AFM423: Report 2 - Realized Volatility Modelling

Zhenning Pei[1], Yasien Sayeed[2], and Zixi Tao[3]

[1] *ID: 20625199 Email: z8pei@uwaterloo.ca*
[2] *ID: 20616222 Email: ygsayeed@uwaterloo.ca*
[3] *ID: 20599273 Email: reginald.tao@uwaterloo.ca*

Abstract

In this report, we model the realized volatility in the market with several learning algorithms. Distributional properties of the S&P500 realized volatility are discussed. Elastic Net and LSTM Neural Network are fitted to the empirical data and are compared against HAR models.

Keywords: Realized Volatility, Elastic Net, Neural Network, HAR Model, Financial Time Series

1 Overview

There have been a variety of models proposed by the researchers in the past years to model realized volatility in the market. Report 1 outlines some popular ones in machine learning and deep learning framework. In this report, we further investigate the models proposed in report 1.

We start first with a description and a rudimentary analysis on the dataset. Models such as elastic net and Long Short-Term Memory (LSTM) based Recurrent Neural Network are then discussed. The models are then compared against the popular HAR based model proposed in Corsi (2009) and its slight variation which includes the usage of bagging. Finally, we propose some potential improvements on the topic.

2 Data

2.1 Description

The original dataset comes from Bloomberg Terminal (Bloomberg 2020) and consists of 15-min frequency intraday trading data of the S&P500 Index. The relevant time horizon is about 11 years long, from the beginning of 2008 to the early months of 2020. The trading data are then transformed into 2 different measures of volatility: the usually realized volatility, and the realized Lower Partial Standard Deviation (LPSD). The resulting datasets each has 3060 data points. This study looked at predicting the realized volatility measures by only looking at the past data for that one feature, and did not look at additional features in the models.

In our study, natural based logarithm is taken in the fit to make sure the predictions on the original scale will always be positive. We will simply use RV/LPSD to denote $\log(RV)$, $\log(LPSD)$ as well.

2.2 EDA

From Table 3 we can see clearly the positive skewness and the excess kurtosis as compared to a normal distribution. This result conforms with the theoretical framework of realized volatility. Clustering is also present as shown in Figure 1.

In both the QQ-Plots in Figure 2(a) and 2(b) and the empirical density fit in Figure 3, we find that the realized volatility is not normally distributed, yet lognormal distribution provides a decent fit on the data set. The result is further supported by a formal Djang-Box Test and is in line with the findings in Andersen *et al.* (2001). However, the realized LPSD is no longer lognormally distributed as indicated by the QQ-Plot in Figure 5.

In Figure 4 and 6, the long memory of RV and realized LPSD are presented. This is the central point of realized volatility modelling. Since traditional popular time series models like AR (up to order 5) and GARCH (usually GARCH(1,1)) model would fail to capture the slowly decreasing correlation of the data points, we turn to the machine learning and deep learning space.

3 Elastic Net

3.1 Model

Elastic Net combines the 2 sets of models in LASSO (Tibshirani 1996) and Ridge (Hoerl and Kennard 1970) with a dynamic weight α. In our model, lag terms up to order 29 are considered. Our goal is trying to optimize the HAR model proposed in Corsi (2009) by considering a similar lag order and utilizing the shrinkage methods. Formal representation of the model can be found in Report 1.

3.2 Model Fit and Results

In the Elastic Net modelling, the data is best fit with $\alpha = 1, \lambda = 0.005$ based on methods in Kuhn (2020). That is, the LASSO model best fits the realized volatility series in the concerned time period, as shown in Figure 7.

We thus turn to the LASSO family and further examine the fit based in Friedman, Hastie, and Tibshirani (2010). Variable importance is shown in Table 4. It is not surprising that lag1 and lag2 are the 2 most significant predictors. However, the fit differs from the HAR model in a variety of ways. We can also see that lag29, which is the highest order lag in our fit, is somewhere in the middle in terms of importance. There might be shock effects that persist way longer than we usually assume, which conform with the result presented in Audrino, Huang, and Okhrin (2019).

The change in Cross Validation Error is not significant until only about 20 predictors remain, as shown in Figure 9. Some of the predictors have negligible effect on RV and can be ignored. The Elastic Net family, in general, may provide a better fit than the HAR model due to its flexibility.

We now take a quick look at the standardized residuals of the model in the testing set shown in Figure 10. Approximately 95% of them are within the 2 standard deviation lines. There are a few points exceeding the 6σ threshold, indicating the standard residuals violate the normality assumption to some extent. We turn to RNN models to see if this can be resolved.

It is not surprising that similar results are also obtained for LPSD data. However, in the standard residuals plot in Figure 11, we found that the model consistently underestimate the realized LPSD value. One possible solution is to give an indicator variable of positive daily returns to the regressor set, yet it is beyond the original scale of the Elastic Net model and this report.

4 LSTM

4.1 Model

LSTM models are a form of recurrent neural networks which aim to learn long-term dependencies, allowing them to identify non-linear relationships while also being able to preserve crucial information over time. For predicting realized volatility, we have examined a few neural networks which made use of LSTM units.

Models with 1, 2, and 3 layers of LSTM units were examined to see if they would be effective in predicting RV and realized LPSD. Each model consists of 1, 2, or 3 pairs of 1 LSTM later and 1 Dropout layer, followed by a linear activation function. The dropout units were used to reduce overfitting. There were 3 hyperparameters considered, which are the dropout rate, the number of lags examined, and the number of neurons in each LSTM layer. The dropout rate was set at 50%, as suggested in the original paper (Srivastava *et al.* (2014)). For lags, we examined lags of 1 (daily), 5 (weekly), 22 (monthly), and 29 (as with the Elastic Net model). For the number of neurons, those were kept constant among layers, and we considered 32, 64, 128, or 256 neurons at each layer. To tune those hyperparameters, we used a validation set where we trained each possible model and validated it to find the ones which were most effective.

The models all have the following architecture:

Quantity	Layer	Description
1	Input	Data is standardized before being fed in. Uses one of {1, 5, 22, 29} lags for the amount of features.
1-3	LSTM Dropout	LSTM layer has one of {32, 64, 128, 256} neurons. Dropout layer drops 50% of connections. This pair of layers is repeated up to 3 times in a row, depending on the model.
1	Output	Uses one neuron with a linear activation function. The ADAM optimizer is used on the model with the default Keras learning rate.

The data of RV/LPSD was split chronologically into training, validation, and test sets, at a ratio of 80:10:10. It was trained in batches of 4 with the default Keras learning rate for the ADAM optimizer, for 50 epochs per model. Before entering the data into each model, it was normalized. For validation and testing purposes, we use RMSE as the measure of which model was better, choosing the model with the lowest MSE. This generated 6 models in total (3 amounts of layers x 2 volatility measures), which were run on the test data.

4.2 Model Fit and Results

One observation regarding LSTM that can be found among this experiment is that there is not any significant change obtained by increasing the number of layers in this LSTM architecture. Regardless of how many LSTM layers were used, all of the predictions are very similar, with little to no change in metrics such as RMSE, MAE, and explained variance (Table 5). This applies to both forms of volatility, RV and LPSD. As shown both from the graphs of predicted values (Figure 12) and the metrics for each model chosen in Table 5, there are not many differences or improvements obtained by adding additional LSTM and dropout layers to the model. From this, it indicates that this architecture is unlikely to benefit by adding in further LSTM layers.

For LSTM modelling, we found that the the higher-lag models were not very effective at making predictions on the validation data, choosing models which had weekly lags, although the number of neurons varied and was not consistent (Table 6). From this, it is indicating that the higher-lag models are likely overfitting to the training set, and during validation, it is preferring the simpler models. This contrasts with the HAR model, which was most effective with more lags, but in this case, the simpler models tended to be more accurate.

Looking at the residual data, they are a slight improvement over the Elastic Net model. Similarly, for each point, 95% of them are within 2 standard deviations (Figure 13). However, the problems present in the Elastic Net model remain, where some residuals are extremely high, beyond 6 standard deviations, although the frequency has been reduced. The residuals do not follow a normal distribution, which can be seen in the histograms for how they are distributed (Figure 14), with these models tending to have only 20% of the values outside 1 standard deviation for RV and 17% for LPSD. There were fewer extreme values in this case, although the issue was not completely eliminated.

For most of the measurements, the LPSD tends to correlate with the RV in similar relationships, as shown in the graphs for the residuals (Figures 13 and 15), which are very similar. Furthermore, both the RMSE and MAE of the residuals are also nearly identical among both types of volatility. However, for LPSD, these models are not able to explain as much of the variance, only accounting for about 37%, while the

RV measures were around 50% (see Table 7).

Although the LSTM models did improve upon the Elastic Net model, reducing the residuals so that there were none past 6 standard deviations, they still have some inaccuracy issues, as they still had a few residuals that would not be expected so frequently normally. Furthermore, they do all have a large portion of unexplained volatility that is not accounted for by the model. It is likely that additional features will need to be included in such a model in order to better predict the realized volatility.

5 HAR and Bagged HAR

5.1 Model

We now establish the HAR model as the benchmark for this study. Ultimately, the classic HAR model is a linear regression model with independent variables being average lags over 1, 5, and 22 days, corresponding to average volatility over the previous day, the previous week, and the previous month. The model was proposed by Corsi (2009), and because of its simplicity and remarkably good forecasting performance, it has been widely accepted in the industry. The classic HAR model also served as the benchmark in many papers reviewed in the previous report.

Here, we examine the performance of the classic HAR; we also take the opportunity to examine the effects of bagging on this model.

To add bagging to the HAR model, for some n (which would be optimized by hypertuning), we randomly draw n sets of samples with replacement from training set and obtain n linear regressors based on these samples; then, to forecast a new value, we take the average the n forecasts from the n regressors. In this study, in order to find the optimal n, we run the algorithm with a range of n's and pick the one that results in the lowest validation error. Bagging does not fundamentally change the nature of the HAR model, as it ultimately preserves the linearity of the 3 features. However, it could reduce the effect of overfitting on one single data set.

5.2 Model Fit and Results

For the classic HAR(1, 5, 22) model described by Corsi (2009), there are no hyperparameters to tune. We simply train a linear regression model on the first 90% of the available time period and use the remaining 10% as the test set.

The p-values for each of the daily, weekly and monthly lags are 1.35e-53, 9.67e-16 and 2.25e-26 respectively, suggesting that all these lags have extremely strong statistical significance.

Observing the forecasted volatilities plotted over true volatilities for both the entire data period and the test period in Figure 18 and Figure 19, we see that the forecast mostly captures the same trend as the true values, although toward the end of the testing period, there was a sharp increase in volatility and the model significantly underestimated the magnitude of the increase, indicating that the model's performance on predicting sudden volatility spikes could be further improved. In fact, over the whole testing period, the model predictions show a "smoother" trend than the true value, indicating that the model tend to underestimate the magnitude of the volatilities.

The standardized residuals of the classic HAR model on the test set is plotted in Figure 20. From the plot we can see most of them are within 2 standard deviations. There are some points towards the end that are well beyond the 2 sigma threshold which happen to correspond to the sudden spike in volatility in the end of testing period. This confirms our observation in Figure 19 that the model's performance is still quite weak for sudden volatility jumps.

For the bagged HAR model, we split the data set into training, validation and test sets, each having weight 80%, 10% and 10%. We pick the size of each sub-sample to be 1/3 of the training set size, which is the same parameter used by McAleer and Medeiros (2010). To pick the optimal number of regressors, we iterate n through 1 to 20 and pick the n that gives the lowest RMSE over the validation set. The tuning RMSEs are plotted in Figure 16. In our study, we noticed quite a bit of variability in Cross Validation RMSE and we have chosen 13 as the optimal number of regressors.

Just like classic HAR, we see that bagged HAR successfully captures the trend of true volatility from the plots in Figure 19. Comparing the RMSEs (Table 9), we see that the test RMSE for bagged HAR is slightly lower than that for classic HAR, while the training RMSE for bagged HAR is lightly higher, suggesting that bagging does indeed correct for overfitting when training the HAR model with one single training set. Despite this difference, we also see the remarkable similarity between plots of standardized residuals for bagged HAR and classic HAR (Figure 20), which have almost the exact shape, even with the same outliers occurring at almost identical spots. This is not surprising as the bagged HAR model does preserve the fundamental structure of the classic HAR, since it is ultimately still a linear combination of the daily, weekly and monthly volatilities.

Finally, we perform the same analysis on LPSD. We notice that just like the case for realized volatility, the residual plots for classic and bagged HAR models look almost identical as shown in graphs in Figure 22. In both graphs, while there are no points exceeding the 6 sigma threshold line, there are slightly more residuals outside the 2 sigma threshold, indicating that the models may not be as suited for forecasting LPSDs as for realized volatilities. This finding is further corroborated by the plotting test predictions against true values. From the plots in Figure 21, we can clearly see that the model predictions have a much 'smoother' trend than real values, suggesting that the model tend to underestimate the magnitude of the LPSDs, which is also similar to our finding for realized volatility.

6 Model Comparisons and Diebold-Mariano Test results

From the above model fit results, we could see that while all three models proposed capture the general trend of the true realized volatility, the residuals for all the models are quite different. In order to evaluate whether there is a statistically significant difference in the models' predictive power, we employ the Diebold-Mariano Test (DM Test).

6.1 The Diebold-Mariano Test

The Diebold-Mariano Test aims to conclude whether two models have the same forecast accuracy based on their residuals when evaluated against the same test set described in Diebold and Mariano (1995). In R's forecast package, a modified version of the test is implemented with the null hypothesis being that two given models have the same forecast accuracy.

	Enet	HAR	Bagged HAR	1 Layer LSTM
Enet	n/a	0.0159	0.0164	0.0398
HAR	0.0159	n/a	0.215	0.624
Bagged HAR	0.0164	0.215	n/a	0.666
1 Layer LSTM	0.0398	0.624	0.666	n/a

Table 1. DM Test P-values for Each Pair of Models

	1 Layer	2 Layers	3 Layers
1 Layer	n/a	0.349	0.829
2 Layers	0.349	n/a	0.345
3 Layers	0.829	0.345	n/a

Table 2. DM Test P-values for LSTM Models with 1, 2, 3 Layers

Looking at Table 1 which outlines the DM test p-values for each pair of models, we arrive at the following conclusion:

1. The elastic net model has p-values below 0.05 when compared with both classic HAR and 1 layer neural network models. This suggests that there is statistically significant evidence that the elastic net model has different forecast accuracy from both the classic HAR and 1 layer LSTM models.
2. The p-values for comparing the LSTM models with different number of layers are in the range of 0.34 to 0.83, which are considerably larger than 0.05. This suggests that there is no statistically significant evidence to support that changing the number of layers in our neural network setup would fundamentally change the forecast accuracy. Furthermore, it is interesting to note that the p-value for comparing the neural network with 1 layer vs 3 layers (0.83) is much larger than that for comparing neural network with 1 or 3 layers to that with 2 layers (0.345-0.349). This could suggest that the residuals for the neural network with 1 and 3 layers are more similar than that with 2 layers.
3. With a p-value of 0.215 > 0.05, we conclude that although the bagged HAR model slightly improves the test RMSE over the classical HAR model in our study, the improvement is not statistically significant enough to support that bagging has a substantial improvement on forecasting accuracy.

7 Conclusion

In this report, we fitted and contrasted 3 set of models in Elastic Net, LSTM Neural Network and HAR. Elastic Net, and penalized methods in general, can be shown to avoid overfitting and provide very interpretable solutions to RV modelling. LSTM Neural Network preserves more theoretical soundness regarding the extremely long term memory of RV, and yet on the other hand do not offer much more accuracy or interpretability.

However, we must note that all 3 methods presented provide decent fit to the quantity in mind. Machine learning and deep learning methods are powerful tools in financial modelling. The flexibility of the learning methods also gives us ways to trade off between accuracy and interpretability.

There are possible improvements to the methods presented. As said, we can allow indicator variables for better prediction accuracy regarding LPSD. Also, weighted methods can be utilized to account for the higher 'volatility of volatility' when the value of RV is high.

A Tables and Figures

	Realized.Volatility
Observations	3060.0000
NAs	0.0000
Minimum	0.0011
Quartile 1	0.0046
Median	0.0068
Arithmetic Mean	0.0089
Geometric Mean	0.0088
Quartile 3	0.0106
Maximum	0.0910
SE Mean	0.0001
LCL Mean (0.95)	0.0086
UCL Mean (0.95)	0.0091
Variance	0.0001
Stdev	0.0074
Skewness	3.5325
Kurtosis	20.4020

Table 3. Summary of Dataset

Order	Most Important	Least Important
1	lag1	lag23
2	lag2	lag22
3	lag13	lag7
4	lag14	lag12
5	lag8	lag24
6	lag10	lag27
7	lag17	lag20
8	lag3	lag28
9	lag5	lag21
10	lag9	lag26

Table 4. Variable Importance

Layers	RMSE	MAE	Explainable Variance	Residuals outside 1σ	Residuals outside 2σ
1	0.0037	0.0025	0.4926	19.02%	3.28%
2	0.0037	0.0025	0.5027	20.66%	3.28%
3	0.0037	0.0025	0.4864	19.34%	2.95%

Table 5. Metrics for RV LSTM models

Layers	Lags/Neurons	32	64	128	256
1	**1**	0.003164	0.003164	0.003146	0.003236
	5	0.003129	<u>0.003085</u>	0.003105	0.003149
	22	0.003229	0.003236	0.003219	0.003399
	29	0.003245	0.003251	0.003386	0.003336
2	**1**	0.003179	0.003213	0.003141	0.003197
	5	0.003208	0.003146	<u>0.003109</u>	0.003205
	22	0.003276	0.00345	0.003262	0.003595
	29	0.003451	0.003315	0.00341	0.003376
3	**1**	0.003197	0.003158	0.003159	0.003156
	5	0.003209	0.00317	<u>0.003126</u>	0.003163
	22	0.003384	0.003442	0.003607	0.003268
	29	0.003341	0.003318	0.003268	0.003192

Table 6. Validation test results (RMSE) for RV. Tends to perform best on models fewer lags, while the amount of neurons varies. Underlined results are the best for each amount of layers, used on the test set.

Layers	RMSE	MAE	Explainable Variance	Residuals outside 1σ	Residuals outside 2σ
1	0.0036	0.0024	0.3691	16.72%	4.59%
2	0.0036	0.0024	0.3787	16.72%	5.57%
3	0.0036	0.0025	0.3703	17.05%	5.57%

Table 7. Metrics for LPSD LSTM models

Layers	Lags/Neurons	32	64	128	256
1	1	0.002981	0.003012	0.002951	0.00307
	5	0.002911	0.002857	0.002896	0.002861
	22	0.002981	0.003004	0.003034	0.003117
	29	0.002959	0.002973	0.003193	0.003137
2	1	0.002961	0.003007	0.002909	0.00295
	5	0.002887	0.002893	0.002917	0.00299
	22	0.002899	0.002934	0.002984	0.002965
	29	0.003014	0.002942	0.003075	0.003056
3	1	0.002964	0.002949	0.002956	0.002938
	5	0.002959	0.00294	0.00292	0.002958
	22	0.002994	0.003031	0.002928	0.003191
	29	0.00303	0.002955	0.002949	0.003002

Table 8. Validation test results (RMSE) for LPSD. Tends to perform best on models fewer lags, while the amount of neurons varies. Underlined results are the best for each amount of layers, used on the test set.

	classic HAR	Bagged HAR
RMSE (Train + Test)	0.00456	0.00457
RMSE (Test Set)	0.00393	0.00390

Table 9. DM Test P-values for Each Pair of Models

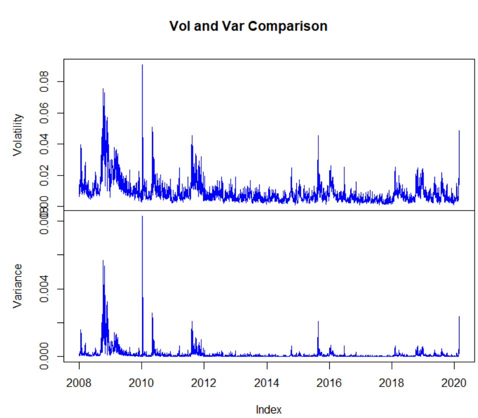
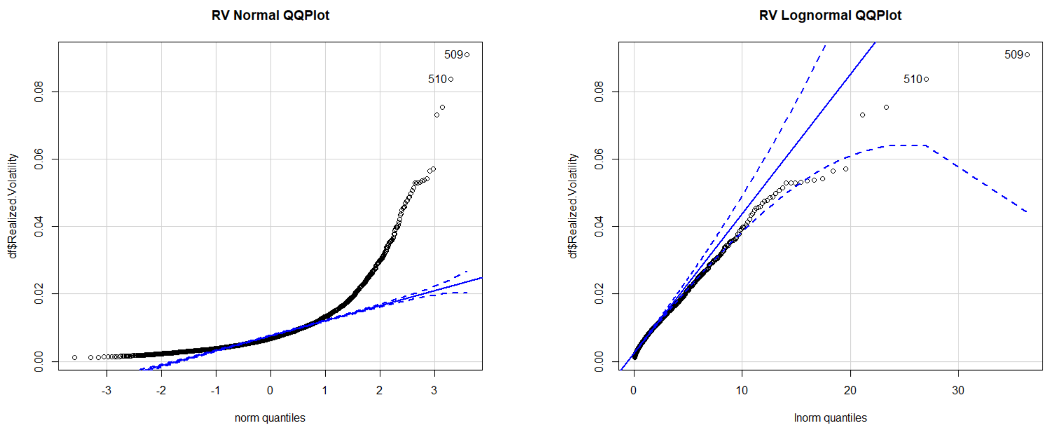


Figure 1. Realized Volatility and Realized Variance. Clustering clearly seen in the figure



(a) Normal QQ (b) Lognormal QQ

Figure 2. Normal and LogNormal QQPlot Comparison. RV is clearly not normally distributed.

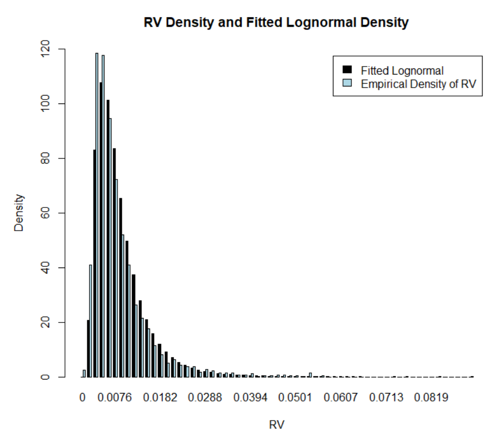


Figure 3. Lognormal provides decent fit on the empirical data.

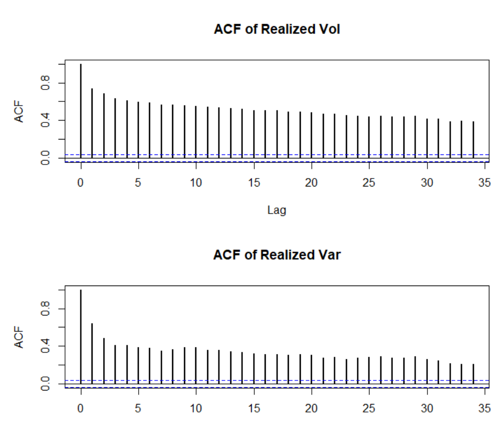


Figure 4. Slowly decreasing ACF. Correlations still significant until lag 35

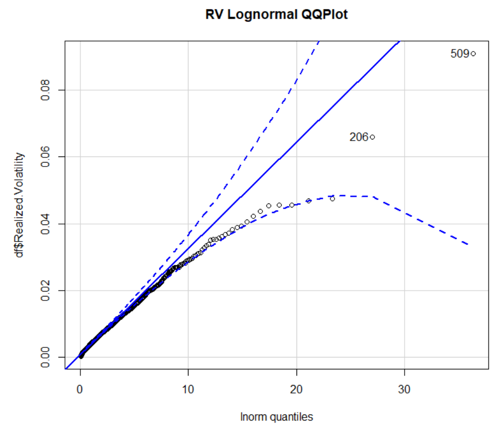


Figure 5. Lognormal QQ Plot. Empirical Quantile deviate away from Theoretical Quantile.

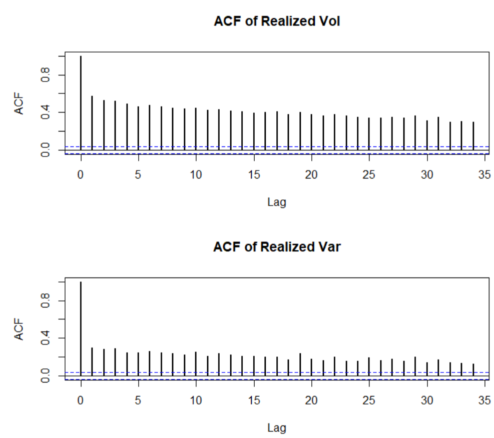


Figure 6. Slowly decreasing ACF. Correlations still significant until lag 35

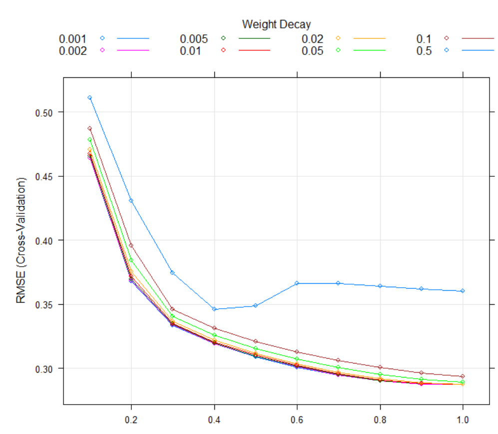


Figure 7. Elastic Net Tuning. Best model chosen is LASSO with $\lambda = 0.005$

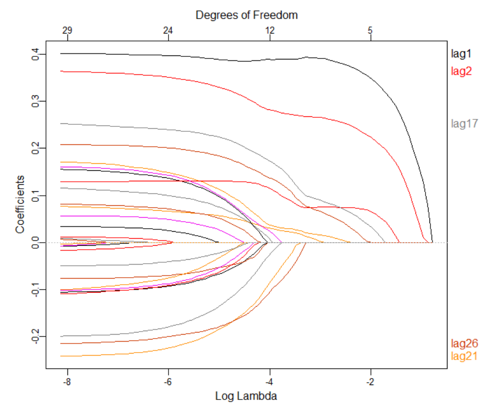


Figure 8. LASSO Tuning.

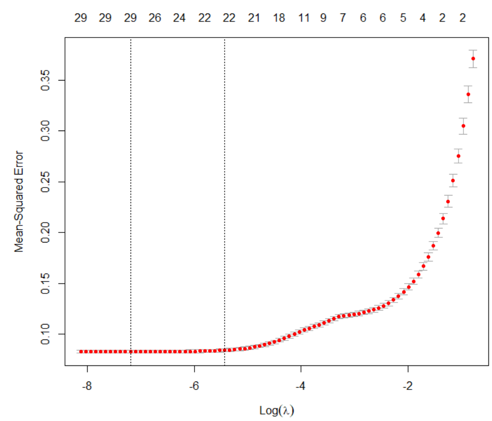


Figure 9. LASSO CV Error. CV error will not increase much until about 20 predictors left.

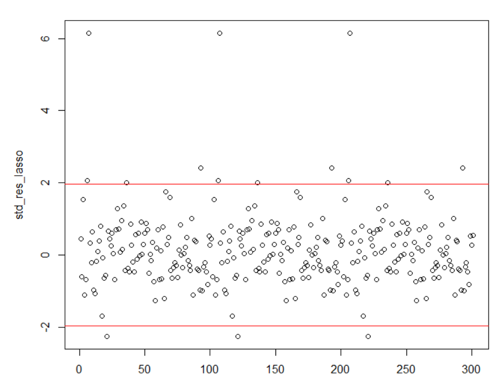


Figure 10. Standardized Residual in LASSO model. Mostly within 2σ.

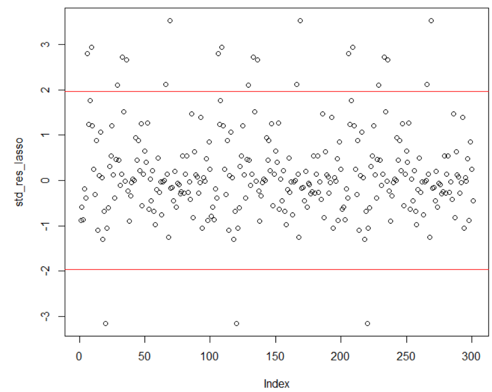


Figure 11. Standardized Residual in LASSO model for LPSD. The model is underestimating the realized value.

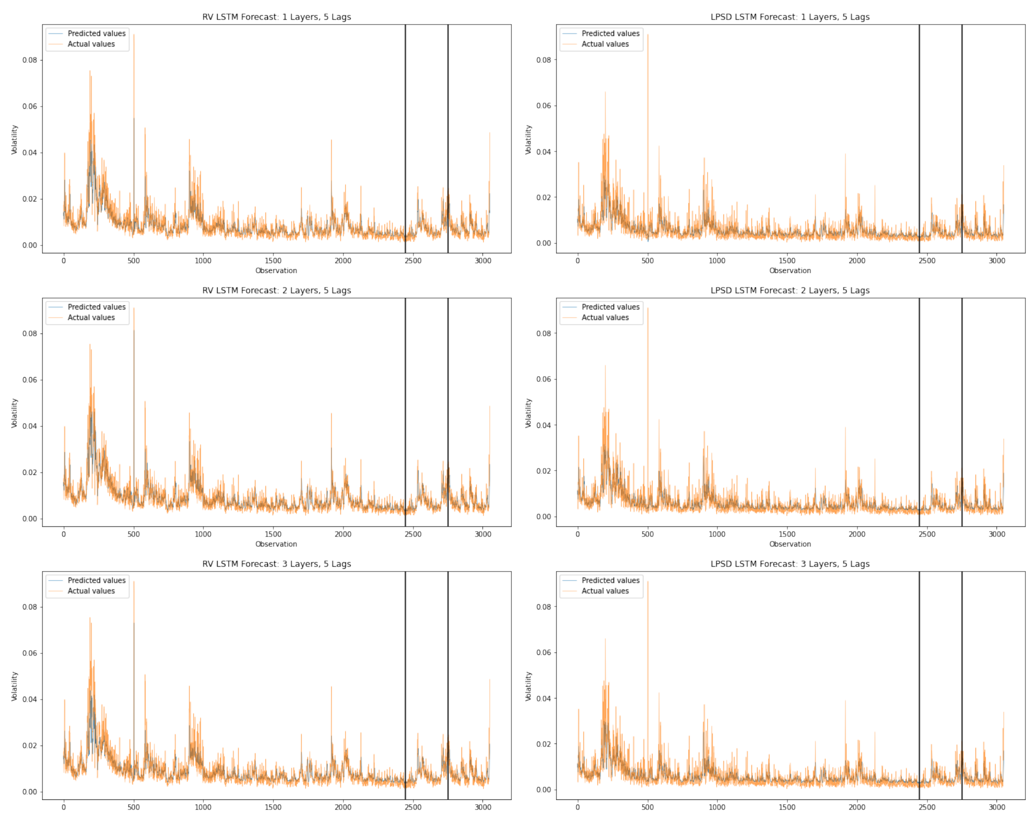


Figure 12. LSTM model predictions vs actual values, separated into training, validation, and test data

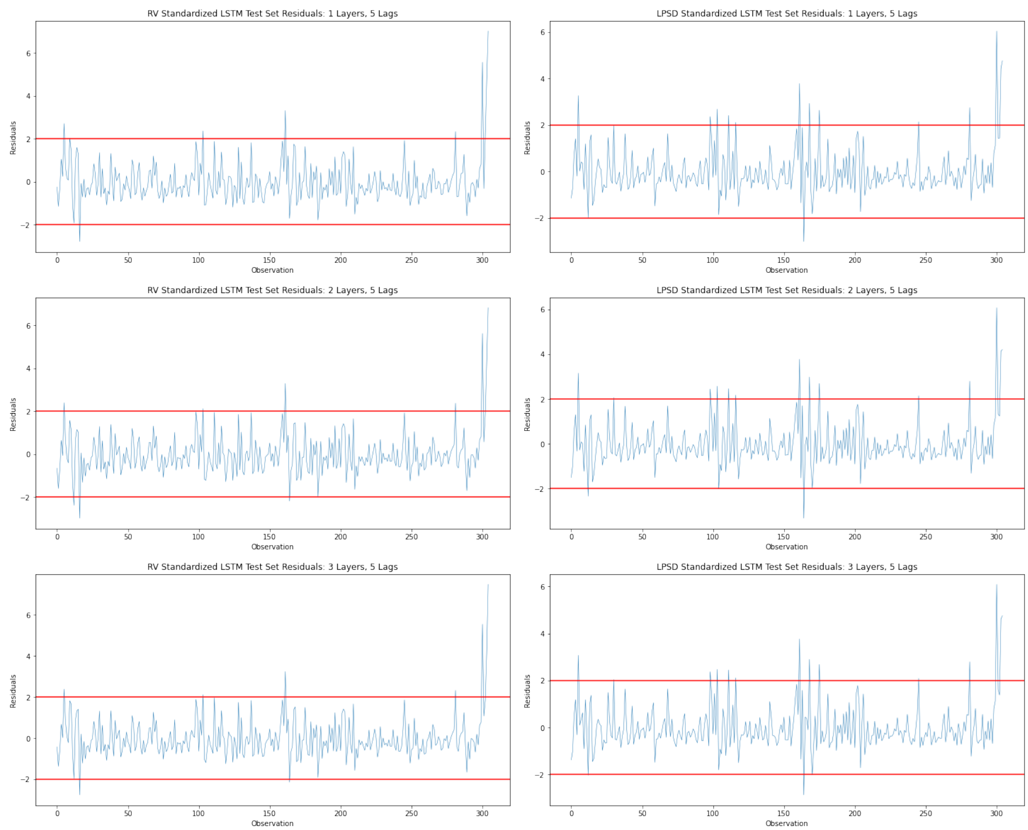


Figure 13. LSTM model Test set residuals

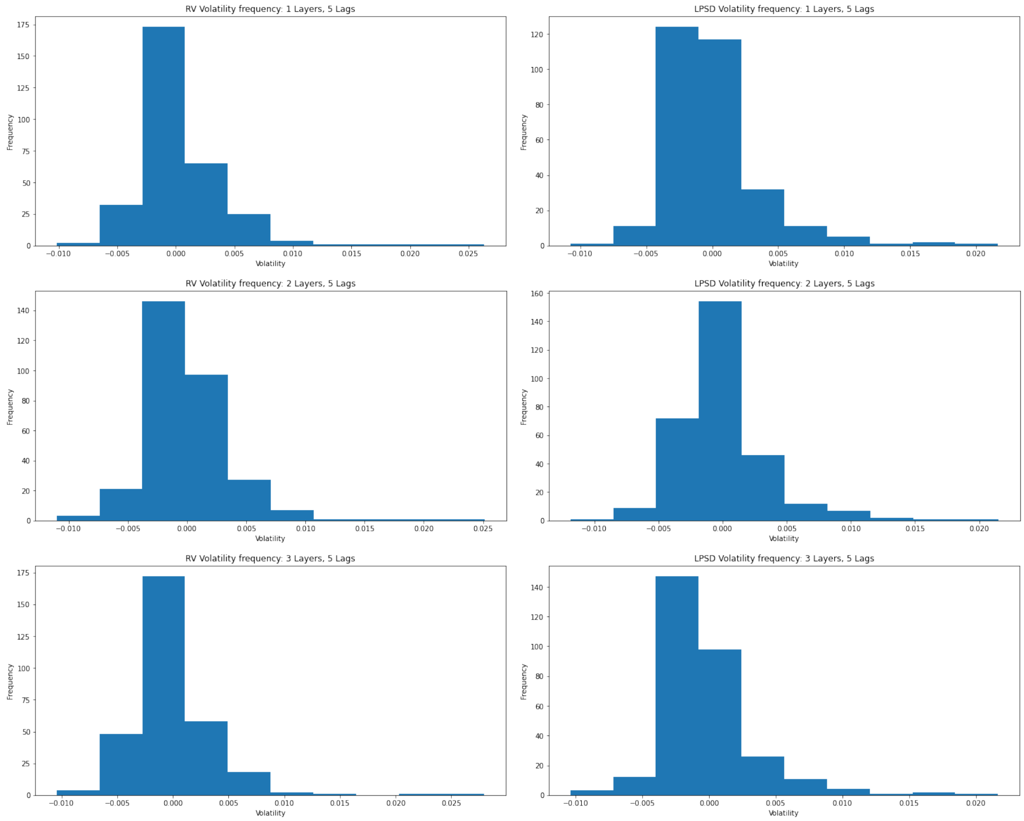


Figure 14. Histograms of LSTM residuals. These tend to be slightly negative, although most results are near the mean

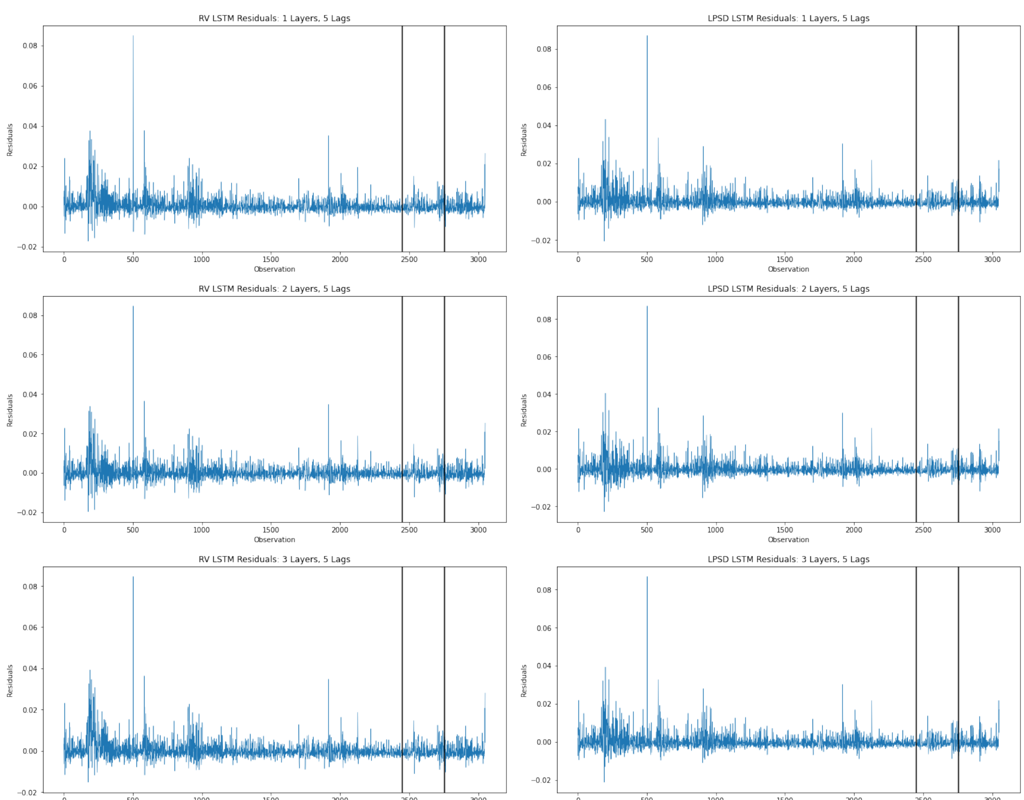


Figure 15. LSTM model residuals

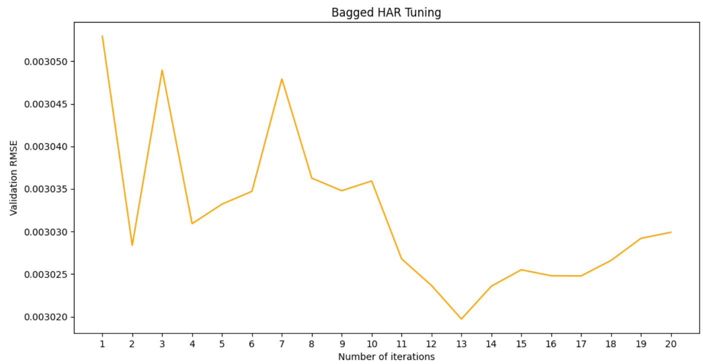


Figure 16. RMSE over Validation set for bagged HAR; 13 appears to be the number of iterations corresponding to the lowest validation RMSE

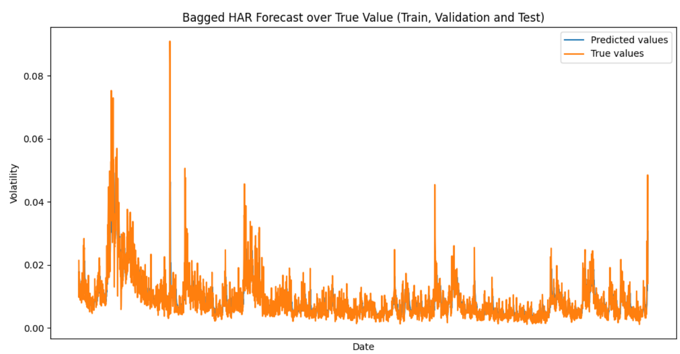


Figure 17. Bagged HAR Forecast over True Value (Train and Test)

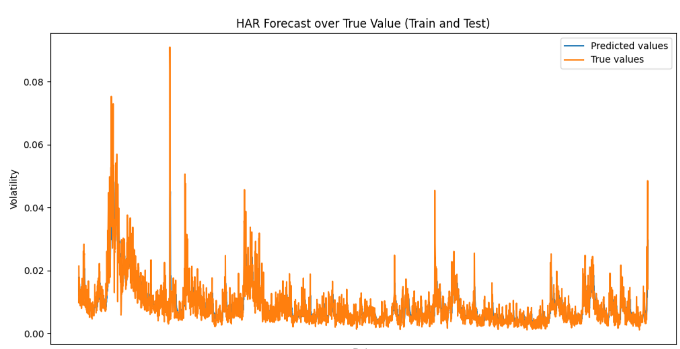


Figure 18. HAR Forecast over True Value (Train and Test)

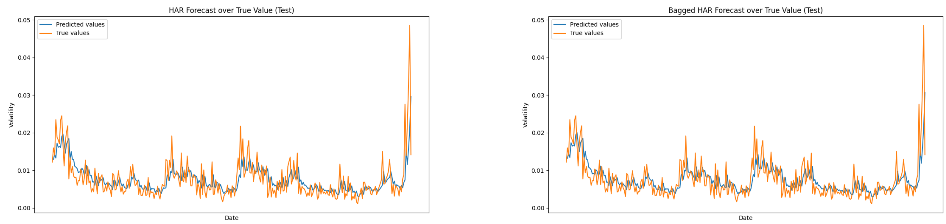


Figure 19. Classic and Bagged HAR Forecast over True Value (Test Set)

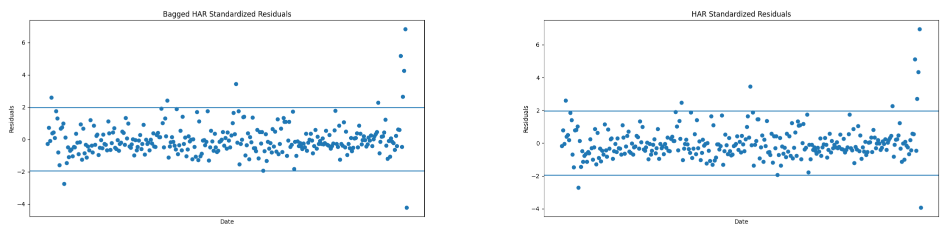


Figure 20. Standardized Residuals for Classic and Bagged HAR (Test Set)

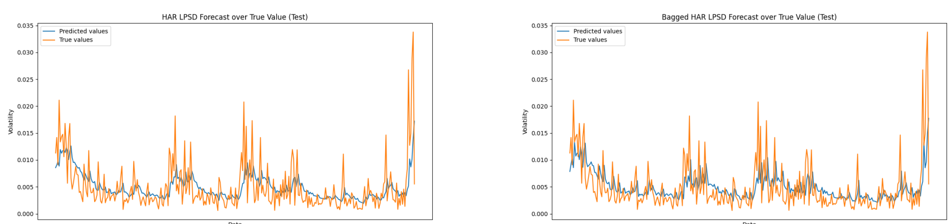


Figure 21. HAR LPSD Forecast over True Value (Test Set)

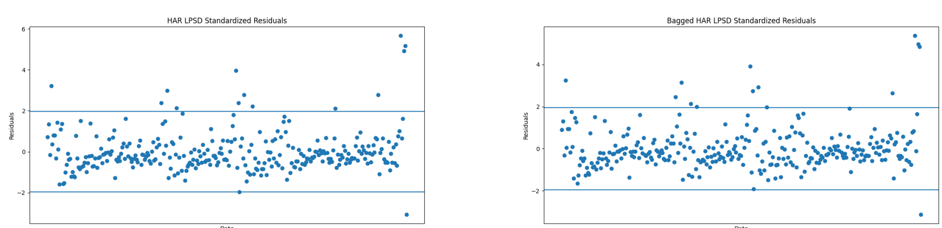


Figure 22. LPSD Standardized Residuals for HAR and Bagged HAR (Test Set)

B Relevant R Snippets and Results

B.1 JB Test

```r
jarque.bera.test(df$Realized.Volatility)
```

```
Jarque Bera Test
data:  df$Realized.Volatility
X-squared = 59435, df = 2, p-value < 2.2e-16
```

B.2 Box Test

```
Box.test(df, type="Ljung-Box", lag = 21)
```

```
Box-Ljung test
data:  df
X-squared = 20157, df = 21, p-value < 2.2e-16
```

B.3 6σ points

```
row.names(tail(df,.1*nrow(df)))[which(std_res_lasso >3)]
```

```
"2018-12-20" "2019-05-16" "2019-10-08"
```

C LPSD

The daily realized Lower Partial Standard Deviation is calculated as below.

$$RLPSD = \sqrt{\sum_i r_i^2 \cdot \mathcal{I}(r_i < 0)}$$

where r_i's are the high frequency log returns.

References

Andersen, T., T. Bollerslev, F. Diebold, and H. Ebens. 2001. "The Distribution of Stock Return Volatility." *Journal of Financial Economics* 61 (July): 43–76. doi:10.1016/S0304-405X(01)00055-1.

Audrino, F., C. Huang, and O. Okhrin. 2019. "Flexible HAR Model for Realized Volatility." *Studies in Nonlinear Dynamics & Econometrics* 23 (August). doi:10.1515/snde-2017-0080.

Bloomberg. 2020. *S&P500 Trading Data.*

Corsi, F. 2009. "A Simple Long Memory Model of Realized Volatility." *Journal of Financial Econometrics* 7 (February): 174–196. doi:10.1093/jjfinec/nbp001.

Diebold, F., and R. Mariano. 1995. "Comparing predictive accuracy[J]." *J. Bus. Econ. Statist.* 13 (January): 134–144.

Friedman, J., T. Hastie, and R. Tibshirani. 2010. "Regularization Paths for Generalized Linear Models via Coordinate Descent." *Journal of Statistical Software* 33 (1): 1–22. http://www.jstatsoft.org/v33/i01/.

Hoerl, A., and R. Kennard. 1970. "Ridge Regression: Biased Estimation for Nonorthogonal Problems." *Technometrics* 12:55–67. doi:10.1080/00401706.1970.10488634.

Kuhn, M. 2020. *caret: Classification and Regression Training.* R package version 6.0-85. https://CRAN.R-project.org/package=caret.

McAleer, M., and M. Medeiros. 2010. "Forecasting Realized Volatility with Linear and Nonlinear Univariate Models." *University of Canterbury, Department of Economics and Finance, Working Papers in Economics* 25 (January). doi:10.1111/j.1467-6419.2010.00640.x.

Srivastava, N., G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. 2014. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting." *Journal of Machine Learning Research 15* (June). http://jmlr.org/papers/volume15/srivastava14a.old/srivastava14a.pdf.

Tibshirani, R. 1996. "Regression Shrinkage and Selection Via the Lasso." *Journal of the Royal Statistical Society: Series B (Methodological)* 58 (January): 267–288. doi:10.1111/j.2517-6161.1996.tb02080.x.