Machine learning for realised volatility forecasting

Eghbal Rahimikia and Ser-Huang Poon*

October 8, 2020

Abstract

This paper examines, for the first time, the performance of machine learning models in realised volatility forecasting using big data sets such as *LOBSTER* limit order books and news stories from 'Dow Jones News Wires' for 28 NASDAQ stocks over a sample period of June 28, 2007, to November 17, 2016. We find strong evidence to support ML forecasting power dominating an extended CHAR and all other HAR-family of models using evaluation measures such as MSE, QLIKE, MDA and RC values. The LOB-ML has very strong forecasting power and adding News sentiment variables to the data set only improves the forecasting power marginally. However, the good forecasting performance of ML models is relevant only for normal volatility days (i.e. 90% of the out-of-sample period). Throughout the study, we find a persistent trade-off between normal vs jump day forecasting; one model serves well for normal days performs poorly for jump days, and *vice versa*.

Keywords: Realised Volatility Forecasting, Machine Learning, Long Short-Term Memory, Heterogeneous AutoRegressive (HAR) models, Limit Order Book (LOB) Data, Dow Jones Corporate News, Big Data.

JEL: C22, C45, C51, C53, C55, C58

*Eghbal Rahimikia (corresponding author) (eghbal.rahimikia@manchester.ac.uk) and Ser-Huang Poon (ser-huang.poon@manchester.ac.uk) are at the University of Manchester, Alliance Manchester Business School, UK. Our special thanks go to Yoichi Otsubo at Alliance Manchester Business School who provided dataset for this study. Special thanks also to Steve Roberts, Stefan Zohren, Jan-Peter Calliess, and Matthias Qian at the Oxford-Man Institute of Quantitative Finance, the University of Oxford, for their invaluable comments and hints. All models are run on the computational shared facility of the University of Manchester. We must express our sincere appreciation to the IT services of the University of Manchester for their constant and continued support and providing the computational infrastructures for this study.

1 Introduction

Volatility forecasting plays a critical role in financial modelling and financial decision making. This paper studies the effectiveness of machine learning (ML) method in volatility forecasting by extracting information from Big Data sets such as high-frequency limit order book (LOB) and news coverage for 23 NASDAQ stocks over a sample period of June 28, 2007, to November 17, 2016. Using the same data set, Rahimikia and Poon (2020) demonstrated the forecasting power of LOB and news sentiment data when augmented to CHAR, the best performing HAR-family of models. No study to date has tested the ML models volatility forecasting performance in this context. It is important to consider machine learning method for several reasons. First, the HAR-family of models, and more broadly all conventional econometric models, are not capable of handling a large number of variables common in Big Data sets such as the limit order book and news stories. Second, there is no satisfactory econometric theory that can be used to assess model validity in the Big Data environment. Third, classical econometric models cannot fully capture the nonlinear and highly complex relationships among the variables in this high dimensional context. Here, we test if the ML models, without these three weaknesses, could be a better choice for RV forecasting compared to the HAR-family of models.

ML methods have been used in many other fields for many years. More recently, Chen et al. (2019) exploit ML method in the estimation of stochastic discount factor and Gu et al. (2020) showed superior performance of ML models for empirical asset pricing. A subgroup of ML models has been developed for sequential data such as video, music, text, and, in our case, time series. Recurrent neural network (RNN) and its extension, long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) are among the most widely used ML models in both academia and industry for this type of data. By employing a chain structure, these models can perform sequential processing and, in the case of LSTM, learning long-term dependencies of time series. In general, ML models have emerged as a powerful tool in a Big Data environment.

The availability of high-frequency financial data makes realised volatility (RV) a popular model-free and low measurement error proxy for actual volatility (Andersen and Bollerslev, 1998; Barndorff-Nielsen and Shephard, 2001). With the popularity of RV, the heterogeneous autoregressive (HAR) model (Corsi, 2009) and its variations are commonly used in forecasting RV. The well-known variants include Corsi and Reno (2009) HAR-J (HAR with jumps) and CHAR (continuous HAR), Patton and Sheppard (2015) SHAR (semivariance-HAR) that separates the impact of negative and positive returns on the subsequent RV, and Bollerslev et al. (2016) HARQ model adjusting the forecasts for measurement error based on realised quarticity (RQ). In a comprehensive study of the HAR-family of models based on 23 NASDAQ stocks, Rahimikia and Poon (2020) find CHAR model to be the best performing model among

the HAR-family of models. Hence, in this study, we will use CHAR as the benchmark model for comparing MSE and QLIKE. For Reality Checks, the comparisons will be made between ML models against all models in the HAR-family of models. Rahimikia and Poon (2020) also showed that separating normal volatility days and volatility jump days provides essential insights which may not be noticeable when the performance statistics are calculated for both volatility regimes. Hence, this study aims to investigate the potential strengths and weaknesses of forecasting models by separating the forecasts on normal volatility days and volatility jump days in the out-of-sample period.

Based on the high-frequency data for 23 NASDAQ stocks, this paper provides strong statistical evidence that ML models outperform the HAR-family of models and the extended CHAR (CHARx in Rahimikia and Poon (2020)) in RV forecasting according to MSE, QLIKE, and MDA (mean directional accuracy). From the large volume of individual stocks LOB data and news coverage, we find the LOB information provides substantial improvement to volatility forecasting. Adding news sentiments to the information set only improves the forecasting performance marginally. Using a linear model, Rahimikia and Poon (2020) find news sentiment variables to have more information content than LOB variable for volatility forecasting. Our findings here suggest that such conclusion might change when the complex and non-linear relationships are considered in the ML models.

However, the substantial and statistically significant performance improvement is restricted to normal volatility days. The ML models trained using minimisation of MSE as the objective function in the training period (2046 days), provide great improvement in RV forecasting on normal volatility days but caused performance degradation on volatility jump days which is about 10% of the out-of-sample forecasting period (300 days). This finding of stronger forecasting performance on normal volatility days (90% of the out-of-sample period) remains unchanged during robustness checks when the amount of historical information is restricted, the input variables are reduced to a single RV variable, and the machine learning is reduced to a single layer in the fully connected neural network (FCNN).

Throughout this study, we find a persistent trade-off between RV forecasting of normal days vs RV forecasting on volatility jump days. Reducing the number of units in the ML model, and having more comprehensive information sets, all help to improve volatility forecasts on normal days. On volatility jump days, the ML model demands more number of units and much-reduced information set. Indeed, on volatility jump days, the ML models are dominated by a simpler linear model with simple RV history. This finding highlights the need to study normal-day volatility forecasting and volatility jump-day forecasting separately. Based on the findings here and those in Rahimikia and Poon (2020), one should use ML models for volatility forecasting

on normal days, and use the extended CHAR (CHARx) model for volatility forecasting on volatility jump days.

Detailed individual stock analysis of the sensitivity of ML hyperparameters (the number of units and the number of epochs) in the RV forecasting performance suggest optimal performance is achieved with a small number of units (about 5) for most stocks on normal volatility days; the optimal number of epochs (about 20) is the same for most stocks. But on volatility jump days, this is much more variations among the optimal number of units/epochs. This means that it is easier to generalise the findings on normal volatility days, but to use ML models to forecast volatility on jump days, might require much laborious effort and computationally intensive estimations. Again, this highlight to consider normal-day and jump-day volatility forecasting separately.

The remaining of this paper is organised as follows: Section 2 provides the background of machine learning; the RNN model in Subsection 2.1, the LSTM model in Subsection 2.2, and the regularization techniques in Subsection 2.3. Section 3 gives a brief review of the RV and HAR-family of models. Section 4 describes the data, variable definitions and the structure of the proposed ML models used in this study. Section 5 presents the results of the primary experiments. Section 6 performs a series of robustness checks and assesses the sensitivity of the RV forecasting performance with respect to the number of input variables used, the complexity of the ML models and the changing tuning parameters. Finally, Section 7 concludes with a discussion of the findings in this study.

2 Machine Learning for Volatility Forecasting

This section is composed of three subsections. Subsection 2.1 gives a brief review of the RNN as an ML model for modelling sequential data. As an extension of this ML model, Subsection 2.2 discusses the LSTM, and Subsection 2.3 draws together a brief discussion of the applied regularization techniques in this study.

2.1 Recurrent neural network

Figure 1 is a representation of the RNN with t input vectors (X^1,X^2,X^3,...,X^t) and one output (Y^t). W_x and b_x are the shared weights and biases between inputs and the neural network layers for time step 1 to t, W_a and b_a are the shared weights and biases between different layers, and W_y and b_y are the shared weights and biases between the last layer of neural network and the single output (Y^t). Also, in this representation, every layer has an arbitrary number of units which are demonstrated by the hatched circles, a^1 to a^t are the transferred information

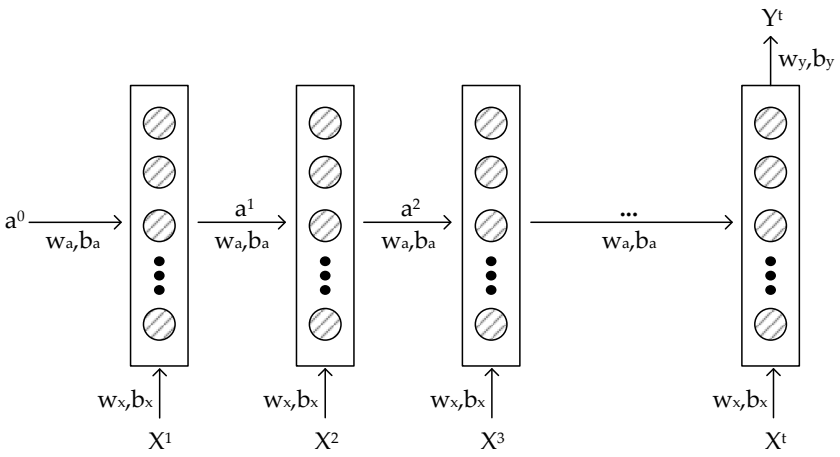


Figure 1: Recurrent neural network abstract representation

Notes: This representation of the RNN has t input vectors through time (X^1,X^2,X^3,...,X^t) and one output (Y^t). W_x and b_x are the shared weights and biases between inputs and the neural network layers for time step 1 to t, W_a and b_a are the shared weights and biases between different layers, and W_y and b_y are the shared weights and biases between the last layer of neural network and the single output (Y^t). Also, in this representation, every layer has an arbitrary number of units which are demonstrated by the hatched circles, a^1 to a^t are the transferred information from one layer to the subsequent layer, and a^0 is the zero input vector for the first layer.

from one layer to the subsequent layer, and a^0 is the zero input vector for the first layer. a^t is defined as follows:

$$a^t = tanh(w_{ax}[a^{t-1}, X^t] + b_{ax}), \tag{1}$$

where w_{ax} is the stacked matrix of w_a and w_x, b_{ax} is the stacked matrix of b_a and b_x, and $tanh$ is the hyperbolic tangent activation function. Also, the output (Y^t) is defined as follows:

$$Y^t = tanh(w_y a^t + b_y), \tag{2}$$

where w_y and b_y are the output weights and biases, and $tanh$ is the hyperbolic tangent activation function. The objective function of minimising MSE is used for optimising the weights and biases in Equation (1) and Equation (2) using the gradient descent with backpropagation as an optimisation algorithm.

Despite the innovative and powerful structure of RNN for modelling complex and nonlinear sequential data through time, this model suffers from the problem of vanishing gradient. As a result, gradients of the loss function may be very small or near zero, complicating the training process of the earlier layers of RNN (in Figure 1, the left-hand side layers). This is the reason why RNN does not work well in capturing long-dependencies in the sequential data, especially for longer sequences. Subsection 2.2 defines an extended structure of the RNN in order to mitigate this weakness.

2.2 Long short-term memory

In 1997, Hochreiter and Schmidhuber demonstrated that adding new gates to the RNN led to a better structure for capturing long-dependencies in the sequential data. The candidate memory cell ($\tilde{c}^t$) is defined as:

$$\tilde{c}^t = tanh(w_c[a^{t-1}, X^t] + b_c), \tag{3}$$

where, w_c and b_c are the weights and biases of the candidate memory cell and $tanh$ is the hyperbolic tangent activation function. Also, the update, forget, and output gates are defined as:

$$G_u = \sigma(w_u[a^{t-1}, X^t] + b_u), \tag{4}$$

$$G_f = \sigma(w_f[a^{t-1}, X^t] + b_f), \tag{5}$$

$$G_o = \sigma(w_o[a^{t-1}, X^t] + b_o), \tag{6}$$

where w_u and b_u are the weights and biases of the update gate, w_f and b_f are the weights and biases of the forget gate, and w_o and b_o are the weights and biases of the output gate. Also, σ is the sigmoid activation function. Taking together, the memory cell (c^t) is defined in the following way:

$$c^t = G_u \times \tilde{c}^t + G_f \times c^{t-1}, \tag{7}$$

where c^{t-1} is the memory cell at time $t-1$ and $\tilde{c}^t$ is the candidate memory cell at time t. Finally, a^t is defined as:

$$a^t = G_o \times tanh(c^t), \tag{8}$$

where G_o is the output gate and c^t is the memory cell in Equation (7) and $tanh$ is the hyperbolic tangent activation function.

Collectively, the role of the update gate in the LSTM is to control the flow of the input activation into the memory cell, and the role of the output gate is to control the output flow of the cell activation into the rest of this neural network. Also, the forget gate in Equation (8) scales the internal state of the cell before adding it as an input to the cell. This extended definition of the RNN prevents this ML model from the vanishing gradient by memorizing and forgetting the past states over time, so theoretically, it is capable of capturing long-dependencies in the sequential data. Subsection 2.3 describes the techniques used in this study for handling over-fitting, one of the most critical steps in the design of ML models.

2.3 Regularization

One of the critical issues of ML modelling is over-fitting when a trained ML model is perfectly fitted to the train data with low training error, but a high error rate in the test data. In general, this means that the trained model memorized all the complexities and non-linearities of the train data but has little flexibility for the unseen test data. Over-fitting is not restricted to ML models only, but because of the highly nonlinear structure of ML models, these models are more prone to the over-fitting problem, which we plan to curtail using regularization techniques.

As Goodfellow et al. (2016) states: 'Regularization is any modification we make to a learning algorithm that is intended to reduce its generalization error but not its training error'. While a variety of regularization techniques for preventing ML models from overfitting have been suggested, this study focuses on the L^2 regularization and dropout techniques, two of the most widely used regularization techniques in the last few years. The L^2 regularization is defined by adding a complexity term to the loss function as follows:

$$\mathcal{L}'(w; X, Y) = \mathcal{L}(w; X, Y) + \lambda.||w||^2, \tag{9}$$

where, $\mathcal{L}$ and $\mathcal{L}'$ are the initial and the modified loss functions with the L^2 regularization, w, X, and Y are, respectively, the weights, inputs, and output of ML model, $||.||^2$ is the L^2 norm, and λ is the regularization factor.

As a hyperparameter, increasing λ leads to a smaller w. Hence, this added term, $\lambda.||w||^2$, to the loss function, helps to reduce ML model overfitting, allowing for higher generalization. This L^2 regularization and the weight decay are useful regularization techniques for the stochastic gradient descent learning algorithms, such as the adaptive gradient algorithms (Loshchilov and Hutter, 2017) we use the in this study.

Turning now to dropout, a regularization technique introduced by Hinton et al. (2012), which has shown superior performance in speech recognition, object detection and other fields. The purpose of dropout is to add noise to the neural network, making the training process more difficult, and consequently decrease the amount of over-fitting by removing some randomly selected units and their incoming and outgoing connections during the training process. As a rule of thumb, optimal dropping out rate typically ranges between 20% and 50% (of the input and hidden units) (Srivastava et al., 2014).

3 Realised Volatility and HAR-family of Models

Suppose that P_t is the price process of an asset and it follows a stochastic differential equation in the following way:

$$dlog(P_t) = \mu_t dt + \sigma_t dw_t, \tag{10}$$

where, μ_t is the drift (continuous function), σ_t is the volatility process (càdlàg function), and w_t is the standard Brownian motion. For time $t-1$ to t, the integrated variance is defined as follows:

$$IV_t = \int_{t-1}^{t} \sigma_s^2 ds. \tag{11}$$

This integrated variance is not observable; therefore, RV is used to proxy IV as follows:

$$RV_t \equiv \sum_{i=1}^{M} r_{t,i}^2, \tag{12}$$

where M is the sampling frequency and $r_{t,i} \equiv log(P_{t-1+i\delta}) - log(P_{t-1+(i-1)\delta})$. For $\delta \to 0$, RV_t is a consistent estimator for IV_t (Andersen and Bollerslev, 1998).

3.1 Realised Volatility Estimation and Statistics

Following Rahimikia and Poon (2020), a sample of 23 more liquid tickers are selected based on the availability of data in $LOBSTER$ for the period from June 28, 2007 to November 17, 2016. (See Subsection 4.1 for a description of the data and the data cleaning procedures.) RV as the dependent variable is calculated according to Equation (12) using 5-minute sampling frequency consistent with Li and Xiu (2016). Table 1 provides the RV descriptive statistics for the 23 tickers.

3.2 HAR-family of models

Turning now to the HAR-family of models as the most popular RV forecasting models. The HAR model is defined by Corsi (2009) as follows:

$$RV_{t+1} = \beta_0 + \beta_1 RV_t + \beta_2 \overline{RV}_{t-1}^{w} + \beta_3 \overline{RV}_{t-1}^{m} + \epsilon_{t+1}, \tag{13}$$

where RV_{t+1} is used for forecasting RV at time $t+1$, RV_t is the first lag of RV, $\overline{RV}_{t-1}^{w}$ is the average daily RV of the last week, and $\overline{RV}_{t-1}^{m}$ is the average daily RV of the last 21 days. Corsi (2009) showed that this easy to estimate linear model with a simple set of historical RVs produced some remarkable forecasting performance. It was since the work of Corsi (2009) that the research on improving RV forecasting performance has gained momentum. Researchers

Table 1: RV descriptive statistics

Ticker[a]	Min	Max	1st quantile	Median	3rd quantile	Mean	STD	Kurtosis	Skewness
AAPL	0.102	198.574	1.021	1.879	3.987	4.953	12.664	83.884	8.028
MSFT	0.067	133.679	0.935	1.577	2.918	3.289	7.268	104.476	8.786
INTC	0.146	130.167	1.165	1.920	3.709	4.065	7.596	71.609	6.875
CMCSA	0.148	153.376	0.951	1.827	3.751	4.014	8.419	95.178	8.139
QCOM	0.122	373.543	0.959	1.872	3.873	4.673	13.778	280.384	13.730
CSCO	0.163	343.946	1.030	1.791	3.438	4.348	12.995	262.347	13.440
EBAY	0.215	252.608	1.461	2.592	4.946	5.525	12.785	139.670	9.850
GILD	0.222	259.489	1.383	2.179	3.900	4.719	13.706	173.238	11.815
TXN	0.183	287.897	1.111	1.999	3.987	4.107	9.249	398.325	15.651
AMZN	0.206	547.030	1.683	2.882	5.720	7.562	23.925	185.115	11.593
SBUX	0.164	192.629	1.086	1.968	4.308	4.714	11.255	114.155	9.331
NVDA	0.317	1104.351	2.180	4.382	9.414	9.591	29.432	837.584	24.558
MU	0.563	359.620	4.204	7.137	13.584	14.355	26.204	61.344	6.711
AMAT	0.292	114.376	1.715	2.812	5.150	5.153	8.149	61.231	6.438
NTAP	0.257	290.647	1.594	2.903	5.743	6.283	14.419	149.830	10.163
ADBE	0.216	569.720	1.153	2.081	3.952	5.059	15.730	693.479	21.309
XLNX	0.229	251.383	1.224	2.184	4.258	4.359	9.382	265.118	12.977
AMGN	0.159	214.156	1.006	1.727	3.126	3.468	9.764	221.110	13.295
VOD	0.134	219.033	0.780	1.487	3.369	4.252	11.788	115.204	9.471
CTSH	0.246	485.894	1.214	2.162	5.266	6.103	17.479	315.162	14.555
KLAC	0.154	499.808	1.278	2.514	5.126	5.689	17.915	395.464	17.684
PCAR	0.214	389.930	1.285	2.563	5.800	6.014	13.514	294.091	12.810
ADSK	0.358	693.772	1.637	2.823	5.256	6.833	24.263	413.773	17.814

[a] Tickers are ranked based on the liquidity (high to low) and availability of data for the selected timespan from June 28, 2007, to November 17, 2016.

investigated expanding the basic HAR model with various information sets and high-frequency data to enhance RV forecasting performance.

In Corsi and Reno (2009), the impact of adding a jump component to the basic HAR model was analysed. First, with the use of BPV (bipower variations) below, the jump at time t is $J_t = Max[RV_t - BPV_t, 0]$:

$$BPV_t = \frac{1}{\mu_1^2} \sum_{i=1}^{M-1} |r_{t,i}||r_{t,i+1}|, \tag{14}$$

where M is the maximum value of sampling frequency, $r_{t,i}$ is the return at the day t, and sampling frequency of i, and $\mu_1 = \sqrt{2/\pi}$. As a variation to Equation (13), the CHAR model replaces the predictive variables with BPV_t, the first lag of BPV, $\overline{BPV}^w_{t-1}$, the average daily BPV of the last week, and $\overline{BPV}^m_{t-1}$, the average daily BPV of the last 21 days. Rahimikia and Poon (2020) conduct an extensive comparison, and find CHAR to be the best performing model among all HAR-family of models for forecasting RV.

Patton and Sheppard (2015) devise the SHAR model to investigate the impact of negative and positive intraday returns on the subsequent RV. In the proposed SHAR model, the first lag of RV in the HAR model (Equation (13)) is replaced with the positive return (RV_t^+) and

negative return (RV_t^-) variables where $RV_t^+ = \sum_{i=1}^{M} r_{t,i(r>0)}^2$ and $RV_t^- = \sum_{i=1}^{M} r_{t,i(r<0)}^2$. The authors find the subsequent volatility is more strongly related to the volatility of the past negative returns compared to the positive ones.

From the asymptotic theory for RV forecasting, Bollerslev et al. (2016) study the impact of measurement error on volatility forecasting. First, they defined the integrated quarticity, $IQ_t = \int_{t-1}^{t} \sigma_{\sigma_s}^4 ds$ and its discrete time equivalent, realised quarticity $RQ_t \equiv (\frac{M}{3}) \sum_{i=1}^{M} r_{t,i}^4$. Next, they introduced the ARQ, HARQ, and HARQ-F models. For example, the ARQ model is defined as follows:

$$\begin{aligned} RV_{t+1} &= \beta_0 + (\beta_1 + \beta_{1Q} RQ_t^{1/2}) RV_t + \epsilon_{t+1} \\ &= \beta_0 + \beta_1 RV_t + \beta_{1Q} RQ_t^{1/2} RV_t + \epsilon_{t+1}. \end{aligned} \tag{15}$$

For $\beta_{1Q} < 0$, RV_t has a lower impact when measurement error is large, and a higher impact when the measurement error is smaller. If $\beta_{1Q} = 0$, Equation (15) reduces to the AR model. When $\overline{RV}_{t-1}^{w}$ and $\overline{RV}_{t-1}^{m}$ are added to the RHS together with $\overline{RQ}_{t-1}^{w}$ and $\overline{RQ}_{t-1}^{m}$, we get HARQ. Bollerslev et al. (2016) find HARQ model to have better forecasting performance, producing more volatility persistency in 'normal times' and a quicker volatility mean reversion in 'erratic times'. For a comprehensive review of the HAR-family of models and their forecasting performance, see Rahimikia and Poon (2020).

3.3 Volatility Forecasts and Evaluations

For the full sample period of June 28, 2007, to November 17, 2016, the training period is from June 28, 2007, to September 10, 2015 (2046 days), and the out-of-sample forecasting period is from September 11, 2015, to November 17, 2016 (300 days). Following Poon and Granger (2003), the model forecast power is judged based on the out-of-sample forecasting performance only. Rahimikia and Poon (2020) argued that reporting the forecasting performance metrics without discriminating between normal days and jumps can be misleading for the RV forecasting model comparison. To separate normal and volatility jump days, a day is defined as a jump when its RV value is greater than $Q3 + 1.5IQR$; otherwise, this day is considered as a normal day. In this definition, $Q1$ and $Q3$ are the first and third quantiles, and IRQ is equal to $Q3 - Q1$. All these values are calculated for every ticker separately using all the observations in the out-of-sample data.[1]

Consistent with Rahimikia and Poon (2020) and Bollerslev et al. (2016), the modified reality check (RC) (Sheppard, 2009) is applied to the forecast results of ML model against the benchmark HAR-family of models. In line with White (2000), the stationary bootstrap of

[1]Rahimikia and Poon (2020) reported that based on this definition, from the 300 days of out-of-sample data, on average, 10% (30 days) are selected as volatility jumps, and the remaining, normal volatility days.

Politis and Romano (1994) with 999 re-samplings and the average block length of 5 are used for this test. The hypotheses are defined in the following way:

$$\begin{aligned} H_0 &: \underset{k=1,\ldots,n}{Min} \ \mathbb{E}[L^k(RV, X) - L^0(RV, X)] \leq 0, \\ H_1 &: \underset{k=1,\ldots,n}{Min} \ \mathbb{E}[L^k(RV, X) - L^0(RV, X)] > 0, \end{aligned} \tag{16}$$

where L^k is the loss of the benchmark models for the first to the n^{th} model and L^0 is the loss of the desired model. A rejection of the null hypothesis implies that the loss of the desired model is significantly smaller than the benchmark models. This test is applied to all proposed ML models here.

Following Patton (2011) and Rahimikia and Poon (2020), the MSE and QLIKE loss functions are chosen for reporting the RV forecasting performance. According to Patton (2011), the MSE and QLIKE loss functions are among a family of robust and homogeneous loss functions for volatility forecasting; for a review of the volatility performance metrics, see Poon and Granger (2003). Therefore, rankings of volatility forecasts based on these loss functions are robust to noise in the proxy, and also, they are invariant to the choice of units of measurement. These loss functions are defined as follows:

$$MSE(RV_t, \widehat{RV_t}) \equiv (RV_t - \widehat{RV_t})^2, \tag{17}$$

$$QLIKE(RV_t, \widehat{RV_t}) \equiv \frac{RV_t}{\widehat{RV_t}} - log(\frac{RV_t}{\widehat{RV_t}}) - 1, \tag{18}$$

where RV_t is the true RV at time t, and $\widehat{RV_t}$ is the fitted (forecasted) RV at time t.

4 Data and ML Model Structure

This section covers two main areas, viz. data and ML model specifications. Subsection 4.1 describes the data and data cleaning process, while Subsection 4.2, Subsection 4.3, and Subsection 4.4 describe the variables from, respectively, HAR-family of models, limit order book, and news sentiment. These are the three core sets of information used in volatility forecasting. The forth information set is the amalgamation of these three information sets. Finally, Subsection 4.5 describes the proposed ML model structure implemented in this study.

4.1 LOBSTER Database and Data Cleaning

The LOBSTER dataset is used for extracting the HAR-family variables in Subsection 4.2, and the Limit Order Book variables in Subsection 4.3. Before calculating these variables, for the

preprocessing of *the LOBSTER* dataset, the modified proposed cleaning steps in Rahimikia and Poon (2020) based on Barndorff-Nielsen et al. (2009) are used in this study. These steps are applied to both the limit order book and message data in the following way (the step names, such as P2, T4, ..., are consistent with the names in Barndorff-Nielsen et al. (2009)):

- **P2:** Delete entries with a bid, ask or transaction price equal to zero.
- **T4:** Delete entries with prices that are above the 'ask' plus the bid-ask spread, or below the 'bid' minus the bid-ask spread.
- **Q1:** When multiple quotes share the same timestamp, they are replaced by a single entry using the median bid price, median ask price, and the sum of all volumes from these multiple quotes. For messages with the same direction (buy or sell), the mentioned procedure is applied to the message data, and the last snapshot of the LOB is selected as the LOB associated with the merged message data. For messages with different directions, the message data and the LOB with the same direction are grouped, and the mentioned procedure is applied separately to buy-side and sell-side.
- **Q2**: Delete entries for which the spread is negative.
- **Q3**: Delete entries for which the spread is more than 50 times the median spread on that day.
- **Q4**: Delete entries for which the mid-quote deviated by more than 10 mean absolute deviations from a rolling centred median (excluding the observation under consideration) of 50 observations (25 observations before and 25 after).

In this study and following Rahimikia and Poon (2020), a sample of 23 more liquid tickers is selected based on the availability of data from June 28, 2007, to November 17, 2016. The data cleaning summary statistics is summarised in Table 2. From this table, it is apparent that the T4 and Q1 cleaning steps have the greatest cleaning percentages among all cleaning steps.

4.2 HAR-family variables

Table 3 lists the variables used in the HAR-family of models. The first column ('Description') contains the name of the variables. 'RV' is defined in Section 3, 'BPV' and 'BPV jump' are defined in Barndorff-Nielsen and Shephard (2004), 'negative; positive RV' denotes the negative and positive RV measures in Patton and Sheppard (2015), and 'realised quarticity' is defined and used in Bollerslev et al. (2016) to investigate the role of measurement error. The second column ('#') contains the number of variables used for the defined variable in the first column. The formula to compile the defined variable is shown in the last column ('Characteristic'). This

Table 2: Data cleaning summary statistics

Name[a]	Ticker	Sample size	Cleaned (%)	P2 (%)	T4 (%)	Q1 (%)	Q2 (%)	Q3 (%)	Q4 (%)
Apple	AAPL	2237122415	35.58	0.01	30.04	5.53	0.00	0.00	0.01
Microsoft	MSFT	2014344642	40.95	0.01	33.04	7.89	0.00	0.00	0.01
Intel	INTC	1589557456	42.86	0.01	34.76	8.07	0.01	0.00	0.01
Comcast	CMCSA	1548795613	50.29	0.01	44.54	5.74	0.00	0.00	0.01
Qualcomm	QCOM	1444079951	44.60	0.01	38.82	5.77	0.00	0.00	0.01
Cisco Systems	CSCO	1369100802	43.97	0.02	35.67	8.28	0.00	0.00	0.01
eBay	EBAY	1125822804	43.23	0.01	37.90	5.30	0.00	0.00	0.01
Gilead Sciences	GILD	928876476	43.22	0.01	39.73	3.48	0.00	0.00	0.01
Texas Instruments	TXN	986266021	44.05	0.01	38.36	5.68	0.00	0.00	0.01
Amazon.com	AMZN	732847153	28.73	0.01	25.51	3.20	0.00	0.00	0.02
Starbucks	SBUX	912122148	48.62	0.01	43.80	4.80	0.00	0.00	0.01
Nvidia	NVDA	862724899	45.87	0.01	39.02	6.82	0.00	0.00	0.01
Micron Technology	MU	846541573	39.21	0.01	32.90	6.27	0.01	0.00	0.01
Applied Materials	AMAT	839175351	45.40	0.01	37.99	7.38	0.00	0.00	0.01
NetApp	NTAP	775503643	47.05	0.01	42.66	4.37	0.00	0.00	0.02
Adobe	ADBE	750683192	46.75	0.01	42.58	4.14	0.00	0.00	0.02
Xilinx	XLNX	764214024	47.21	0.01	42.79	4.39	0.00	0.00	0.02
Amgen	AMGN	662265397	42.81	0.02	38.38	4.40	0.00	0.00	0.02
Vodafone Group	VOD	709434297	46.60	0.01	43.71	2.87	0.00	0.00	0.02
Cognizant	CTSH	630525626	48.83	0.01	45.58	3.21	0.00	0.00	0.02
KLA Corporation	KLAC	636397484	47.86	0.01	44.80	3.04	0.00	0.00	0.02
Paccar	PCAR	601367391	48.85	0.01	45.89	2.92	0.00	0.00	0.02
Autodesk	ADSK	609555304	47.97	0.01	44.84	3.09	0.00	0.00	0.02
Average		1025101028	44.3700	0.0108	39.2743	5.0713	0.0008	0.0000	0.0143

Notes: **P2:** Delete entries with a bid, ask or transaction price equal to zero, **T4:** Delete entries with prices that are above the 'ask' plus the bid-ask spread, or below the 'bid' minus the bid-ask spread, **Q1:** When multiple quotes share the same timestamp, they are replaced by a single entry using the median bid price, median ask price, and the sum of all volumes from these multiple quotes. For messages with the same direction (buy or sell), the mentioned procedure is applied to the message data, and the last snapshot of the LOB is selected as the LOB associated with the merged message data. For messages with different directions, the message data and the LOB with the same direction are grouped, and the mentioned procedure is applied separately to buy-side and sell-side, **Q2**: Delete entries for which the spread is negative, **Q3**: Delete entries for which the spread is more than 50 times the median spread on that day, **Q4**: Delete entries for which the mid-quote deviated by more than 10 mean absolute deviations from a rolling centred median (excluding the observation under consideration) of 50 observations (25 observations before and 25 after) [a] Tickers are ranked based on the liquidity (high to low) and availability of data for the selected timespan from June 28, 2007, to November 17, 2016.

Table 3: HAR-family variables

Description	#	**Characteristic**
RV	1	$RV_t \equiv \sum_{i=1}^{M} r_{t,i}^2$
BPV	1	$BPV_t = \frac{1}{\mu_1^2} \sum_{i=1}^{M-1} \lvert r_{t,i} \rvert \lvert r_{t,i+1} \rvert$
BPV jump	1	$J_t = MAX(RV_t - BPV_t, 0)$
Negative; positive RV	1	$RV_t^+ \equiv \sum_{i=1}^{M} r_{t,i}^{2+}$; $RV_t^- \equiv \sum_{i=1}^{M} r_{t,i}^{2-}$
Realised quarticity	1	$RQ_t \equiv (\frac{M}{3}) \sum_{i=1}^{M} r_{t,i}^4$

'RV' is defined in Section 3, 'BPV' and 'BPV jump' are defined in Barndorff-Nielsen and Shephard (2004), 'negative; positive RV' denotes the negative and positive RV measures in Patton and Sheppard (2015), and 'realised quarticity' is defined and used in Bollerslev et al. (2016) to investigate the role of measurement error. The second column ('#') contains the number of variables used for the defined variable in the first column. The most commonly used 5-minute sampling frequency is used for calculating these variables.

group of HAR-family has five defined variables in total. The most commonly used 5-minute sampling frequency is also used for calculating these variables.

4.3 Limit order book variables

The LOBSTER dataset is used for extracting the limit order book variables. This dataset contains the limit order book data and message data of the NASDAQ market. The *CRSP* database is used for correcting the limit order book data and the message data for the effect of stock splits, stock dividends, spin-offs, stock distributions, and rights on price and volume. Table 4 lists the LOB variables as defined in Kercheval and Zhang (2015) for up to $N = 10$ levels of the limit order book giving rise to 134 variables in total. The fourth column ('Parameter') provides the value for the time parameter. For the variables extracted from the limit order book and message data, which include execution, submission, cancellation, and deletion of orders, the cleaning steps in Subsection 4.2 are applied.

To examine the impact of the limit order book variables on the RV forecasting performance, Rahimikia and Poon (2020) consider only the slope and depth of the limit order book. Here, we extend Rahimikia and Poon (2020) by considering a much larger set of LOB variables compiled from different types of orders. Such analysis is feasible because of the power and flexibility of ML models for handling big and complex data.

4.4 News variables

The database, *Dow Jones Newswires*, covers the Wall Street Journal, MarketWatch, and Barron's news. In this database, every story is tagged with 'significant', 'about', or 'mention'. 'Significant' denotes news story that is important to a referenced ticker; 'about' denotes a story about a ticker, but of no particularly significant, while 'mentioned' denotes cases where the ticker is referenced but is not the main subject of the news story. As 'Significant' is not implemented for the entire sample period, the tag ('about') is used for extracting related news stories for the sample of 23 tickers in this study.

Table 5 lists the nine daily news variables extracted from the news data. Apart from 'News count', the other eight sentiment variables, 'negative', 'positive', 'uncertainty', 'litigious', 'weak modal', 'moderate modal', 'strong modal', and 'constraining', are compiled based on the latest (2018) version of ML dictionary (Loughran and McDonald, 2011).[2] The averaging is done across the sentiment measures of all news stories appearing in the *Dow Jones Newswires* during the day. To address the importance of a word in a story, the term frequency-inverse document

[2]The latest version of LM dictionary (2018) is downloaded from Software Repository for Accounting and Finance, University of Notre Dame.

Table 4: Limit order book variables

Description	#	**Characteristic**[a]	**Parameter**
Bid-ask spreads	10	$[(P_i^{ask} - P_i^{bid})]_{l=1}^{N}$	-
Mid prices	10	$[(P_i^{ask} + P_i^{bid})/2]_{l=1}^{N}$	-
Price differences	18	$[P_N^{ask} - P_1^{ask}, P_N^{bid} - P_1^{bid}]_{l=1}^{N}$	-
Absolute price differences	20	$[\lvert P_{l+1}^{ask} - P_i^{ask}\rvert, \lvert P_{l+1}^{bid} - P_i^{bid}\rvert]_{l=1}^{N}$	-
Mean prices	2	$[\frac{1}{n}\sum_{l=1}^{N} P_i^{ask}, \frac{1}{n}\sum_{l=1}^{N} P_i^{bid}]$	-
Mean volumes	2	$[\frac{1}{n}\sum_{l=1}^{N} V_i^{ask}, \frac{1}{n}\sum_{l=1}^{N} V_i^{bid}]$	-
Price/volume accumulated differences	2	$[\sum_{l=1}^{N}(P_i^{ask} - P_i^{bid}), \sum_{l=1}^{N}(V_i^{ask} - V_i^{bid})]$	-
Price/volume derivatives	40	$[dP_i^{ask}/dt,\, dP_i^{bid}/dt,\, dV_i^{ask}/dt,\, dV_i^{bid}/dt]$	$dt = 1day$
Average intensity[b]	10	$[\lambda_{\Delta t}^{(E)bid(ask)}, \lambda_{\Delta t}^{(S)bid(ask)}, \lambda_{\Delta t}^{(C)bid(ask)}, \lambda_{\Delta t}^{(D)bid(ask)}]$	$\Delta_t = 1day$
Relative intensity[c]	10	$[1_{\{\lambda_{\Delta t}^{(E)bid(ask)} > \lambda_{\Delta T}^{(E)bid(ask)}\}}, 1_{\{\lambda_{\Delta t}^{(S)bid(ask)} > \lambda_{\Delta T}^{(S)bid(ask)}\}}, 1_{\{\lambda_{\Delta t}^{(C)bid(ask)} > \lambda_{\Delta T}^{(C)bid(ask)}\}}, 1_{\{\lambda_{\Delta t}^{(D)bid(ask)} > \lambda_{\Delta T}^{(D)bid(ask)}\}}]$	$\Delta_T = 1day$, $\Delta_t = 15mins$
Accelerations[d]	10	$[d\lambda^{(E)bid(ask)}/dt,\, d\lambda^{(S)bid(ask)}/dt,\, d\lambda^{(C)bid(ask)}/dt,\, d\lambda^{(D)bid(ask)}/dt]$	$dt = 1day$

This table contains the proposed variables in Kercheval and Zhang (2015). The first column ('Description') contains the name of the added variables from the limit order book data. The second column ('#') contains the number of variables extracted from every defined variable in the first column for ten levels of the limit order book. The characteristic of every defined variable is shown in the third column ('Characteristic'). The fourth column ('Parameter') provides the value for the time parameter. [a] (E), (S), (C), and (D) stand for execution, submission, cancellation, and deletion of orders, respectively. Also, 'P' and 'V' stand for price and volume. [b] The ratio of every defined variable as the nominator to the total number of orders during that day. [c] This value is a binary number (one or zero). [d] The nominators are the calculated variables in the 'average intensity' group of variables.

Table 5: News data variables

Description	#	Characteristic[a]
News count	1	*Number of news*
Positive sentiment	1	*Average of positive sentiments*
Negative sentiment	1	*Average of negative sentiments*
Uncertainty sentiment	1	*Average of uncertainty sentiments*
Litigious sentiment	1	*Average of litigious sentiments*
Weak modal sentiment	1	*Average of weak modal sentiments*
Moderate modal sentiment	1	*Average of moderate modal sentiments*
Strong modal sentiment	1	*Average of strong modal sentiments*
Constraining sentiment	1	*Average of constraining sentiments*

The first column ('Description') contains the name of the added variables from the news data. The second column ('#') contains the number of variables extracted from every defined variable in the first column. The characteristic of every defined variable is shown in the last column ('Characteristic'). All sentiments in this table are calculated based on the latest version (2018) of ML dictionary (Loughran and McDonald, 2011).
[a] 'News count' variable contains the daily number of stories. Also, average values are the average of the specified sentiment in the first column for all stories during that day.

frequency (tf.idf) weighting scheme is applied to the variable calculation in Subsection 4.4. The steps for preprocessing the news data follow those in Rahimikia and Poon (2020) and Loughran and McDonald (2011).

4.5 ML model structure

A single layer LSTM model is chosen here because it has several advantages over the more complex ML models. First, as mentioned in Section 1, LSTM is well-known for analysing sequential data both in academia and industry, and it's among the top ML choices for modelling time series data. Second, a simple vanilla LSTM is employed here as it is easier to follow the behaviour of this model compared to other more complex ML structures. Third, a simpler LSTM model helps to overcome the heavy reliance on a large training dataset needed by more complex models for optimising a large number of parameters. Finally, complex ML models have more combinations of hyperparameters to test and choose from, which is overly complex at this stage of the research.

Figure 2 presents the structure of the ML models used in this study. For every explanatory/predictive variable (X), n lags of that variable (X^t,X^{t-1},...,X^{t-n}) are used as input to the LSTM model. The number of outputs from the LSTM (D_O) is equal to the number of units of this model. An FCNN is used for converting the outputs from the LSTM model to RV forecast ($\widehat{RV}^{t+1}$).

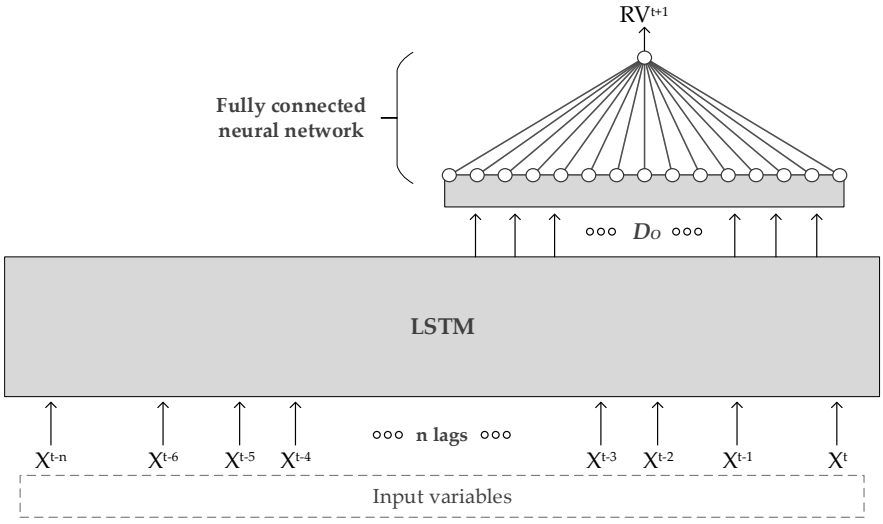


Figure 2: ML model structure

Notes: For every input variable (X), n lags of that variable (X^t,X^{t-1},...,X^{t-n}) are considered as the input variables of the LSTM model. The number of outputs of the LSTM (D_O) is equal to the number of units of this model. A fully connected neural network (FCNN) is used for converting outputs of the LSTM to the subsequent RV (RV^{t+1}).

The ML model specifications in this study are as follows: the selected learning algorithm is ADAM (Kingma and Ba, 2014), one of the most commonly used learning algorithms. ADAM is an adaptive learning rate optimisation algorithm based on stochastic gradient descent. The learning rate and weight decay are set to 0.001 and $1e-5$, respectively. For both the LSTM and FCNN, the kernel and bias L^2 regularisation techniques are applied with $1e-5$ as the decay parameter. As another regularisation technique, the dropout is applied between the LSTM and FCNN with the rate of 0.5. For the LSTM, the sigmoid activation function is chosen for the cell and hidden states, and the tangent hyperbolic activation function is chosen for the input, forget, and output gates. Also, for the FCNN, the rectifier activation function[3] is chosen in order to prevent the model from generating negative output. For the primary experiments in Section 5, minimising MSE is chosen as the objective function, and the number of epochs is set equal to 50. Also, the LSTM is tested with 5, 10, 15, 20, and 25 units for increasing complexities. In this proposed ML structure, the FCNN has one layer (1 unit).

For the primary experiments in Section 5, the use of 21 lags (n in Figure 2) of all variables is consistent with the HAR-family of models in Section 3. As the number of variables tested in the ML model is far greater than the HAR-family of models, even with the restriction to 21 lags. In the second stage (robustness checks in Section 6), we examine the sensitivity of the findings to the amount of input data used by the ML models in formulating the volatility forecasts. Finally, all input variables are standardised by removing the mean and scaling to unit variance. This standardisation is executed at every rolling window estimation. Of the two window sizes tested in Rahimikia and Poon (2020) (1023 days and 2046 days), the larger

[3]A unit with the rectifier activation function is called a rectified linear unit (ReLU).

Table 6: Number of independent variables and parameters of models

Group	**HAR-family of models**							
Model	AR	CHAR	HAR-J	CHAR	SHAR	ARQ	HARQ	HARQ-F
Number of variables	1	3	4	3	4	2	4	6
Number of parameters	1	4	5	4	5	3	5	7

Group	**ML models**[a]			
Model	HAR-ML	News-ML	OB-ML	News/OB-ML
Number of variables	$6 \times$ NoL[b]	$15 \times$ NoL	$138 \times$ NoL	$147 \times$ NoL
Number of parameters (5 units)[c]	246	426	2886	3066
Number of parameters (10 units)	691	1051	5971	6331
Number of parameters (15 units)	1336	1876	9256	9796
Number of parameters (20 units)	2181	2901	9256	13461
Number of parameters (25 units)	3226	4126	16426	17326

The HAR-ML contains the HAR-family variables (described in Subsection 4.2). The News-ML contains the limit order book variables (described in Subsection 4.3). The OB-ML contains the news variables (described in Subsection 4.4), and finally, the News/OB-ML includes all the mentioned variables together. [a] All the prosed ML models contain the HAR-family variables. [b] Number of lags (21 lags for the primary experiments). [c] Number of units of the LSTM model.

sample size (2046 days) is used in this study.

Table 6 lists the number of independent variables and the number of parameters for all HAR-family (ML) models in the top (bottom) panel. It is cleared that the ML models use far more input variables than the HAR-family of models. The News/OB-ML model, combining all data set, has 147×21 variables compared to only one variable in the AR model. The ability to deal with nonlinear and complex relationships between variables is a strong feature of ML models. Note also that adding more lags to the ML models does not change the number of parameters in the LSTM model; only the number of variables and the number of units determine the number of variables to be modelled.

In accordance with Rahimikia and Poon (2020), by applying a rolling window procedure, for 300 out-of-sample days, 300 ML models are trained and applied on the out-of-sample data for forecasting the subsequent RV. Also, for every selected ticker, just the data for that ticker is used for training the models. For having reproducible results, the same seed for the random number generator (RNG) is used for all models in this study. Another way to obtain reproducible results is by repeating the training process many times for every model and calculate the average of the forecast outputs, which is not computationally feasible in this study.

5 Out-of-Sample Forecasting Results

This section compares the forecasting performance of four ML models, differ by their information sets, against the $CHAR$ model, which is the best performing HAR-family model in Rahimikia and Poon (2020) estimated using OLS. If we define, for stock i, and for each out-of-

sample forecast from ML model j,

$$\Delta_{MSE,i,j} = MSE_i^{ML}\ (Model_j) - MSE_i^{OLS}\ (CHAR)\ \ and \tag{19}$$

$$\Delta_{QLIKE,i,j} = QLIKE_i^{ML}\ (Model_j) - QLIKE_i^{OLS}\ (CHAR), \tag{20}$$

where $Model_j$ $(j = 1, 2, 3, 4)$ is one of the four ML models with $j = 1$ as 'HAR-ML' consisting of the HAR-family variables described in Subsection 4.2. The second group ('News-ML') contains the LOB variables (described in Subsection 4.3). The third group ('OB-ML') contains the News variables (described in Subsection 4.4), and finally, the fourth group ('News/OB-ML') includes both LOB and News variables. It is worth noting that 'News-ML', 'OB-ML' and 'News/OB-ML' also contain the HAR-family variables from the base case.

In equations (19) and (20), a negative $\Delta_{MSE/QLIKE,i,j}$ indicates an improvement from using ML, and a positive value indicates degradation in MSE/QLIKE. Next, model $j's$ average $\Delta_{MSE,j}$ and average $\Delta_{QLIKE,j}$ of the 23 stocks are:

$$Average\ \Delta_{MSE,j} = \frac{1}{23}\sum_{i=1}^{23}\ \Delta_{MSE,i,j}, \tag{21}$$

$$Average\ \Delta_{QLIKE,j} = \frac{1}{23}\sum_{i=1}^{23}\ \Delta_{QLIKE,i,j}, \tag{22}$$

while $Median\ \Delta_{MSE,j}$ and $Median\ \Delta_{QLIKE,j}$ being the median equivalent of Equation (21) and Equation (22) respectively.

Table 7 reports the average and median $\Delta_{MSE/QLIKE,j}$ for the four ML models with 21 lags of every variables included, and different number of units (5, 10, 15, 20, and 25). The corresponding RC value is the percentage of tickers with better ML performance, in terms of MSE or QLIKE, at the 5% and 10% significance levels compared to all HAR-family of models estimated using OLS. First, consider the top half of Table 7 for normal volatility days which constitute 90% of the out-of-sample forecast period. All four ML models outperformed the OLS CHAR model as average/median MSE/QLIKE are all negative. The RC values, when MSE measure is used, are above 91% and are often 100%. When QLIKE is used, the minimum RC value is 69.57%. This is strong evidence supporting the forecasting power of ML against all HAR-family of models. Across the four ML models, 'OB-ML' predicts better than 'News-ML', but 'News/OB-ML' using all information set, appears to be the best specification. In general, the lower the number of units, the better the forecasting performance for normal volatility days.

The results for the volatility jump days in the bottom half of Table 7 present a very different picture. All four ML models under-performed the OLS CHAR model as average/median

Table 7: Out-of-sample performance and RC results for ML models (primary experiment)

			HAR-ML					News-ML					OB-ML					News/OB-ML				
Normal days		Units	5	10	15	20	25	5	10	15	20	25	5	10	15	20	25	5	10	15	20	25
MSE		Avg	-2.417[a]	-1.641	-1.433	-1.415	-1.296	-2.381	-1.503	-1.248	-1.119	-0.685	-2.726	-3.118	-2.555	-2.105	-1.540	-2.590	***-3.176***	-2.557	-2.055	-1.381
		Med	***-2.540***[b]	-1.105	-0.971	-0.969	-0.861	-2.272	-1.077	-1.011	-0.664	-0.363	-2.239	-2.361	-2.182	-1.672	-0.695	-2.307	-2.306	-1.980	-1.779	-1.448
RC	0.05		***100***[c]	100	95.65	100	100	100	95.65	91.30	91.30	91.30	100	100	100	100	95.65	100	***100***	100	95.65	95.65
	0.10		***100***[c]	100	100	100	100	100	100	95.65	91.30	100	100	100	100	100	95.65	100	***100***	100	100	95.65
QLIKE		Avg	-0.033	-0.012	-0.012	-0.019	-0.026	-0.014	-0.008	-0.017	-0.029	-0.020	0.046	-0.077	-0.048	-0.056	-0.053	0.039	***-0.077***	-0.057	-0.054	-0.046
		Med	-0.052	-0.026	-0.028	-0.024	-0.028	-0.076	-0.011	-0.015	-0.018	-0.008	-0.064	-0.063	-0.041	-0.051	-0.048	-0.060	***-0.067***	-0.047	-0.042	-0.038
RC	0.05		91.30	78.26	86.96	91.30	91.30	91.30	69.57	78.26	82.61	82.61	69.57	95.65	91.30	100	95.65	69.57	***100***	100	91.30	86.96
	0.10		86.96	78.26	82.61	91.30	91.30	91.30	78.26	86.96	86.96	95.65	73.91	100	95.65	100	95.65	78.26	***100***	100	100	91.30
Jumps																						
MDA		Avg	72.442	39.434	26.251	19.385	16.860	75.312	37.747	20.337	14.473	***9.001***	89.475	52.391	34.211	23.828	13.989	87.998	52.321	30.532	22.318	13.782
		Med	26.728	10.679	8.105	5.370	3.014	33.849	7.035	5.769	***0.984***	4.350	41.329	19.836	18.445	12.777	10.884	42.052	24.925	14.697	9.357	7.726
RC	0.05		4.35	13.04	26.09	43.48	26.09	4.35	8.70	39.13	***52.17***	39.13	0.00	13.04	21.74	30.44	26.09	0.00	21.74	21.74	39.13	30.44
	0.10		8.70	34.78	47.83	60.87	60.87	13.04	21.74	65.22	***73.91***	52.17	8.70	26.09	39.13	39.13	47.83	8.70	39.13	39.13	56.52	43.48
MDA		Avg	1.477	0.579	0.483	0.370	0.426	1.918	0.581	0.511	***0.401***	0.724	6.392	1.985	1.946	1.275	1.190	5.883	1.929	1.396	1.875	1.778
		Med	0.957	0.355	0.277	0.235	0.339	1.341	0.383	***0.360***	0.394	0.415	3.342	1.192	1.711	1.185	1.043	3.123	1.277	1.280	1.429	1.059
RC	0.05		0.00	8.70	17.39	30.44	26.09	0.00	8.70	17.39	***30.44***	26.09	0.00	17.39	17.39	13.04	13.04	0.00	13.04	13.04	17.39	17.39
	0.10		4.35	39.13	39.13	56.52	56.52	13.04	26.09	47.83	***56.52***	47.83	4.35	26.09	34.78	30.44	39.13	0.00	26.09	26.09	30.44	34.78

Notes: [a] The difference between the average of the out-of-sample MSEs of the specified ML model and the CHAR model (the best performing HAR-family model in Rahimikia and Poon (2020)) for 23 tickers (negative value shows improvement, and positive value shows degradation in performance). [b] The difference between the median of the out-of-sample MSEs of the specified ML model and the CHAR model (the best performing HAR-family model in Rahimikia and Poon (2020)) for 23 tickers (negative value shows improvement, and positive value shows degradation in performance). [c] Percentage of tickers with outstanding performance at the 0.05 and 0.10 significance levels of the RC compared to the HAR-family of models as the benchmark for every specified ML model considering different number of units (5, 10, 15, 20, and 25) for both the MSE and QLIKE loss functions.

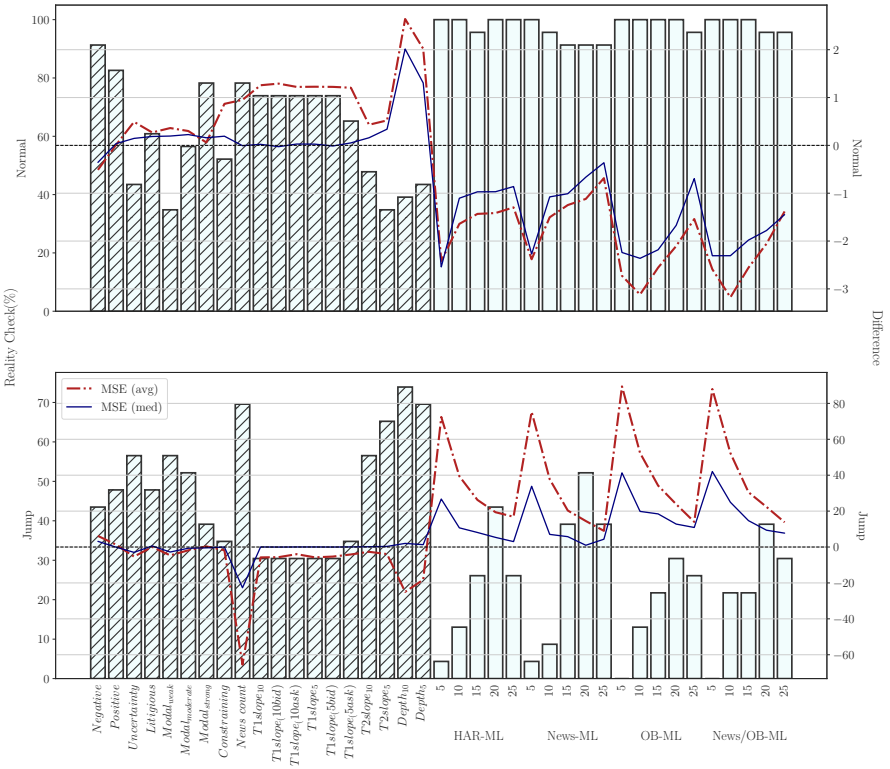
MSE/QLIKE are all positive. The RC values, when MSE measure is used, are very low and the majority of which are well below 50%, and even worse when QLIKE is used. In general, the smaller the number of units, the worse performance from the ML models. This contrasting results to normal volatility days suggest ML models, as they are currently specified here, are not appropriate for forecasting extreme volatility. One should exercise caution when volatility becomes extreme. Under such circumstance, one could either switch back to the simple linear model estimated using OLS or devise a better ML specification that has volatility jumps as the main target in the training period.

The findings in Table 7 contradict those reported in Rahimikia and Poon (2020) that shows News variables have stronger predictive power than LOB variables. The different finding here may be due to the ability of the ML model in dealing with a large number of variables and complex nonlinear relationships simultaneously. Indeed, in Rahimikia and Poon (2020), 10 LOB variables and 9 News sentiment measures are studied, but only one variable was added to the $CHAR$ model at a time. In the ML models tested here, we have 134 LOB variables, 9 News variables and 5 HAR-family variables; where all the variables from the same set are added to the base equation simultaneously. So the 'News/OB-ML' model has 148 variables (i.e. 5 HAR-family, 9 News, and 134 LOB) in total in the ML training and forecasting periods.

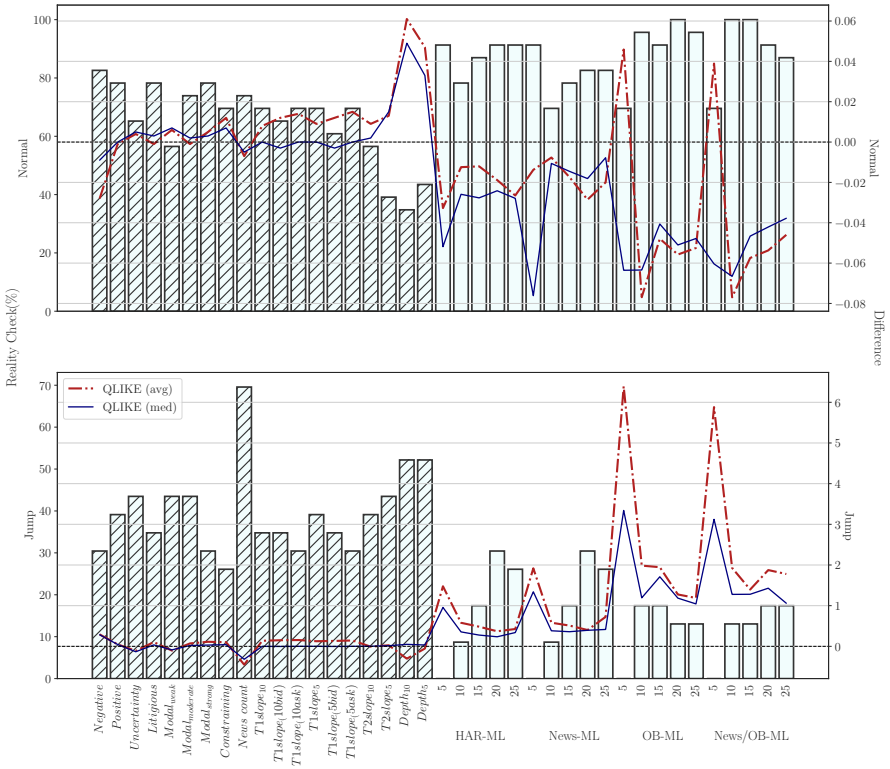
Figure 3 provides a comparison of the four ML models (white bars) in this study and the extended $CHARx$ models (hatched bars) in Rahimikia and Poon (2020). The white (hatched) bar reports the RC values of a specific ML ($CHARx$) model against all HAR-family of models. The dashed (solid) line represents the out-of-sample average (median) $\Delta_{MSE,j}$ in (a) and average (median) $\Delta_{QLIKE,j}$ in (b). Among the $CHARx$ models (hatched bars), the News variables are defined in Subsection 4.4. The LOB variables include the type 1 modified slope measure (Næs and Skjeltorp (2006)), the type 2 slope measure (Kalay et al. (2004)), and the LOB depth. A variable name, $T1slope(5bid)$, means type 1 slope measure aggregated from the first five levels on the bid side of the LOB. If a LOB variable name does not contain 'ask' or 'bid', it means that it is calculated using both bid and ask sides of the LOB. The $CHARx$ model, includes by default, RV^d_{t-1}, RV^w_{t-1} and RV^m_{t-1}.

The most striking result in Figure 3 is the superior forecasting performance of the four ML models (white bars) compared to $CHARx$ (hatched bars) estimated using OLS, for normal volatility days in the upper graphs. In contrast, all four ML models performed poorly in the volatility jump days in the lower graphs. News count and LOB depth are the only variables that, when added to the $CHARx$ model, help to forecast volatility on jump days. The results for the Mean Directional Accuracy (MDA) presented in Figure A1 of Appendix A, broadly support these findings here, viz. a great improvement in RV forecasting performance for normal

Figure 3: ML models and extended CHARx models comparison



(a) Forecast Evaluation Under MSE



(b) Forecast Evaluation Under QLIKE

Notes: The bar chart is the percentage of tickers with the outstanding performance considering the MSE loss function at the 0.05 significance level of the RC compared to the all HAR-family of models as the benchmark for every specified extended CHARx model (hatched bars) and the ML model (white bars). The values for the bar chart can be read from the left-hand axis. The dashed (solid) line represents the difference between the average (median) of the out-of-sample MSEs of the specified model with the CHAR model (the best performing HAR-family model in Rahimikia and Poon (2020)) for 23 tickers (negative value shows improvement, and positive value shows degradation in performance). The values for the dashed and solid lines can be read from the right-hand axis. The horizontal dashed line represents no improvement.

volatility days when switching from $CHARx$ to ML, but significant degradation in performance for days with volatility jumps.

5.1 Individual Stocks Radar Plots

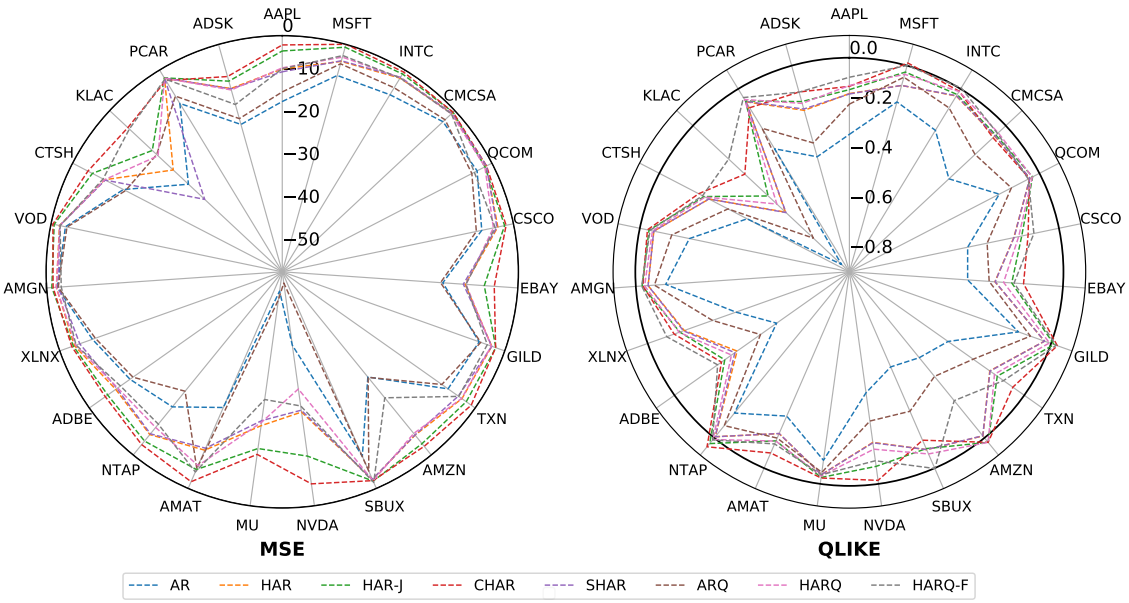
To understand the changes in forecasting performance at the ticker and model levels, Figure 4 presents the $\Delta_{MSE,i}$ and $\Delta_{QLIKE,i}$ between News/OB-ML (10 units) and each and every model in the HAR-family for models for every ticker, i, using radar plots.[4] The MSE (QLIKE) results are presented on the left (right), and the normal (jump) volatility days results are at the top (bottom). A negative Δ value shows improvement, and a positive Δ indicates degradation. The bold circle inside of the radar chart represents $\Delta = 0$; there is no improvement from using the ML model over HAR-family model.

For normal volatility days in Figure 4a, it is apparent that 'News/OB-ML' outperformed every model in the HAR-family of models as $\Delta_{MSE,i}$ and $\Delta_{QLIKE,i}$ lie inside the bold circle for almost every ticker. Consistent with the findings in Rahimikia and Poon (2020), the red dotted line, representing CHAR, lies closest to the bold circle suggesting that CHAR has the best forecasting performance among the HAR- family of models, albeit not as good as ML. For volatility jump days presented in Figure 4b at the bottom, most $\Delta_{MSE,i}$ and $\Delta_{QLIKE,i}$ are positive and lie outside the bold circle; switch from HAR-family model to ML caused performance degradation for almost every ticker and every HAR-family model. Since QLIKE placed greater emphasis on the mis-forecast of larger volatility, the degradation is less severe in the case of QLIKE. For MSE, the $\Delta_{MSE,i}$ is extremely large for 'MU', and then 'ADSK'. From the descriptive statistics of RV in Table 1, 'MU' has the greatest maximum, mean, standard deviation, and the lowest kurtosis. Moreover, 'ADSK' is among the tickers with the extreme minimum, maximum, mean, standard deviation, kurtosis, and skewness. This is a clear indication that 'News/OB-ML' (10 units) as specified and trained here are ill-fitted for forecasting these realised volatilities with extreme distributions. Rahimikia and Poon (2020) document similar findings among $CHARx$ augmented with News sentiment or LOB variable. These more sophisticated models perform poorly for extreme RV for stocks with extreme RV distributions.

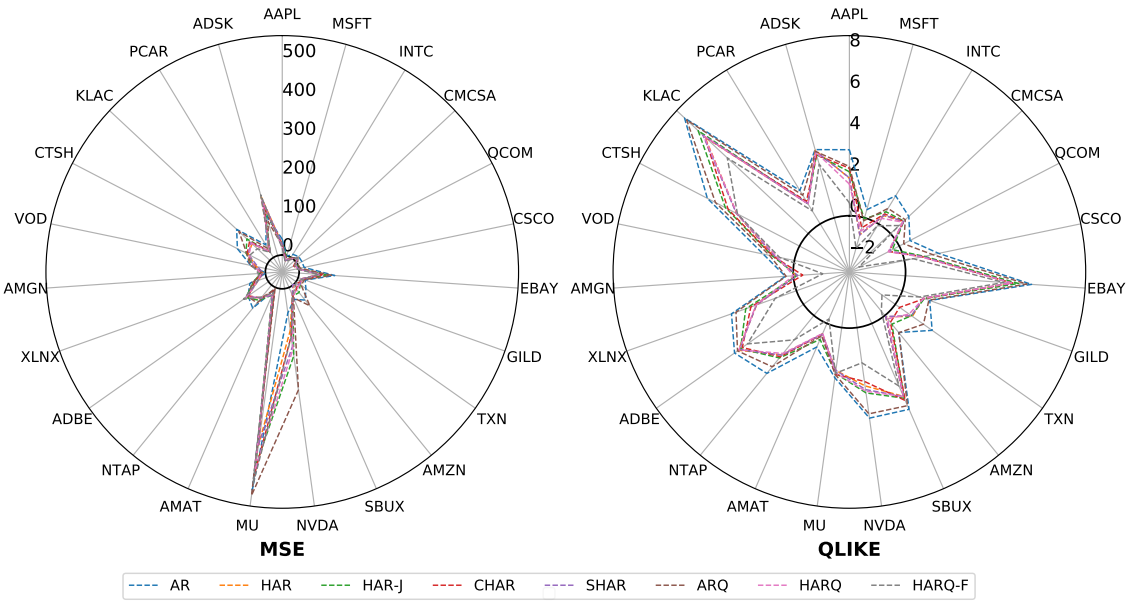
As noted in Section 5, adding more units to the ML model improved the RV forecasting performance on jump days but degraded the performance on normal volatility days. Figure 6 uses box chart to present the distribution of true RV for normal days, and the forecast RVs from News/OB-ML model with 10 and 25 units, and CHAR respectively. For the sake of clarity, RV on the y-axis is truncated at 100%. The points above each box chart are jumps. There

[4]That is AR, HAR, HAR-J, CHAR, SHAR, ARQ, HARQ, and HARQ-F.

Figure 4: $\Delta_{MSE,i}$ and $\Delta_{QLIKE,i}$ between News/OB-ML (10 units) and HAR-family of Models



(a) Normal Volatility Days



(b) Volatility Jump Days

Notes: Every radar chart contains the difference between the performance of the News/OB-ML model with ten units with the AR, HAR, HAR-J, CHAR, SHAR, ARQ, HARQ, and HARQ-F models for the mentioned tickers considering normal volatility days (a) and volatility jump days (b). The left and right radar charts contain results for the MSE and QLIKE loss functions, respectively. For these loss functions, the negative value shows improvement. The bold circle inside of these radar charts show no improvement (zero).

are some important observations from Figure 6. First, CHAR (the best performing model in Rahimikia and Poon (2020)) produced some forecasts that are classified as jumps. None of the ML models produced any jumps. Both CHAR and ML models produced RV forecasts that are more contained, and hence they performed poorly on jump days. Second, from the true RV in the first chart on the left, it is apparent that 'MU' has a distribution that is very extreme when compared to the other tickers. Third, an increase in ML model complexity (with more units) increased the magnitudes of forecast RVs.

5.1.1 Increasing the number of units in News/OB-ML

Figure 5 shows the impact of increasing the number of units from 10 to 25 in the ML model. Comparing Figure 5a with Figure 4b, it is clear that a more complex ML model with a larger number of units improved the forecasting performance of RV on volatility jump days; the improvement is more marked for stocks that have extreme distribution such as 'MU'. Figure 5b compares the true (dashed line) and the forecast (solid line) RV in the out-of-sample period for 'ADSK' from three models, viz. News/OB-ML with 10 and 25 units and $CHAR$. For the sake of clarity, the RV values at the y-axis are truncated at 100%.

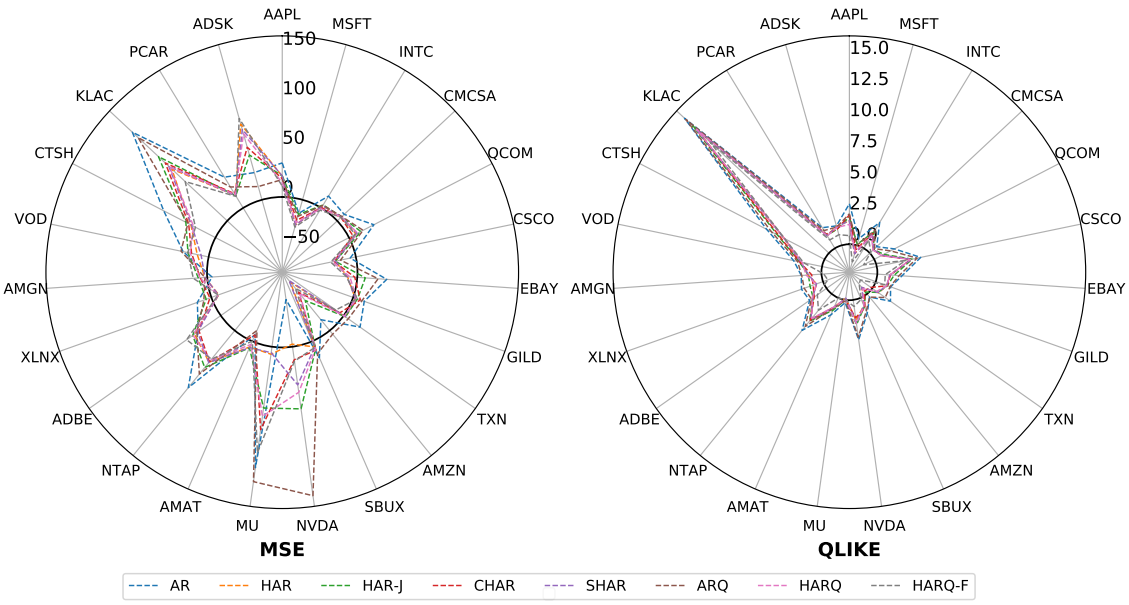
What stands out in Figure 5b is the difference between $CHAR$, the best performing HAR-family model in Rahimikia and Poon (2020), and the two ML models. On high volatility days, the forecast RV from $CHAR$ moves much closer to the true RV than the forecast RV from the two ML models. Closer inspections of the other HAR-family of models revealed the same pattern. In general, the HAR-family of models produced forecasts that have higher means and higher standard deviations in the out-of-sample period than those from the ML models.[5] This is one of the reasons for their better forecasting performance for volatility jump days. Indeed, increasing the number of units in the ML model also has the effect of increasing the mean and the standard deviation of the RV forecasts, resulting in better forecasting performance on volatility jump days.

5.2 Directional forecasting performance

It is a widely held view that ML models are excellent for classification and directional forecasts. Hence, this section investigates the MDA of ML models compare with the HAR-family of models

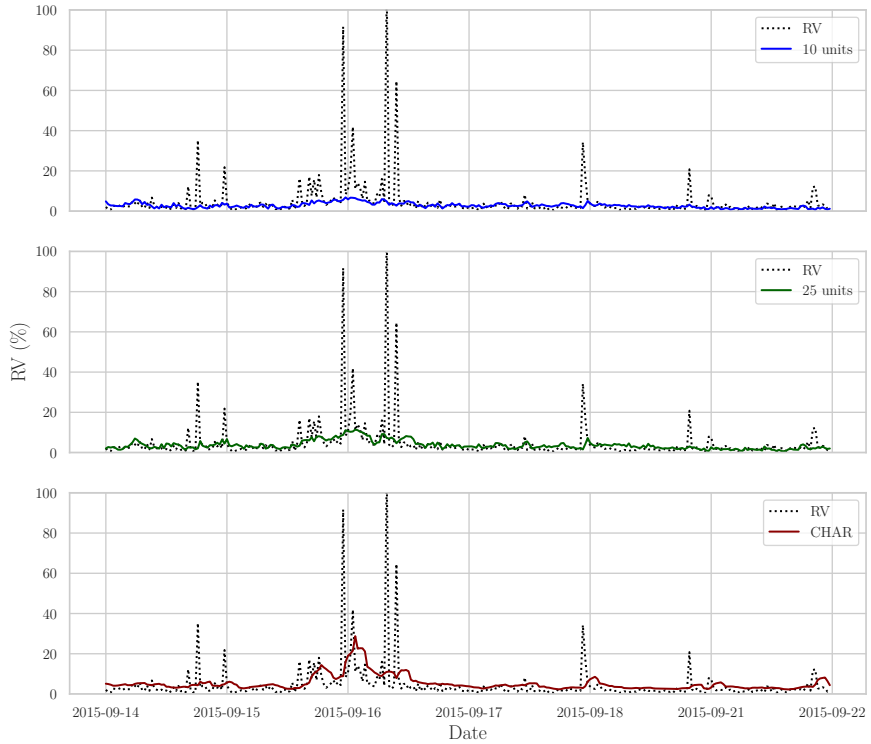
[5]For the 23 stocks, the forecasts from AR, HAR, HAR-J, CHAR, SHAR, ARQ, HARQ, and HARQ-F are higher than the true RV, respectively, 88.3%, 84.0%, 84.7%, 84.9%, 83.8%, 87.1%, 83.7%, and 77.6% of the time, while the forecasts from News/OB-DL with 10 and 25 units are higher than the true RV, 63.0% and 67.3% of the time respectively. The standard deviations of the forecasts from the eight HAR-family of models are higher than the standard deviation of the forecast from News/OB-DL with 10 units (25 units) for, respectively, 65.2%, 95.6%, 95.6%, 91.3%, 95.6%, 95.6%, 95.6%, and 100% (56.5%, 69.6%, 65.2%, 52.2%, 69.6%, 91.3%, 82.6%, and 95.6%) of the 23 tickers.

Figure 5: Increasing the number of units in News/OB-ML



(a) News/OB-ML (25 units) performance on volatility jump days

Notes: The radar chart plots $\Delta_{MSE/QLIKE}$ between the ML and each of the HAR-family of models. A negative value inside the bold circle indicates improvement; a positive value outside the circle indicates a performance degradation.



(b) True vs forecast RV for 'ADSK'

Notes: The top, middle, and bottom charts show the true (dashed) and out-of-sample forecast (solid) RV of 'ADSK' for News/OB-ML with 10 and 25 units and CHAR respectively. For the sake of clarity, the RV(%) on the y-axes are truncated at 100%.

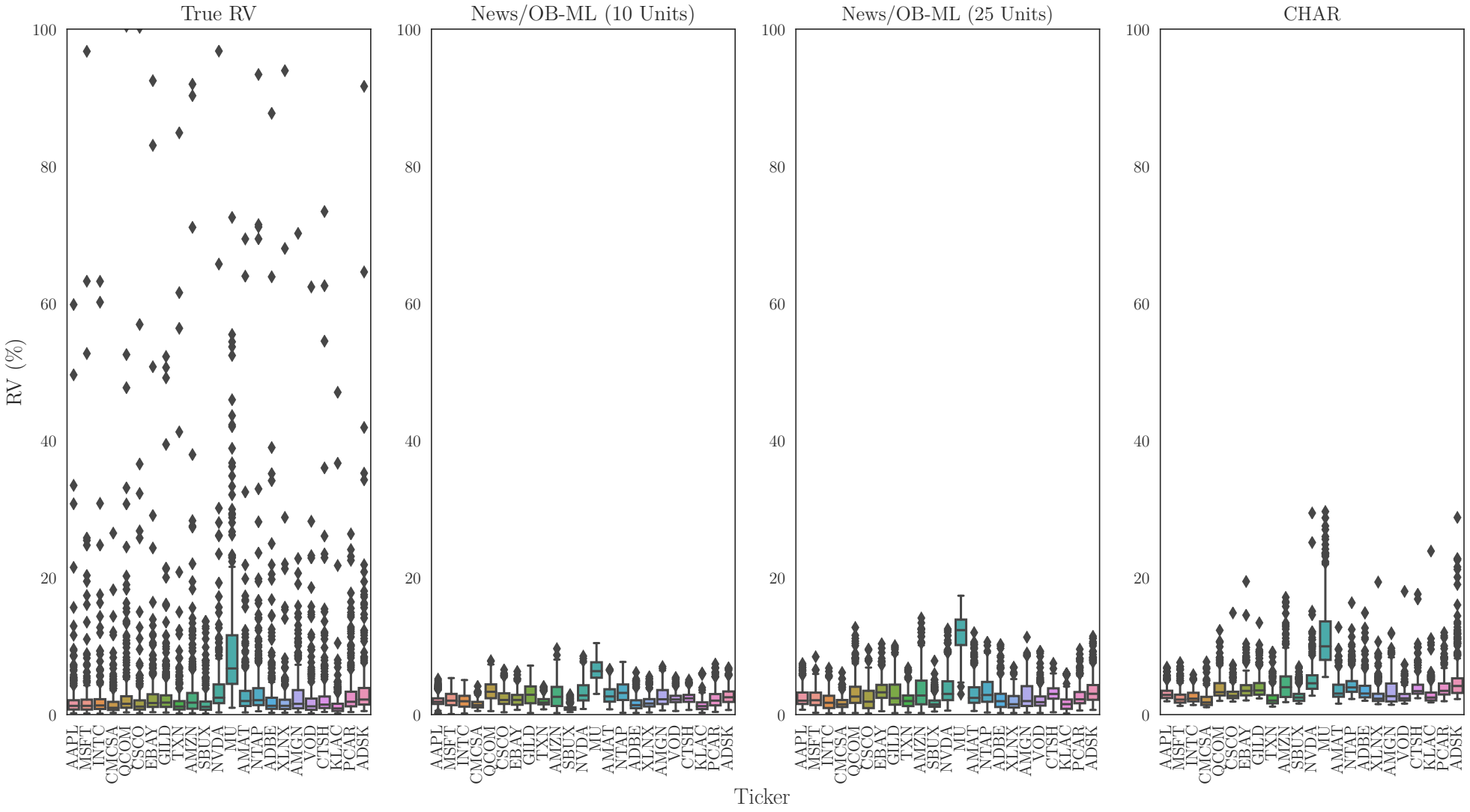


Figure 6: True RV and forecast RVs from News/OB-ML and CHAR

Notes: From left to right, box charts show the distribution of the true RV, the forecast RVs from News/OB-ML models with 10 and 25 units, and CHAR respectively. For the sake of clarity, RV at the y-axis is truncated at 100%. The points above each box chart are jumps.

using the MDA measures defined below:

$$MDA = \frac{1}{N}\sum_{t=1}^{N} 1_{sign(RV_t - RV_{t-1}) == sign(\widehat{RV_t} - RV_{t-1})}, \tag{23}$$

$$\Delta_{MDA,i,j} = MDA_i^{ML}\ (Model_j) - MDA_i^{OLS}\ (CHAR), \ \ and \tag{24}$$

$$Average\ \Delta_{MDA,j} = \frac{1}{23}\sum_{i=1}^{23} \Delta_{MDA,i,j}, \tag{25}$$

where, RV_t is the true RV at time t, $\widehat{RV_t}$ is the forecast RV at time t, N is the number of days in the out-of-sample period, $sign(\cdot)$ and 1 are the sign and indicator functions, and $Median\Delta_{MDA,j}$ is the median equivalent of $Average\Delta_{MDA,j}$ for model j.

Table 8 presents the out-of-sample directional forecasting performance and the RC results for the four ML models with 5, 10, 15, 20 and 25 units. First, consider the normal volatility days; the results show clearly that all four ML models outperformed $CHAR$ in terms of average and median Δ_{MDA}. The RC values at 5% and 10% are all very large, ranging between 78-100%. As before, 'OB-ML' performs better than 'News-ML', but 'News/OB-ML', encompassing all information set, has the best performance. Also, ML models with a smaller number of units perform better. In contrast, on volatility jump days, all four ML models under-performed the $CHAR$ model. All the average and median Δ_{MDA} are negative, and the RC values at the 5% significant level range between 0-35%. The larger the information set, and the smaller the number of units, the worse is the performance. Hence, forecasts from 'HAR-ML' with the smaller number of units is the least harmful ML model on volatility jump days.

Figure 7 presents the $\Delta_{MDA,i,k}$ from the News/OB-ML (10 units) against the HAR-family of models for stock i on normal volatility days on the left, and on volatility jump days on the right. The $\Delta_{MDA,i,k}$ is calculated for every HAR-family model k. In these radar charts, a positive value indicates improvement by switching from HAR to ML, and a negative value indicates a performance degradation. The bold circle inside the radar chart marked the border of no difference in performance. Consistent with Table 8, for normal volatility days on the left, almost all the Δ_{MDA} lie outside the bold circle, where ML outperformed every HAR-family model. In some cases, this improvement is nearly 20%. In contrast, for volatility jump days on the right, switching from HAR to ML caused degradation in forecasting performance as almost all the Δ_{MDA} lie inside the bold circle. Collectively, the MDA results correspond to those from MSE and QLIKE discussed previously.

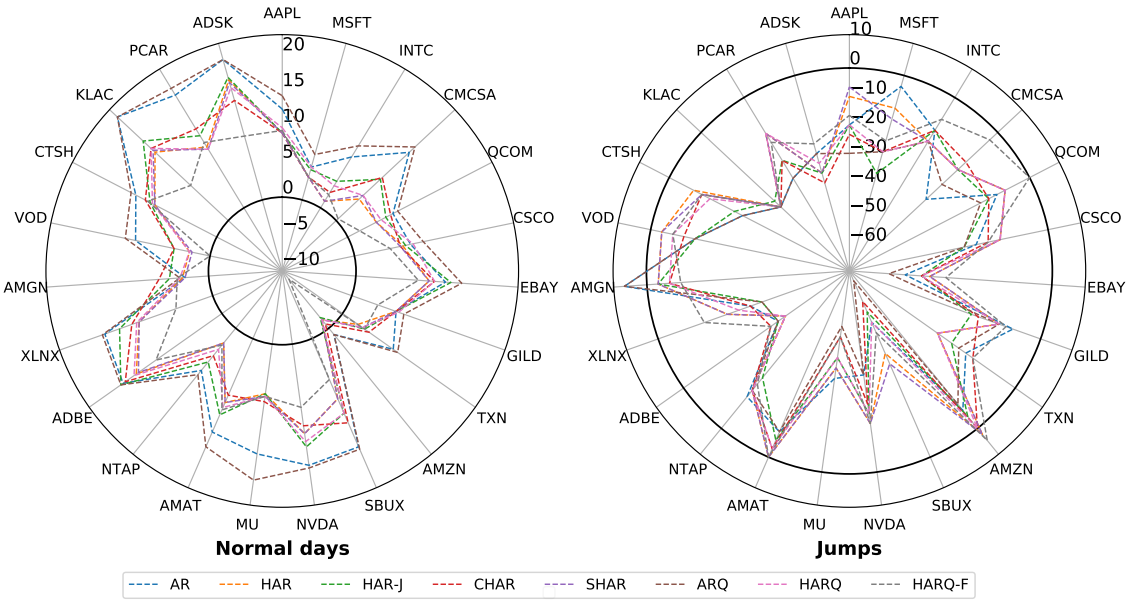


Figure 7: News/OB-ML (10 units) MDA performance for individual stocks

Notes: Every radar chart contains the difference between the MDA performance of the News/OB-ML model with ten units with the AR, HAR, HAR-J, CHAR, SHAR, ARQ, HARQ, and HARQ-F models for the mentioned tickers. The left and right radar charts contain results of normal days and jumps, respectively. For these radar charts, the positive value shows improvement, and the negative value shows degradation in performance. The bold circle inside of these radar charts shows no improvement (zero).

Table 8: Out-of-sample directional forecasting performance and RC results for four ML models (primary experiment)

			HAR-ML					News-ML					OB-ML					News/OB-ML				
Normal days		Units	5	10	15	20	25	5	10	15	20	25	5	10	15	20	25	5	10	15	20	25
MDA		Avg	5.96[a]	3.66	3.13	2.46	2.60	6.92	3.90	3.85	4.62	4.35	8.85	8.40	6.82	6.96	5.78	9.12	8.62	7.18	6.54	6.54
		Med	5.64[b]	2.93	3.42	2.56	2.55	7.35	3.30	3.80	4.36	3.77	9.89	8.24	6.56	6.08	5.99	9.70	8.61	8.12	6.25	6.32
RC	5%		91.30[c]	82.61	78.26	82.61	82.61	95.65	86.96	91.30	95.65	95.65	100	100	100	100	95.65	100	95.65	100	100	95.65
	10%		91.30[c]	82.61	86.96	86.96	82.61	95.65	91.30	91.30	100	100	100	100	100	100	100	100	95.65	100	100	100
Jumps																						
MDA		Avg	-19.76	-8.24	-7.74	-6.25	-7.02	-22.54	-13.19	-11.14	-9.94	-9.95	-38.59	-21.92	-18.25	-15.71	-15.25	-37.91	-23.79	-16.01	-17.79	-15.51
		Med	-15.38	-3.85	-4.35	-4.17	-4.17	-19.44	-11.54	-11.54	-7.69	-9.38	-34.78	-16.13	-18.18	-16.67	-16.67	-6.67	-22.50	-12.90	-17.24	-17.86
RC	5%		4.35	13.04	26.09	34.78	30.44	4.35	17.39	13.04	13.04	26.09	0	4.35	13.04	13.04	8.70	0	13.04	13.04	13.04	13.04
	10%		13.04	43.48	43.48	60.87	65.22	4.35	34.78	30.44	26.09	39.13	0	8.70	17.39	21.74	17.39	0	13.04	13.04	21.74	17.39

Notes: [a] $Average\Delta_{MDA,j}$ and [b] $Median\Delta_{MDA,j}$ of 23 stocks, with a positive value indicates improvement, and a negative value indicates performance degradation.
[c] Percentage of tickers with outstanding ML performance at the 5% and 10% significance levels of the RC compared to all HAR-family of models based on MDA.

6 Robustness Checks

Up to now, the primary experiments above show that, in comparison with the HAR-family of models, the ML model enhanced with LOB and News variables works well for RV forecasting on days with normal volatility, but registered a performance degradation on volatility jump days (about 10% of the sample period). However, it is not clear if the superior performance of ML is due to the much larger data sets used, the more complex algorithms, or the much heavier computer-intensive method that is employed. In this section, we perform a series of secondary robustness tests, to see if equipped with a smaller data set, and a simpler data structure, the ML method can continue to outperform the HAR-family of models on normal volatility days.

The secondary experiments in this section include restricting input information sets, a simpler ML algorithm, and a large combination of tuning parameters. Apart from these experimented parameters, all other model specifications are the same as those adopted in Section 5.

6.1 Restricted Information Set, $\Phi_t = \{t-1,\ ..., t-5\}$

One may argue that the superior ML performance is solely due to the large amount of information set that it uses. Here, we restrict the information set to contain only five lags (equivalent to the last week) and one lag (i.e. the previous day) of all predictive variables. An LSTM with one lag (one input and one output) has a similar structure as a standard neural network. Figure 8 presents the average (solid) and median (dotted) Δ_{MSE} and Δ_{QLIKE}, as well as the RC values (bar chart). The results show that restricting the information set from 21 days (in Table 7) to 5-days and 1-day here did not change the superior performance of the ML models for normal volatility days when evaluated using MSE (see Figure 8b for 5-days MSE and Figure 8c for 1-day MSE). The 1-day result is weaker than the 5-day result, possibly due to some outliers since the average is worse than the median. As before, the ML methods are not suited for forecasting on volatility jump days. The results in the bottom half of Figure 8 for volatility jump days, and the results based on QLIKE in the bottom halves of Figure 8e and Figure 8f (which gives more weights to under-forecasts) all support this conclusion.

The MDA results presented in Figure B1 of Appendix B also consistent with the findings here. For normal volatility days, the RC values of ML models are high, and many are reaching 100%. The improvement in MDA by switching from CHAR to ML ranges approximately between 2% to 10% for ML with different number of units and different number of lagged variables included in the information variables. In summary, these results show that, for normal volatility days, with a smaller number of lags, ML still outperforms the HAR-family of models in RV forecasting.

Figure 8: ML Models With Restricted Information Set $\Phi_t = \{t-1, ..., t-5\}$

(a) Forecasts Evaluated Under MSE

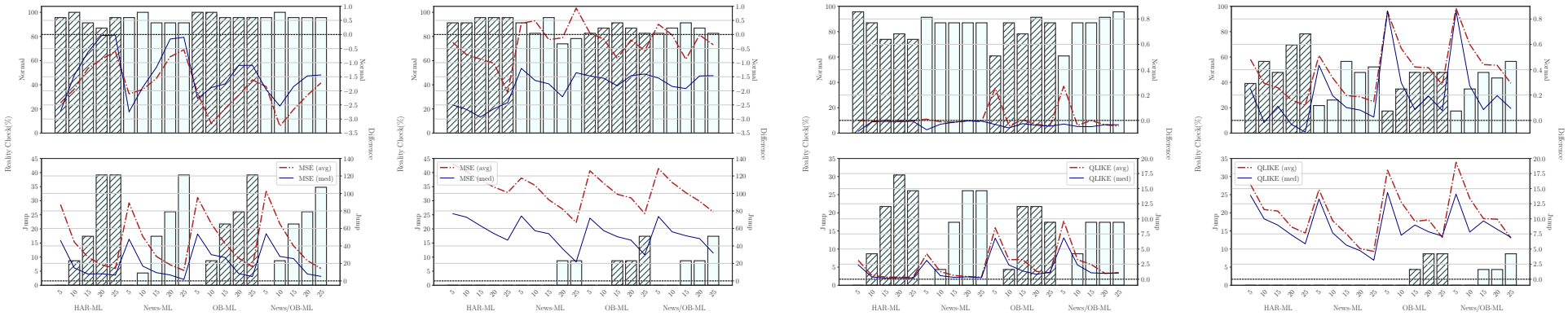
(b) Five lags

(c) One lag

(d) Forecasts Evaluated Under QLIKE

(e) Five lags

(f) One lag



Notes: The left and right figures display the results considering five lags (last week) and one lag (last day). The bar chart is the percentage of tickers with the outstanding performance considering the MSE/QLIKE loss function at the 0.05 significance level of the RC compared to the HAR-family of models as the benchmark for every HAR-ML, News-ML, OB-ML, and News/OB-ML model. The values for the bar chart can be read from the left-hand axis. The dashed (solid) line represents the difference between the average (median) of the out-of-sample MSEs of the HAR-ML, News-ML, OB-ML, and News/OB-ML models and the CHAR model (the best performing HAR-family model in Rahimikia and Poon (2020)) for 23 tickers (negative value shows improvement, and positive value shows degradation in performance). The values for the dashed and solid lines can be read from the right-hand axis. The horizontal dashed line represents no improvement.

6.2 21-day Historical RV, $\Phi_t = \{RV_{t-1},\ ..., RV_{t-21}, RV^d_{t-1}, RV^w_{t-1}, RV^m_{t-1}\}$

In this section, the information set is restricted to 21 lags of RV only. The results are presented in Figure 9, where $\Phi_t = \{RV_{t-1},\ ..., RV_{t-21}\}$ for the LSTM model in Figure 9a, and $\Phi_t = \{RV^d_{t-1}, RV^w_{t-1}, RV^m_{t-1}\}$ for the simpler FCNN model in Figure 9b.[6] Within each subgraph, MSE (QLIKE) is on the left (right), and normal (jump) volatility days are at the top (bottom).

In the top-left (North-West corner) of both Figure 9a and Figure 9b under MSE for normal volatility days, ML model with a smaller number of units continue to outperform CHAR and HAR-family of models. The average and median Δ_{MSE} are negative when the number of units is 5. The RC values are between 75-95%. As before, the ML models perform poorly on volatility jump days. Looking at these figures, it is clear that even with only one explanatory variable, the ML models can outperform HAR-family of models.

The MDA results presented in Appendix B (Figure B2 for LSTM, and Figure B3 for FCNN) show a cohesive picture; ML provides a substantial MDA improvement over HAR-family of models for normal volatility days. The improvement is greater for smaller number of units, and the RC values are as high as 90%. Nevertheless, the ML performance is poor on volatility jump days.

6.3 MSE vs QLIKE as Objective Function

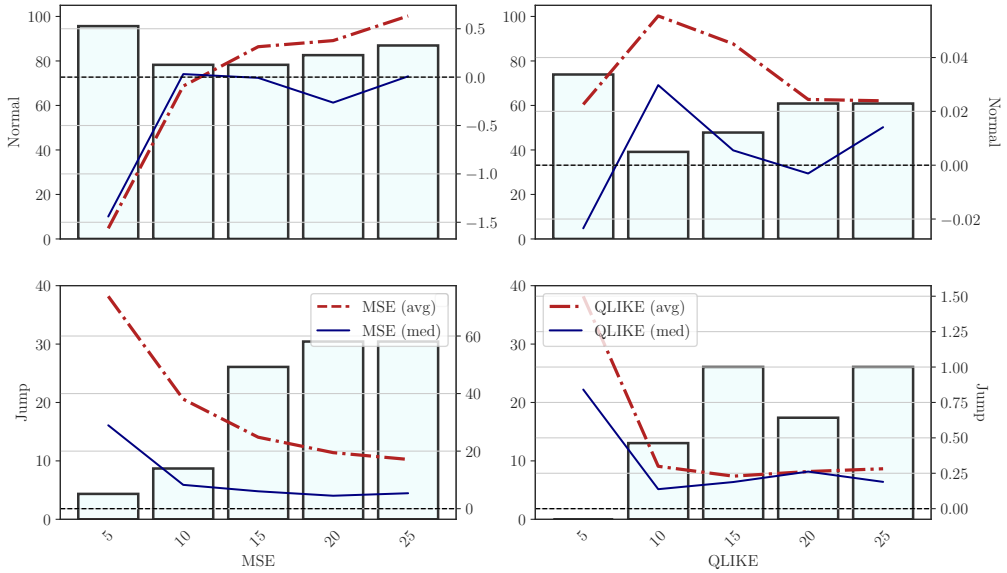
All the ML model results reviewed so far are based on minimising MSE as the objective function in the training period while using MSE and QLIKE for forecast evaluation. Here, we test if changing the objective function to minimising QLIKE will change the results and conclusions. According to Patton (2011), MSE and QLIKE are members of a family of robust and homogeneous loss functions below:

$$L(RV, \widehat{RV}; b) = \begin{cases} \frac{1}{(b+1)(b+2)}(RV^{b+2} - \widehat{RV}^{b+2}) - \frac{1}{b+1}h^{b+1}(RV - \widehat{RV}), & \text{for } b \notin \{-1, -2\} \\ \widehat{RV} - RV + RV(log(\frac{RV}{\widehat{RV}})), & \text{for } b = -1 \\ \frac{RV}{\widehat{RV}} - log(\frac{RV}{\widehat{RV}}) - 1, & \text{for } b = -2, \end{cases} \tag{26}$$

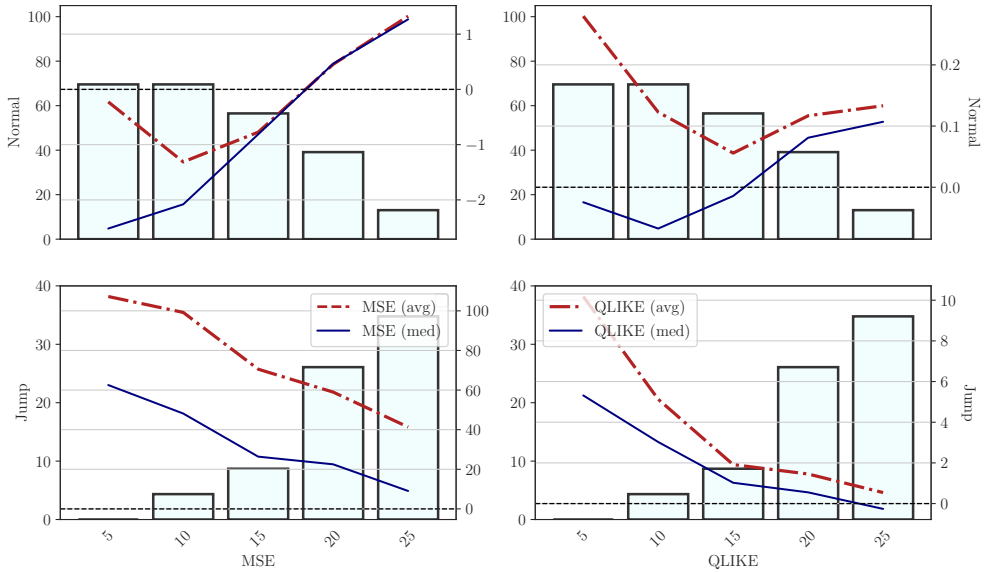
where L is the loss function, RV is the true RV, $\widehat{RV}$ is the fitted (forecasted) RV, and b is the scalar parameter. For $b = 0$, L becomes MSE , and L becomes QLIKE if $b = -2$.

[6]The FCNN is a simplified structure without the LSTM layer. It consists of three layers, viz. input, hidden and output layers. The size of the input layer is the same as the number of input variables. The size of the hidden layer varies from 5, 10, 15, 20, to 25. The size of the output layer is equal to the number of output(s), which is one in this study. The dropout rate between the hidden layer and the output layer is set equal to 0.5. The activation functions are sigmoid and rectifier, respectively, for the hidden and the output layers. The other ML model specifications remained unchanged.

Figure 9: ML models with 21-day information set $\Phi = \{RV_{t-1}, ..., RV_{t-21}, RV_t, RV_t^w, RV_t^m\}$



(a) LSTM with $\Phi = \{RV_{t-1}, ..., RV_{t-21}\}$

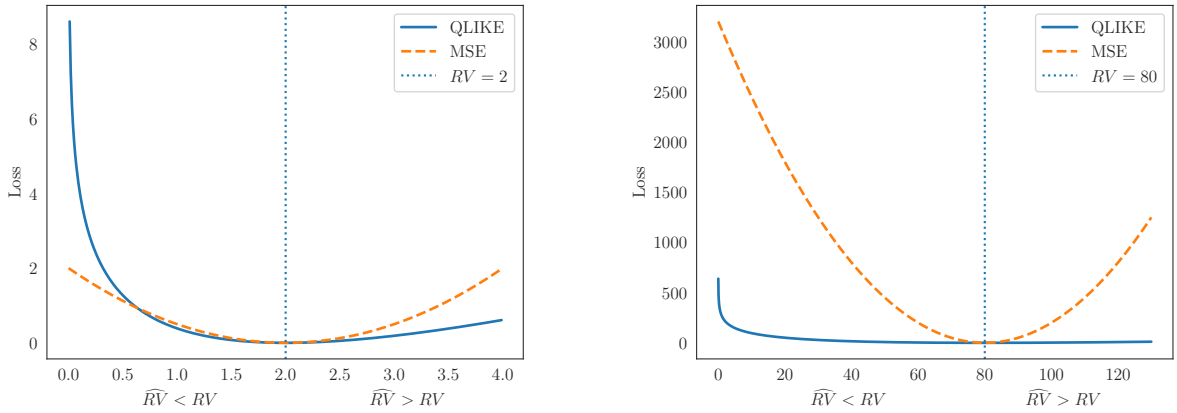


(b) FCNN with $\Phi = \{RV_t, RV_t^w, RV_t^m\}$

Notes: The bar chart is the percentage of tickers with the outstanding performance considering the MSE (left figures) and QLIKE (right figures) loss functions at the 0.05 significance level of the RC compared to the HAR-family of models as the benchmark for every specified number of units of the ML model. The values for the bar chart can be read from the left-hand axis. The dashed (solid) line represents the difference between the average (median) of the out-of-sample MSEs and QLIKEs of the specified ML models and the CHAR model (the best performing HAR-family model in Rahimikia and Poon (2020)) for 23 tickers (negative value shows improvement, and positive value shows degradation in performance). The values for the solid and dashed lines can be read from the right-hand axis. The horizontal dashed line represents no improvement.

Figure 10: A representation of MSE and QLIKE loss functions

(a) Normal Volatility (b) Volatility jumps



Notes: This graph presents the shape of the MSE and QLIKE loss functions. In this example, the true RV (vertical dashed line) is equal to 2 for (a) Normal volatility, or 80 for (b) volatility jumps. To the left of true RV, $\widehat{RV} < RV$, and to the right of the true RV, $\widehat{RV} > RV$.

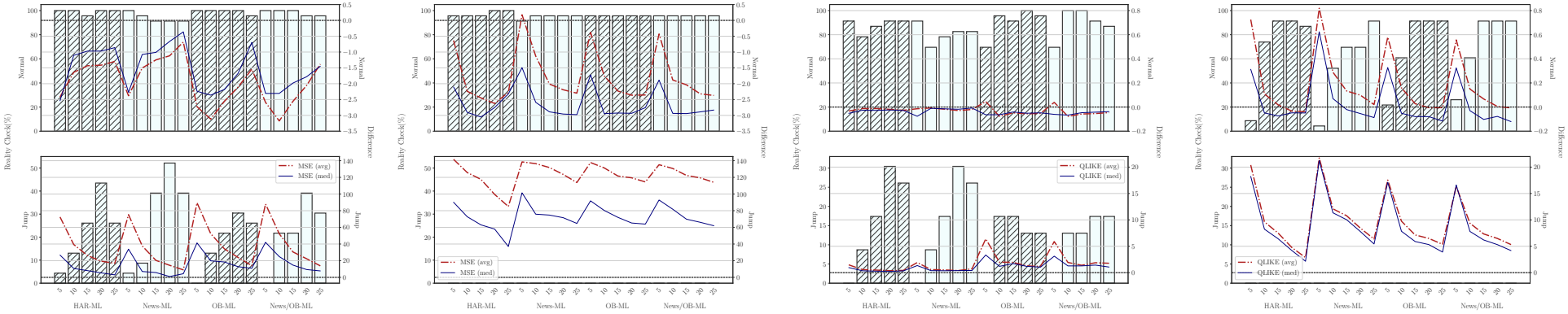
Figure 10 presents the shape of these two loss functions setting the true RV (vertical dashed line) equal to 2 (Figure 10a) and 80 (Figure 10b). To the left (right) of the true RV, $\widehat{RV} < RV$ ($\widehat{RV} > RV$). Comparing to MSE, QLIKE is asymmetry, penalises large under-forecasts more than large over-forecasts. Patton and Sheppard (2009) find, in volatility forecasting, that the power of DMW tests (Diebold and Mariano (1995) and West (1996)) are higher when the loss function is QLIKE than MSE, suggesting that QLIKE might be a better loss function for ranking competing volatility forecasting models.

As a robustness check, we compare MSE vs QLIKE as the objective function in the training period for the ML models in Section 5, while all other model specifications remain unchanged. QLIKE (as in Equation (18) and Equation (26)) is undefined when $\widehat{RV} \leq 0$. To avoid this limitation, we add a lambda layer as the last layer in FCNN, making the $\widehat{RV} \geq 0.01$, i.e. always greater than zero.[7] The results are presented in Figure 11. Figure 11c (Figure 11f) evaluates the forecasts using MSE (QLIKE), while the left (right) figures within each subgraph (Figure 11c or Figure 11f) show the results for the ML models trained using minimising MSE (QLIKE) as the objective function. The bar chart is the RC percentage of tickers with outstanding ML performance at the 5% significance level against all the HAR-family of models. The values for the bar chart can be read from the left-hand axis. The red dashed (blue solid) line represents the average (median) $\Delta_{MSE/QLIKE}$ for each of the four ML models (HAR-ML, News-ML, OB-ML, and News/OB-ML) against CHAR (the best performing HAR-family model

[7]Another way of making sure $\widehat{RV} > 0$ is to change the scalar parameter, b, very close to -2 like -1.99 or -2.01 in Equation (26). In order to keep consistency with the other experiments, this method is not utilised in this study.

Figure 11: Minimising MSE vs QLIKE as the Objective Function in Training Period

(a) MSE loss function (b) QLIKE loss function (d) MSE loss function (e) QLIKE loss function



(c) Forecasts Evaluated Under MSE (f) Forecasts Evaluated Under QLIKE

Notes: For each of the two subgraphs, the left (right) figure presents the results for the four ML models (HAR-ML, News-ML, OB-ML, and News/OB-ML) trained using minimising MSE (QLIKE) as the objective function. The other model specifications are the same as in Section 5. The bar chart is the RC percentage of tickers with outstanding ML performance at the 5% significance level against all the HAR-family of models. The values for the bar chart can be read from the left-hand axis. The red dashed (blue solid) line represents the average (median) $\Delta_{MSE/QLIKE}$ for each of the four ML models (HAR-ML, News-ML, OB-ML, and News/OB-ML) against CHAR (the best performing HAR-family model in Rahimikia and Poon (2020)) for 23 tickers; a negative value indicates improvement, and a positive value indicates performance degradation). The values for the dashed and solid lines can be read from the right-hand axis. The horizontal dashed line represents no difference in performance between ML and CHAR.

in Rahimikia and Poon (2020)) for 23 tickers; a negative value indicates improvement, and a positive value indicates performance degradation). The values for the dashed and solid lines can be read from the right-hand axis. The horizontal dashed line represents no difference in performance between ML and CHAR.

As before, there are huge gains from ML models, improving volatility forecasts on normal days but a performance degradation on volatility jump days. But, the choice of MSE or QLIKE as the objective function leads to different ML model choices. First, Figure 11c shows that when MSE is the objective function, the average (red dashed line) Δ_{MSE} is more negative than the median (solid blue line) Δ_{MSE} for all four ML models, noting that a negative Δ_{MSE} indicates improvement. The reverse is true for QLIKE in Figure 11f. In other words, MSE as an objective function leads to improvements for most tickers, but a few tickers have big under-forecasts. In contrast, QLIKE as an objective function avoided big under-forecasts, but the average improvement is less than that from MSE. This finding corresponds to the property of QLIKE in avoiding large under-forecasts and the weighting pattern in Figure 10.

Next, Figure 11c also shows MSE as the objective function leads to ML models with a smaller number of units while QLIKE is the objective function leads to ML models with a larger number of units. While QLIKE as an objective function makes the ML models focus more on large RV, it fails to outperform CHAR and HAR-family of models on volatility jump days. The mean and median $\Delta_{MSE/QLIKE}$ are all positive, and the RC values are zero; ML was dominated by all HAR-family of models for forecasting RV on volatility jump days. Overall, it highlights the fact that minimising QLIKE is a bad choice as the objective function.

So far, all models under-forecast RV on volatility jump days for all 23 tickers. Among the HAR-family of models, AR, HAR, JHAR, CHAR, SHAR, ARQ, HARQ, and HARQ-F under-forecast, respectively, 88.03%, 88.71%, 86.28%, 87.59%, 88.87%, 84.79%, 85.99%, and 84.14% of all volatility jump days in the out-of-sample period. The News/OB-ML models with 5, 10, 15, 20, and 25 units and with minimising MSE as the objective function under-forecast, respectively, 99.71%, 99.20%, 97.35%, 92.74%, and 92.39% of volatility jump days in the out-of-sample period. What remains unclear, however, is why when minimising QLIKE as the objective function, the ML models under-perform those ML models with minimising MSE as the objective function, when QLIKE is designed to avoid large under-forecasts? The clue lies in the QLIKE weighting function when true RV is very high. Returning to Figure 10 where the MSE and QLIKE loss functions are presented for true $RV = 2$ on the left, and true $RV = 80$ on the right. It is evident that when the true RV is very high, the weights of QLIKE becomes very flat on both sides of the true RV; only when $\widehat{RV} << RV$, the weight begins to rise. This means that when true RV is very high, QLIKE becomes insensitive to the size of the

forecast errors (except for extremely big under-forecasts), and hence lose the ability to learn to forecast accurately. This evidence suggests that QLIKE, without further modification, is not appropriate as an objective function for training ML models. Hence, in this study, only minimising MSE is used as the objective function in training.

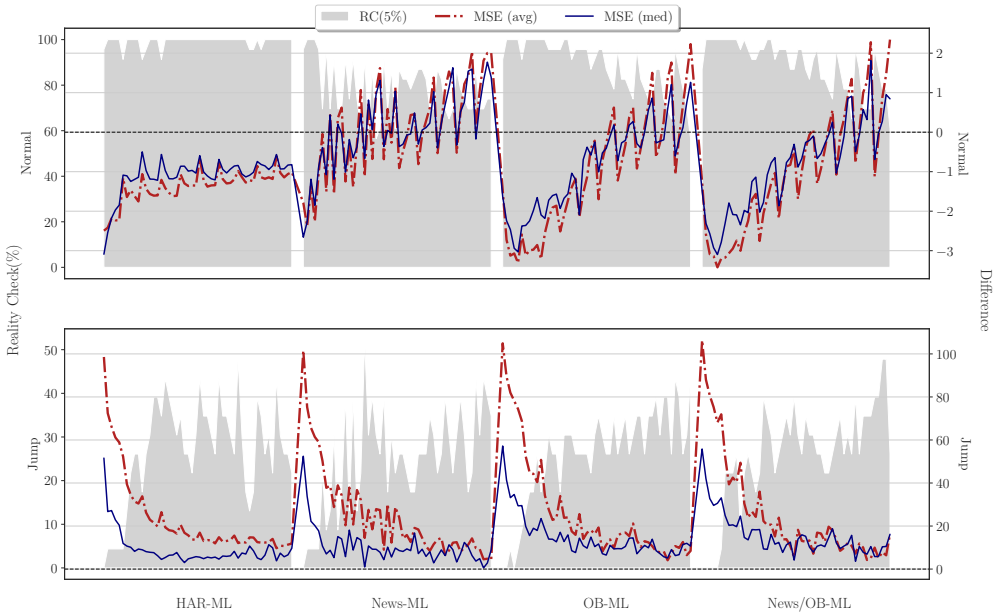
6.4 No of Units vs No of Epochs

This section tests the sensitivity of the number of units and number of epochs in affecting the forecasting performance of ML models on normal volatility days as well as on volatility jump days. etWe test the number of units ranging from 5, 10, 15, 20, 25, 30, 35, 40, 45, to 50, and for the number of epochs, from 25, 50, 75, 100, to 125. The other specifications are the same as those in Section 4. For 23 tickers, four groups of information sets, and 300 days in the out-of-sample period, 5,768,400 ($209 \times 23 \times 4 \times 300$) ML models are trained and tested which is substantially higher than the 138,000 (5 choices of units $\times 23 \times 4 \times 300$) ML models trained in Section 5.

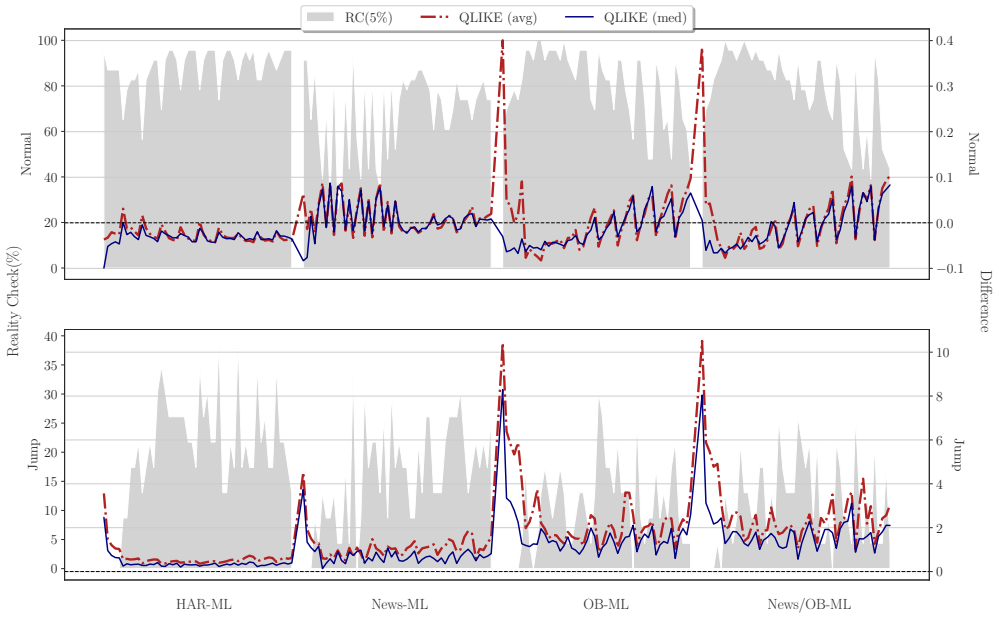
The results are presented in Figure 12a for MSE evaluation function, and Figure 12b for QLILE evaluation function, for the four ML models (i.e. from left to right, HAR-ML, News-ML, OB-ML, and News/OB-ML). For each ML tested, the results are shown in the following order from left to right (number of units-number of epochs): 5-25, 5-50, 5-75, 5-100, 5-125, 10-25, 10-50, 10-75, 10-100, 10-125, 15-25, 15-50, 15-75, 15-100, 15-125, 20-25, 20-50, 20-75, 20-100, 20-125, 25-25, 5-50, 25-75, 25-100, 25-125, 30-25, 30-50, 30-75, 30-100, 30-125, 35-25, 35-50, 35-75, 35-100, 35-125, 40-25, 40-50, 40-75, 40-100, 40-125, 45-25, 45-50, 45-75, 45-100, 45-125, 50-25, 50-50, 50-75, 50-100, and 50-125. For the sake of clarity, these values are not shown in these figures. Also, the grey area is the RC result at the 5% level showing the percentage of tickers with the outstanding ML performance against each and every HAR-family of models. The red dashed (blue solid) line represents the average (median) $\Delta_{MSE/QLIKE}$ between ML and CHAR (the best performing HAR-family model in Rahimikia and Poon (2020)) for 23 tickers and in the out-of-sample period. A negative value indicates improvement, and a positive value indicates degradation in performance.

First, consider the case of MSE and QLIKE as the evaluation functions in Figure 12a and Figure 12b. As before, after a substantial number of combinations of units/epochs test, the results confirm that, for normal volatility days, generally a smaller number of units produced the best performance in terms of average (median) $\Delta_{MSE/QLIKE}$ and the RC values. Second, the number of units has a greater influence on performance than the number of epochs; after fixing the number of units, increasing the number of epochs improves the ML forecasting performance. Third, ML outperformed CHAR and HAR-family of models only on normal volatility days;

Figure 12: No. of Units vs No. of Epochs



(a) Forecasts Evaluated Under MSE



(b) Forecasts Evaluated Under QLIKE

Notes: From left to right, this figure consists of the HAR-ML, News-ML, OB-ML, and News/OB-ML variable groups. For every group, the results are shown in the following order from left to right (number of units-number of epochs): 5-25, 5-50, 5-75, 5-100, 5-125, 10-25, 10-50, 10-75, 10-100, 10-125, 15-25, 15-50, 15-75, 15-100, 15-125, 20-25, 20-50, 20-75, 20-100, 20-125, 25-25, 5-50, 25-75, 25-100, 25-125, 30-25, 30-50, 30-75, 30-100, 30-125, 35-25, 35-50, 35-75, 35-100, 35-125, 40-25, 40-50, 40-75, 40-100, 40-125, 45-25, 45-50, 45-75, 45-100, 45-125, 50-25, 50-50, 50-75, 50-100, and 50-125. For the sake of clarity, these values are not shown in this figure. The grey area is the percentage of tickers with the outstanding performance considering the MSE loss function in the top part (QLIKE at the bottom) at the 0.05 significance level of the RC compared to the HAR-family of models as the benchmark for every specified ML model. The values for this area can be read from the left-hand axis. The red dashed (blue solid) line represents the average (median) $\Delta_{MSE/QLIKE}$ between ML and CHAR (the best performing HAR-family model in Rahimikia and Poon (2020)) for 23 tickers and in the out-of-sample period. A negative value indicates improvement, and a positive value indicates degradation in performance. The values for the dashed and solid lines can be read from the right-hand axis. The horizontal dashed line represents no improvement.

when switching to the volatility jump days, ML models under-performed with positive average (median) $\Delta_{MSE/QLIKE}$, and lower than 50% RC values; increasing the number of units/epochs reduce the amount of under-performance on volatility jump days. Forth, the results for QLIKE as the evaluation function show the same, but less marked, patterns.[8] Finally, of the four ML models tested, OB-ML is outstanding. The News/OB-ML model only slightly improved the performance of OB-ML when all the News variables are added to the combined information set.

Overall, the results provide strong evidence that extended ML models exhibit strong volatility forecasting performance for normal volatility days, but not for volatility jump days. As there is an important trade-off between normal and jump day performance, more care is needed for forecasting jump day volatility; a simpler HAR-family model is preferred. If an ML model is to be used for forecasting volatility jump, a more complex ML model has to be trained specifically for this purpose.

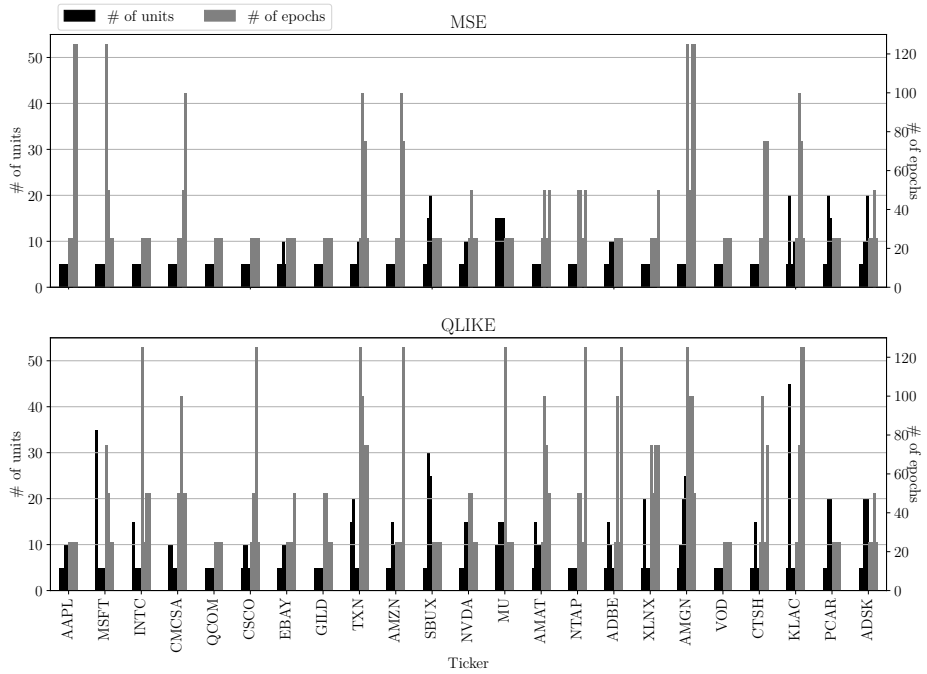
6.5 Units vs Epochs: Individual Stock Analysis

This subsection concern the best combination of the number of units and epochs for the best volatility forecasting performance for each ticker. The results are presented in Figure 13a and Figure 13b for normal volatility days and volatility jumps respectively. For each ticker, there are two sets of four bars correspond to the four ML models, viz. from left to right, HAR-ML, News-ML, OB-ML, and News/OB-ML. The first four black bars correspond to their optimal number of units, while the next four grey bars correspond to their optimal number of epochs. The top (bottom) half is for normal volatility (volatility jump) days. Within each subfigure, the top (bottom) graph reports the results considering the MSE (QLIKE) loss function. The number of units (# of units) can be read from the left axis, and the number of epochs (# of epochs) can be read from the right axis.

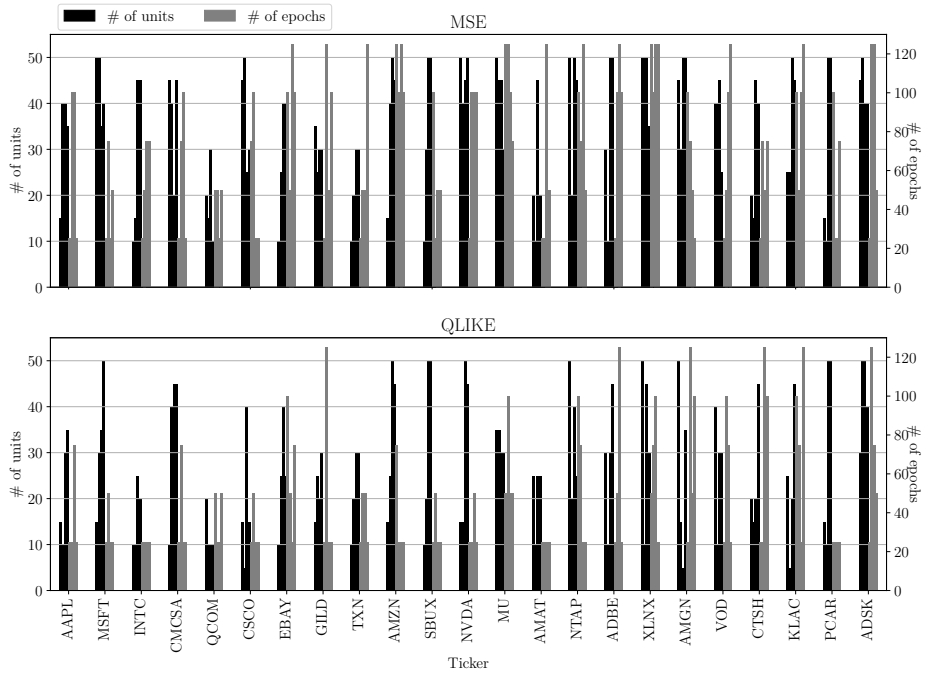
First, the optimal numbers of units and epochs are much larger for volatility jump days than those needed for normal volatility days correspond to the findings in Section 5 and Section 6. Second, for normal volatility (volatility jump) days, the average optimal numbers of units and epochs, are higher (lower) under QLIKE than under MSE loss function. The average number of units for the MSE and QLIKE loss functions for normal volatility days are 7 and 10; however, these values for volatility jumps are 35 and 27. Also, the average number of epochs for the MSE and QLIKE loss functions of normal volatility days are 40 and 51, and these values for volatility jumps are 73 and 47. Third, there are some variations among the tickers, especially for normal volatility days in Figure 13a. When MSE is the loss function, 'MU' optimal number

[8]The results for the MDA tests in Figure C1 of Appendix C corroborate the findings here.

Figure 13: The best performing hyperparameters (units and epochs) for four extended ML models



(a) Normal Volatility Days



(b) Volatility Jump Days

Notes: For each ticker, there are two sets of four bars that correspond to the four ML models, viz. HAR-ML, News-ML, OB-ML, and News/OB-ML. The first four black bars correspond to their optimal number of units, while the next four grey bars correspond to their optimal number of epochs. The top (bottom) half is for normal volatility (volatility jump) days. Within each subfigure, the top (bottom) graph reports the results considering the MSE (QLIKE) loss function. The number of units (# of units) can be read from the left axis, and the number of epochs (# of epochs) can be read from the right axis.

of units (black bars) are high for all four ML models, correspond to the findings in Rahimikia and Poon (2020) and Subsection 5.1. On the other hand, 'AMGN' has an optimal number of epochs (grey bars) much higher than all other tickers. When QLIKE is the loss function, the variations are across all tickers, with no specific idiosyncratic behaviours. Turning now to results for volatility jump days in Figure 13b; there are great variations in the optimal numbers of units and epochs across tickers, and there is no discernible pattern. Because of this, it seems that for forecasting on volatility jump days, a more accurate hyperparameter tuning is needed for each ticker; there is no one size fits all solution.

The individual ticker analysis gives important insight into the importance of hyperparameter tuning for ML models. In an ideal world of unlimited computing resources and time, one can test all combinations for all models using a variety of optimisation algorithms. But the real world is full of constraints. According to the analysis and findings here, we can infer that if the goal is to train ML models for forecasting volatility on normal days, a fixed model specification such s that in Section 5 works well for a majority of tickers. In contrast, if the goal is to forecast volatility on volatility jump days, the task is more challenging and needs careful tuning of hyperparameters for every group of considered variables; otherwise, relying on universal results and common performance metrics could be misleading.

7 Conclusions

This study investigates the strengths and weaknesses of ML models for RV forecasting. Three types of daily data are used, viz. 5 variables from HAR-family of models, 134 limit order book variables and 9 news sentiment variables, for 23 NASDAQ stocks over the period from June 28, 2007, to November 17, 2016. The stock returns data, and the limit order book data are extracted from *LOBSTER*, while the news sentiment variables are compiled from the *Dow Jones Newswires* based on the latest version of LM dictionary ((Loughran and McDonald, 2011)). The sample period was split into training period (2046 days from June 28, 2007, to November 17, 2016) and out-of-sample forecasting period (300 days from September 11, 2015, to November 17, 2016).

Using an LSTM combined with an FCNN layer and four sets of variables each with 21 lags, 138,000 ML models were trained based on the objective function of minimising MSE. These experiments provide strong evidence for the stronger forecasting power of ML models than all HAR-family of models, including the extended CHAR (CHARx) models in Rahimikia and Poon (2020). The findings remain qualitatively the same when the forecasts are evaluated using MSE, QLIKE, MDA (Mean Directional Accuracy) or the RC (Reality Check) values. However,

this substantial and statistically significant improvement was relevant only for normal volatility days, not for days with volatility jumps.

Throughout this study, we find a persistent trade-off between the forecasting performance on normal volatility days and days with volatility jumps; the best model for forecasting normal day volatility is the worst model for jump day forecast and *vice versa*. For normal day volatility forecasting, an LSTM model with 5 units and augmented with LOB performs well. For volatility jump days, the ML models trained here performed poorly; one should re-train a more complex ML model explicitly designed for jump day forecast, or use a simpler linear HAR model augmented with, e.g. news count (Rahimikia and Poon (2020)). Surprisingly, only a minority of tickers produced the forecasting improvement with ML models over HAR-family of models for volatility jump days in the extended experiment.

The findings and conclusions remain the same after a series of robustness checks. By reducing the length of lagged variables, or by reducing the number of variables to just the history of RV, or by restricting the ML structure to a simple FCNN layer, the ML models continue to dominate CHAR and all HAR-family of models in volatility forecasting on normal days. By considering an extensive range of the number of units and the number for epochs, 5,768,400 ML models were trained and tested. The findings reveal big variations in optimal ML specification (number of units and number of epochs) across stocks for volatility jump days, but more common optimal values for forecasting volatility on normal days. This means that ML models for jump day forecasting are stock-specific and would require laborious training. The alternative is to switch to a simpler HAR-family model with a more straightforward structure. A less noticeable finding is the issue with minimising QLIKE as an objective function. Despite the emphasis on preventing large under-forecast, QLIKE is a poor objective function as the weights become flat at high volatility, making it harder for the optimisation algorithm to reach the true RV.

It is interesting to note that the LOB-ML model outperformed News-ML model here, but Rahimikia and Poon (2020) find better forecasting performance with CHAR augmented with news variables. This is a reminder that results from previous studies based on linear models may change when a more complex ML model is used. The ML model can deal with complex nonlinear connections and high dimensional relationships. Apart from the bigger challenge of volatility jumps, making the analysis idiosyncratic also, the ML model has much potential in improving volatility forecast on normal volatility days. The empirical findings in this study provide a new understanding of the complexities behind developing ML models for financial forecasting. Therefore, this study highlights that even a standard structure of ML model needs a substantial amount of effort to fully understand the behaviour of the ML models, tuning the hyperparameters, and also analysing the information content of different sets of variables. In

summary, this study is the first comprehensive assessment of a well-known and state-of-the-art ML model for RV forecasting and lays the foundation for future ML research in this area. The present study has demonstrated that although ML models show superior performance in many cases for RV forecasting, these findings should be interpreted with caution for other financial forecasting applications regarding the properties of the under investigation financial series. Future research could study other financial series and other types of ML models for financial forecasting.

References

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

Barndorff-Nielsen, O. E., P. R. Hansen, A. Lunde, and N. Shephard (2009). Realized kernels in practice: Trades and quotes.

Barndorff-Nielsen, O. E. and N. Shephard (2001). Non-gaussian ornstein–uhlenbeck-based models and some of their uses in financial economics. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 63*(2), 167–241.

Barndorff-Nielsen, O. E. and N. Shephard (2004). Power and bipower variation with stochastic volatility and jumps. *Journal of financial econometrics 2*(1), 1–37.

Bollerslev, T., A. J. Patton, and R. Quaedvlieg (2016). Exploiting the errors: A simple approach for improved volatility forecasting. *Journal of Econometrics 192*(1), 1–18.

Chen, L., M. Pelger, and J. Zhu (2019). Deep learning in asset pricing. *Available at SSRN 3350138*.

Corsi, F. (2009). A simple approximate long-memory model of realized volatility. *Journal of Financial Econometrics 7*(2), 174–196.

Corsi, F. and R. Reno (2009). Har volatility modelling with heterogeneous leverage and jumps. *Available at SSRN 1316953*.

Goodfellow, I., Y. Bengio, and A. Courville (2016). *Deep learning*. MIT press.

Gu, S., B. Kelly, and D. Xiu (2020). Empirical asset pricing via machine learning. *The Review of Financial Studies 33*(5), 2223–2273.

Hinton, G. E., N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov (2012). Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

Hochreiter, S. and J. Schmidhuber (1997). Long short-term memory. *Neural computation 9*(8), 1735–1780.

Kalay, A., O. Sade, and A. Wohl (2004). Measuring stock illiquidity: An investigation of the demand and supply schedules at the tase. *Journal of Financial Economics 74*(3), 461–486.

Kercheval, A. N. and Y. Zhang (2015). Modelling high-frequency limit order book dynamics with support vector machines. *Quantitative Finance 15*(8), 1315–1329.

Kingma, D. P. and J. Ba (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Li, J. and D. Xiu (2016). Generalized method of integrated moments for high-frequency data. *Econometrica 84*(4), 1613–1633.

Loshchilov, I. and F. Hutter (2017). Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Loughran, T. and B. McDonald (2011). When is a liability not a liability? textual analysis, dictionaries, and 10-ks. *The Journal of Finance 66*(1), 35–65.

Næs, R. and J. A. Skjeltorp (2006). Order book characteristics and the volume–volatility relation: Empirical evidence from a limit order market. *Journal of Financial Markets 9*(4), 408–432.

Patton, A. J. (2011). Volatility forecast comparison using imperfect volatility proxies. *Journal of Econometrics 160*(1), 246–256.

Patton, A. J. and K. Sheppard (2015). Good volatility, bad volatility: Signed jumps and the persistence of volatility. *Review of Economics and Statistics 97*(3), 683–697.

Politis, D. N. and J. P. Romano (1994). The stationary bootstrap. *Journal of the American Statistical association 89*(428), 1303–1313.

Poon, S.-H. and C. W. Granger (2003). Forecasting volatility in financial markets: A review. *Journal of economic literature 41*(2), 478–539.

Rahimikia, E. and S.-H. Poon (2020). Big data approach to realised volatility forecasting using har model augmented with limit order book and news. *Available at SSRN 3684040*.

Sheppard, K. (2009). Mfe matlab function reference financial econometrics. *Unpublished paper, Oxford University, Oxford. Available at: http://www. kevinsheppard. com/images/9/95/MFE_Toolbox_Documentation. pdf*.

Srivastava, N., G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research 15*(1), 1929–1958.

White, H. (2000). A reality check for data snooping. *Econometrica 68*(5), 1097–1126.

A ML models and extended CHARx models comparison (MDA)

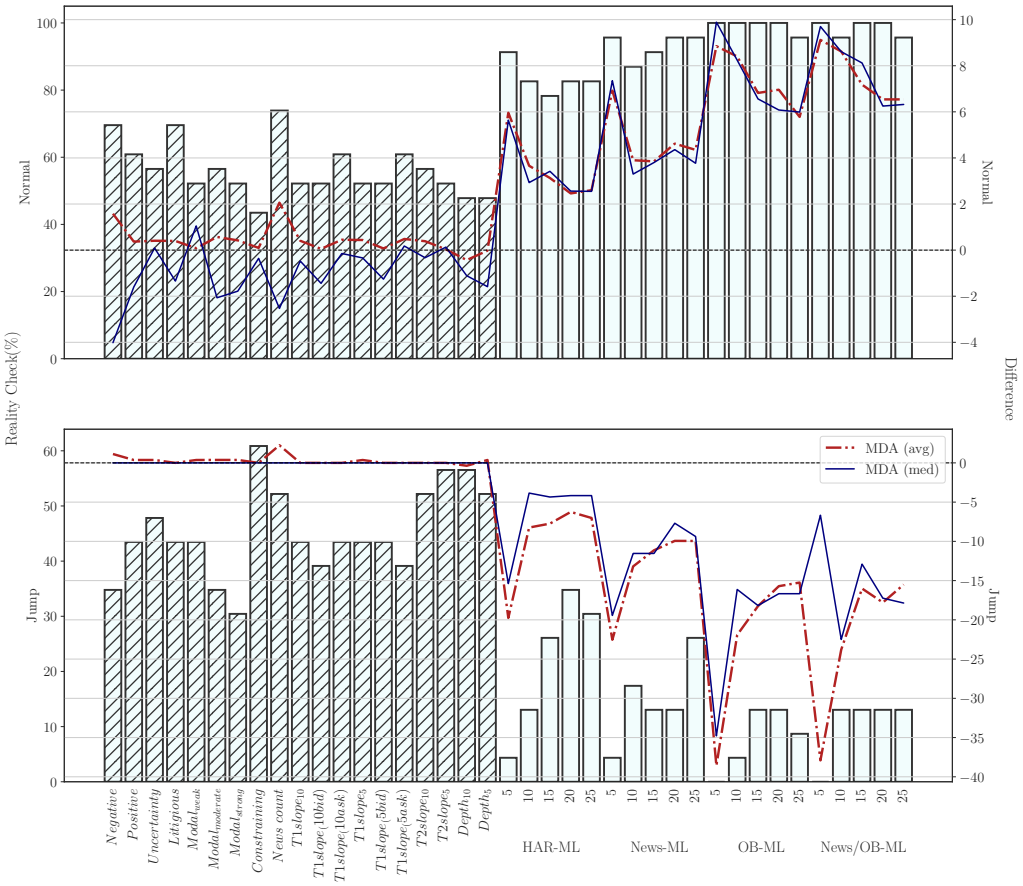
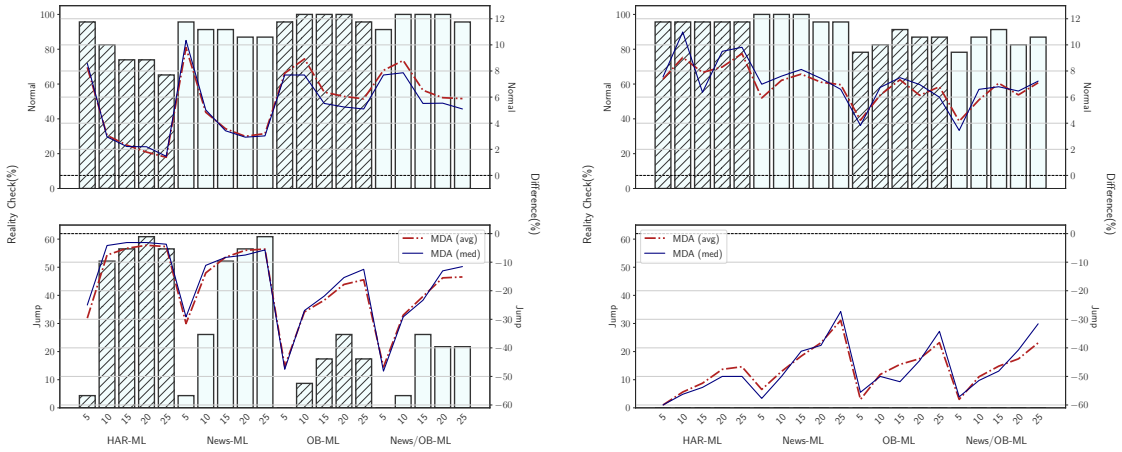


Figure A1: ML models and extended CHARx models comparison (MDA)

Notes: The bar chart is the percentage of tickers with the outstanding performance considering the MDA loss function at the 0.05 significance level of the RC compared to the all HAR-family of models as the benchmark for every specified extended CHAR model (hatched bars) and the ML model (white bars). The values for the bar chart can be read from the left-hand axis. The dashed (solid) line represents the difference between the average (median) of the out-of-sample MDAs of the specified model with the CHAR model (the best performing HAR-family model in Rahimikia and Poon (2020)) for 23 tickers (positive value shows improvement, and negative value shows deterioration of performance). The values for the dashed and solid lines can be read from the right-hand axis. The horizontal dashed line represents no improvement.

B MDA results for robustness checks



(a) Five lags (b) One lag

Figure B1: ML Models With Restricted Information Set $\Phi_t = \{t-1, \ ..., t-5\}$ (MDA)

Notes: The left and right figures display the results considering five lags (last week) and one lag (last day). The bar chart is the percentage of tickers with the outstanding performance considering the MDA loss function at the 0.05 significance level of the RC compared to the HAR-family of models as the benchmark for every HAR-ML, News-ML, OB-ML, and News/OB-ML model. The values for the bar chart can be read from the left-hand axis. The dashed (solid) line represents the difference between the average (median) of the out-of-sample MDAs of the HAR-ML, News-ML, OB-ML, and News/OB-ML models and the CHAR model (as the best performing HAR-family model in Rahimikia and Poon (2020)) for 23 tickers (positive value shows improvement, and negative value shows deterioration of performance). The values for the dashed and solid lines can be read from the right-hand axis. The horizontal dashed line represents no improvement.

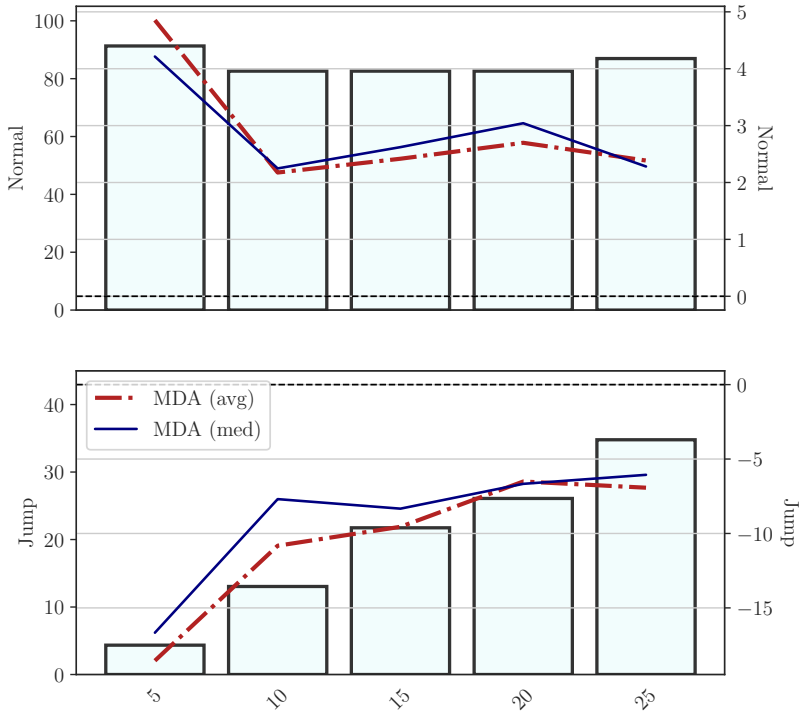


Figure B2: LSTM with $\Phi = \{RV_{t-1}, ..., RV_{t-21}\}$ (MDA)

Notes: The bar chart is the percentage of tickers with the outstanding performance considering the MDA loss function at the 0.05 significance level of the RC compared to the HAR-family of models as the benchmark for every specified number of units of the ML model. The values for the bar chart can be read from the left-hand axis. The dashed (solid) line represents the difference between the average (median) of the out-of-sample MDAs of the specified ML models and the CHAR model (as the best performing HAR-family model in Rahimikia and Poon (2020)) for 23 tickers (positive value shows improvement, and negative value shows degradation in performance). The values for the solid and dashed lines can be read from the right-hand axis. The horizontal dashed line represents no improvement.

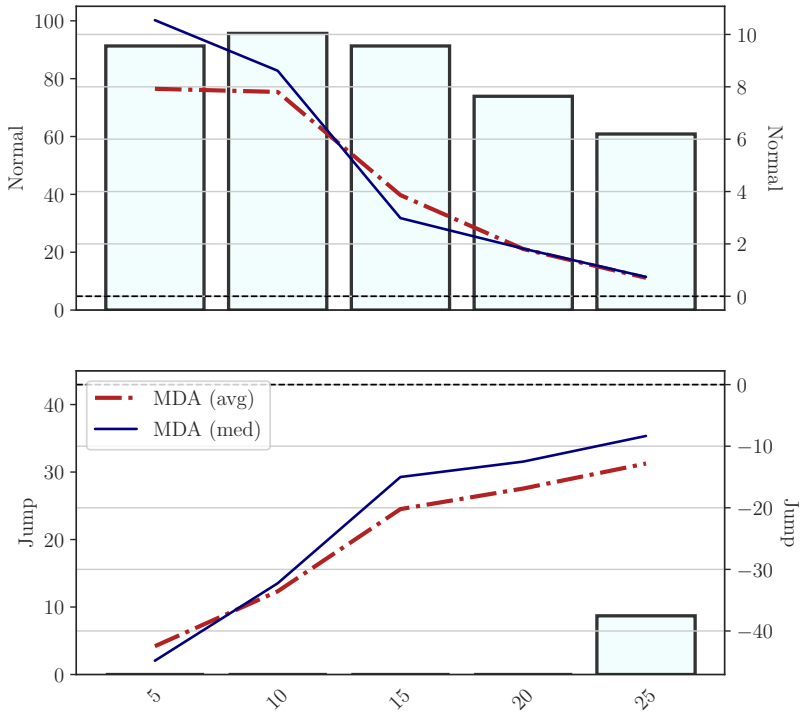


Figure B3: FCNN with $\Phi = \{RV_t, RV_t^w, RV_t^m\}$ (MDA)

Notes: The bar chart is the percentage of tickers with the outstanding performance considering the MDA loss function at the 0.05 significance level of the RC compared to the HAR-family of models as the benchmark for every specified number of units of the ML model. The values for the bar chart can be read from the left-hand axis. The dashed (solid) line represents the difference between the average (median) of the out-of-sample MDAs of the specified ML models and the CHAR model (as the best performing HAR-family model in Rahimikia and Poon (2020)) for 23 tickers (positive value shows improvement, and negative value shows deterioration of performance). The values for the solid and dashed lines can be read from the right-hand axis. The horizontal dashed line represents no improvement.

C Complementary results for extended ML models

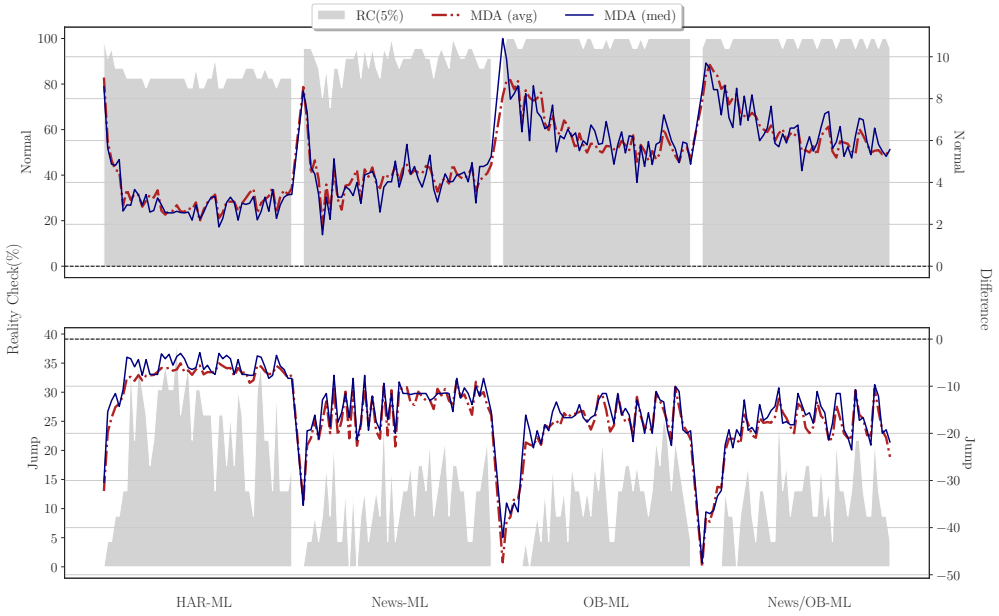


Figure C1: No. of Units vs No. of Epochs (MDA)

Notes: From left to right, this figure consists of the HAR-ML, News-ML, OB-ML, and News/OB-ML variable groups. For every group, the results are shown in the following order from left to right (number of units-number of epochs): 5-25, 5-50, 5-75, 5-100, 5-125, 10-25, 10-50, 10-75, 10-100, 10-125, 15-25, 15-50, 15-75, 15-100, 15-125, 20-25, 20-50, 20-75, 20-100, 20-125, 25-25, 5-50, 25-75, 25-100, 25-125, 30-25, 30-50, 30-75, 30-100, 30-125, 35-25, 35-50, 35-75, 35-100, 35-125, 40-25, 40-50, 40-75, 40-100, 40-125, 45-25, 45-50, 45-75, 45-100, 45-125, 50-25, 50-50, 50-75, 50-100, and 50-125. For the sake of clarity, these values are not shown in this figure. The grey area is the percentage of tickers with the outstanding performance considering the MDA loss function at the 0.05 significance level of the RC compared to HAR-family of models as the benchmark for every specified ML model. The values for this area can be read from the left-hand axis. The dashed (solid) line represents the difference between the average (median) of the out-of-sample MDAs of the specified ML model with the CHAR model (the best performing HAR-family model in Rahimikia and Poon (2020)) for 23 tickers (positive value shows improvement, and negative value shows deterioration of performance). The values for the dashed and solid lines can be read from the right-hand axis. The horizontal dashed line represents no improvement.