Bitcoin Return Volatility Forecasting: A Comparative Study of GARCH Model and Machine Learning Model

Ze Shen, Qing Wan, David J. Leatham

Abstract:

One of the well-known features of bitcoin is its extreme volatility. The modeling and forecasting of bitcoin volatility is crucial for bitcoin investors' decision making analysis and risk management. All the previous studies of bitcoin volatility were founded on economic models. However, research on bitcoin volatility forecasting using machine learning algorithms is still void. In this article, both conventional economic models and machine learning model are used to forecast the volatility of bitcoin return. The objective of this study is to compare their out-of-sample performance in forecasting accuracy and risk management efficiency. The results demonstrate recurrent neural network method outperforms the economic GARCH model and simple moving average model in terms of statistical forecasting accuracy, but is less efficient in risk management in the framework of Value at Risk. This article proposes an alternative way of bitcoin volatility analysis and provides more motivation for economic researchers to apply machine learning methods to the financial and economics world. Meanwhile, it also shows that the machine learning approaches are not always more advanced than economic models, which is contrary to what is widely believed by people.

Keyword: bitcoin, GARCH, machine learning, recurrent neural network, volatility, risk management

JEL codes: G000, G170

Since Satoshi Nakamoto proposed the first cryptocurrency in 2009, the cryptocurrency market has received much attention. Bitcoin is the most successful and popular one in the market, which accounts for over fifty percent of the current whole cryptocurrency market capitalization.[1] The bitcoin enthusiasm is due to its innovational features of decentralization, anonymity and zero transaction cost. Researchers' analysis of bitcoin has recently received growing interests. David Yermack (2015) studied the features and functions of bitcoin, and concludes that bitcoin appears to be more like a speculative investment than a real currency due to its high volatility. If people look at the bitcoin price history from 2009 till now, its violent fluctuations will be discovered. As a financial asset, bitcoin is famous for its extreme volatility. The modeling and forecasting of bitcoin volatility is crucial for bitcoin investors' decision making analysis and risk management.

Earlier studies mainly explored bitcoin volatility by using GARCH family models. Bouoiyour and Selmi (2015, 2016) compared different GARCH type model on sub-period bitcoin volatility, and Paraskevi (2017) compared the GARCH family models over the whole period. Mehmet el at. (2017) found the bitcoin trading volume fails to predict bitcoin volatility by studying their causal relationship.

These early studies of bitcoin volatility were founded on economic models. However, research on bitcoin volatility forecasting using machine learning algorithms is still void. Susan Athey (2018) pointed out that the machine learning would have a dramatic impact on the field of economics in the near future. Unlike the economic models, where researcher picks a specific model based on economic principles and estimates the parameters, machine learning algorithm is usually a data

[1] From coinmarket.com, the total cryptocurrency market capitalization was around $1,800 billion and bitcoin market capitalization was over $97 billion on April 2019.

driven modeling focused on the selection process. Thus a model of machine learning algorithm is not fixed or predetermined but will be refined during a training process. Applying machine-learning methods to solve for economic issues can potentially make a difference in the economic and financial field.

In this article, both conventional economic models and a machine learning model are used to forecast the volatility of bitcoin return, and their forecasting performance are evaluated. The aim of this article is to compare their performance, and to discover if machine learning can improve economic time series forecasting. The booming development of machine learning techniques in time series forecasting encourages people to apply it in financial market. Moreover, the success of machine learning on stock market prediction leads us to believe that it may also work well for cryptocurrency price forecasting. In addition, the empirical studies show that the machine learning method is more efficient than ARIMA model in bitcoin price prediction. Sean, Jason and Simon (2018) compared the forecasting performance of recurrent neural network (RNN), long short term memory (LSTM) network and ARIMA on bitcoin price, and reported that the machine learning models outperformed ARIMA. Laura A. et at (2018) examined the forecasting performance on cryptocurrency portfolio, and reported the same conclusion that machine learning methods overwhelms the standard benchmark simple moving average. It makes sense for machine learning method to be superior to traditional economic model (such as simple moving average and ARIMA), because machine learning model is constructed in a more general scope that considers both linear and nonlinear features. It also preserves more temporal information of a time series during training.

As discussed above, the machine learning methods are more advanced than some traditional economic models in time series forecasting, both theoretically and empirically. However, caution must be considered with this assertion. First of all, the economic models involve economic intuition while machine learning mainly deals with data. In the economic world, the economic intuition is the key to economic analysis. In contrast, the machine learning captures information only from data. However, the information contained in data is limited in analyzing economic issues. Secondly, the performance of machine learning depends on the amount of data. Its performance is dramatically improved as the data amount getting larger. However, in this article, the bitcoin market history is quite short and only daily data is available. Finally, machine learning is sensitive to the fluctuations. Compared to other approaches, machine learning is more efficient in identifying time series trends and patterns. However, this leads to a problem that a shock or abnormal perturbation will be treated more seriously. But in the real world, there are many factors that affect the market reaction to the shock or abnormal perturbation, and the fluctuation sensitivity might cause an overreaction problem in the forecasting, especially for the volatility analysis.

The objective of this study is to compare the forecasting performance between traditional economic models and machine learning method in terms of both forecasting accuracy and risk management efficiency. Investors are interested in the bitcoin volatility forecasting accuracy performance because they need information on how volatile the market will be in the future. Meanwhile, because the investors in bitcoin market are facing large risk every day, they are also concerned about risk management. The contribution of this article is to investigate whether the machine learning method is more advanced in bitcoin volatility forecasting and bitcoin market risk management, and how advanced it is going to be. This article is structured as follows. First, the economic models are presented. The article starts with the naive model, simple moving average model (SMA) as a benchmark, and then move to a more complex but conventionally applied model, generalized autoregressive conditional heteroscedasticity (GARCH) model, to forecast bitcoin return volatility. Then a machine learning model based on Recurrent Neural Network (RNN) is proposed. The next step is to evaluate the out-of-sample performance of the three models. The root mean squared error (RMSE) and mean absolute error (MAE) are used to evaluate their forecasting accuracy performances and the Value at Risk (VaR) is used to compare their risk management efficiency. Since the true conditional volatility of bitcoin return is unobservable, the bitcoin daily squared return and Garman-Klass volatility (Garman and Klass, 1980) are used as proxies for the realized volatility.

Data

In this study, the bitcoin return time series is used rather than the raw bitcoin price data. The bitcoin daily return is defined as the difference of the natural logarithm of the daily bitcoin closing price. Bitcoin daily opening, high, low and closing price are used to estimate the realized bitcoin volatility. All the data are available in website: CoinMarketCap.com. The data ranges from April 30, 2013 to November 20, 2018, 2031 observations totally. Figure 1 illustrates the bitcoin daily return and bitcoin daily squared return respectively. Table 1 shows the descriptive statistics of the bitcoin daily return.

Before going further to the economic modeling, the stationary of time series is checked. The augmented Dickey-Fuller- test (ADF) and Phillips-Perron (PP) unit root test are used to check for the stationary of bitcoin daily return series, and table 2 indicates that the financial time series is stationary.

Methodology

In this section, the economic methodology is discussed first, and then the recurrent neural network model, which is a machine learning methodology will be presented. Engle in 1982 proposed the autoregressive conditional heteroscedasticity model (ARCH), which assumes that the volatility of asset returns is time varying instead of a constant. Bollerslev (1986) generalized the ARCH model and developed a more commonly used GARCH model. In this study, the GARCH model is applied as the economic method.

Economic methodology

First look at figure 1, as it shows notable fluctuations in bitcoin daily return. It is also found that large changes follow large turbulence and small changes follow calm periods. This phenomena in time series asset return is known as "volatility clustering". The plot of bitcoin daily squared return in figure 1 provides more evidence that changes tend to be clustered together.

Figure 2 shows the autocorrelation function (ACF) and partial autocorrelation function (PACF) of bitcoin daily squared return. Table 3 shows the results of Ljung-Box Q-test for the bitcoin daily squared return. Figure 2 and table 3 indicate that the bitcoin daily squared return is serially correlated, which suggests the existence of the conditional heteroscedasticity in bitcoin price volatility. Thus, the economic model needs to capture the feature of heteroscedasticity.

The basic structure of the economic model is as follows:

$r_t = \mu_t + Z_t$ (1)

$\mu_t = E(r_t | \digamma_{t-1})$ (2)

$h_t^2 = Var(r_t | \digamma_{t-1}) = E[(r_t - \mu_t)^2 | \digamma_{t-1}] = E(Z_t^2 | \digamma_{t-1})$ (3)

Where $r_t = \frac{\log P_t}{\log P_{t-1}}$, μ_t is the conditional mean and h_t^2 is the conditional variance, $\digamma_{t-1}$ denotes for the past information.

Conditional mean

ARMA(p, q) process is applied to model the conditional mean:

$\mu_t = \phi_0 + \sum_{i=1}^{p} \phi_i r_{t-i} + \sum_{j=1}^{q} \theta_j Z_{t-j}$ (4)

With the autoregressive order p and moving average order q.

After applying the ARMA(p, q) process, the estimated parameters and the residuals are obtained. As it is discussed above, the bitcoin daily return exhibits volatility clustering, which indicates the conditional heteroscedasticity volatility. The ARCH effects of the residuals are tested. If there is ARCH effect in the residuals, the conditional variance models will be specified in the next section.

Conditional variance

Given the conditional mean model and using (3), the residuals $Z_t = r_t - \mu_t$ are obtained. Then the condition variance models are able to be built. Two different models are presented in the following section. It starts with SMA model, then move the GARCH model, to forecast bitcoin return volatility.

Simple moving average (SMA)

Even though simple moving average is the simplest model for volatility forecasting, it models the time varying variance and captures the past information and historical variance. Although the simple moving average model incorporates neither conditional mean nor conditional variance in the sense of GARCH, it is presented here as a benchmark to evaluate the performance of the other models.

The simple moving average model is presented as:

$\sigma_{k+1}^2(n) = \frac{1}{n} \sum_{i=0}^{n-1} r_{k-i}^2$ (5)

Where k is the forecast origin, and r_t^2 is the bitcoin daily squared return. This article sets $n = 10$.

GARCH model

The generalized autoregressive conditional heteroscedasticity model (GARCH) was developed by Bollerslev in 1986. Both ARCH process and GARCH process model the variations of a financial assets' volatility, and the GARCH process allows the conditional variance to be an ARMA process. The GARCH (m, n) process is as follows:

$Z_t = h_t \varepsilon_t, \{\varepsilon_t\} \sim IID(0,1)$ (6)

$h_t^2 = \alpha_0 + \sum_{i=1}^{m} \alpha_i Z_{t-i}^2 + \sum_{j=1}^{n} \beta_j h_{t-1}^2$ (7)

$\{Z_t\}$ is the residual series of the best fitting ARMA(p, q) model, thus the conditional variance of the residual series essentially acts like an ARMA process. It is expected that the standardized squared residuals obtained from best fitted ARMA-GARCH model should not be autocorrelated and there should not remain any ARCH effects. The ARCH LM test is used to check whether this is true or not in this study.

Recurrent neural network (RNN)

The sequencing model for predicting bitcoin return volatility is built on the concept of Recurrent Neural Network (RNN). RNN deals well with sequence problems, which thanks to its special architecture that takes the order of data into consideration. Each RNN has a type of memory unit concatenated into multi stages and each of which will turn previous states and current input to activations and pass necessary information forward to the next stage. In this study, a GRU (Gated Recurrent Units) cell is employed for serving as the memory unit. The cost function is redesigned based on a tangent function. This model doesn't build any embedding or probability layer inside that are usual configurations that exist in some engineering task. In addition, by considering some uncertainty of the volatility, the range is equally cut into 250 intervals to convert a real volatility value to a vector with a dimension of 250. This conversion serves as an encoder for a RNN cell's input. A whole architecture of RNN model is listed as figure 3. In general, the encoding process will turn a fixed length of sequential data into the same length of vectors for RNN, which is fed into multiple layers of perceptron (MLP). The MLP will decode states from RNN into sequential vectors and transfer them to a predictor for output.

Empirical results

In this section, the forecasting results of the SMA model, GARCH model and RNN model will be presented. Then, their out-of-sample forecasting accuracy performance will be evaluated and compared. Before the evaluation, appropriate proxies for the realized volatility have to be found.

Forecasting

The sample data is divided into two parts, in sample period from April 30, 2010 to April 30, 2018 (1827 observations) and out of sample period from May 1st, 2018 to November 20, 2018 (204 observations).

In economic GARCH model, the ARMA (p, q) order are selected by AIC and BIC, and the best fitted conditional mean model was found to be ARMA$(2,2)$. Then, the ARCH effects of the residuals are tested, and the result indicates there remains ARCH effect in the residual series. Finally the best fitted ARMA $(2,2)$-GARCH$(1,2)$ model is obtained. Table 4 presents the estimated parameters of ARMA $(2,2)$-GARCH$(1,2)$ model.

The autocorrelation in the standardized residuals of the fitted ARMA-GARCH model is checked, and the result indicates that there is no remaining ARCH effect in the residuals.

For the RNN model, 30 days samples of the volatility are used to predict the next 1 day, 5 days and 10 days with an out of sample method. For example, the first 30 days of volatility values was used to predict the 31^{st}. The sequential data generated by this process is called as tuple 1; then the 2^{nd} to 31^{st} volatility values are used to predict the 32^{th}, and it is called tuple 2. The total data length was 2031. By rolling this process, 1994 tuples were generated. In the out of sample method, the first 1794 (90%) observations were appointed to training and the remaining 200 observations were used as a test volume. Also, due to value of volatility being very small, each value was scaled by 10^4.

A more detailed implementation is illustrated in figure 4. Two layers of RNN with GRU cell are built as core. The first layer has 512 units while the second shrinks to 256 units. Sequential data were fed in cells on the bottom from left to right. The predicated data were collected on the top from left to right.

Both training and testing were taken on GTX 1070 GPU. A SGD (Stochastic Gradient Descend) algorithm that shuffles the whole dataset is used in each iteration; RMSProp gradient update algorithm was chosen as an optimizer; learning rate and batch size was set to 0.0001 and 20, respectively. As it stated before, the model 1000 epochs is trained on the 1794 tuples and the 200 tuples are tested every five epochs.

Volatility proxies

One difficulty of evaluating the forecasting performance is that the true conditional volatility of bitcoin return is unobservable. Thus, a proxy for the realized bitcoin return volatility has to be estimated. The most common used proxy for the volatility is the bitcoin daily squared return. Thus, the first volatility proxy in this study is the daily squared return. However, it may lead to a poor out-of-sample performance (Anderson and Bollerslev, 1998). To get a more robust forecasting performance comparison result, a second volatility proxy is necessary. The cumulative squared intra-day returns is a more efficient proxy for volatility (Chou et al., 2010), but it requires high frequency bitcoin prices in one day, which is not available in this case. Then, the Garman-Klass volatility (Garman and Klass, 1980) is used as the second proxy for bitcoin return volatility. This proxy includes the information of daily high, low, opening and closing prices. Garman and Klass (1980)'s estimator in practical is presented as:

$$\hat{\sigma}_{GK}^2 = 0.5[\ln(BTC_{Ht}/BTC_{Lt})]^2 - [2\ln 2 - 1][\ln(BTC_{Ct}/BTC_{Ot})]^2 \quad (8)$$

Where $BTC_{Ht}$and BTC_{Lt} is the highest bitcoin price and lowest bitcoin price at the trading day; BTC_{Ct} and BTC_{Ot} is the closing price and opening price respectively.

Out-of-sample performance

To compare the out-of-sample performance of the three models, the root mean squared error (RMSE) and mean absolute error (MAE) are used to evaluate and rank them. RMSE and MAE are the most commonly used metric for model evaluation. MAE is a good indicator of average model performance (Willmott & Matsuura, 2005), while RMSE deals well with outliers by penalizing large errors more (Chai & Draxler, 2014). Table 5 exhibits the out-of-sample performance of the SMA (benchmark) model, GARCH model and RNN model. Different horizon forecasting performance, 1 day ahead, 5 days ahead and 10 days ahead are presented in table 5. The RNN model performs best with the lowest RMSE and MAE, and the benchmark SMA performs worst. When using the first proxy, the 1 (5, 10) day ahead RMSE and MAE of GARCH model are 20% (13.5%, 15.8%) and 37.5% (35.3%, 38.9%) larger than those of RNN model; when calculated with the second proxy, the 1 (5, 10) day ahead RMSE and MAE of the GARCH model are 25.5%

(26.8%, 28.4) and 8% (6.6%, 8.7) larger than those of RNN model, respectively. The 1 (5, 10) day ahead RMSE and MAE of the benchmark SMA are 22.9% (16.2%, 18.4%) and 25% (17.6%, 16.7%) higher than those of RNN model by using the first proxy; the 1 (5, 10) day ahead RMSE and MAE of SMA are larger than those of RNN by 26.4% (27.8%, 29.3%) and 10% (9.9%, 12.8%) by using the second proxy.

Figure 5 presents the standard deviation of the first proxy (daily squared return) and the standard deviation of 1 day ahead volatility forecasting of each model. Figure 6 shows the standard deviation of the second proxy (Garman-Klass volatility) and the standard deviation of 1 day ahead volatility forecasting of each model.[2] SMA1/5/10, GARCH1/5/10 and RNN1/5/10 denote for the 1/5/10 day(s) ahead volatility forecasted by each model.

It is can be seen from figure 5 and 6 that the RNN model does better in capturing the volatility trends and clustering than the economic models. The second proxy are less volatile than the first one, thus the Garman-Klass volatility proxy performs not as good as bitcoin daily squared return proxy. As expected, the RNN model is more efficient in bitcoin return volatility forecasting. The GARCH model improves the forecasting accuracy over the SMA, but the machine-learning model does much better.

Value at Risk (VaR)

The machine learning model performs better than the economic models in volatility forecasting in terms of statistics. However, investors are also concerned about which method performs better in

[2] The standard deviation of the first and second proxy and the standard deviation of the 5 days ahead and 10 days ahead volatility forecasting of each model are presented in appendix.

risk management. To provide more useful information about the bitcoin market in terms of risk management, the Value at Risk (VaR) is a good measurement. The bitcoin Value at Risk is defined as the maximum amount of money that the bitcoin investors could potentially lose at a given confidence level over a defined period of time. The Value at risk concept was proposed by J.P. Morgan in 1994. There are two categories of methodology for VaR calculation, parametric models and nonparametric models. The most used nonparametric VaR model is Monto Carlo method, which is complicated while the parametric VaR are simpler for investors to apply. The parametric models are mainly considered and discussed in this article. Mathematically, the forecasted VaR of bitcoin is defined by:

$$Pr(r_t \leq VaR_t(\alpha)|F_{t-1}) = \alpha \quad (9)$$

Where r_t is the bitcoin daily return, and F_{t-1} is the past information. The α is the given confidence level, 1%, 2.5% and 5%. Thus the VaR estimation involves the assumption of bitcoin daily return distribution. The traditional VaR calculation assumes the portfolio return is normally distributed. However, it is empirically documented that financial asset return distributions always exhibit heavy tails. Table 1 lists the summary statistics for bitcoin daily returns, which shows there is excess kurtosis in the sample data. Therefore, the Student's t-distribution assumption is applied in this article instead of the normal one. The forecasting of daily bitcoin Value at Risk under Student's t-distribution is estimated as:

$$VaR_t(\alpha) = E(r_t|F_{t-1}) + t_\alpha^v \sigma_t \sqrt{\frac{v}{v-2}} \quad (10)$$

Where t_α^v is the Student's t-distribution critical value at confident level α (1%, 2.5% and 5%); $E(r_t|F_{t-1})$ is the conditional mean generated by ARMA (2,2) and forecasted by RNN; v is the estimated degree of freedom, and σ_t is the forecasted standard deviation at time t. Following

Heikkinen and Kanto (2002) and Andreev and Kanto (2004), the degree of freedom is allowed to be non-integer. Applying method of moments, the consistent estimator of the degree of freedom is estimated by:

$$\hat{v} = 4 + \frac{6}{\hat{k}},\ \forall v > 4 \quad (11)$$

Where $\hat{k}$ is the sample excess kurtosis.

Then the 1 day ahead, 5 days ahead and 10 days ahead VaR at 1%, 2.5% and 5% confidence level are calculated by the conditional expected return and forecasted volatility, which are generated from SMA model, ARMA (2,2)-GARCH (1,2) model and RNN model, respectively. Figure 7 presents the realized returns and 1 day ahead VaR (1%)[3] forecasts for each of the three models.[4]

In order to compare the Value at Risk forecasting performance, Table 6 gives the out of sample coverage $\hat{\alpha}$ for each model. The $\hat{\alpha}$ is calculated by the number of realized losses that exceeds the forecasted Value at risk on that day divided by the number of total out of sample observations:

$$\hat{\alpha} = \frac{No.(loss > VaR_t(\alpha))}{No.(out\ of\ sample\ obs.)} \quad (12)$$

It is obvious that the closer $\hat{\alpha}$ to α, the more accurate VaR would be, which makes it easier for investors to manage the risk in bitcoin market. Thus the model with the smallest $|\alpha - \hat{\alpha}|$ provides the best risk coverage.

[3] The figures of realized return and 5/10 days ahead VaR (1%) of each model are presented in appendix.

[4] The figures of realized return and 1/5/10 day(s) ahead VaR (2.5% and 5%) of each model are presented in appendix.

Table 6 indicates that the sample coverage $\hat{\alpha}$ of GARCH model are closest to the given confidence level α, while the sample coverage of RNN model appears to be the most volatile with largest distance between $\hat{\alpha}$ and α. Thus the RNN model performs poorly in Value at Risk forecasting, even worse than the benchmark SMA model.

Conclusion

Bitcoin is the most successful and popular cryptocurrency in the market, with around 130 billion daily trading volume as of April 2019. Bitcoin has historically had a larger fluctuations in price than most other financial assets. Therefore the analysis of bitcoin return volatility is crucial for investors' decision making and risk management. Both economic models and machine learning method are used to forecast the bitcoin return volatility.

The machine learning method in time series forecasting are expected to be superior to the traditional economic models. The earlier empirical studies in stock price forecasting and cryptocurrencies prices forecasting provided evidence of this statement. Comparing the out-of-sample performance of each volatility forecasting model, the result indicates that compared to the traditional economic model, the machine learning method is more accurate in bitcoin return volatility forecasting, which is consistent with the results of financial price forecasting studies.

One advantage of recurrent neural network is that it is a nonlinear model, and the model is learned through past experience. If there is enough data for RNN to learn, it should outperform the linear models, such as SMA and GARCH model. In this study, 1826 in sample observations are sufficient for the model to learn from experience and use them to predict the future.

However, one concern is that the economic models consider economic backgrounds while machine learning models only deal with data. Thus, machine learning models may be less efficient in risk management. Although it performs better in statistical forecasting accuracy, it is beaten by the conventional economic models in the framework of Value at Risk. One reason is that in risk management, the Value at Risk takes additional market information – the asset return distribution- into consideration. In this article, the machine learning approach outperforms conventional economic models, SMA and GARCH, in volatility forecasting in terms of forecasting accuracy, but is overwhelmed in terms of bitcoin market risk management efficiency.

This study proposed an alternative way of volatility analysis. It illuminates the feasibility and potential to apply machine learning methods to economic time series forecasting. It is widely believed and empirically proved by earlier studies in financial market that machine learning approach is more advanced in time series forecasting. However, this article shows something different. None of the methods studied in this article outperformed others in all aspects. If the bitcoin market investors need information on how volatile the market would be in the future and the data amount is large enough, RNN is a good method as it captures the trend and volatility clustering better than other models; however, if additional market information is available and investors are interested in bitcoin market risk management, the economic models can be more efficient.

Tables and figures

Table 1. Summary Statistics of Bitcoin Daily Returns in the Sample Period

	BTC
Sample size	2031
Mean	0.000733
Variance	0.000361
Std. Dev.	0.019004
Skewness	-0.195558
Kurtosis	8.017161

Note: This table displays the summary statistics of bitcoin daily return from April 30, 2013 to November 20,2018; BTC denotes for bitcoin daily return

Table 2. Unit Root Tests

	Without Trend		With Trend	
	ADF	PP	ADF	PP
BTC daily return	-44.52	-44.74	-44.51	-44.73
Critical values (1%)	-3.43	-3.43	-3.96	-3.96

Note: ADF and PP test statistics are much smaller than the 1% critical value, indicating the bitcoin daily return from April 30, 2013 to November 20, 2018 is stationary at 1% significance level.

Table 3. Ljung-Box Q-Test for Bitcoin Daily Return

No. of lags	Lag 10	Lag 15	Lag 20
P-value	0.006623***	0.005274***	0.00005***

Note: Triple asterisk (***) denotes variable significant at 1% level.

Table 4. ARMA (2,2)-GARCH(1,2)

Parameters	Estimated value	t-value	p-value
ϕ_0	0.001	3.734	0.000***
ϕ_1	1.495	133.042	0.000***
ϕ_2	-0.948	-102.038	0.000***
θ_1	-1.513	-121.275	0.000***
θ_2	0.957	222.547	0.000***
α_0	0.000	1.842	0.066*
α_1	0.224	7.720	0.000***
β_1	0.367	2.441	0.015**
β_2	0.408	3.014	0.003***

Note: Asterisk (*), double asterisk (**) and triple asterisk (***) denote variables significant at 10%, 5% and 1% levels respectively.

Table 5. Out-of-sample Performance

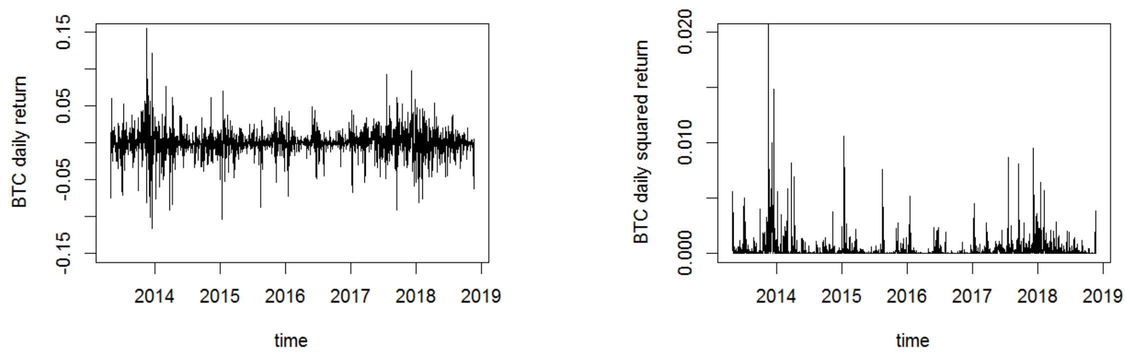
	RMSE		MAE	
	1st Proxy	2nd Proxy	1st Proxy	2nd Proxy
1 day ahead				
SMA	0.00043	0.00402	0.00020	0.00165
GARCH	0.00042	0.00399	0.00022	0.00162
RNN	**0.00035**	**0.00318**	**0.00016**	**0.00150**
5 days ahead				
SMA	0.00043	0.00405	0.00020	0.00167
GARCH	0.00042	0.00402	0.00023	0.00162
RNN	**0.00037**	**0.00317**	**0.00017**	**0.00152**
10 days ahead				
SMA	0.00045	0.00410	0.00021	0.00168
GARCH	0.00044	0.00407	0.00025	0.00162
RNN	**0.00038**	**0.00317**	**0.00018**	**0.00149**

Note: The lowest RMSE and MAE of each model is in bold.

Table 6. Out of Sample Coverage of Each Model

	SMA	GARCH	RNN
1 day ahead			
α =1%	5.88%	**3.43%**	41.0%
α =2.5%	7.35%	**4.41%**	42.0%
α =5%	10.78%	**7.84%**	44.0%
5 days ahead			
α =1%	7.50%	**3.00%**	42.5%
α =2.5%	11.00%	**5.00%**	43.0%
α =5%	13.00%	**8.50%**	43.5%
10 days ahead			
α =1%	3.59%	**2.56%**	43.0%
α =2.5%	7.69%	**4.10%**	43.5%
α =5%	12.31%	**6.67%**	43.5%

Note: The smallest $|\alpha - \hat{\alpha}|$ in each model is in bold.



BTC daily return

BTC daily squared return

Figure 1. Bitcoin daily return and bitcoin daily squared return

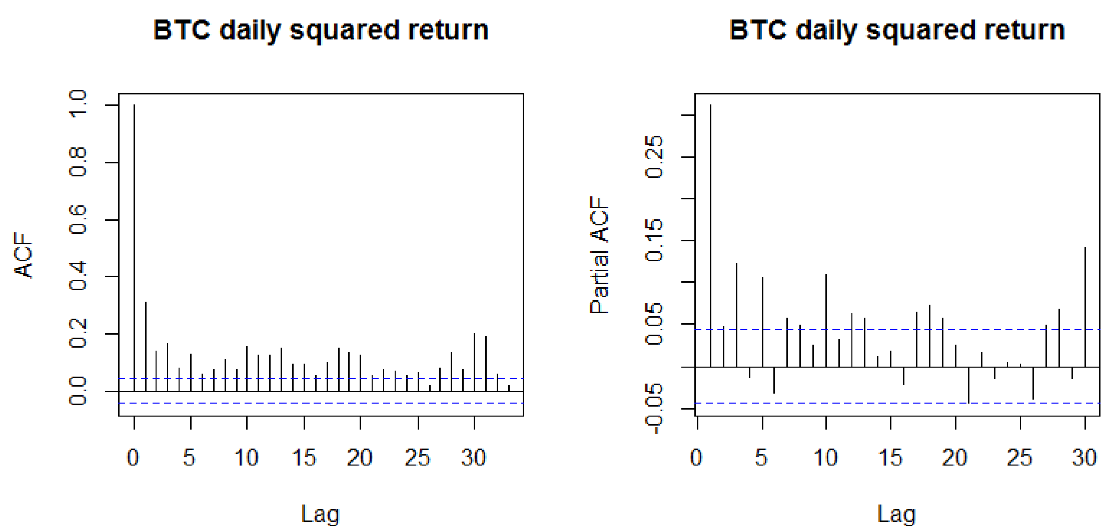


Figure 2. ACF and PACF of bitcoin daily squared return

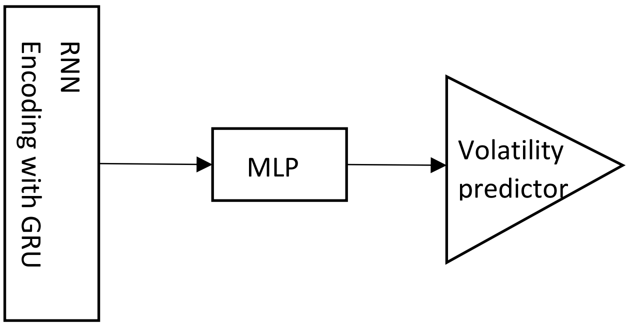


Figure 3. Architecture of recurrent neural network (RNN) model

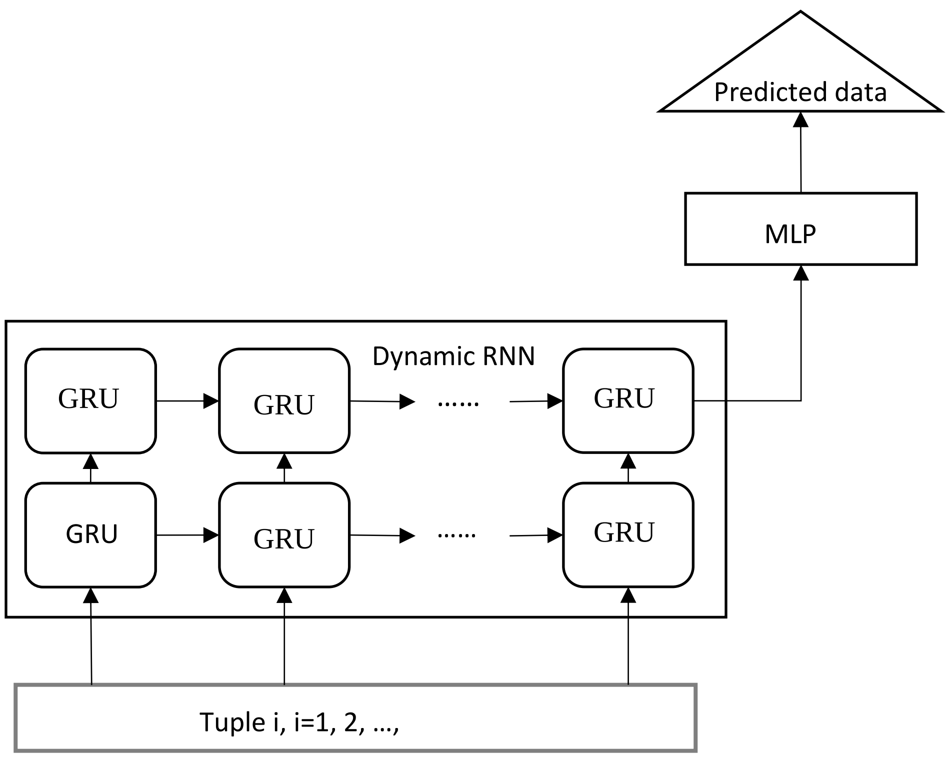


Figure 4. Detailed implementation of recurrent neural network (RNN) model

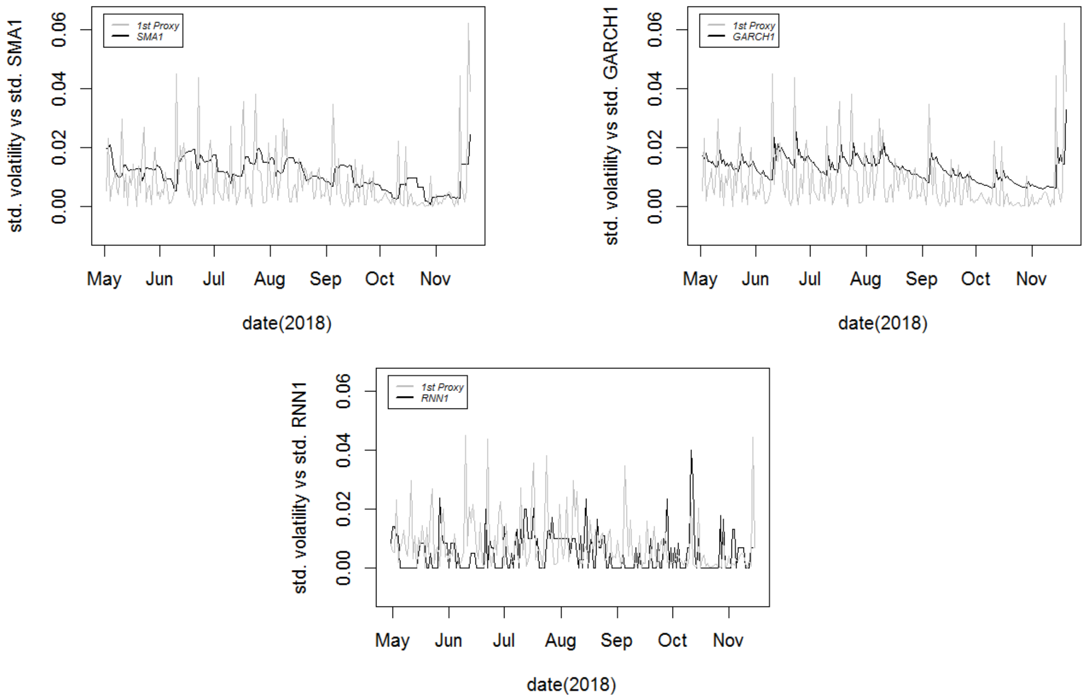


Figure 5. Out of sample standard deviation of realized volatility (1st proxy) vs standard deviation of 1 day ahead volatility forecasting of SMA, GARCH and RNN model

Note: SMA1, GARCH1 and RNN1 denote for the 1 day ahead forecasting of each model.

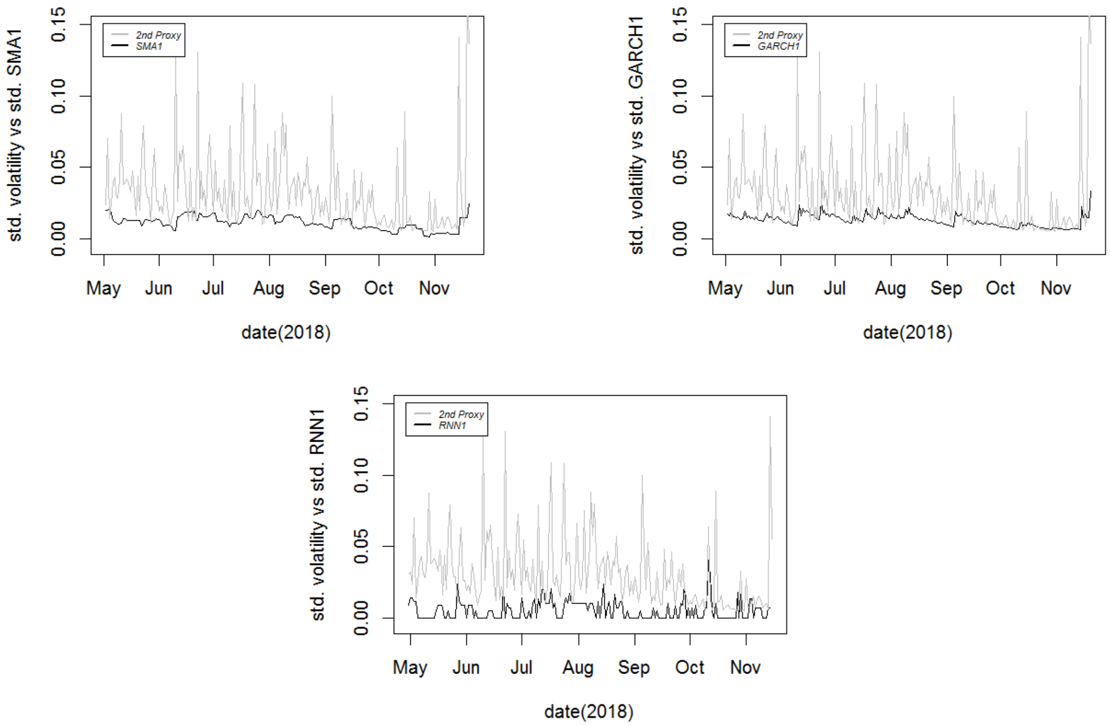


Figure 6. Out of sample standard deviation of realized volatility (2nd proxy) vs standard deviation of 1 day ahead volatility forecasting of SMA, GARCH and RNN model

Note: SMA1, GARCH1 and RNN1 denote for the 1 day ahead forecasting of each model.

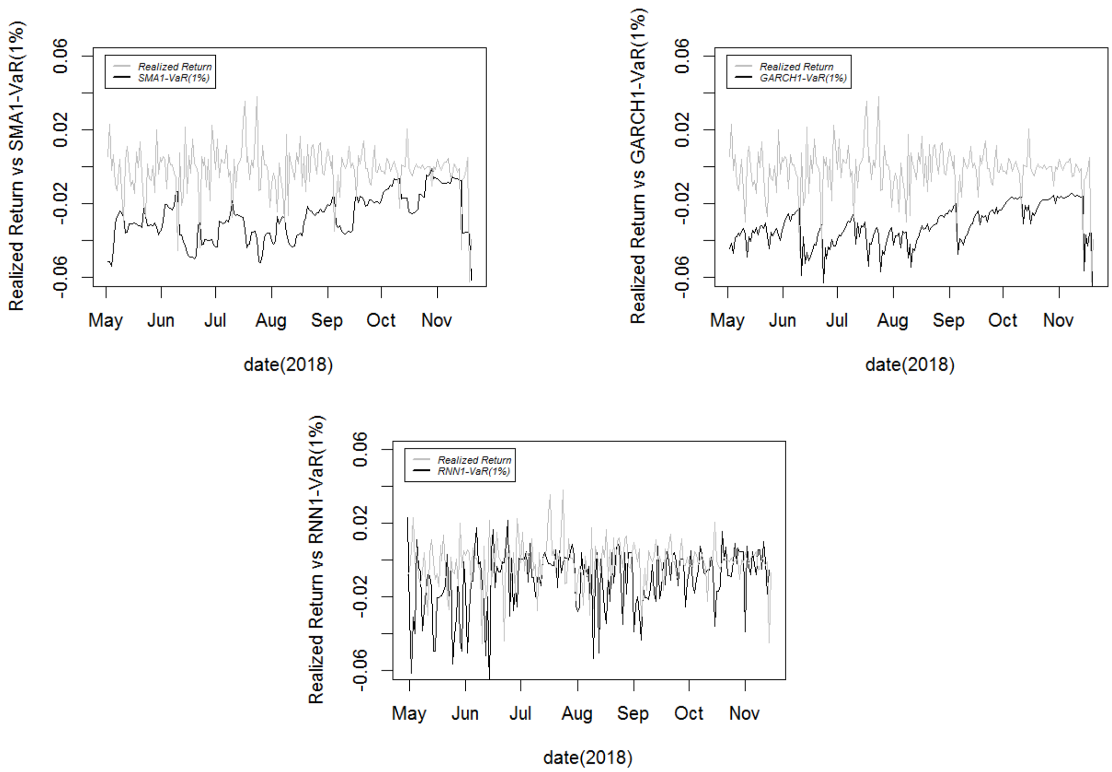


Figure 7. Out of sample realized return vs 1 day ahead VaR(1%) of SMA, GARCH and RNN model

Note: SMA1, GARCH1 and RNN1 denote for the 1 day ahead forecasting of each model.

Reference

Andersen, T.G., and T. Bollerslev "Answering the skeptics: Yes, standard volatility models do provide accurate forecasts." *International economic review* (1998):885-905.

Athey, S. "The impact of machine learning on economics." *The Economics of Artificial Intelligence: An Agenda.* Anonymous University of Chicago Press, 2018.

Balcilar, M., E. Bouri, R. Gupta, and D. Roubaud "Can volume predict Bitcoin returns and volatility? A quantiles-based approach." *Economic Modelling* 64 (2017):74-81.

Bertolami, R., H. Bunke, S. Fernandez, A. Graves, M. Liwicki, and J. Schmidhuber "A Novel Connectionist System for Improved Unconstrained Handwriting Recognition." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31 (2009):

Bollerslev, T. "Generalized autoregressive conditional heteroskedasticity." *Journal of Econometrics* 31 (1986):307-27.

Bouoiyour, J., and R. Selmi "Bitcoin: A beginning of a new phase." *Economics Bulletin* 36 (2016):1430-40.

Bouoiyour, J., R. Selmi, A.K. Tiwari, and O.R. Olayeni "What drives Bitcoin price." *Economics Bulletin* 36 (2016):843-50.

Chou, R.Y., H. Chou, and N. Liu "Range volatility models and their applications in finance." *Handbook of quantitative finance and risk management.* Anonymous pp. 1273-1281. Springer, 2010.

Einarsen, R.H. *A comparative study of volatility forecasting models* (2014):

Garman, M.B., and M.J. Klass "On the estimation of security price volatilities from historical data." *Journal of business* (1980):67-78.

Katsiampa, P. "Volatility estimation for Bitcoin: A comparison of GARCH models." *Economics Letters* 158 (2017):3-6.

Makridakis, S., E. Spiliotis, and V. Assimakopoulos "Statistical and Machine Learning forecasting methods: Concerns and ways forward." *PloS one* 13 (2018):e0194889.

McClelland, J.L., D.E. Rumelhart, and PDP Research Group "Parallel distributed processing." *Explorations in the Microstructure of Cognition* 2 (1986):216-71.

McNally, S., J. Roche, and S. Caton *Predicting the price of Bitcoin using Machine Learning*IEEE, 2018.

Robbins, H., and S. Monro "A stochastic approximation method." *The annals of mathematical statistics* (1951):400-7.

Schmidhuber, J. "Habilitation thesis: System modeling and optimization." *Page 150 ff demonstrates credit assignment across the equivalent of 1,200 layers in an unfolded RNN* (1993):

Wennström, A. "Volatility Forecasting Performance: Evaluation of GARCH type volatility models on Nordic equity indices." (2014):

Appendix

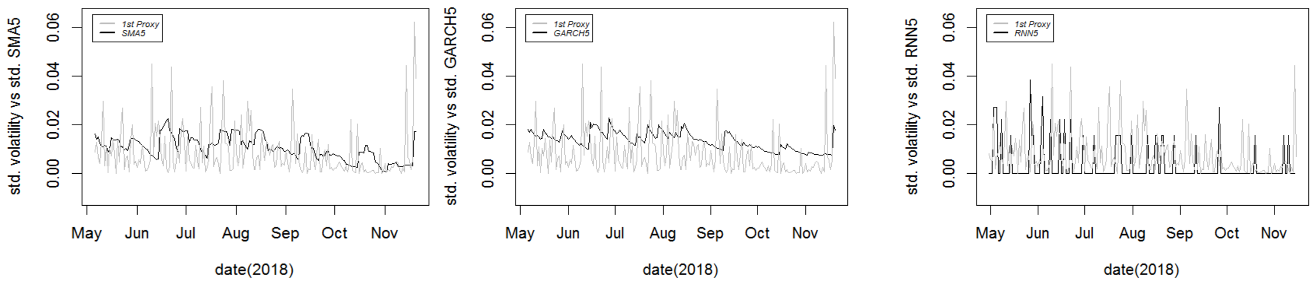


Figure 1. Out of sample standard deviation of realized volatility (1st proxy) vs standard deviation of 5 days ahead volatility forecasting of SMA, GARCH and RNN model

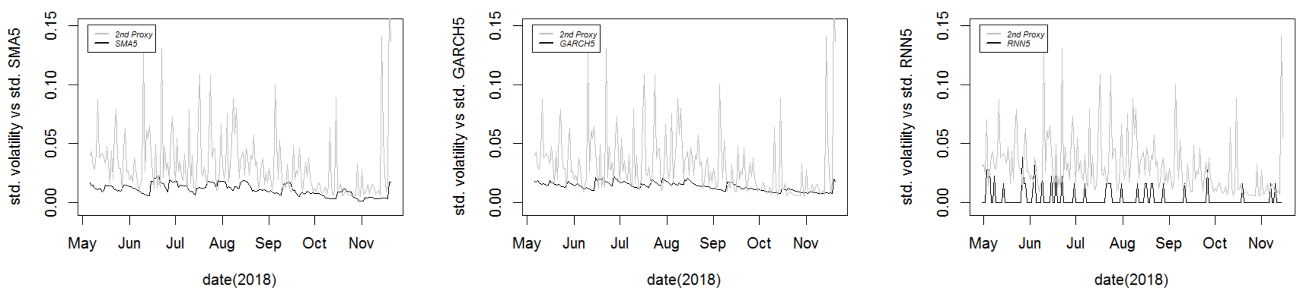


Figure 2. Out of sample standard deviation of realized volatility (2nd proxy) vs standard deviation of 5 days ahead volatility forecasting of SMA, GARCH and RNN model

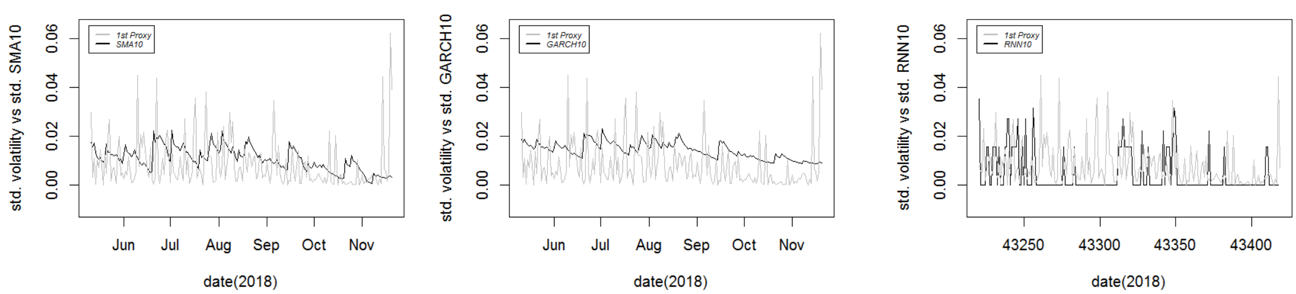


Figure 3. Out of sample standard deviation of realized volatility (1st proxy) vs standard deviation of 10 days ahead volatility forecasting of SMA, GARCH and RNN model

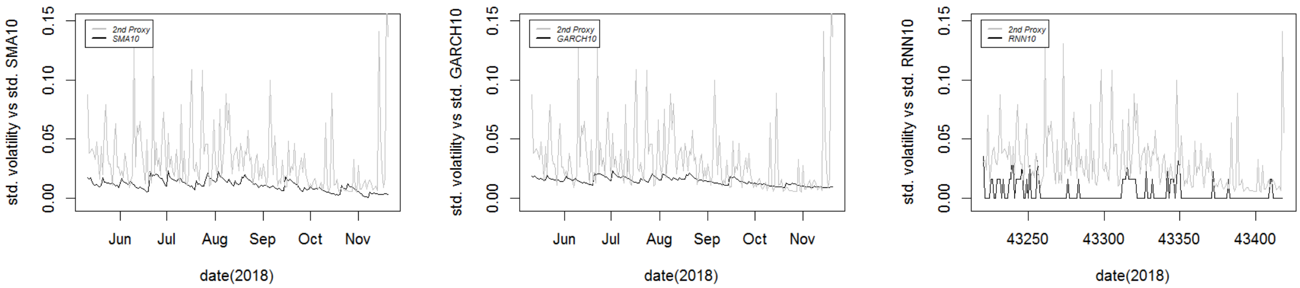


Figure 4. Out of sample standard deviation of realized volatility (2nd proxy) vs standard deviation of 10 days ahead volatility forecasting of SMA, GARCH and RNN model

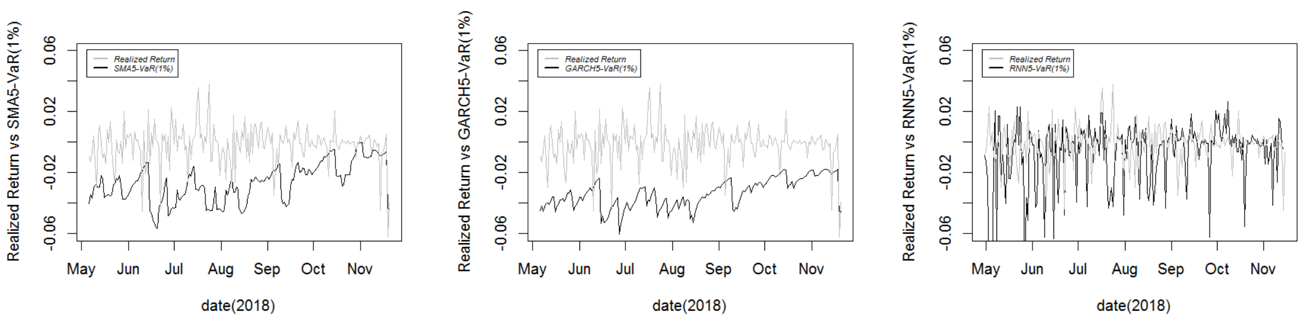


Figure 5. Out of sample realized return vs 5 days ahead VaR(1%) of SMA, GARCH and RNN model

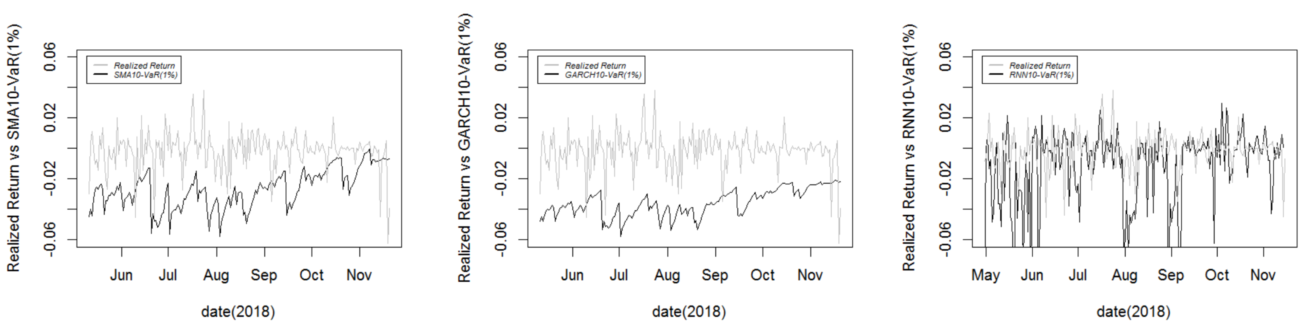


Figure 6. Out of sample realized return vs 10 days ahead VaR(1%) of SMA, GARCH and RNN model

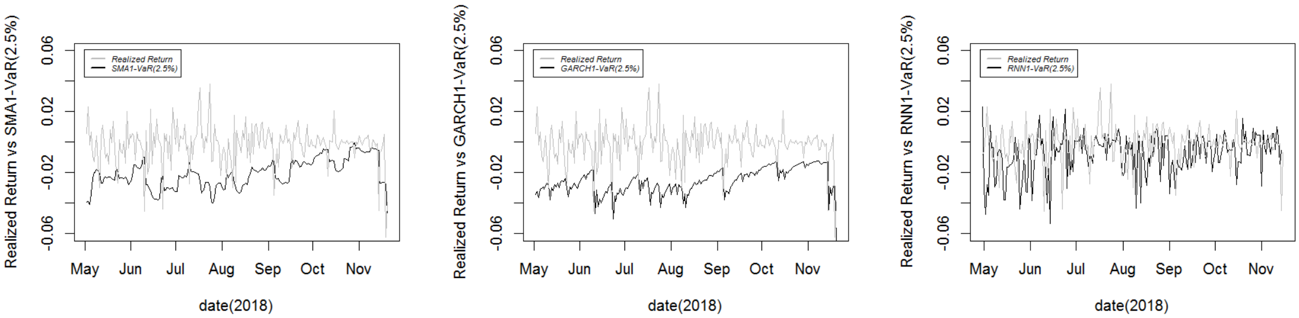


Figure 7. Out of sample realized return vs 1 day ahead VaR(2.5%) of SMA, GARCH and RNN model

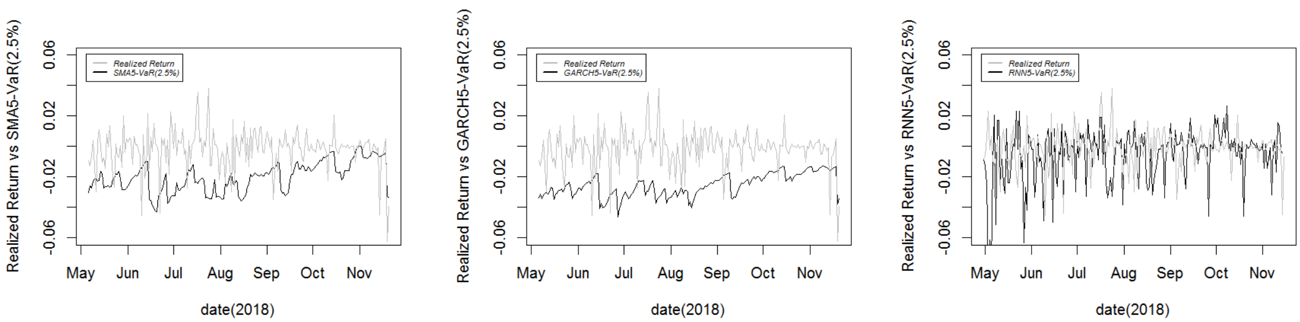


Figure 8. Out of sample realized return vs 5 days ahead VaR(2.5%) of SMA, GARCH and RNN model

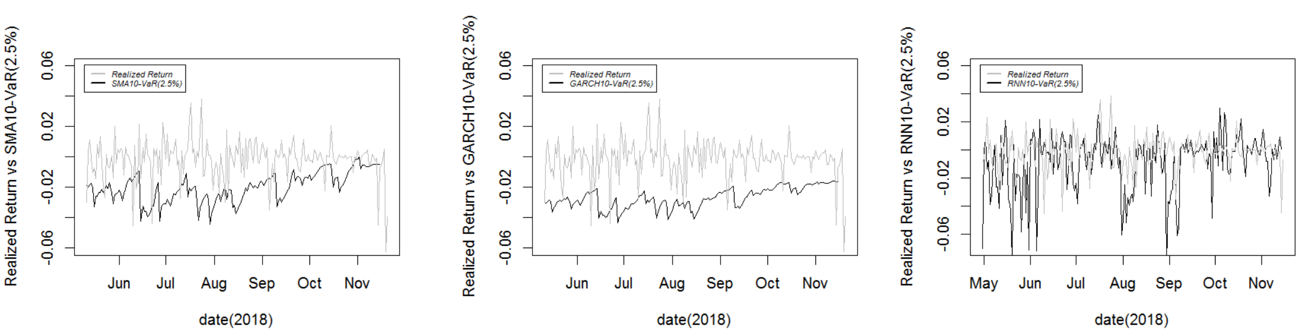


Figure 9. Out of sample realized return vs 10 days ahead VaR(2.5%) of SMA, GARCH and RNN model

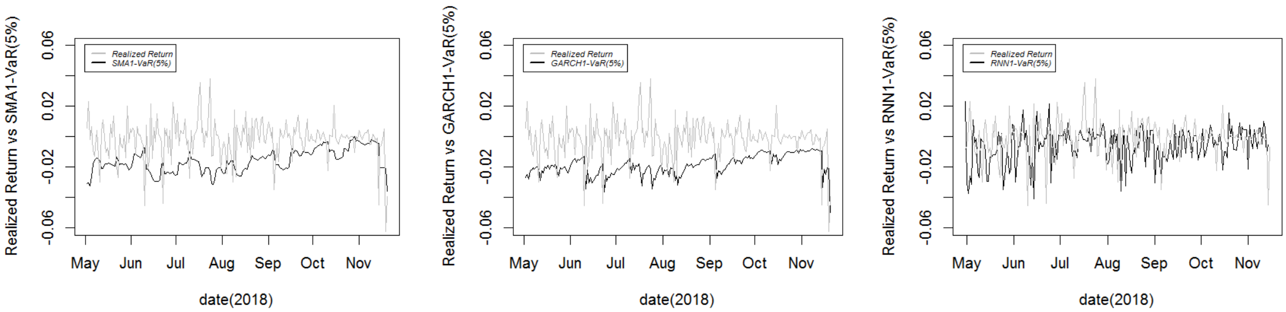


Figure 10. Out of sample realized return vs 1 day ahead VaR(5%) of SMA, GARCH and RNN model

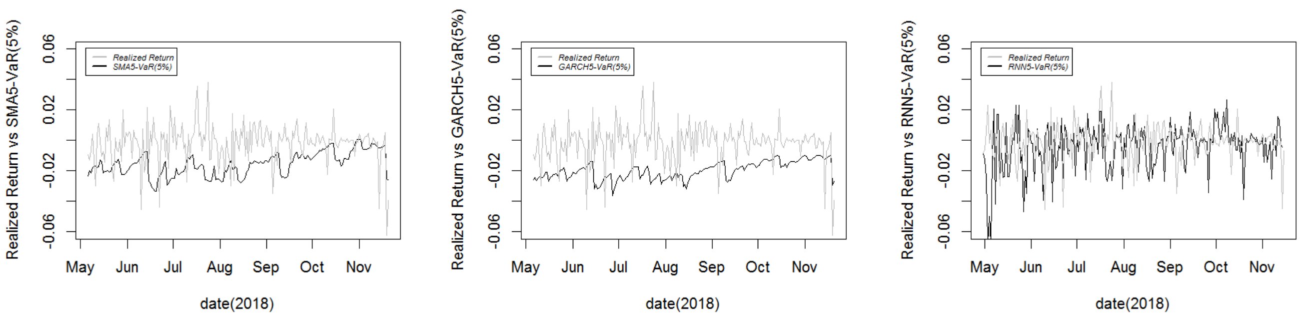


Figure 11. Out of sample realized return vs 5 days ahead VaR(5%) of SMA, GARCH and RNN model

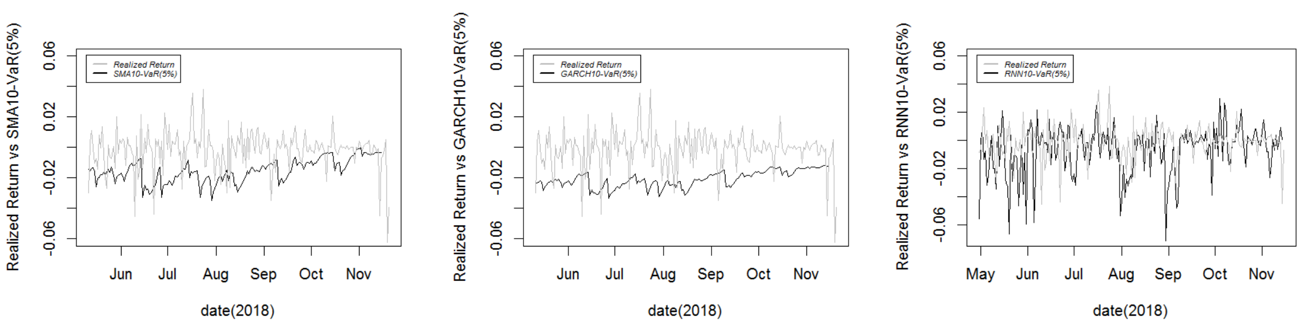


Figure 12. Out of sample realized return vs 10 days ahead VaR(5%) of SMA, GARCH and RNN model