Forecasting South African Financial Market Volatility with Machine Learning

Jonathan Rossouw[1]

Abstract

Volatility modelling is an important problem in financial econmetrics. The recent proliferation of powerful machine learning techniques offer models that are well suited to this complex problem. In this paper volatility forecasting of the JSE All Share index for different machine learning techniques are compared. The Support Vector Regression (SVR) and Long-Short Term Memory Recurrent Neural Networks (LSTM) are compared. An EGARCH models is used as a baseline model. The LSTM model performed best at one-day ahead and the SVR at three-day ahead forecasting.

1. Introduction

Volatility modelling is an important and complex problem in financial econometrics. The volatility the returns of an asset are an important measure for the risk of an asset. Volatility itself is a key factor in options pricing and in asset allocation (Tsay, 2012). The Value-at-Risk calculations made for risk management rely on measures of volatility. There has been a recent proliferation of machine learning techniques that have been applied to financial data includig volatility forecasting (Henrique, Sobreiro & Kimura, 2019). The traditional forecasting techniques build on the generalized autoregressive conditional heteroscedastic (GARCH) class of models are well suited for uncovering the true patterns of volatility. However, their ability to accurately forecast volatility is limited.

The use of machine learning techniques including Support Vector Regression (SVR) and artificial neural network models has grown in recent years and have improved the precision of volatility forecasting Sheta, Ahmed & Faris (2015). Recently proposed volatility models using the long-short term memeroy recurrent neural network (LSTM) archiceture have futher improved forecasting precision (Liu, 2019). While these models have been used to model volatility in the US, only the SVR and GARCH models have been applied to South African return volatility forecasting, metal price forecasting and exchange rates forecasting Chai, Zhao, Hu & Zhang (2021).

In this paper, volatility of the JSE ALSI total returns index (TRI) is modelled. This paper finds that the GARCH models perform poorly on pure forecasting precision. The SVR and LSTM models perform similarly well on both one-day and three-day ahead forecasting. For one-day ahead forecasting

Email address: 20858345@sun.ac.za (Jonathan Rossouw)

the LSTM is marginally better while for three-day ahead forecasting the SVR is marginally better than the LSTM model.

The remainder of the paper is organised as follows: the data and methodology are described, the results are analysed followed by the conclusion.

2. Methodology

The methodology section is split into a description of the data, a description of each class of model and the sets taking in analysis.

2.1. Data

The data used in this paper were the total return index (TRI) of the JSE ALSI. The TRI is price adjusted for dividends, stock splits and other corporate actions. The data was split into training, validation and test series. The training set was from 05-01-2009 to 30-12-2016, the validation set was from 03-01-2017 to 31-12-2018, and the test set was from 03-01-2019 to 31-12-2019. Dlog returns were calculated using the log difference of TRI given in equation 2.1.

$$r_t = \ln(TRI_t) - \ln(TRI_{t-1}) \tag{2.1}$$

2.2. GARCH

The returns series often follow a mean equation given by

$$r_t = \alpha + \mu_t + \varepsilon_t \tag{2.2}$$

which can be modelled using an ARIMA class of models. The residual, ε_t, has $E(\varepsilon_t) = 0$ and constant unconditional $E(\varepsilon_t^2) = c$. This means the equation 2.2 has homoscedastic errors. However, conditionally the equation 2.2 displays heteroscedastic errors since there are periods of autocorrelation displayed by the error variance (Engle, 1982). The GARCH class of models apply a ARMA type of model to the squared residuals of the mean equation. This allows for the residual to be multiplicatively decomposed into a white noise, η_t component and a cleaned volatility compoenent h_t. The GARCH formula is given by

$$y_t = \alpha + \mu_t + \varepsilon_t$$

$$\varepsilon_t = h_t \eta_t$$

$$h_t = \sqrt{\alpha_0 + \sum_{i=1}^{p} \beta_i h_{t-i}^2 + \sum_{i=1}^{q} \alpha_i \varepsilon_{t-i}^2} \;\; ; \qquad \eta_t \sim N(0,1)$$

$$\text{with constants:} \quad \alpha_0 > 0, \;\; \alpha_i > 0, \;\; \beta_i > 0$$

$$0 < \sum_{i=1}^{p} \beta_i + \sum_{i=1}^{q} \alpha_i < 1$$

The unconditional mean of the residual is given by

$$E_t(\varepsilon_t) = 0$$

and the unconditional variance is given by the following constant

$$E_t(\varepsilon_t^2) = \frac{\alpha_0}{1 - (\sum_{i=1}^{p} \beta_i + \sum_{i=1}^{q} \alpha_i)}$$

with $0 < \sum_{i=1}^{p} \beta_i + \sum_{i=1}^{q} \alpha_i < 1$. The conditional variance is given by

$$E_{t-1}(\varepsilon_t^2) = h_t^2 = \alpha_0 + \sum_{i=1}^{p} \beta_i h_{t-i}^2 + \sum_{i=1}^{q} \alpha_i \varepsilon_{t-i}^2$$

Many additions have been made to the original GARCH formulation. This includes the exponential GARCH (EGARCH) model. EGARCH controls for the assyemtry in volatility between positive and negative movements in the returns series (Nelson, 1991). The EGARCH model is given by

$$\ln(h_t^2) = \beta_0 + \beta_1 \ln(h_{t-1}^2) + \gamma_1 \cdot \frac{\varepsilon_{t-1}}{h_{t-1}} + \gamma_2 \left\{ \left| \frac{\varepsilon_{t-1}}{h_{t-1}} \right| + \sqrt{\frac{2}{\pi}} \right\}$$

with the assumption of a normally distributed noise component, $E\left(\frac{\varepsilon_{t-1}}{h_{t-1}}\right) = \sqrt{\frac{2}{\pi}}$.

2.3. LSTM

The LSTM architecture specifically refers to a type of recurrent cell designed specifically for sequential data (Gers, Schmidhuber & Cummins, 2000). The LSTM can be viewed as a type of encoder that encodes the features of a sequence that can then be fed to a dense multilayer perceptron which then produces predictions or forecasts. The LSTM is special in that along with taking in the current observation and the encoded previous observation it also keeps track of the cell state of the previous cell. Together the three inputs are fed through gates to determine the importance of each input in

the encoding of the current observation. The output is an encoded observation and the current cell state. The steps of a single LSTM cell are give by

Step a: compute forget and input gates

$$\text{forget gate: } \mathbf{g}^{(f)} = \boldsymbol{\sigma}_{sig}(\mathbf{b}^{(f)} + \mathbf{V}^{(f)}\mathbf{h}^{-} + \mathbf{W}^{(f)}\mathbf{x}) \tag{2.3}$$

$$\text{input gate: } \mathbf{g}^{(i)} = \boldsymbol{\sigma}_{sig}(\mathbf{b}^{(i)} + \mathbf{V}^{(i)}\mathbf{h}^{-} + \mathbf{W}^{(i)}\mathbf{x}) \tag{2.4}$$

Step b: compute the Elman update

$$\mathbf{h}^{+} = \boldsymbol{\sigma}_{tanh}(\mathbf{b} + \mathbf{V}\mathbf{h}^{-}\mathbf{W}\mathbf{x}) \tag{2.5}$$

Step c: compute output gate

$$\text{output gate: } \mathbf{g}^{(o)} = \boldsymbol{\sigma}_{sig}(\mathbf{b}^{(o)} + \mathbf{V}^{(o)}\mathbf{h}^{-} + \mathbf{W}^{(o)}\mathbf{x}) \tag{2.6}$$

Step d: update cell state

$$\mathbf{c}^{+} = \mathbf{g}^{(f)} \odot \mathbf{c}- + \mathbf{g}^{(i)} \odot \mathbf{h}^{+} \tag{2.7}$$

Step d: update memory

$$\mathbf{h}^{+} \leftarrow \mathbf{g}^{(o)} \odot \boldsymbol{\sigma}_{tanh}(\mathbf{c}^{+}) \tag{2.8}$$

where $\mathbf{V}, \mathbf{W}$ are weight matrices, $\mathbf{b}$ are bias vectors, $\boldsymbol{\sigma}$ are activation functions, $\mathbf{h}$ is the hidden or encoded state, $\mathbf{c}$ is the cell state and $\mathbf{x}$ is the current observation. Figure 2.1 shows a diagram of a single LSTM cell.

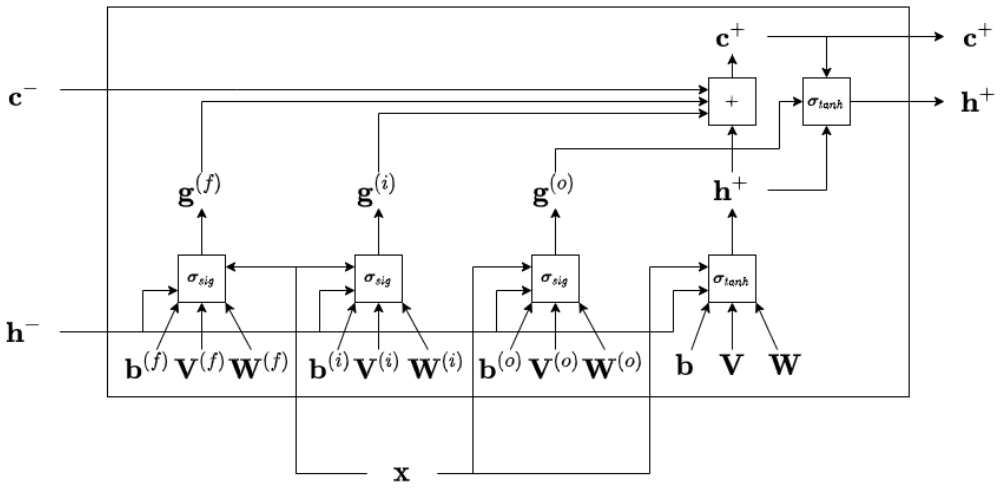

Figure 2.1: LSTM model architecture

LSTM models are highly flexible as the LSTM only refers to a single cell that can be used in any configuration of neural network architecture. The following hyperparameters, amongst others, can be tuned: the loss function, the number of LSTM layers, the number of LSTM units in each layer, the dropout rate, the number of epochs, and whether the first LSTM layer outputs a sequence. The LSTM models were implemented using the Keras API for Tensorflow.

2.4. SVR

The SVR is an extension to the Support Vector Machine for regression problems. The SVR attempts to fit a "hyper-tube" in high dimensional space around the true function which is analogous to the SVM hyperplane (Vapnik, 1999). Observations within the "hyper-tube" are not penalised while observations on the "hyper-tube" or outside the "hyper-tube" are penalised and become the support vectors for the "hyper-tube". By applying a basis expansion through the use of a kernel function, the "hyper-tube" can be expressed in a higher dimensional function space. This allows for extremely non-linear "hyper-tubes" to be formed.

The SVR can be expressed as a Lagrangian optimisation problem. The primal optimisation problem is given as

$$\begin{aligned} \underset{\beta_0,\boldsymbol{\tau},\boldsymbol{\xi},\boldsymbol{\xi^*}}{\text{minimize}} \quad & \tfrac{1}{2}\boldsymbol{\tau'K\tau} + C\textstyle\sum_{i=1}^{N}(\xi_i + \xi_i^*) \\ \text{subject to} \quad & \beta_0 + \textstyle\sum_{j=1}^{N}\tau_j K(\mathbf{x}_i,\mathbf{x}_j) - y_i \leq \varepsilon + \xi_i \text{ for all } i = 1,2,...,N \\ & y_i - \beta_0 - \textstyle\sum_{j=1}^{N}\tau_j K(\mathbf{x}_i,\mathbf{x}_j) \leq \varepsilon + \xi_i^* \text{ for all } i = 1,2,...,N \\ & \xi_i \geq 0 \text{ for all } i = 1,2,...,N. \end{aligned} \tag{2.9}$$

The dual optimisation problem is given by

$$\begin{aligned} \underset{\boldsymbol{\alpha},\boldsymbol{\alpha^*}}{\text{maximize}} \quad & -\varepsilon\mathbf{1}'(\boldsymbol{\alpha}+\boldsymbol{\alpha^*}) + \mathbf{y}'(\boldsymbol{\alpha^*}-\boldsymbol{\alpha}) - \tfrac{1}{2}(\boldsymbol{\alpha^*}-\boldsymbol{\alpha})'\mathbf{G}(\boldsymbol{\alpha^*}-\boldsymbol{\alpha}) \\ \text{subject to} \quad & 0 \leq \alpha_i, \alpha_i^* \leq C; \text{ for all } i = 1,2,...,N \\ & \textstyle\sum_{i=1}^{N}(\alpha_i^* - \alpha_i) = 0 \end{aligned} \tag{2.10}$$

where $\mathbf{G}_{ij} = K(\mathbf{x}_i,\mathbf{x}_j)$. The optimisation of the SVR can be achieved through quadratic programming. In the implementation of liquidSVM, only a Gaussian radial basis function kernel is possible. The hyperparameters for the SVR are $\lambda = \frac{1}{C}$ from equation 2.9. Additionally, the hyperparameter γ is given as the bandwidth of the kernel.

2.5. Analysis

Since the true volatility, σ_t, of the returns series is not observable, a proxy was needed (Sun & Yu, 2020). Following the framework of (Li, Liang, Li, Wang & Wu, 2009), a proxy of the volatility was calculated as the following

$$\sigma_t = \sum_{i=0}^{4} r_{t-i} \tag{2.11}$$

which is the 5-day rolling average of the squared return. This was used as the target variable in the machine learning models. A one-day ahead forecast and a three-day ahead forecast, where there was an interval of two days between the inputs and the target variable, were performed.

In order to test whether a GARCH class of models was appropriate, the autoregressive nature of the returns series, r_t, squared returns series r_t^2 and the absolute value of the returns series, $|r_t|$, were tested using the autocorrelation functions (ACFs). Once the autoregressive nature of the squared residuals was determined, a AR(1) model was fit tp the return series for the mean equations. The residuals ε_t were modelled using various GARCH models. The AIC was used to determine which model fit best. The best fitting model was used to forecast one-day ahead using the training and validation sets. The performance of the forecast was determine using the MSE given as

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(\hat{\sigma}_t - \sigma_t)^2$$

where $\hat{\sigma}_t$ is the forecasted volatility at time t.

The machine learning techniques were used for one-day ahead forecasts and three-day ahead forecasts. For the one-day ahead forecast, the machine learning models were trained using the $[\sigma_{t-1}, \varepsilon_{t-1}^2]$ where ε_{t-1} is the residual of an AR(1) process of r_t (Bezerra & Albuquerque, 2017). For the three day ahead forecasts, the machine learning models were trained using the $[\sigma_{t-1}, \sigma_{t-2}, \varepsilon_{t-1}^2, \varepsilon_{t-2}^2]$. The performance of the forecasts of the machine learning models were determined by MSE. The machine learning techniques were tuned for the appropriate combination of hyperparameter using a grid-search. The values for the grid search are given in table 2.1.

Table 2.1: Hyperparameter tuning grid

Class of Models	Hyperparameters
SVR	gamma: 0.082, 0.368, 1.649, 7.389, 33.115, 148.413, 665.142, 2980.958 lambda: 0.082, 0.368, 1.649, 7.389, 33.115, 148.413, 665.142, 2980.958
LSTM	epochs: 10, 20, 50 loss: mse, mae lstm layers: 1, 2 lstm units: 10, 20, 40 return sequences: TRUE, FALSE dropout rate: 0, 0.1

The performance of the validation forecasts were used to determine the best hyperparameter combination. The best combinations for each class of model were compared with the best performing model being selected for the test set forecast. The forecast performance of the test set is given as the overall performance.

3. Results

Following the plots in figure 3.1, there appears to be periods of autoregression in the squared and absolute values of the returns series. The ACFs are given in figure 3.2. The Lung-Box-Pierce Test for has a test statistic of 632.57 and a p-value of 0. The null hypothesis of no conditional heteroscedasticity in the squared returns series was rejected. The LBQ test and GARCH-LM tests were not conducted.

An AR(1) model was used for the mean equation. Table 3.1 shows the results of a number of different GARCH class models. Each model was fit with GARCH(1,1) specification. The EGARCH(1,1) model had the lowest AIC of -6.4901. The coefficients of the EGARCH(1,1) fit to the validation set are given in table 3.2. Plots of the EGARCH forecasts are given in the appendix.

The best hyperpameter combinations for both the one-day ahead and three-day ahead LSTM models were: mse, 20 epochs, 2, layers, 40 units, return sequence and dropout rate of 0. The combination of hyperparameters had no effect on the SVR models thus the hyperparameters were both set to 0.082.

The best performing model from each class of models is given in table 3.3. The EGARCH(1,1) model was only used for the one-day ahead forecasts. The LSTM model had the lowest MSE for the one-

day ahead forecasting while the SVR model had the lowest MSE for the three-day ahead forecasting. Figure 3.3 shows the LSTM models performance on the validation set where the red indicates the forecasted values and the grey is the true volatility. Figure 3.4 shows the SVR models performance on the three-day ahead forecast. Clearly the models were much better and predicting one-day ahead than three-days ahead.

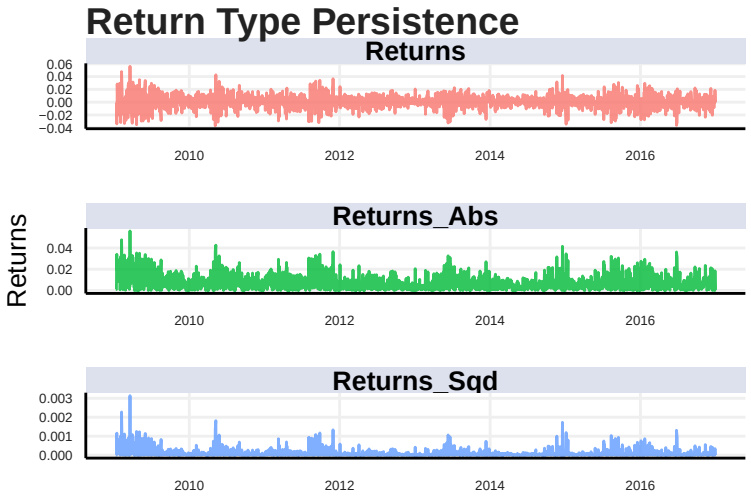

Figure 3.1: Plots of returns, squared returns and absolute value of returns

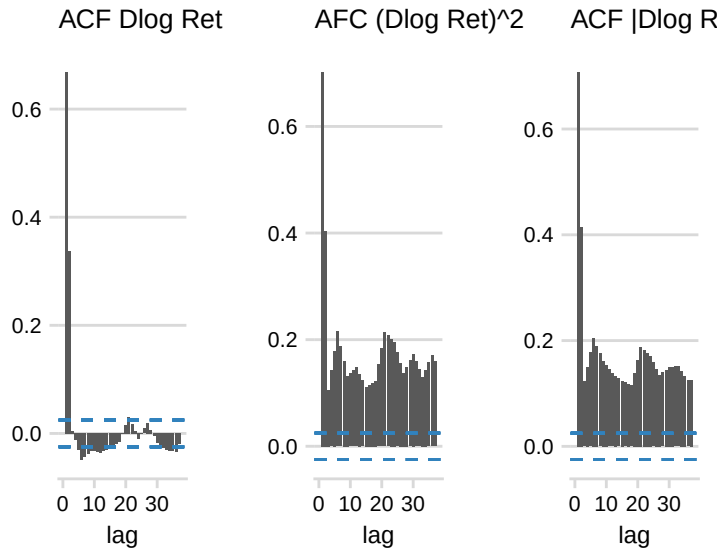

Figure 3.2: ACF plots of returns, squared returns and absolute value of returns

Table 3.1: Selection statistics of different GARCH based models

	sGARCH	gjrGARCH	eGARCH	apARCH
Akaike	-6.452	-6.482	-6.490	-6.476
Bayes	-6.439	-6.466	-6.474	-6.457
Shibata	-6.452	-6.482	-6.490	-6.476
Hannan-Quinn	-6.447	-6.476	-6.484	-6.469

Table 3.2: EGARCH validation model coefficients

	Estimate	Std. Error	t value	Pr(>|t|)
mu	0.0003	0.0003	0.951	0.342
ar1	0.026	0.044	0.599	0.549
omega	-0.324	0.007	-46.199	0
alpha1	-0.149	0.022	-6.639	0
beta1	0.967	0.0001	8,874.838	0
gamma1	0.061	0.027	2.257	0.024
shape	8.276	2.892	2.862	0.004

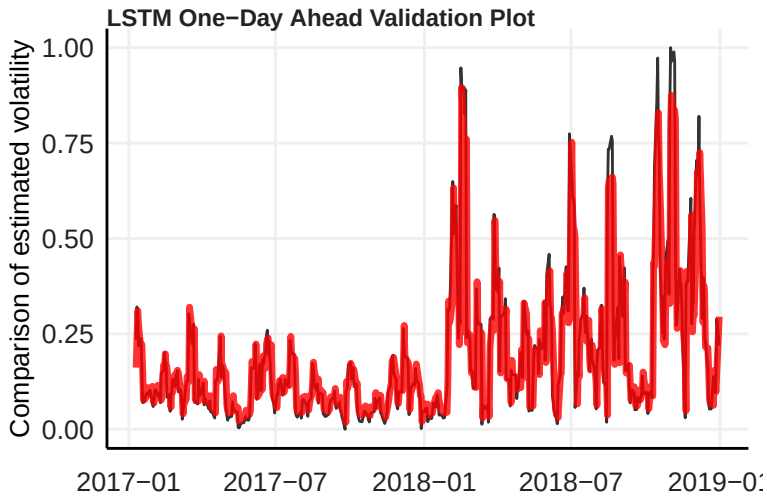

Figure 3.3: LSTM One-Day Ahead Validation Forecast

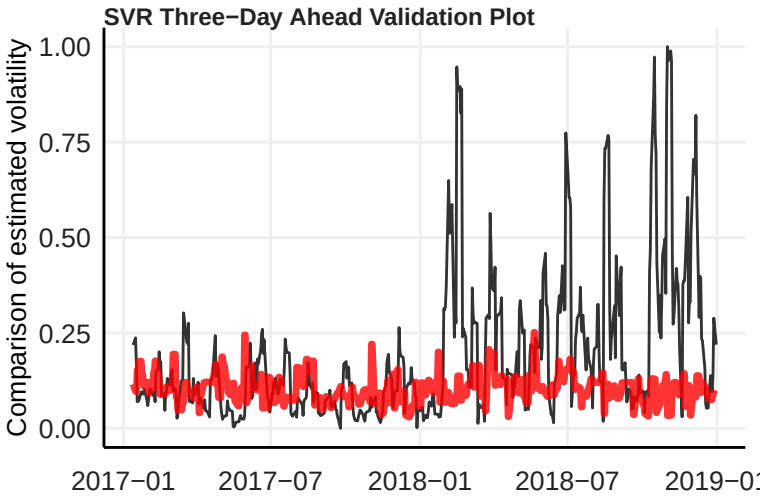

Figure 3.4: SVR Three-Day Ahead Validation Forecast

Table 3.3: Best hyperparameters training and validation performance

Model	Training MSE	Validation MSE
One-day ahead		
EGARCH	0.0115	0.0264
LSTM	0.0035	0.0096
SVR	0.0034	0.0097
Three-day ahead		
LSTM	0.0152	0.0477
SVR	0.0081	0.0462

The results from the best models forecasts using the test set are given in table 3.4. Figure 3.5 shows the LSTM model's one-day ahead test performance. Figure 3.6 shows the SVR model's three-day ahead test performance. The figures display that the one-day ahead forecast was far more precise than the three-day ahead forecasts. This could be due to the weak autocorrelation over that size interval. Thus the inputs three and four days prior to the forecast date having insufficient information. Using a longer inputs series may yield improved results especially for the LSTM which relies on large training data sets. The poor three-day forecasting performance illustrates the difficulty of forecasting volatility far into the future. Recently, new artificial neural network architectures including transformer models have offered potential to improved volatility forecasting results (Ramos-Pérez, Alonso-González &

Núñez-Velázquez, 2021).

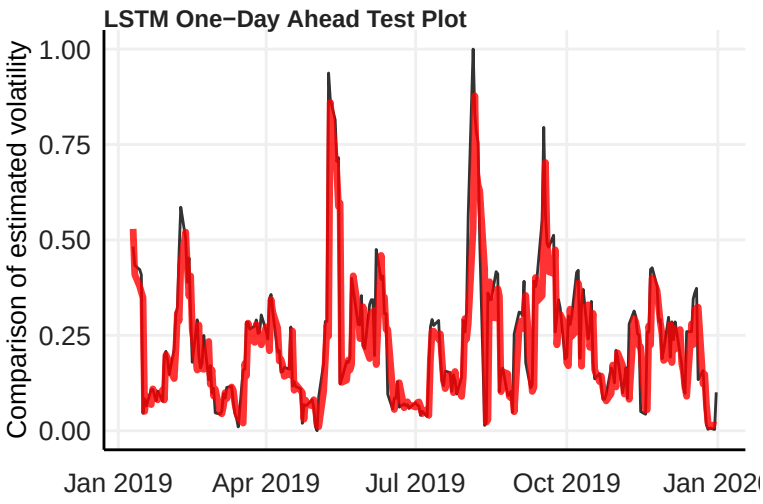

Figure 3.5: LSTM One-Day Ahead Test Forecast

Table 3.4: Forecasting full training and test performance

Model	Full Training MSE	Test MSE
	One-day ahead	
LSTM	0.0031	0.0121
	Three-day ahead	
SVR	0.0037	0.0523

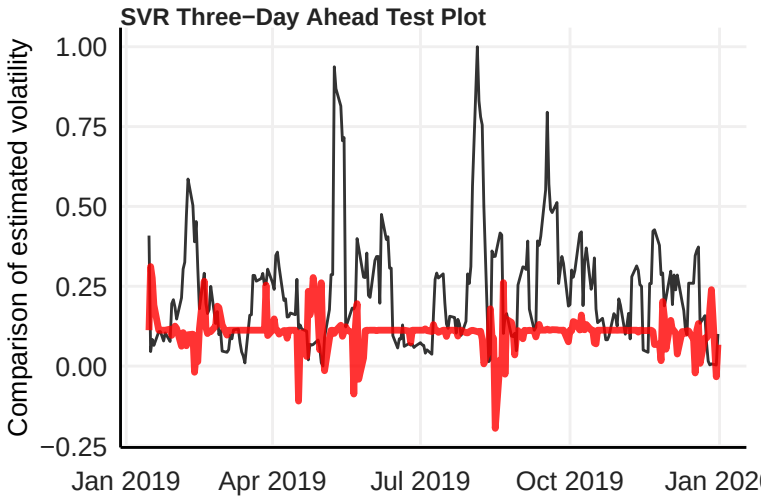

Figure 3.6: LSTM Three-Day Ahead Test Forecast

4. Conclusion

While forecasting returns series is of little use due to the potential for arbitrage opportunities, forecasting volatility is an important task in financial econometrics. In this paper, three different classes of volatility models were compared. For one-day ahead volatility modelling, the LSTM model performs best with the lowest MSE. The plots of the forecasts illustrate a very close fit to the data. For three-day ahead forecasting, the SVR model performs best. However, as illustrated by plots of the forecasts, the three-day ahead forecasts do not perform as well. The novel approach to using these machine learning techniques to the South African context show that their high levels of performance translates to this context. While volatility modelling, especially with long time intervals, still posses a significant challenge, machine learning techniques offer a method for more accurate forecasting.

References

10 Babikir, A., Gupta, R., Mwabutwa, C. & Owusu-Sekyere, E. 2012. Structural breaks and GARCH models of stock return volatility: The case of south africa. *Economic Modelling*. 29(6):2435–2443.

Bezerra, P.C.S. & Albuquerque, P.H.M. 2017. Volatility forecasting via SVR–GARCH with mixture of gaussian kernels. *Computational Management Science*. 14(2):179–196.

Chai, J., Zhao, C., Hu, Y. & Zhang, Z.G. 2021. Structural analysis and forecast of gold price returns. *Journal of Management Science and Engineering*. 6(2):135–145.

Engle, R.F. 1982. Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. *Econometrica: Journal of the econometric society*. 987–1007.

Gers, F.A., Schmidhuber, J. & Cummins, F. 2000. Learning to forget: Continual prediction with LSTM. *Neural computation*. 12(10):2451–2471.

Henrique, B.M., Sobreiro, V.A. & Kimura, H. 2019. Literature review: Machine learning techniques applied to financial market prediction. *Expert Systems with Applications*. 124:226–251.

Ismail, M.T., Audu, B. & Tumala, M.M. 2016. Volatility forecasting with the wavelet transformation algorithm garch model: Evidence from african stock markets. *The Journal of Finance and Data Science*. 2(2):125–135.

Li, N., Liang, X., Li, X., Wang, C. & Wu, D.D. 2009. Network environment and financial risk using machine learning and sentiment analysis. *Human and Ecological Risk Assessment*. 15(2):227–252.

Liu, Y. 2019. Novel volatility forecasting using deep learning–long short term memory recurrent neural networks. *Expert Systems with Applications*. 132:99–109.

Nelson, D.B. 1991. Conditional heteroskedasticity in asset returns: A new approach. *Econometrica*. 59(2):347–370. [Online], Available: http://www.jstor.org/stable/2938260.

Pérez-Cruz, F., Afonso-Rodriguez, J.A. & Giner, J. 2003. Estimating GARCH models using support vector machines. *Quantitative Finance*. 3(3):163.

Ramos-Pérez, E., Alonso-González, P.J. & Núñez-Velázquez, J.J. 2021. Multi-transformer: A new neural network-based architecture for forecasting s&p volatility. *Mathematics*. 9(15):1794.

Sheta, A.F., Ahmed, S.E.M. & Faris, H. 2015. A comparison between regression, artificial neural networks and support vector machines for predicting stock market index. *Soft Computing*. 7(8):2.

Sun, H. & Yu, B. 2020. Forecasting financial returns volatility: A GARCH-SVR model. *Computational Economics*. 55(2):451–471.

Tsay, R. 2012.

Vapnik, V. 1999. *The nature of statistical learning theory*. Springer science & business media.

Appendix

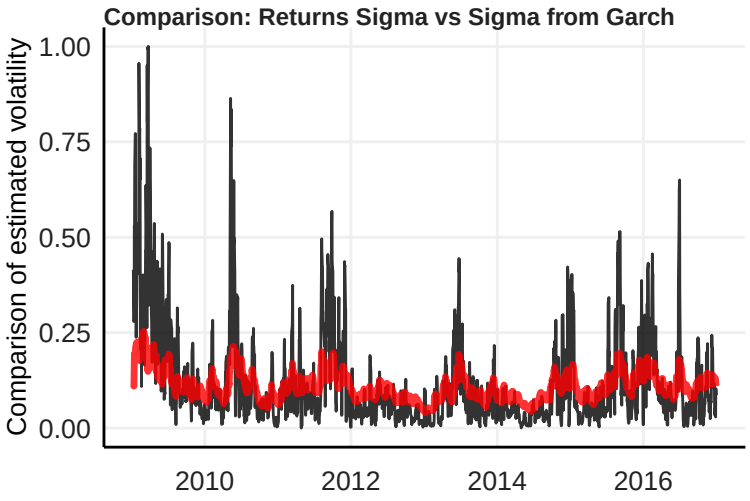

Figure 4.1: EGARCH(1,1) One-Day Ahead Training Forecast

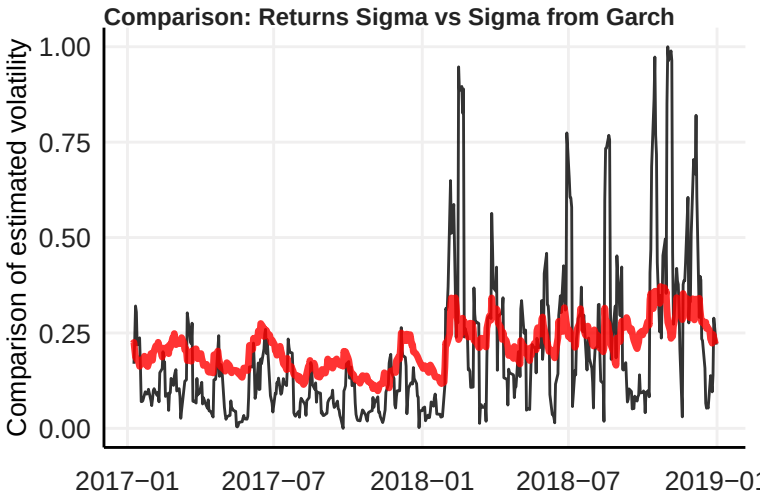

Figure 4.2: EGARCH(1,1) One-Day Ahead Validation Forecast

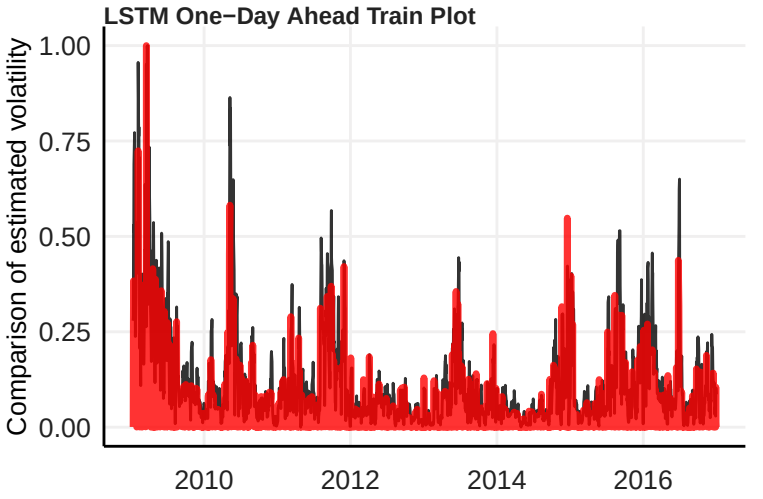

Figure 4.3: LSTM One-Day Ahead Training Forecast

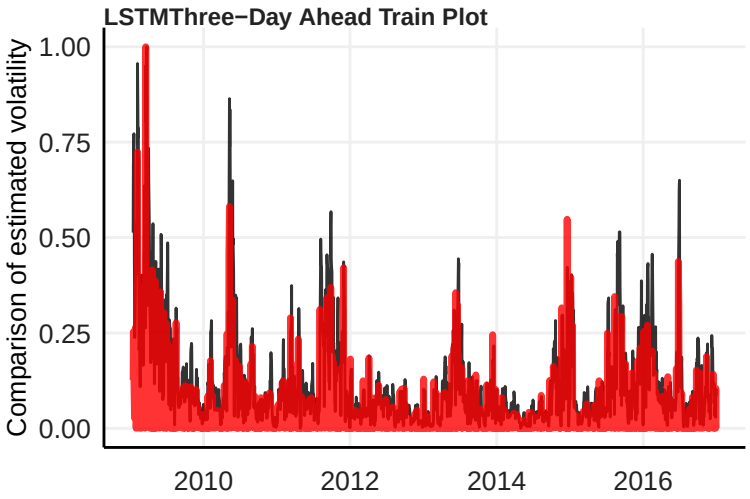

Figure 4.4: LSTM Three-Day Ahead Training Forecast

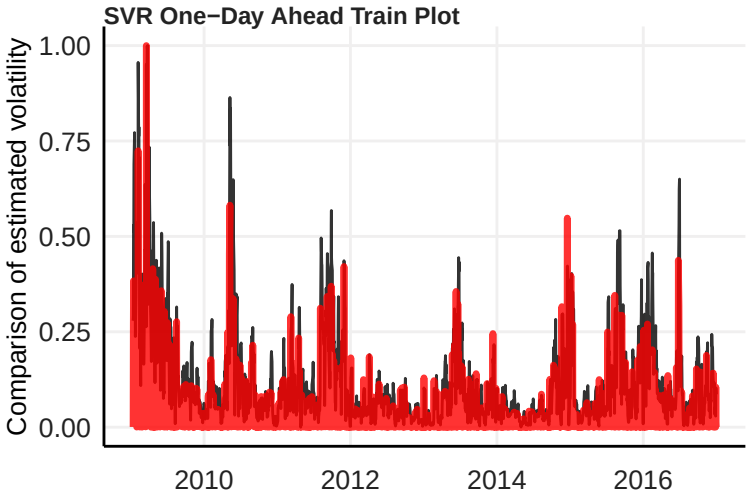

Figure 4.5: SVR One-Day Ahead Training Forecast

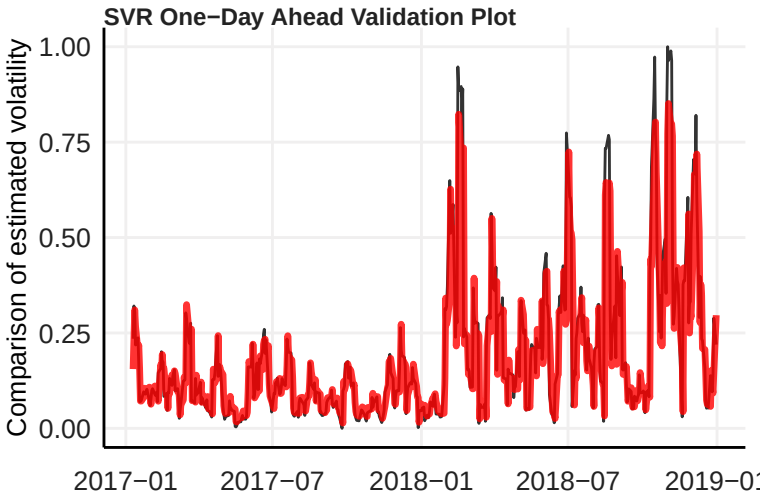

Figure 4.6: SVR One-Day Ahead Validation Forecast

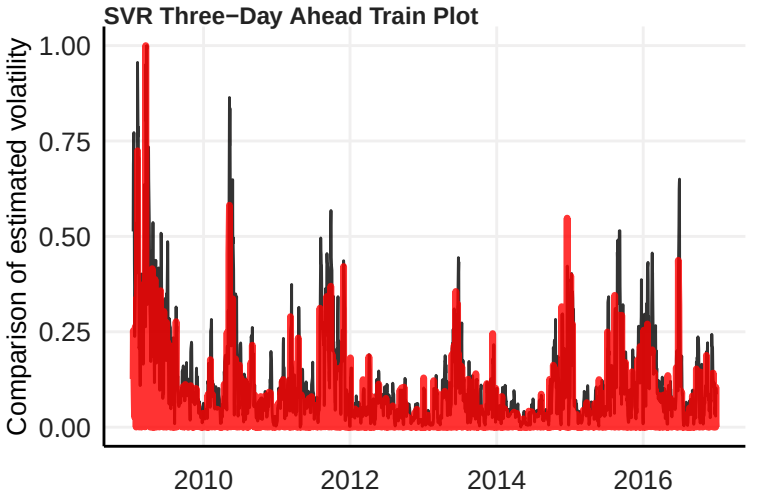

Figure 4.7: SVR Three-Day Ahead Training Forecast

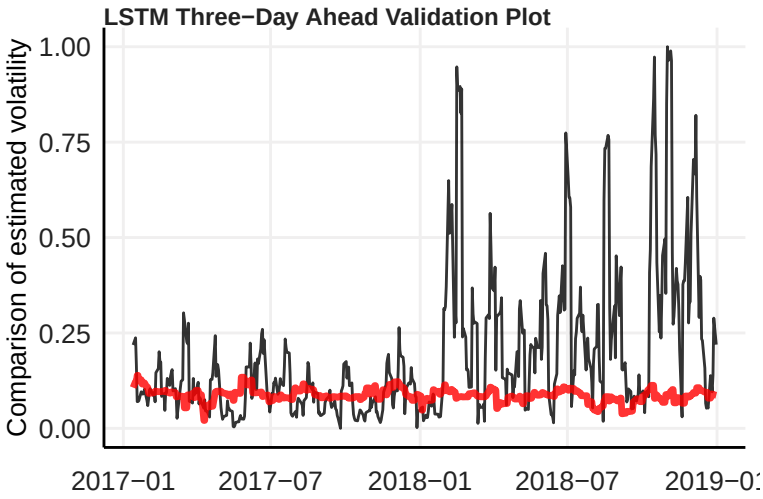

Figure 4.8: LSTM Three-Day Ahead Validation Forecast

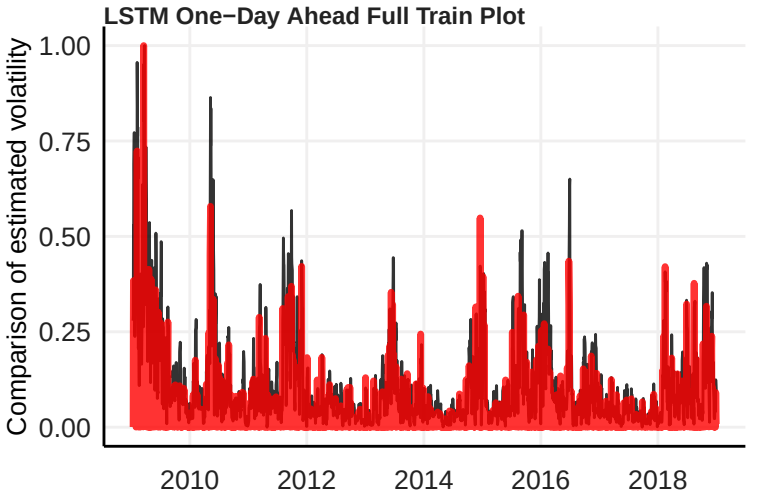

Figure 4.9: LSTM One-Day Ahead Full Training Forecast

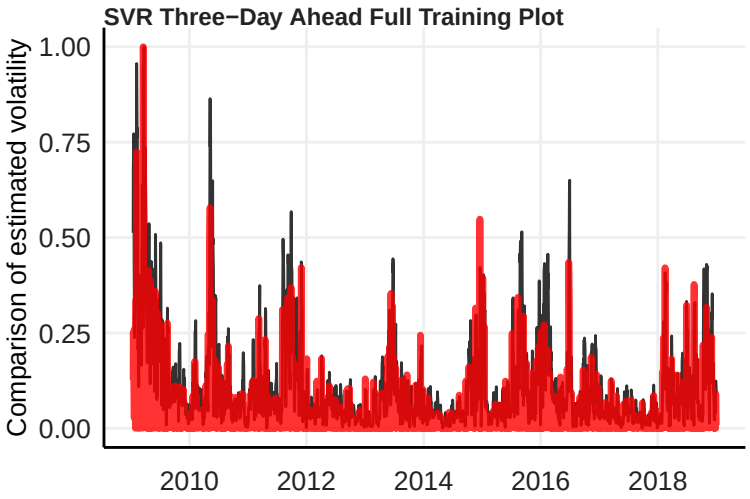

Figure 4.10: SVR Three-Day Ahead Full Training Forecast