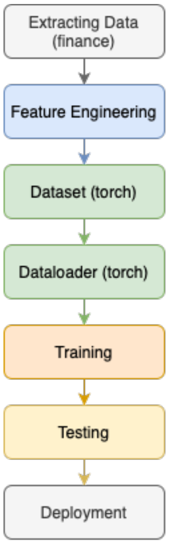


Fig. 2: Training-Testing Pipeline

The approach method is shared by both BiLSTM and Transformers, the only difference is on the creation of the Data loader (Batch size - Transformers is 32, BiLSTM is 64), and of course on the models themselves.

2.1 Feature Engineering

There where two approaches for this problem. In the first instance the data was used with only normalization. The second method applies the moving average of 10 days, before the normalization takes place.

The normalization returns the percentage of change in the data columns (high, low, open, close prices and the volume) normalized with min-max (0-1).

2.2 Transformers

Transformers have been gaining popularity since they've been showing outstanding results. The combination of self-attention [4], [5], parallelization and positional encoding [6] under one usually provides better results than BiLSTM. The model is constituted of:

- Embedding / Time to Vector Layer [6];
- Encoder Layer [5] / Encoder;
- Decoder.

2.2.1 Positional Encoding, Time2Vec

The time to vector layer is necessary to make sure the position isn't lost. This layer returns two new features, the time linear and the time periodic, the implementation was based on the paper **"Time2Vec: Learning a Vector Representation of Time"** [6].
Mathematically it is represent as Eq. 1.

$$\mathbf{t2v}(\tau)[i] = \begin{cases} \omega_i \tau + \varphi, if i = 0. \\ F(\omega_i \tau + \varphi_i), if 1 \leq i \leq k. \end{cases} \tag{1}$$

- k is the time2vec dimension
- τ is raw time series
- F is a periodic activation function
- ω and φ are set of learnable parameters

F is sine function based on the study presented on the paper [6] (Fig 3), its clear that the ReLU performs the worst and Sine outperforms any other.

To justify this positional encoding we can also take a look at some of the performance enhancements presented by the paper Fig. 4, it is possible to observe that it presents quite better results.

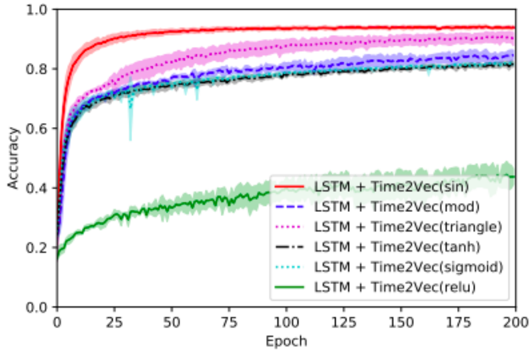
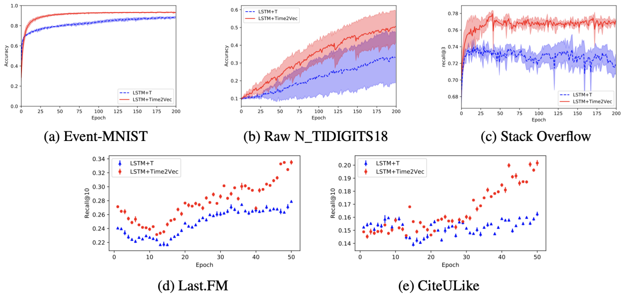


Fig. 3: Performance comparison of non-linear functions [6]



(a) Event-MNIST

(b) Raw N_TIDIGITS18

(c) Stack Overflow

(d) Last.FM

(e) CiteULike

Fig. 4: Performance comparison [6]

2.2.2 Transformer Encoder Layer

The transformer encoder layer is made of self attention and feed forward, based on **"Attention is all you need"** [5]. This allows to focus on the relevant parts of the time-series data. Self-attention consists of a single head attention and multi head attention, connecting all time series steps at once.

After implementation of the time embeddings (time2vec), as explained previously it will return the non periodic and the periodic time features, this features will be combined with the 5 previous features (open, high, low, close, volume) resulting in a total of 7 features.

2.3 LSTM & BiLSTM

Long Short-Term Memory (LSTM) and Bidirectional LSTM (BiLSTM) are types of Recurrent Neural Networks (RNN), designed to improve the long-term dependencies in sequential data. RNNs are a type of Neural Network that process sequential data, like time series data.

LSTMs where designed to address the vanishing gradient problem, a common issue in

RNNs where the information from the error function becomes very weak as it is propagated by a large amount of steps. LSTMs use a Memory Cell to better retain information for long periods of time, improving the long-term dependencies learning.

BiLSTMs are just LSTMs trained in both directions of the input sequence and the outputs from both directions are combined. This way the network can capture contextual information from the past and the future (of the input sequence).

LSTM Equations

$$\begin{aligned} i_t &= \sigma(W_{ii}x_t + b_{ii} + W_{hi}h_{t-1} + b_{hi}) \\ f_t &= \sigma(W_{if}x_t + b_{if} + W_{hf}h_{t-1} + b_{hf}) \\ g_t &= \tanh(W_{ig}x_t + b_{ig} + W_{hg}h_{t-1} + b_{hg}) \\ o_t &= \sigma(W_{io}x_t + b_{io} + W_{ho}h_{t-1} + b_{ho}) \\ c_t &= f_t \odot c_{t-1} + i_t \odot g_t \\ h_t &= o_t \odot \tanh(c_t) \end{aligned} \tag{2}$$

The BiLSTM implemented makes use of three layers, two being BiLSTM and one the Fully Connected which agglomerates all the information coming from the BiLSTM nodes and returns the pretended predictions.

3 Problem Description

Given an input of length n of days it will predict the $n + 1$ day, providing some insight on the stock. To chive our goals, we used the LTSMs and Transformers models, taking into consideration the following assumptions about the market:

- Isn't fully perfect;
- Historical data influence the future;
- Follows mostly rational behaviour;
- Has volatility;
- Very difficult to predict.

4 Experiments

The same tests were applied to both the Transformer [3], [5] and the BiLSTM [1], [2] with the same conditions, to be able to measure their performances.

The data is constituted of 8368 days with a train-validation-test split of 80%-10%-10%, being time series, we get first 80% of the data to the training, the following 80-90% to the validation and from 90-100% reserved for testing, Fig. 5.

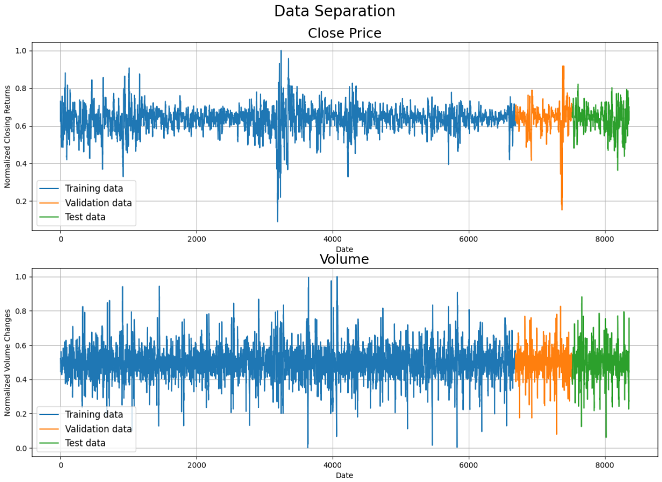


Fig. 5: Data Split

In both architectures, the sequence length is of size 128 and for training around 1000 epochs were used for Transformer networks, and 100 for BiLSTM networks. The Transformers and the BiLSTM took about 30 minutes.

Transformers generated 1 186 689 trainable parameters, and the BiLSTM generated 2 116 097.

As evaluation metrics, for the model criterion we selected *Mean Squared Error Loss (MSELoss)*. However we also evaluated the *Mean Absolute Error (MAE)* and the *Mean Absolute Percentage Error (MAPE)* in order to compare the performance of both models.

5 Experimental Results

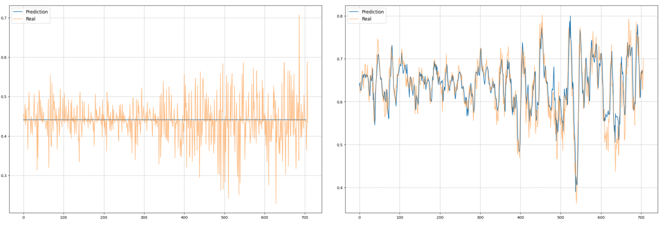
After training, both models passed through the test data in order to be evaluated, in the 5 the test data is the green part (the last 10 percent of the data).

To evaluate the differences, as presented above, we'll look at the loss, mean absolute error (MAE) and mean absolute percentile error (MAPE). Due to hardware limitations, we could only perform the last two for the Transformer model.

The speed up learning algorithm, the Adam optimizer was implemented in order to update the network weights. Adam employs two gradient based methodologies, Momentum and Evolution. This algorithm is a replacement for the gradient descent optimizer, although being gradient based.

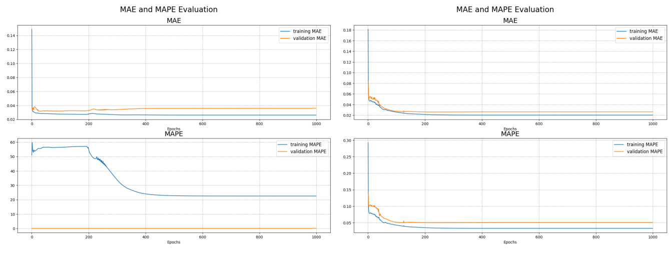
5.1 Transformer Results

Through the comparison of the results presented by the figures 6a and 6b, it is possible to observe that the data with moving average for the same number of epochs presents quite better results. This is backed by the figures 7a and 7b, that explain the data without the moving average would need more epochs to converge. It is also observed that at 200 epochs the model is not training anymore. The results of the evaluation stays the same until the 1000 epochs, concluding that we could use only 200 epochs to train the model the results of the moving average data are consistent throughout various runs, however, the results from the initial data show variability in the MAE and MAPE.



(a) no moving average (b) moving average

Fig. 6: Transformer Prediction vs Real

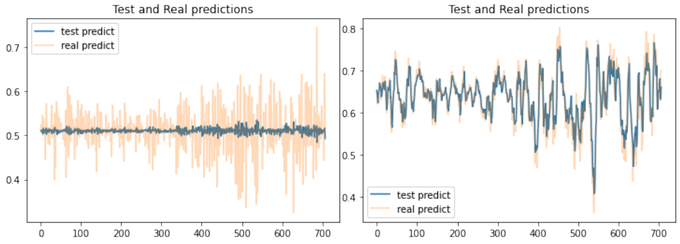


(a) no moving average (b) moving average

Fig. 7: Transformer MAE and MAPE

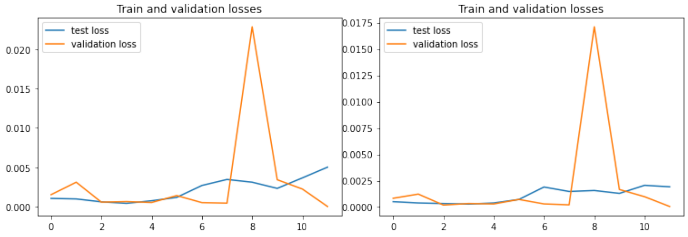
5.2 BiLSTM Results

Through the comparison of the results presented by the figures 8a and 8b, it is possible to obverse a very big difference between the Moving Average and the Singular data, for the same quantity of epochs the predictions based on M.A. data are much better. This is backed by the figures 9a and 9b, that show a difference in order of magnitude of the Validation Loss metric, although the representation is similar. The data starts to converge and hit a plateau at around 50 epochs, so that ammount could be used to train the models instead of the 100 established. This statement is corroborated by various runs with the same result.



(a) no moving average (b) moving average

Fig. 8: BiLSTM Prediction vs Real



(a) no moving average (b) moving average

Fig. 9: BiLSTM Validation Loss

5.3 Results Analysis

In the tables 1 and 2 it is presented the benefit of the moving average instead of the regular data, both models show significant improvements over the data without moving average. However, something interesting was that the BiLSTM performed better than the Transformer when looking at the test loss, the other metrics weren't computed due to lack of computing resources.

TABLE 1: Results without moving average

			Loss	MAE		MAPE	
	Epochs	Time [m]	Test Set	Training Set	Validation Set	Training Set	Validation Set
Transformers	1000	30	0.0020	0.0261	0.0359	22.579	0.1226
BiLSTM	100	30	0.0020	—	—	—	—

TABLE 2: Results with moving average

			Loss	MAE		MAPE	
	Epochs	Time [m]	Test Set	Training Set	Validation Set	Training Set	Validation Set
Transformers	1000	30	0.0050	0.0203	0.0264	0.0331	0.0511
BiLSTM	100	30	0.0010	—	—	—	—

6 Conclusions

Key conclusions

- Development of two distinct networks, them being Transformers and BiLSTM;
- Transformers didn't present very different results from BiLSTM, however when complicated with more data, from more stocks, due to parallelization, transformers will probably perform much better.
- For BiLSTM models, the Moving Average is a must, as without it, the network can only predict the trend of the price movements, with no account to outliers.

Future Work

- Test different hyper parameters, although the results are satisfactory;
- Test pretrained networks, like prophet from Facebook [7] and compare it to our results;
- Train bigger datasets from various stocks and markets
- Deploy the model
- Apply MAE and MAPE to BiLSTM

References

[1] V. K. M. S. K. Hiransha Ma, Gopalakrishnan E.Ab. (2018) Nse stock market prediction using deep-learning models. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050918307828

[2] V. Akhter Mohiuddin Rather, Arun Agarwal. (2014) Recurrent neural network and a hybrid model for prediction of stock returns. [Online]. Available: https://doi.org/10.1016/j.eswa.2014.12.003

[3] M. M. A. M. M. M. M. I. S. I. K. M. S. A. Tashreef Muhammad, Anika Bintee Aftab. (2022) Transformer-based deep learning model for stock price prediction: A case study on bangladesh stock market. [Online]. Available: https://doi.org/10.48550/arXiv.2208.08300

[4] A. R. R. S. Gongbo Tang, Mathias Müller. (2018) Why self-attention? a targeted evaluation of neural machine translation architectures. [Online]. Available: https://doi.org/10.48550/arXiv.1808.08946

[5] N. S. N. P. J. U. A. N. G. K. Ashish Vaswani, Llion Jones. (2017) Attention is all you need. [Online]. Available: https://doi.org/10.48550/arXiv.1706.03762

[6] S. E. J. R. J. S. S. T. S. W. C. S. P. P. M. B. Seyed Mehran Kazemi, Rishab Goel. (2019) Time2vec: Learning a vector representation of time. [Online]. Available: https://doi.org/10.48550/arXiv.1907.05321

[7] B. L. Sean J. Taylor. (2017) Forecasting at scale. [Online]. Available: https://doi.org/10.7287/peerj.preprints.3190v2

Appendix A
Images

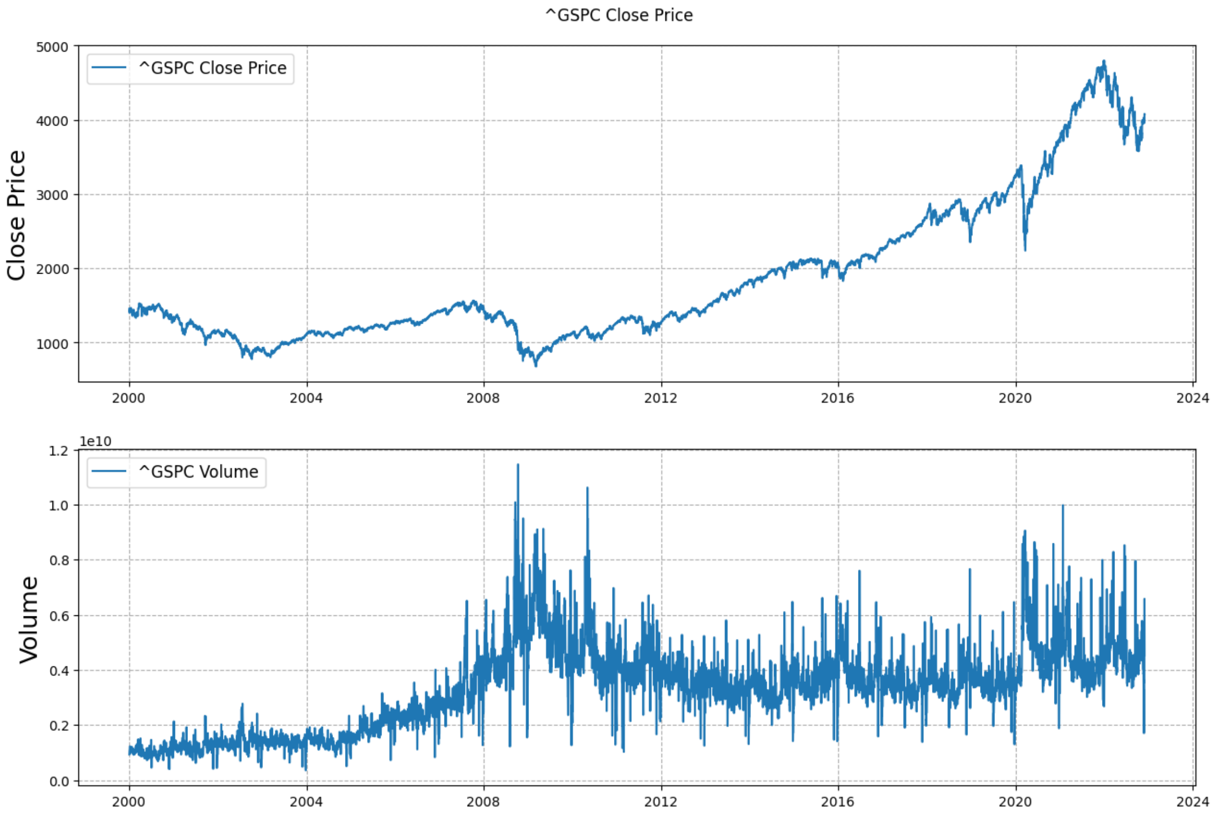


Fig. 10: S&P 500 closing price and volume dataset

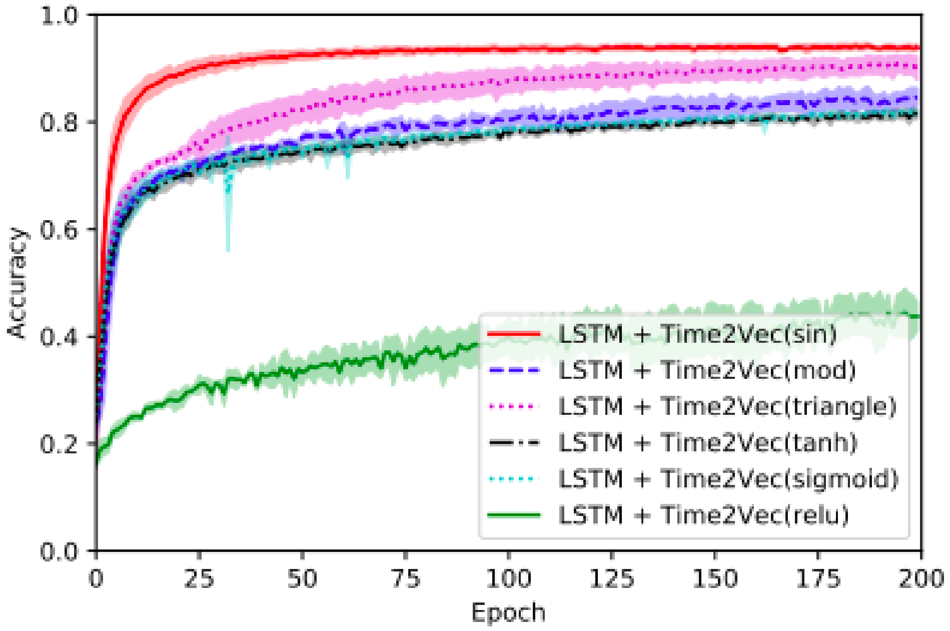
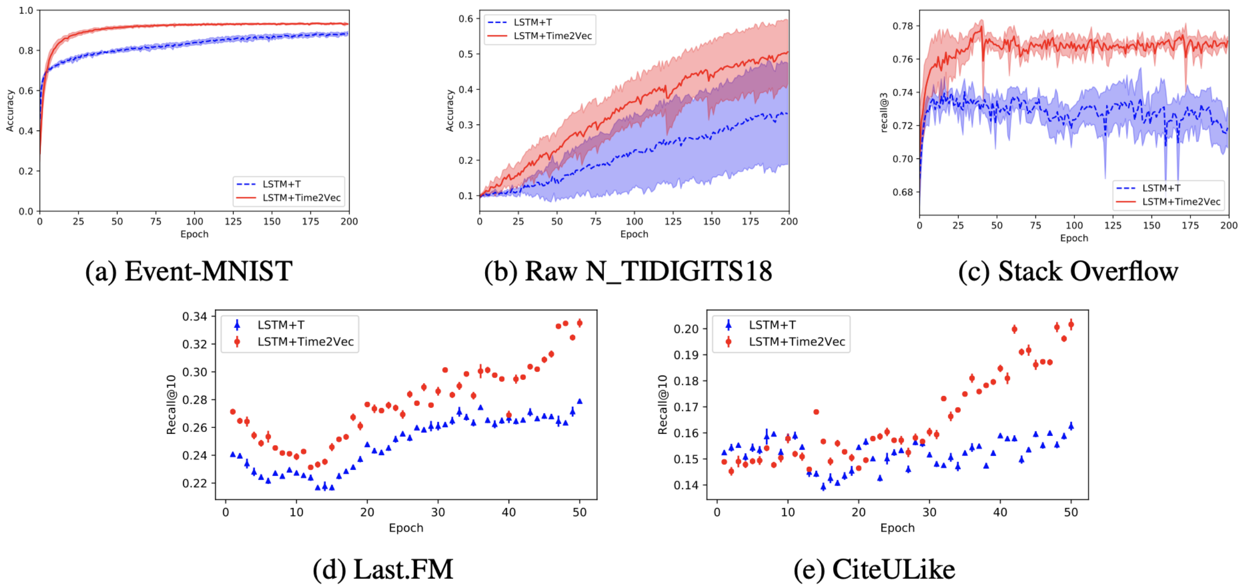


Fig. 11: Performance comparison of non-linear functions [6]



(a) Event-MNIST (b) Raw N_TIDIGITS18 (c) Stack Overflow

(d) Last.FM (e) CiteULike

Fig. 12: Performance comparison [6]

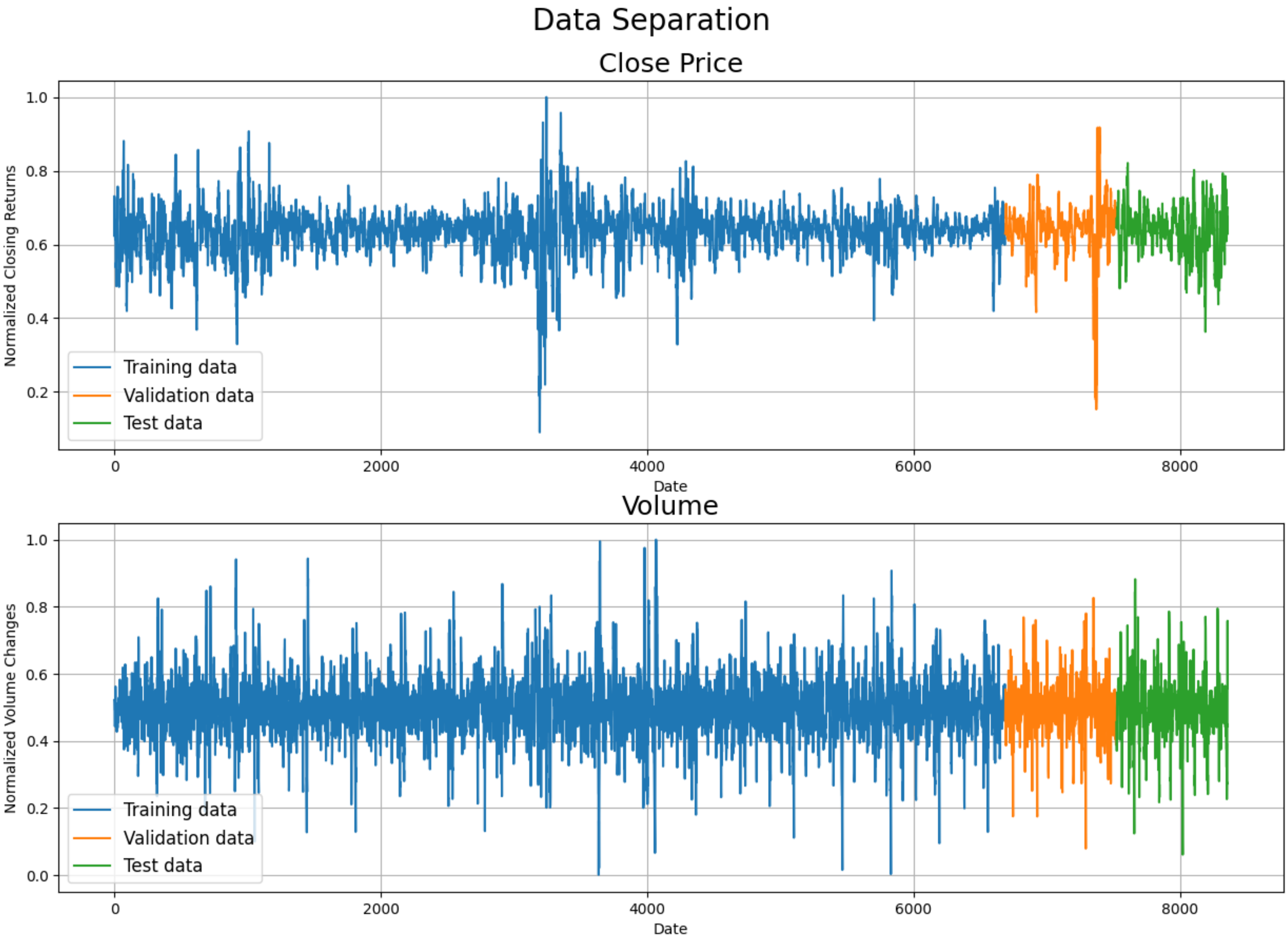


Fig. 13: Data Split

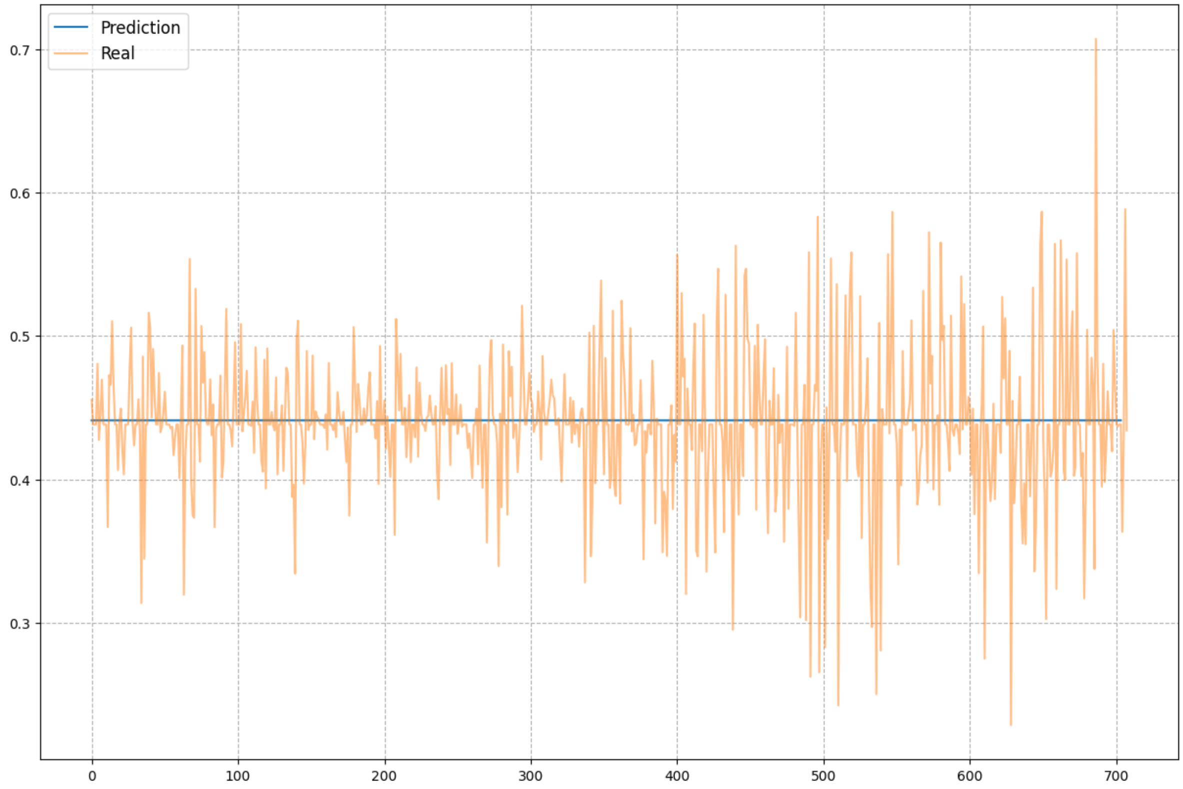


Fig. 14: Transformer with no moving average

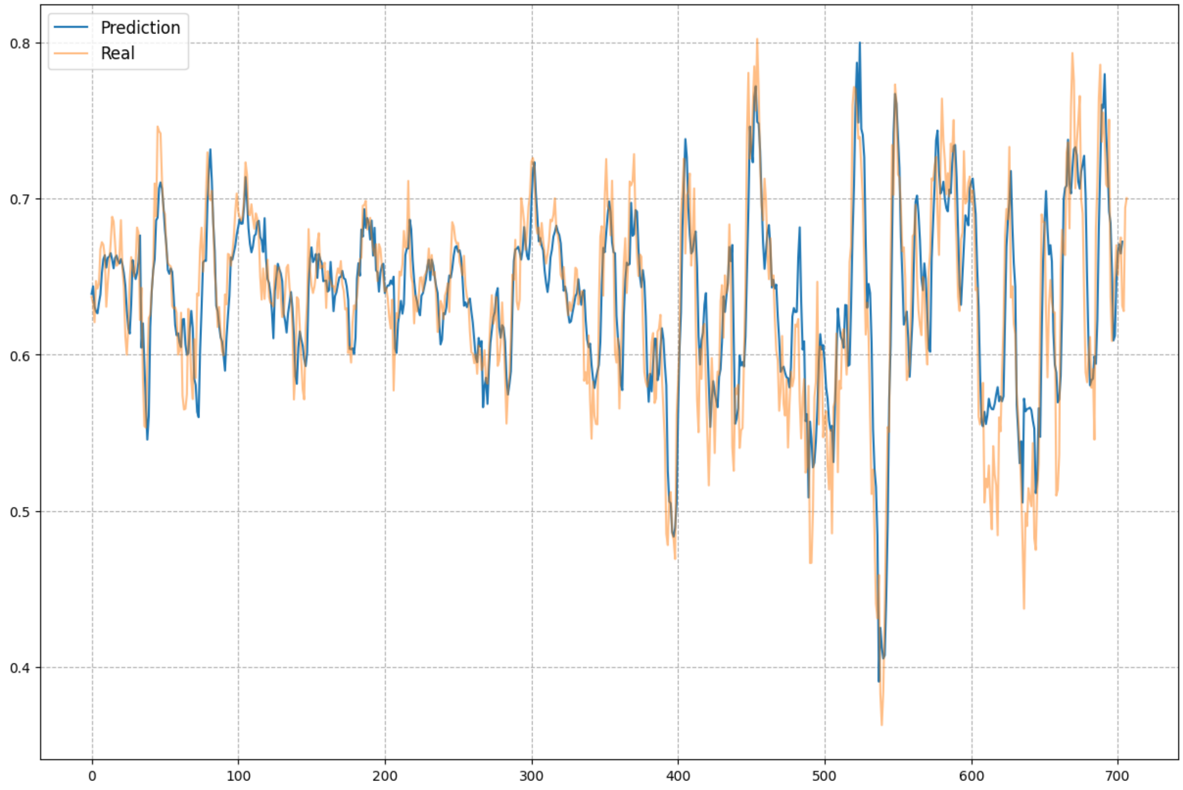


Fig. 15: Transformer with moving average

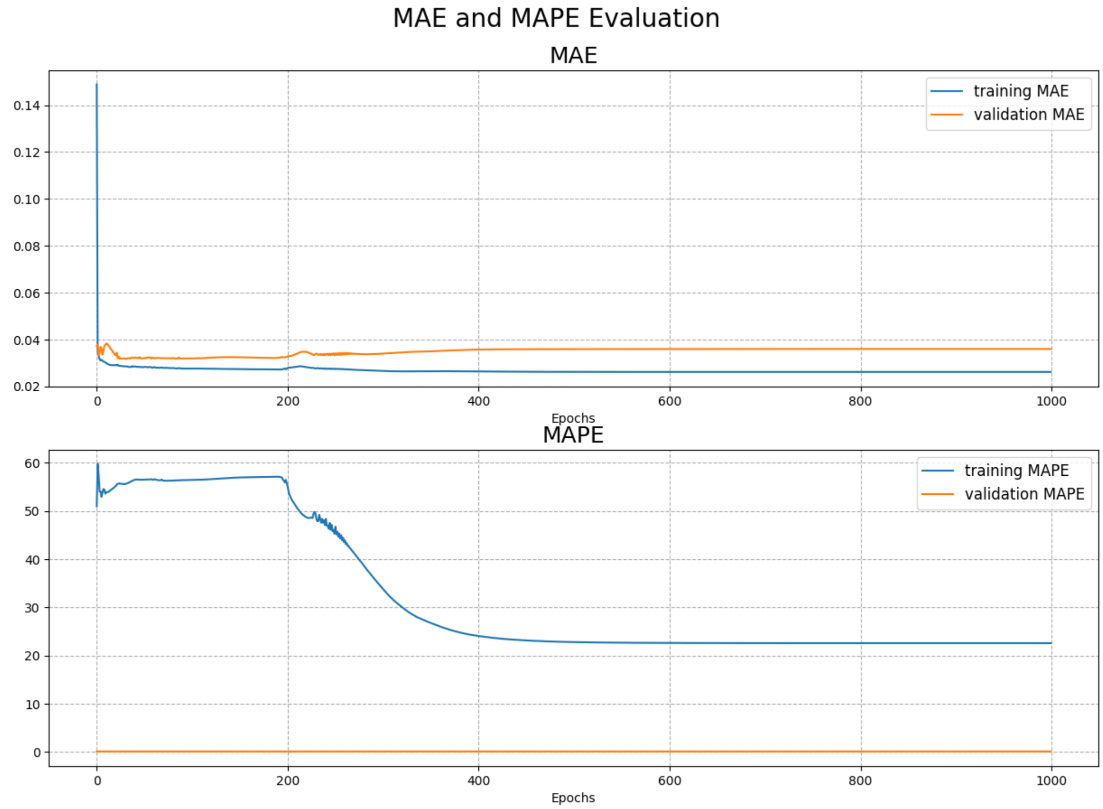


Fig. 16: Transformer with no moving average

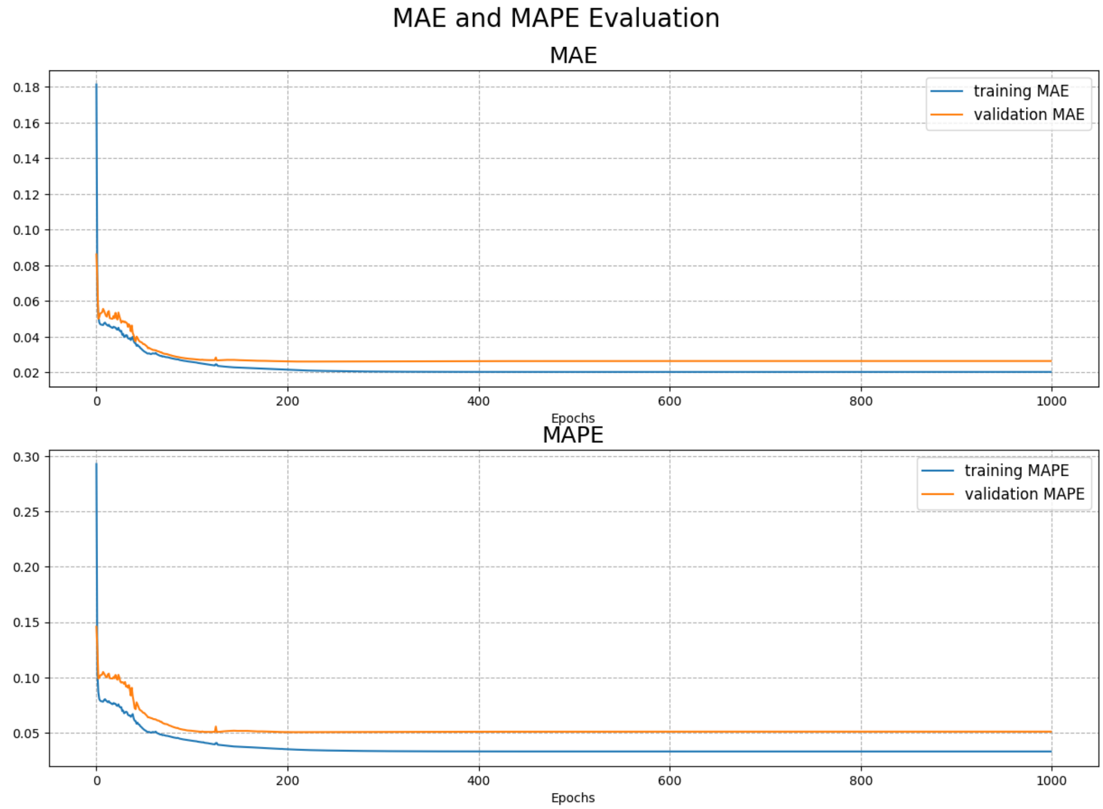


Fig. 17: Transformer with moving average

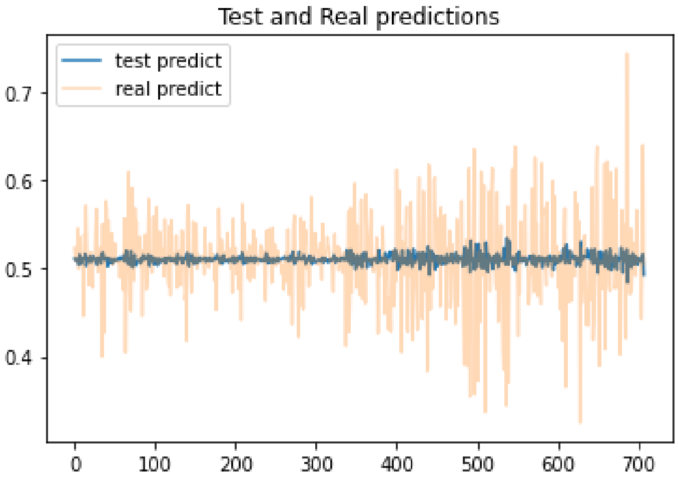


Fig. 18: BiLSTM with no moving average

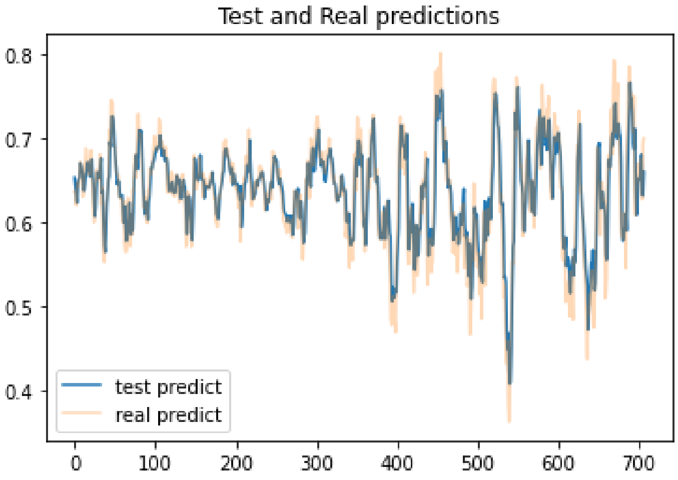


Fig. 19: BiLSTM with moving average

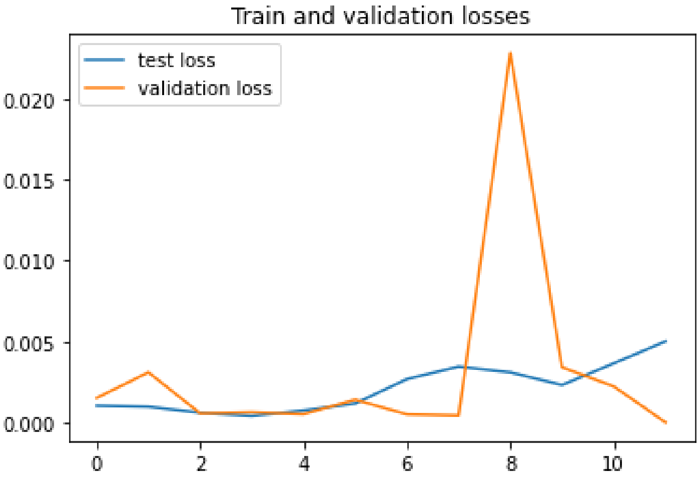


Fig. 20: BiLSTM with no moving average

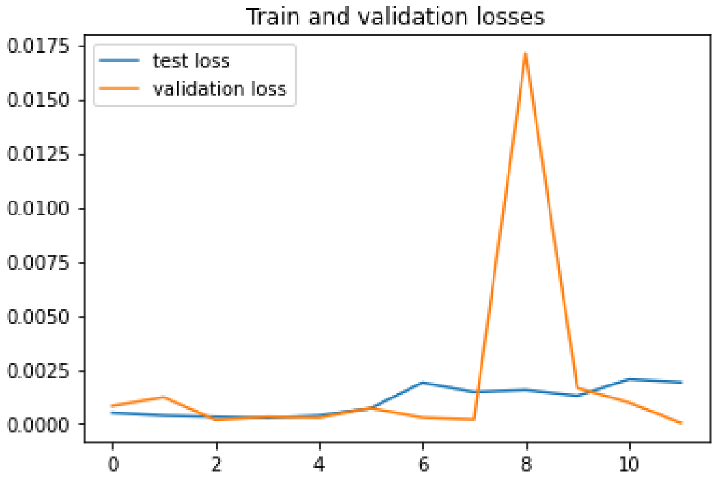


Fig. 21: BiLSTM with moving average

Appendix B
Tables

TABLE 3: Results without moving average

			Loss	MAE		MAPE	
	Epochs	Time [m]	Test Set	Training Set	Validation Set	Training Set	Validation Set
Transformers	1000	30	0.0020	0.0261	0.0359	22.579	0.1226
BiLSTM	100	30	0.0020	——	——	——	——

TABLE 4: Results with moving average

			Loss	MAE		MAPE	
	Epochs	Time [m]	Test Set	Training Set	Validation Set	Training Set	Validation Set
Transformers	1000	30	0.0050	0.0203	0.0264	0.0331	0.0511
BiLSTM	100	30	0.0010	——	——	——	——