Beating persistence, ARIMA and VAR models of financial time series using Recurrent Neural Networks

Garik Vardanyan

2019/09/14

1 Introduction

Advances in deep learning revolutionise economy just like electricity did in the past. From face and speech recognition to electric load time series prediction deep learning is everywhere. Financial market participants have always used state of art techniques to find minor arbitrage opportunities. For example, nowadays hedge- funds use deep learning for satellite image recognition, which helps them with forecasts of earnings. One of the first works considering neural networks dates back to 1996 when in [1] authors built eight steps for designing neural network for forecasting financial and economic time series. Nowadays, there are different types of neural networks which are used for time series forecasting such as LSTMs, GRU's and simple multilayer perceptrons. For the same purpose there are also many machine learning regression models, such as XGBoost or Random Forest or econometric time series models such as ARIMA for univariate time series and VAR for multivariate time series. The goal of this project is to investigate how different types of recurrent neural networks compare with their predictive power to econometric models. There is a versatile academic literature on time-series forecasting. The structure of this work will be similar to [2] in which authors use numerous types of neural networks to forecast time series on electric load. The main difference will be the usage of financial data in our model and the selection of benchmark models (econometric models and persistence model). [3] was one of few articles on deep learning and financial time series modelling, in the work authors alternatively use GANs for time series simulation and forecasting, they also compare their models with econometric benchmark models such as GARCH.

2 Data

In the project proposal I decided that the stock data of 5 companies with biggest market capitalization will be analyzed. The tickers of those companies are "AAPL","GOOGL","AMZN","MSFT","BRK-A". The data was collected using yfinance library. There were not any missing points. Figure 1 summarizes the correlation between stocks. As we can see it ranges from 0.875 to 1 which is very high, thus to have better features for multivariate forecasting it was decided to take other approach when selecting companies for analysis.

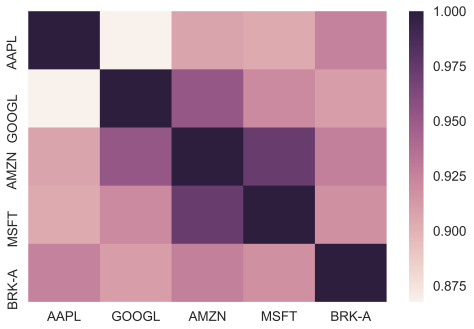

Figure 1: Correlation between stocks of project proposal

S&P500 index contains companies from more than 10 sectors. I decided to take from each sector randomly one company. The tickers of selected companies are:"T","BBY","KO","FTI","BLK","DVA","BA", "IBM","ALB","CBRE". The new correlogram is summarized in figure 2 and as we can see there is a wide range of correlation between stocks.

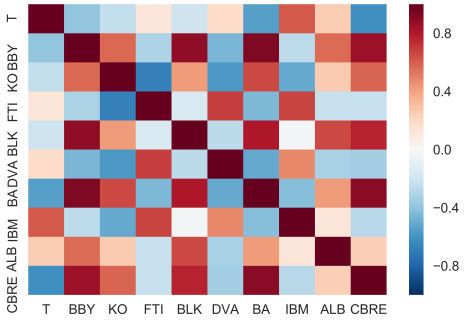

Figure 2: Correlation between final stocks

It is also important to note that for train set time-frame considered included daily stock observations from 01-01-2015 to 01-01-2019. The test set included observations from 04-01-2019 to 06-09-2019. Train contained 1007 observations while test 168. There is an opinion in literature that some econometric models are well suited for monthly forecasting and in order to check robustness of our models we forecast monthly data of considered stocks too(from 2010 to 2019 in this case). 4 types of metrics are used for comparison, MSE, MAE, R-squared and MAPE. There is some discussion about appropriateness of MSE ,MAE or R-squared [4], [5], considering the assumptions behind those metrics, therefore the key measure of comparison will be MAPE, while others will be calculated out of curiosity. Before training models all series were tested for stationarity using Dickey-Fueller test. The VAR model was trained on first difference of series which was stationary. Before feeding data to LSTM and GRU I used MinMaxScaler to scale it into range from 0 to 1 to speed up gradient descend convergence. Figure 3 summarises first 4 tickers, other stocks have similar patterns.

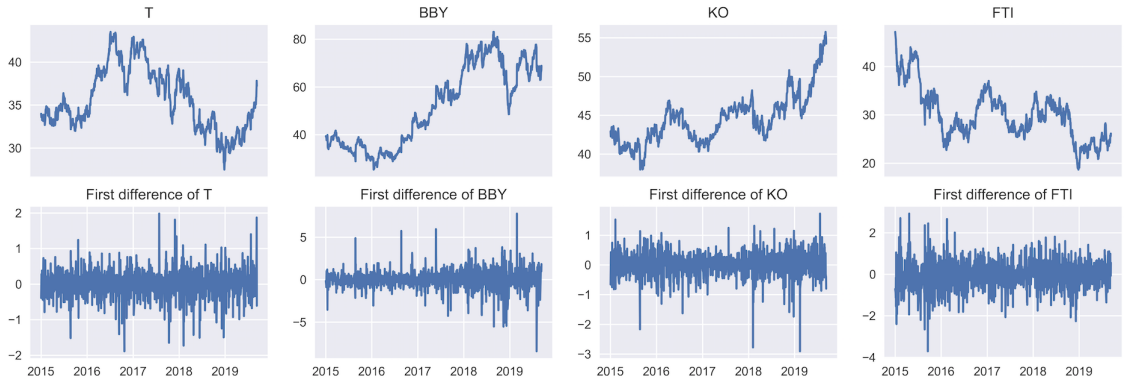

Figure 3: Graph of stock prices and differences

As we can see from difference of series, for majority of observations the change is 0, this means that we should expect persistence model to be quiet precise.

3 Benchmark Models

Each considered model forecasted one step ahead by knowing all present data. This approach is rolling window approach and is more realistic than forecasting several steps ahead by knowing present values. To avoid look ahead bias as discussed in [6] for each step when we added "present" data to train set, new econometric model was trained. As the goal of this work is to compare RNNs with econometric models, for benchmark models I considered ARIMA for univariate modelling and VAR for multivariate modelling. The persistence model, which forecasts that tomorrow's price of stock will be equal to today's one was also used. The critical part about benchmarks was their performance. As we will see in the next section it is very hard to beat persistence model for daily stock prices because it is very precise. This is due to the fact that the daily stock prices do not change significantly (Figure 3). The ARIMA model even further confirmed initial concerns about persistence model, because when I searched for optimal parameters of ARIMA, using AIC criterion and function autoarima from statsmodels library, it returned exactly the persistence model.

4 Forecasting Daily Data

4.1 Univariate models

As it was mentioned in last section, for each model at each step I forecasted one step ahead using available data from past. The performance of persistence model is summarised in table 1.

Metrics	T	BBY	KO	FTI	BLK	DVA	BA	IBM	ALB	CBRE
MSE	0.14	2.03	0.23	0.27	30.34	1.21	49.22	2.77	2.63	0.51
MAE	0.26	0.93	0.34	0.42	4.31	0.78	4.80	1.13	1.20	0.56
R^2	0.95	0.94	0.98	0.87	0.95	0.91	0.93	0.93	0.96	0.94
MAPE (%)	0.82	1.39	0.69	1.77	0.98	1.44	1.30	0.83	1.62	1.13

Table 1: Performance of persistence model

As we can see from MAPE the persistence model is quiet precise and averages to 1.19 % on all stocks. Nextly lets consider the performance of the most optimal ARIMA model: ARIMA (1,1,0).

Metrics	T	BBY	KO	FTI	BLK	DVA	BA	IBM	ALB	CBRE
MSE	0.14	2.08	0.24	0.28	30.87	1.28	52.51	2.84	2.71	0.53
MAE	0.27	0.94	0.35	0.43	4.34	0.81	5.01	1.15	1.22	0.58
R^2	0.95	0.94	0.97	0.87	0.95	0.91	0.92	0.93	0.96	0.94
MAPE (%)	0.83	1.40	0.70	1.79	0.99	1.49	1.36	0.85	1.65	1.17

Table 2: Performance of ARIMA model

The average MAPE of ARIMA model is 1.22 and is higher than for Persistence model. This small difference is due to the noise factor that is added to lagged value in each ARIMA term. As we can see ARIMA model is not able to outperform for any of the series Persistence model. Nextly lets try different specifications of LSTM and GRU recurrent neural networks and see if it is possible to improve quality of Persistence model. The structure of considered models is similar to one in [7]. First model has 64 LSTM units in the first layer and 32 LSTM units in the second layer. After both layers there is 10% dropout to prevent overfitting. Last layer is Dense layer which is simply prediction of series in the next step. As a features LSTM's use only the first lag of time series. The results are summarised in table 3.

Metrics	T	BBY	KO	FTI	BLK	DVA	BA	IBM	ALB	CBRE
MSE	0.14	1.95	0.29	0.27	30.31	1.22	61.47	2.85	2.69	0.69
MAE	0.26	0.92	0.39	0.42	4.30	0.79	5.62	1.15	1.22	0.68
R^2	0.95	0.94	0.97	0.87	0.95	0.90	0.89	0.94	0.96	0.93
MAPE (%)	0.82	1.36	0.78	1.78	0.98	1.47	1.50	0.85	1.64	1.37

Table 3: Performance of LSTM(64,32) model

Interestingly for the majority of tickers LSTM performance is the same or worse than performance of persistence model, while for BLK it has better MAPE. If we compare LSTM model with ARIMA then for 6 of the time series it has better MAPE. For the 3 of tickers the MAPE is worse than one of the ARIMA and for one of the tickers the performance is equal. As we can see univariate LSTM model can improve the ARIMA forecasting. Surprisingly if we compare average MAPE of LSTM (1.25%) and ARIMA (1.23%), ARIMA has advantage. This means that while for majority of time series LSTM is better, on series where it is not better than ARIMA it has much worse performance than ARIMA on the series where LSTM is superior. Nextly lets consider two other architectures of

RNNs, one with fewer hidden layers and one that is more complicated. Second considered RNN has two hidden layers, first one consists of 32 LSTM units, while second one consists of 16 LSTM units. The last layer is again used for prediction and is Dense layer with one unit.

Metrics	T	BBY	KO	FTI	BLK	DVA	BA	IBM	ALB	CBRE
MSE	0.14	1.98	0.32	0.27	34.29	1.29	93.38	2.75	2.64	0.76
MAE	0.27	0.94	0.42	0.42	4.62	0.80	7.52	1.14	1.21	0.72
R^2	0.95	0.94	0.96	0.87	0.94	0.88	0.82	0.94	0.96	0.91
MAPE (%)	0.82	1.39	0.83	1.75	1.05	1.50	1.99	0.83	1.63	1.45

Table 4: Performance of LSTM(32,16) model

For the majority of tickers performance quality decreased and only for IBM MAPE is better by 2 percentage points. The average MAPE for this architecture is equal to 1.32 and is significantly worse than the one of the (64,32) model. Lets double units in the hidden layers and now consider LSTM (128,64). Table 5 summarises results for LSTM (128,64).

Metrics	T	BBY	KO	FTI	BLK	DVA	BA	IBM	ALB	CBRE
MSE	0.14	2.02	0.28	0.27	30.07	1.23	54.95	2.90	2.65	0.62
MAE	0.26	0.94	0.38	0.42	4.29	0.79	5.21	1.17	1.21	0.64
R^2	0.95	0.94	0.97	0.88	0.95	0.91	0.91	0.94	0.96	0.93
MAPE (%)	0.81	1.40	0.77	1.77	0.98	1.46	1.40	0.86	1.63	1.29

Table 5: Performance of LSTM(128,64) model

Average MAPE of LSTM(128,64) is equal to 1.24 it is better than one of the LSTM(64,32). For 7 of the tickers LSTM(128,64) has higher MAPE than initial (64,32) model. In comparison to ARIMA model LSTM(128,64) performs better only on 5 of the tickers, while if we compare LSTM(128,64) to Persistence model, it only performs better on ticker "T". Lets make one step further and look at the model LSTM(256,128), see if additional units can help dominate persistence model.

	T	BBY	KO	FTI	BLK	DVA	BA	IBM	ALB	CBRE
MSE	0.14	1.97	0.28	0.29	30.12	1.21	50.90	2.80	2.69	0.58
MAE	0.27	0.93	0.38	0.44	4.30	0.78	4.94	1.15	1.21	0.62
R^2	0.95	0.94	0.97	0.87	0.95	0.91	0.92	0.94	0.96	0.94
MAPE (%)	0.84	1.38	0.77	1.83	0.98	1.45	1.33	0.84	1.64	1.25

Table 6: Performance of LSTM(256,128) model

In comparison to initial LSTM, for 5 of the tickers the new model is better. The average MAPE is approximately 1.23 and it is still worse than results of persistence model. Only for BBY the current LSTM's prediction is better than prediction of persistence model. Another type of recurrent neural networks are gated recurrent units, which are simpler than LSTMs. The same network architectures were used for GRU's. First of all network with 2 layers one with 64 GRU units, second with 32 units. The results are summarised in Table 7.

Metrics	T	BBY	KO	FTI	BLK	DVA	BA	IBM	ALB	CBRE
MSE	0.14	1.99	0.27	0.29	29.83	1.23	57.91	2.87	2.87	0.67
MAE	0.27	0.93	0.38	0.44	4.28	0.80	5.40	1.16	1.24	0.67
R^2	0.95	0.94	0.97	0.87	0.95	0.91	0.90	0.94	0.95	0.93
MAPE (%)	0.85	1.38	0.76	1.84	0.97	1.47	1.45	0.85	1.68	1.35

Table 7: Performance of GRU(64,32) model

Overall the quality significantly diminshed in comparison to LSTM models with average MAPE of 1.27. I will try making more unit layers to see if the model is underfitting the data. The results for GRU(128,64) are summarised in Table 8.

Metrics	T	BBY	KO	FTI	BLK	DVA	BA	IBM	ALB	CBRE
MSE	0.14	1.98	0.25	0.28	30.02	1.23	50.83	2.75	2.67	0.54
MAE	0.27	0.93	0.36	0.43	4.28	0.79	4.95	1.13	1.21	0.59
R^2	0.95	0.94	0.97	0.87	0.95	0.91	0.91	0.94	0.96	0.94
MAPE (%)	0.83	1.39	0.72	1.80	0.98	1.46	1.33	0.83	1.64	1.19

Table 8: Performance of GRU(128,64) model

Clearly our first GRU model was underfitting the data. GRU(128,64) has average MAPE of 1.217 on test data. Its performance is still lower than one of the persistence model, so lets try to add more units and check everything for GRU(256,128).

	T	BBY	KO	FTI	BLK	DVA	BA	IBM	ALB	CBRE
MSE	0.17	1.96	0.27	0.28	30.52	1.22	52.14	3.09	2.64	0.51
MAE	0.31	0.92	0.38	0.43	4.33	0.79	5.03	1.23	1.21	0.57
R^2	0.94	0.94	0.97	0.88	0.95	0.91	0.91	0.93	0.96	0.95
MAPE (%)	0.96	1.37	0.75	1.79	0.99	1.45	1.35	0.90	1.63	1.15

Table 9: Performance of GRU(256,128) model

The average MAPE increased to 1.23 and thus the model with more GRU units started to have worse performance.

In this part of work we built two-layer LSTM and GRU recurrent neural networks and analyzed if they were able to perform better on test data then persistence and arima models. None of the constructed models dominated the persistence model. The best models by MAPE for each stock are summarised in table 10.

Stocks	T	BBY	KO	FTI	BLK	DVA	BA	IBM	ALB	CBRE
Best model	$L_{128,64}$	$L_{64,32}$	P	$L_{32,16}$	$G_{64,32}$	P	P	$G_{128,64}$	P	P
Best MAPE (%)	0.81	1.36	0.69	1.75	0.97	1.44	1.30	0.83	1.62	1.13

Table 10: Summary of best models

The P in table refers to persistence model, L to LSTMs while G to GRUs. Final results suggest that despite the fact that Persistence model is dominating, some of the RNNs were able to surpass its performance for specific stocks.

4.2 Multivariate models

As a benchmark model for multivariate forecasting I consider VAR model. It was trained on first difference of series, because initial series were not stationary. After training the differenced series were transformed into initial series. The optimal number of lags for VAR model was selected using selectorder from statsmodels library. As it was expected from the Figure 3, the number of lags was equal to 0. That meant that the model was giving constant prediction to changes in series, thus the prediction was same as for persistence model but with some round error.

Metrics	T	BBY	KO	FTI	BLK	DVA	BA	IBM	ALB	CBRE
MSE	0.14	2.00	0.23	0.28	30.19	1.23	49.80	2.80	2.65	0.51
MAE	0.27	0.92	0.34	0.43	4.29	0.79	4.85	1.14	1.21	0.57
R^2	0.95	0.94	0.98	0.88	0.95	0.91	0.93	0.94	0.96	0.95
MAPE (%)	0.83	1.38	0.69	1.79	0.98	1.46	1.32	0.84	1.63	1.15

Table 11: Summary of VAR(0)

The results are nearly identical to ones obtained by persistence model. This is the main reason why I decided to also analyze monthly data of the same time series. I expect that there will be much more volatility in data and thus persistence model won't work that well.

Nextly I will try some neural networks to see if it is possible to get better forecast for daily time series than VAR(0). As a feature RNNs will use lagged values of other stocks. First model is LSTM(64,32) with 1400 iterations. I increased iterations by 400 because now we have more features and model will need more time to converge. The results are summarised in Table 12.

Metrics	T	BBY	KO	FTI	BLK	DVA	BA	IBM	ALB	CBRE
MSE	0.47	2.31	1.18	0.81	88.36	1.84	191.30	13.01	3.58	1.82
MAE	0.57	1.08	0.87	0.70	7.85	1.02	11.89	3.21	1.44	1.17
R^2	0.79	0.93	0.80	0.57	0.84	0.82	0.61	0.69	0.94	0.78
MAPE (%)	1.72	1.59	1.69	2.92	1.77	1.88	3.17	2.33	1.91	2.33

Table 12: Performance of LSTM(64,32) on multivariate time series

As we can see the multivariate model is significantly worse than univariate one. I will increase units to see if it is possible to make model better.

Metrics	T	BBY	KO	FTI	BLK	DVA	BA	IBM	ALB	CBRE
MSE	0.50	3.27	1.12	0.56	93.47	1.86	111.89	8.31	3.82	1.48
MAE	0.59	1.39	0.86	0.59	8.02	1.05	8.52	2.45	1.49	1.04
R^2	0.78	0.90	0.82	0.71	0.82	0.84	0.80	0.81	0.94	0.82
MAPE (%)	1.79	2.03	1.67	2.47	1.81	1.91	2.28	1.79	1.99	2.06

Table 13: Performance of LSTM(128,64) on multivariate time series

Increasing number of hidden units doesn't help as we can see in table 13. Comparing these results with VAR and considering the fact that the most optimal VAR model is one with 0 lags I decided that it is not effective to spend further time on multivariate models. 0 lags in VAR indicate that there is not any significant linear relationship between lags of differenced daily stock prices. As we have seen in figure 3, there is not too much variability in daily data, thus for more robust results I decided to analyse monthly data of the stocks of same companies and see which models perform better.

5 Forecasting Monthly Data

5.1 Univariate models

In this section we will forecast monthly data of price of stocks considered in section 4. Figure 4 shows some of those stocks.

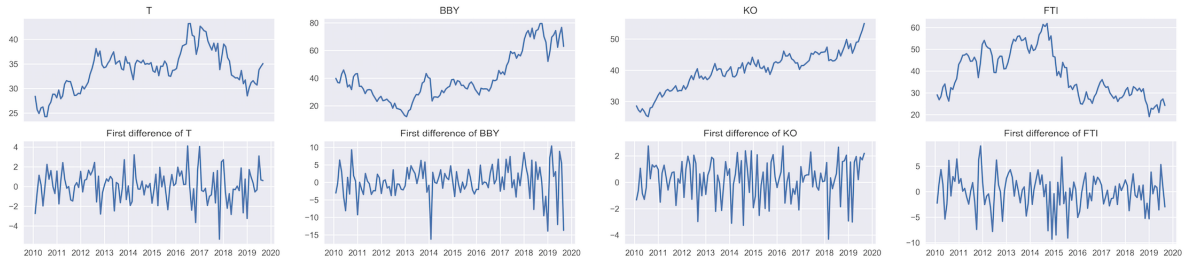

Figure 4: Graph of monthly stock prices and differences

For monthly data the mean percentage change in subsequent observation is equal to 0.21% while for daily data it is equal to 0.00004%, thus monthly data of stocks have much higher variance and the modified choice is justified. The data collected includes 118 observations. Because we have only monthly data, it was decided to collect observations from 2010. The train test will be from 2010 to 2018 (97 observations), while test from 2018 to September of 2019 (21 observations). For monthly data autoarima function returned order of (0,1,0) which es equivalent to persistence model, thus performance of ARIMA and persistence models is the same and is summarised in table 14.

Metrics	T	BBY	KO	FTI	BLK	DVA	BA	IBM	ALB	CBRE
MSE	2.53	55.27	3.96	8.19	1434.64	39.49	1281.95	154.35	85.38	11.90
MAE	1.21	6.28	1.71	2.34	32.41	4.71	30.05	9.34	7.09	2.83
R^2	0.64	-0.28	0.47	0.53	0.41	0.45	-0.35	0.05	0.62	0.06
MAPE (%)	3.74	9.44	3.62	9.19	7.12	8.16	8.31	7.07	8.65	6.09

Table 14: Performance of Persistence and Arima models on monthly time series

As we can see, the MAPE significantly increased in this case averaging to 7.13 %.

Same LSTM neural network architectures were used for monthly time series, the best results with corresponding models are summarised in table 14.

Metrics	T	BBY	KO	FTI	BLK	DVA	BA	IBM	ALB	CBRE
LSTM	(64, 32)	(256, 128)	(32, 16)	(128, 64)	(256, 128)	(64, 32)	(64, 32)	(32, 16)	(32, 16)	(256, 128)
MAPE (%)	3.71	9.65	3.57	8.85	7.73	8.27	9.56	6.01	8.74	5.79

Table 15: Performance of best LSTM models

Although the mean MAPE is 7.19% for LSTM models, for 5 of the ticker we were able to perform better. I do not present results for GRU RNN's but their performance is worse than one of LSTMs.

5.2 Multivariate forecasting

The optimal VAR model again was one with 0 lags. The table 16 shows its performance.

Metrics	T	BBY	KO	FTI	BLK	DVA	BA	IBM	ALB	CBRE
MSE	2.80	56.59	3.67	8.12	1417.59	44.27	1236.43	148.64	88.39	11.29
MAE	1.30	6.34	1.56	2.35	32.29	5.03	29.25	8.96	7.25	2.66
R^2	0.62	-0.19	0.49	0.53	0.41	0.37	-0.03	0.10	0.71	0.12
MAPE (%)	3.94	9.55	3.30	9.14	7.08	8.55	8.16	6.78	8.68	5.74

Table 16: Performance of VAR(0) on monthly data

Interestingly despite that VAR uses differenced series and 0 lags, this model was able to improve some of the metrics in comparison to persistence model. Now we will look at the best multivariate LSTM and GRU models which will include one lag of all of the stocks. The performance is summarized in table 17, only best models are included

Metrics	T	BBY	KO	FTI	BLK	DVA	BA	IBM	ALB	CBRE
Model	LSTM	GRU	LSTM	LSTM	GRU	LSTM	GRU	LSTM	LSTM	LSTM
Layers	64, 32	64, 32	32, 16	128, 64	32, 16	64, 32	64, 32	32, 16	32, 16	256, 128
MAPE (%)	3.71	9.18	3.57	8.85	7.18	8.27	7.91	6.01	8.74	5.79

Table 17: Performance of best multivariate RNN models

Final results show that RNN models improved both VAR model (MAPE - 7.10) and persistence model (MAPE - 7.19) by having average MAPE on all stocks equal to 6.92%.

6 Results

In this work I analysed time series data of 10 stocks from S&P 500 index. In the first two sections I discussed methods used for time series analysis and data that I have collected. I explained how I chose the stocks and presented in the third section benchmark models used.

Fourth section of work was dedicated to daily stock data. There were univariate models in which as features I took only lagged values of stock itself and multivariate models features in which also included lagged values of other stocks. The key finding indicates that while due to specificity of stock data none of the models is able to beat Persistence model, some of the RNNs are able to perform better on some of the stocks. Multivariate models of daily stock data indicate that RNNs are not able to effectively use information about other stocks to have better performance than univariate models. This leads to idea to analyse not daily but monthly data.

Monthly data of the stocks obviously had more volatility and thus was more interesting to forecast. Univariate RNNs again were able to better forecast some of the stocks than persistence or ARIMA models. Multivariate RNNs indeed were able to dominate both persistence model and optimal VAR model by having better forecasts for 7 of the stocks and lower average error on all of the stocks.

The main results indicate that for monthly stock data RNNs are able to perform better than benchmark models, while for daily data because there is so much less volatility multivariate and univariate RNNs are not able to dominate benchmark models.

References

[1] Boyd M. Kaastra, I. Designing a neural network for forecasting financial and economic time series. *Neurocomputing 10(3), 215236*, 1996.

[2] Lukovic S. Alippi C. Gasparin, A. Deep learning for time series forecasting: The electric load case. *arXiv*, 2019.

[3] Knobloch R. Korn R. Kretschmer P. Magnus, W. Quant gans: Deep generation of financial time series. *arXiv*, 2019.

[4] Does r-squared matter in time series data? if not, then why? *https://www.quora.com/Does-R-squared-matter-in-time-series-data-If-not-then-why.*

[5] What is the difference between squared error and absolute error? *https://www.quora.com/What-is-the-difference-between-squared-error-and-absolute-error.*

[6] Avoiding look ahead bias in time series modelling. *https://www.datasciencecentral.com/profiles/blogs/avoiding-look-ahead-bias-in-time-series-modelling-1.*

[7] Fei-Fei L. Karpathy A., Johnson J. Visualizing and understanding recurrent neural networks. *arXiv*, 2016.