Forecasting and Trading High Frequency Volatility on Large Indices

Fei Liu[1,2], Athanasios A. Pantelous[1,2] and Hans-Jörg von Mettenheim[3*]

[1]Department of Mathematical Sciences, University of Liverpool, Liverpool United Kingdom

[2]Institute for Risk and Uncertainty, University of Liverpool, Liverpool United Kingdom

[3]Institut für Wirtschaftsinformatik, Leibniz Universität Hannover, Königsworther Platz 1, D-30167, Hannover Germany

Abstract

The present paper analyzes the forecastability and tradability of volatility on the large S&P500 index and the liquid SPY ETF, VIX index and VXX ETN. Even though there is already a huge array of literature on forecasting high frequency volatility, most publications only evaluate the forecast in terms of statistical errors. In practice, this kind of analysis is only a minor indication of the actual economic significance of the forecast that has been developed. For this reason, in our approach, we also include a test of our forecast through trading an appropriate volatility derivative. As a method we use parametric and artificial intelligence models. We also combine these models in order to achieve a hybrid forecast. We report that the results of all three model types are of similar quality. However, we observe that artificial intelligence models are able to achieve these results with a shorter input time frame and the errors are uniformly lower comparing with the parametric one. Similarly, the chosen models do not appear to differ much while the analysis of trading efficiency is performed. Finally, we notice that Sharpe ratios tend to improve for the longer forecast horizon.

JEL classification: C53; G1; C4

Keywords: Forecasting; Realized Volatility; High-Frequency Data; HAR-RV-J; RNN; Hybrid Model; Trading efficiency.

1 Introduction

Volatility plays central roles in asset pricing and allocation, and in risk management, e.g. value-at-risk and expected shortfall. Modeling and forecasting volatility is important for econometricians, statisticians and practitioners, and for that reason it has gained much interest in the financial and economic literature, however the application of traditional *Generalized AutoRegressive Conditional Heteroskedasticity* (GARCH) and *Stochastic Volatility* (SV) models are not appropriately suited for applications where *high frequency* data has been used. In response

*Corresponding author. Email: mettenheim@iwi.uni-hannover.de

to the increasing availability of those financial data, Andersen and Bollerslev (1998) proposed that the daily volatility, which is normally treated as a latent variable in various parametric models, now can be approximated using intraday data, and their new measure was called *Realized Volatility* (RV).

Undoubtedly, high frequency data contains more information of the daily transaction, and are useful not only in measuring volatility, but also in direct model estimation and forecast evaluation, therefore, by increasing the sampling frequency, the RV is considered as a very good proxy of the true volatility under the assumption of no market microstructure noise; see for details and further discussion in Zhou (1996); Andersen and Bollerslev (1998); Andreou and Ghysels (2002); Zhang et al. (2005) among others. However, a higher frequency leads inevitably to a larger microstructure noise, thus Hansen and Lunde (2006) suggested a 5-min sampling frequency which is now commonly used to compute RV in order to trade off the bias-variance problem.

Many recent studies focusing on high frequency data evaluate the performance of various models of RV. The parametric model termed *Heterogeneous AutoRegressive model of Realized Volatility* (HAR-RV) is the commonly used for prediction of RV, which has a simple structure and also considers the long memory property. Additionally, an alternative model which allowing *jumps* or *discontinuities* in the estimation of RV is proposed by Andersen et al. (2007a), which referred as HAR-RV-J in this paper. Their empirical studies show that incorporating the jumps to the HAR model increase the accuracy of forecasting performance.[1]

The HAR families have been developed to capture certain features of volatility, however, the errors in prediction by using the parametric models are often argued by researchers. This is because the linear models are often based on certain distribution assumptions and the microstructure noise can arise by bid-ask bounce, asynchronous trading, and price discreteness, Barunik and Krehlik (2016). *Artificial Neural Network* (ANN) models offer a potential improvement to earlier approaches because ANNs have the ability to tolerate data with errors and also find nonlinear associations between the parameters of the model, Haykin (2007).

The present paper compares the HAR-RV-J with a Recurrent Neural Network (RNN) and the *hybrid* HAR-RV-J-RNN model to forecast volatility, thereby analyzing the forecastability. The application of machine learning is increasing in the volatility literature, however the studies also on hybrid models which incorporate parametric model and neural network by using high frequency data are limited, therefore our study is also contributed on this gap. What is more, most of the published papers evaluate the volatility forecasting performance of certain models by using traditional statistical accuracy criteria, e.g. *mean square error*, *mean absolute error* and *mean absolute percentage error*. However, the practitioners select appropriate models base on financial rather than statistical criteria. Therefore, the intuitively appealing idea of this paper is to investigate the power of forecasting models from both a statistical and economic point of view. In order to do so, we apply a realistic volatility trading strategy by using the first volatility futures ETNs to be issued were Barclays iPath S&P 500 VIX short-term (Ticker: VXX), launched at the beginning of 2009, which tracks the performance of S&P VIX short-term futures index. To the best of our knowledge, this is the first attempt to apply this trading strategy based on the new approach of volatility forecasting for high frequency data. Furthermore, we emphasize the aspect of building a robust model. Such a model should

[1]For determining whether an asset return process has jumps by using high frequency data, see Carr and Wu (2003); Barndorff-Nielsen and Shephard (2006); Andersen et al. (2003); Huang and Tauchen (2005); Andersen et al. (2007a,b); Fan and Wang (2007); Lee and Mykland (2008); Jiang and Oomen (2008); Aït-Sahalia and Jacod (2009); Andersen et al. (2012); Lee and Hannig (2010); Christensen et al. (2014); Bajgrowicz et al. (2016).

exhibit minimal variation in quality when presented with varying meta-parameters. These include, for example, the lookback or forecast interval or different model variants. Finally, we strive at making the model as data efficient as possible to reduce the need for potentially costly historical intraday data.

The paper is organised as follows. The Section 2 gives a brief literature on volatility modelling, forecasting and trading. The Section 3 discusses the volatility forecasting models including HAR-RV-J, RNN, and hybrid models. The Section 4 presents dataset employed in the empirical study. The Section 5 compares the estimation and forecasting results of the models and also introduces the volatility trading strategies. Moreover, it provides detailed trading results and the discussion of their applications. Section 6 concludes the whole discussion.

2 Literature Research

The development of volatility research has had at least three notable stages. The first stage is GARCH model which was proposed by Bollerslev (1986), see also Bollerslev et al. (1994) and Engle and Patton (2001). The second stage is the so called SV model which contributed to the contemporaneous development in Bayesian statistical analysis using the Markov Chain Monte Carlo procedure, see Taylor (1986) and Harvey et al. (1994), also the recent work of Lux and Moreles-Arias (2013). The third stage was followed by the work of Andersen and Bollerslev (1998) and Barndorff-Nielsen and Shephard (2001), who proposed the use of the sum of the squared intradaily returns at different sampling frequencies as a proxy measure for the corresponding daily volatility. This measure provides a consistent estimator of the latent volatility under an ideal market condition. Barndorff-Nielsen and Shephard (2002a), Andersen et al. (2003) among others, have established some theoretical foundations for RV construction via high frequency data.

Since GARCH-type models were constructed to describe daily data, in high-frequency data environment, they are not suitable to solve this problem. Hansen et al. (2012) considered the so called realized GARCH (RGARCH) model by introducing a measure function to link the latent variances to realized volatility. HAR model became widely used to forecast realized volatility because this model easily capture the long memory property in contrast to RGARCH model. Andersen et al. (2007a) built on the theoretical results of realized variation measures constructed from high frequency returns by involving the so called bipower variation measures. Their study pointed out that volatility jump component is essential and significant jumps were associated with specific macroeconomic news announcements; see also the recent work of Borovkova and Mahakena (2015). In this context, Corsi (2009) found that HAR-RV model is able to reproduce the same volatility persistence observed in the empirical data as well as many from the other main stylized facts of financial data, in spite of its simplicity and the fact that it does not formally belong to the class of long-memory models.

Following Andersen's and his co-authors' works, Celik and Ergin (2014) found that the heterogeneous autoregressive model allowing for discontinuities was the best among high frequency based on the volatility forecasting models. They use Turkey index futures data and proved the superiority of high frequency data based volatility forecasting model over traditional GARCH model. Moreover, Papavassiliou (2016) confirmed the significance of discontinuous jumps in forecasting volatility by studying individual stocks and demonstrated the importance of using high frequency data in model-free, non-parametric financial econometric procedures. For detailed analysis of dynamics of jumps can also be found in Fan and Wang (2007); Lee and Hannig (2010); Lee (2012); Prokopczuk et al. (2015); Borovkova and Mahakena (2015);

Boudt and Zhang (2015); Sevi (2014); Bajgrowicz et al. (2016), among others.

Linear models, which are based on restrictive distribution assumptions, have been developed to capture certain properties of volatility, however, changes in market conditions and many mircostructure noise lead to complex patterns which cannot be captured. Other tools used in the study of return volatility are ANNs. The application of ANNs to modelling economic conditions has been expanding rapidly the last decades, see for instance Dunis and Huang (2002); Bildirici and Ersin (2009); Hajizadeh et al. (2012); Kotkatvuori-Ornberg (2016). Recent studies on stock markets price forecast using ANNs can also be found by Jammazi and Aloui (2012); Panella et al. (2012); Papadimitriou et al. (2014), among others. Kristjanpoller and Minutolo (2015) applied a hybrid ANN-GARCH model to forecast the gold price volatility and concluded that the overall forecasting performance was improved as compared to a GARCH method alone. However, their study focused on daily returns for forecasting the daily volatility, and used daily squared returns which are calculated from closing prices and therefore cannot capture price fluctuations during day. In high-frequency data context, Barunik and Krehlik (2016) proposed an ANN approach that incorporates realized measures with generalised regression to capture the complex patterns hidden in linear models, and evaluated multiple-step-ahead volatility forecasts of energy markets using several popular high frequency measures and forecasting models, concluding that this newly proposed methodology yields both statistical and economic gains.

However, it seems that in the literature most papers evaluate forecasting performance by using traditional statistical accuracy criteria, seldom has applied the forecasting results to volatility products trading. Since the financial crisis exchange-traded products have been developed rapidly and become more popular among investors. Carr and Lee (2009) provided an extensive literature on volatility derivatives. Zhang et al. (2010) explored the relationship between the VIX index and VIX futures and showed that the VIX and VIX futures are high correlated by establishing a mean-reverting variance model. The study of Fassas and Siriopoulos (2012) showed that VIX futures prices can be used as an efficient and unbiased estimator for the spot VIX. More recently, Alexander et al. (2015) overviewed the recent developments in the volatility exchange-traded products that are related to implied volatility.

To summarise, the ANN models continue to provide more accurate forecasting performance, nonetheless, there are still room for improving upon the existing models. In the following sections, the specific methodology is presented and we use empirical data to test our model.

3 Methodology

3.1 The HAR-RV-J Model

We consider an n-dimensional price process defined on a complete probability space, $(\Omega, \mathcal{F}, P)$, evolving in continuous time over the interval $[0,\ T]$, where T denotes a positive integer. Following closely the setup of Andersen et al. (2003, 2007a)'s work, let p_t denote a logarithmic asset price at time t, and incorporating also the theoretical framework of Back (1991), the continuous-time semimartingale jump diffusion process used in asset pricing is as follows:

$$dp(t) = \mu(t)dt + \sigma(t)dW(t) + \kappa(t)dq(t), \quad 0 \leq t \leq T, \tag{1}$$

where $\mu(t)$ is a continuous and locally bounded variation process, $\sigma(t)$ is a positive and cadlag stochastic volatility process, $W(t)$ is a standard Brownian motion, $q(t)$ a counting process with $dq(t) = 1$ corresponding to a jump at time t and $dq(t) = 0$ otherwise with jump intensity $\lambda(t)$,

and $\kappa(t)$ refers the size of the corresponding discrete jumps in the logarithmic price process. The quadratic variation for the cumulative return process, $r(t) = p(t) - p(0)$, is given by:

$$[r, r]_t = \int_0^t \sigma^2(s)ds + \sum_{0<s\leq t} \kappa^2(s). \tag{2}$$

In the absence of jumps, the quadratic variation $[r, r]_t$ is equal to integrated volatility $\int_0^t \sigma^2(s)ds$, see Andersen and Bollerslev (1998); Andersen et al. (2001, 2003, 2006); Barndorff-Nielsen and Shephard (2001, 2002a,b).

Let denote the sampled $\delta-$period returns $r_{t,\delta} = p(t) - p(t-\delta)$, then define the daily RV by summing the corresponding $1/\delta$ high frequency intradaily squared returns:

$$RV_{t+1}(\delta) = \sum_{j=1}^{1/\delta} r^2_{t+j*\delta,\delta}. \tag{3}$$

By the theory of quadratic variation, see Back (1991); Andersen et al. (2003), the realized variation converges uniformly in probability to the increment of the quadratic variation process as the sampling frequency of the underlying returns go to infinity, that is

$$RV_{t+1}(\delta) \sim \int_t^{t+1} \sigma^2(s)ds + \sum_{t<s\leq t+1} \kappa^2(s). \tag{4}$$

Thus, in the absence of jumps the realized variation is consistent for the integrated volatility. However, in order to separate the continuous variation and jump components, Barndorff-Nielsen and Shephard (2004) proposed the *Bipower Variation* (BV), which is defined as follows:

$$BV_{t+1}(\delta) = \frac{2}{\pi} \sum_{j=2}^{1/\delta} |r_{t+j*\delta,\delta}||r_{t+(j-1)*\delta,\delta}|. \tag{5}$$

As $\delta \sim 0$, it is possible to see that:

$$BV_{t+1}(\delta) \sim \int_t^{t+1} \sigma_s^2 ds. \tag{6}$$

Combining the results in Eqs. (4) and (5), the contribution to the quadratic variation process due to jumps in the underlying process can be estimated by:

$$RV_{t+1}(\delta) - BV_{t+1}(\delta) \sim \sum_{t<s\leq t+1} \kappa^2(s). \tag{7}$$

To prevent the right hand-side of Eq. (7) from becoming negative, Andersen et al. (2007a) imposed non-negativity truncation on the jump measurements:

$$J_{t+1}(\delta) = max[RV_{t+1}(\delta) - BV_{t+1}(\delta), 0]. \tag{8}$$

HAR-RV model is introduced by Corsi (2009), and it can be expressed as:

$$RV_{t+1} = \beta_0 + \beta_D RV_t + \beta_W RV_{t-5,t} + \beta_M RV_{t-22,t} + \varepsilon_{t+1}, \tag{9}$$

$t = 1, 2, \ldots, T$. RV_t, RV_{t-5} and RV_{t-22} mark daily, weekly (5 business days) and monthly (22 business days) RV, respectively. Weekly and monthly RV is calculated as: $RV_{t,t+h} =$

$h^{-1}[RV_{t+1} + RV_{t+2} + ... + RV_{t+h}]$, $h = 1, 2, \ldots$. Andersen et al. (2007a) proposed the new HAR-RV-J model, in which included the jump components. Daily HAR-RV-J model is expressed as:

$$RV_{t,t+1} = \beta_0 + \beta_D RV_t + \beta_W RV_{t-5,t} + \beta_M RV_{t-22,t} + \beta_J J_t + \varepsilon_{t,t+1}. \quad (10)$$

Logarithmic and standard deviation form of HAR-RV-J model is given by:

$$(RV_{t,t+1})^{1/2} = \beta_0 + \beta_D (RV_t)^{1/2} + \beta_W (RV_{t-5,t})^{1/2} + \beta_M (RV_{t-22,t})^{1/2} + \beta_J (J_t)^{1/2} + \varepsilon_{t,t+1}, \quad (11)$$

and

$$log(RV_{t,t+1}) = \beta_0 + \beta_D log(RV_t) + \beta_W log(RV_{t-5,t}) + \beta_M log(RV_{t-22,t}) + \beta_J log(J_t + 1) + \varepsilon_{t,t+1}. \quad (12)$$

3.2 Recurrent Neural Networks

ANNs are a powerful non-parametric tool used for signal filtering, recognition of patterns and interpolation, also, can tolerate data with errors and find nonlinear associations between the parameters of the model, see Haykin (2007); Kristjanpoller et al. (2014); Kristjanpoller and Minutolo (2015). In particular, as we discussed it in Section 2, ANNs have been applied with increasing success to economic and financial forecasting. Most of econometric models are developed by capturing specific features of time-series, e.g. long memory, or making an assumption of functional relationship among variables, the major advantages of ANNs is that they contain nonlinearities and incorporate all variables.

Briefly speaking, see also Haykin (2007) among other classical books, each neural network connects a group of *input* variables $\{x_1, x_2, \ldots\}$ with one or more *output* variables $\{y_1, y_2, \ldots\}$ and zero, one or more *hidden* layers. Neurons are connected between the layers for connections that are activated by reaching a threshold. Each layers can have a different number of neurons. A series of weight vectors $\{w_{i,j}, w_{2,j}, \ldots, w_{n,j}\}$ is associated with the input vectors, each node may additionally have also a bias input θ_j, thus the actual outputs of the neurons in the hidden layer is:

$$y_i = sigmoid[\sum_{i=1}^{n} x_i * w_{i,j} - \theta_j],$$

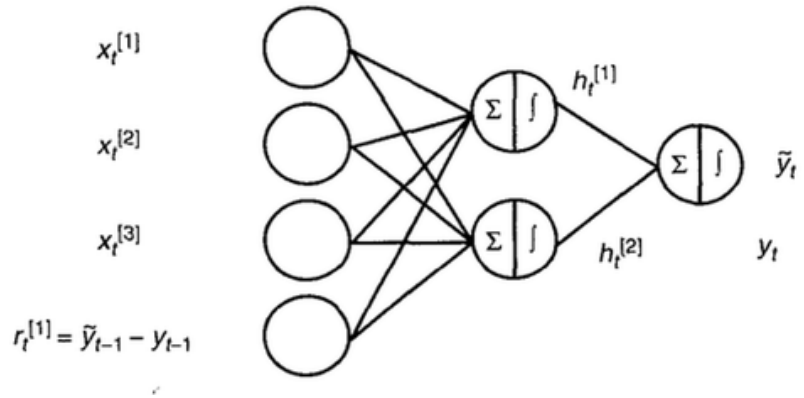
and *sigmoid* is the sigmoid activation function $f(x) = sigmoid(x) = \frac{1}{1+e^{-x}}$.

In this paper, we use *Recurrent Neural Network* (RNN) models which were introduced by Elman (1990), see also an application of RNNs in currency trading by Dunis and Huang (2002). Their only difference from multilayer neural network is that they include a loop back from one layer, either the output or the intermediate layer or the input layer. Figure 1 shows a single output RNN model with one hidden layer and two hidden nodes.

When using ANN, generally, we have to carefully consider the topic of data pre-processing. Indeed, for most ANN the output is limited due to the squashing function being either the hyperbolic tangent or the logistic function. These two sigmoid squashing functions as mentioned above are the most commonly used. While the hyperbolic tangent leads to an output in the interval $]-1, +1[$, the logistic function has an output in the interval $]0, +1[$. In our empirical application, we choose the latter one, the logistic function that is better adapted to the output domain of the numbers we analyze.

However, the generated data is typically relatively small compared to unity. The typical order of magnitude is between 10^{-5} and 10^{-4}, see Table 1. In this case, a simple linear transformation is advised that makes better use of the available output range of the logistic

Figure 1: Single output Recurrent Neural Network (RNN) model with one hidden layer



function. To achieve this, we simply multiply all data with 10^3 for the linear HAR-RV-J model inputs. As we also train ANN on the logarithmic deviation of the HAR-RV-J model a slightly different approach is needed. Indeed, the logarithmic transformation will generate data in the range $[-14, -5]$, see Table 1. This also has to be transformed. An appropriate scaling factor is given by 5×10^{-2} with a shift of $+1$. This simple linear transformation does not change the basic interpretation of the data but makes it easier to be learned by an ANN.

The inputs to our RNN model are the same as to the linear model to allow a fair comparison. That is, we include the three RV and one jump inputs. Also, the question is to the meta-parameters of the neural network that have to be addressed, specifically, the size of the hidden layer and the exact architecture. Determining the number of hidden neurons is often left to experiment, and no single dominant method has emerged. However, taking twice the geometric mean of the input and output layer size is an often used heuristic. Therefore in all our RNN models we determine the number of hidden neurons h in the following way:

$$h = 2 \times \sqrt{i \times o}, \tag{13}$$

where i and o refer to the size of the input and output layer, respectively. Note, that in the case of the input layer we do not take the bias neuron into account.

Upto this point the RNN looks very similar to a standard three-layer perception. However, storing the output in a separate state layer and feeding this back into the hidden layer makes the network state aware. Therefore, when evaluating the network, we have to be careful to store the present state and to carry out any evaluation in sequential order of time. As our rolling-window approach includes just one forecast per model, there is not much potential for confusion in our specific application. Each model is trained on the rolling-window and then immediately used for the corresponding forecast horizon. However, if the network has to be reused for several forecasts, then this issue has to be considered. For example, if network training times were much longer, it might make sense not to train a new model every day. In this latter case it is compulsory to store the network state for later reuse.

The training of ANN is a topic that has been discussed intensely. The architecture of a neural network makes it computationally efficient and straightforward to compute the partial derivatives of the error with respect to the network's weights. Therefore, algorithms that make use of partial derivatives can be used. This includes, for example, simple gradient descent and its variations. However, computing the Hesse matrix of second order partial derivatives is much less straightforward and computationally intensive. Therefore, pure Newton methods are generally avoided. However, quasi-Newton methods can be used instead. In the case of the

training of an ANN our goal is twofold. On the one hand, of course, we want to decrease the error. On the other hand, however, we also strive at achieving a robust model. Therefore, the weight set that leads to the absolute smallest error is not necessarily the one to be preferred, if it is just a lone minimum in a steep valley. Rather, we would prefer minima where the neighboring values also lead to decent results. In the present case, we train our networks using resilient backpropagation, where we only consider the sign of the first partial derivative. This, generally, leads to a robust result.

3.3 Hybrid Model

The hybrid model is also designed as an RNN. However, as an additional input we feed the *forecast* of the linear model to the RNN. We also keep our four basic inputs. Thus, the total number of inputs rises to five in the case of the hybrid model.

All other model parameters are kept the same. Specifically, the number of hidden neurons is determined as above. Also, the model architecture stays identical.

The motivation of using a hybrid model stems from the desire to use each model in a way that exploits its specific abilities. By feeding the linear forecast to the RNN we potentially remove any linear component from the forecasting task. This should leave more room for better matching the non-linear residual of the linear forecast error.

4 Data

Our base dataset consists of tick data from Thomson Reuters Tick History (TRTH) for the S&P500 index that starts on the January, 2nd, 1996 upto June, 2nd, 2016. For the reason we mentioned in Section 1, see Hansen and Lunde (2006), initially we aggregate it to 5 minutes data. Thus, intraday RV is computed based on these 5 minutes blocks. To filter out any half-holidays, we require that a trading day has to have a complete history of data from 10am to 3.55pm in order to be included in our computation. The regular trading hours for our index are 9.30am to 4pm. However, as is commonly done, we exclude the first half hour of trading and the very last five minute interval of the regular trading session to avoid any bias that may be caused by the price determination process at the beginning of the trading day or by any rebalancing trades towards the end of the session.

Of course, it is necessary to strike a balance between unwanted noise and making best use of the intraday data. The specific cut-off times are debatable. However, our preliminary experiments showed that the above procedure produced sensible results.

Our tick data series from TRTH for the SPY ETF, the VIX index and the VXX ETN start on March, 20th, 1996, January, 2nd, 1996, and January, 30th, 2009, respectively. Among these, only the VXX series starts much later, simply, because the corresponding ETN was introduced only in 2009.

5 Empirical Results

Table 1 summarizes the distributional properties of RV and jump series. It is evident that the realized volatility and jump are highly significant serial correlation. This can be confirmed by the Ljung and Box (1978) statistics for up to tenth-order serial correlation. Variables have kurtosis greater than 3 indicating leptokurtic distribution, the distribution of logarithmic transformation of RV are closer to normal than RV and standard deviation form of RV. This

finding is consistent with the study of Andersen et al. (2007a). Figure 2 provides a visual illustration of RV and jumps for S&P500. Also consistent with earlier evidence of Andersen et al. (2007a), many of the largest realized volatility are directly associated jumps in the underlying price process. The largest jump occurred around 2008 when global financial crisis broke out. In the following we present statistical analysis of various models, we also expect our modeling to yield insight into the actual tradeability of volatility with readily available products.

All computations are rather straightforward and can be carried out in acceptable time on a laptop. Our machine is a Lenovo Thinkpad W530 with an Intel Quad Core i7-3720QM CPU, running at 3.6GHz, with 6MB Level 3 cache, and 1600MHz FSB. Due to the generally fast computation no care was taken to parallelize the computation. Therefore, in the end, only one core was used for all computations. As we compute daily updated rolling-window forecasts a single run through the dataset produces 4448 single models to cover a timespan of approximately 18 years. Two years have to be removed to account for the maximum lookback. For each of the 4448 models three variants have to be computed for the linear model, RNN, and hybrid model. Both again are computed using the basic variant and the *log* variant. Each model finally is computed for five different lookbacks and three different forecast horizons. In total this leads to $4448 \times 3 \times 2 \times 5 \times 3 = 400320$ different models that are computed. Or, put differently, for each of our three model types 133440 models are computed. This may sound like a lot, however all our models are comparatively small by today's standards and can therefore be computed quickly. The linear model has a closed form solution and just requires linear algebra. The RNN has to be trained numerically. The computational core of an ANN is, however, also linear algebra with just a very small amount of computation time dedicated to computing the non-linear squashing function. In all cases, the initial data-preprocessing time can be neglected, as it only has to be carried out once per variant.

In total the computation times are 492 seconds for the linear models, 22217 seconds for the RNN and 56582 seconds for the hybrid model. Adding up the numbers leads to a total computation time of a bit more than 22 hours, for the *entire* model ranges. The computation could even have been sped-up by parallelizing it, as this problem is ideally coarsely parallelizable without interdependencies which would have slowed down the computation. However, for simplicity, this was not done presently. Looking at the numbers, this model seems very suitable for real-time applications. Indeed, for computing a single decision (model) we only need to carry out the computation once. Dividing the above numbers by the number of models per variant we achieve computation times of 3.7 ms per model in the linear case, 166.5 ms per model for the RNN, and, finally, 424.0 ms per model for the hybrid case.

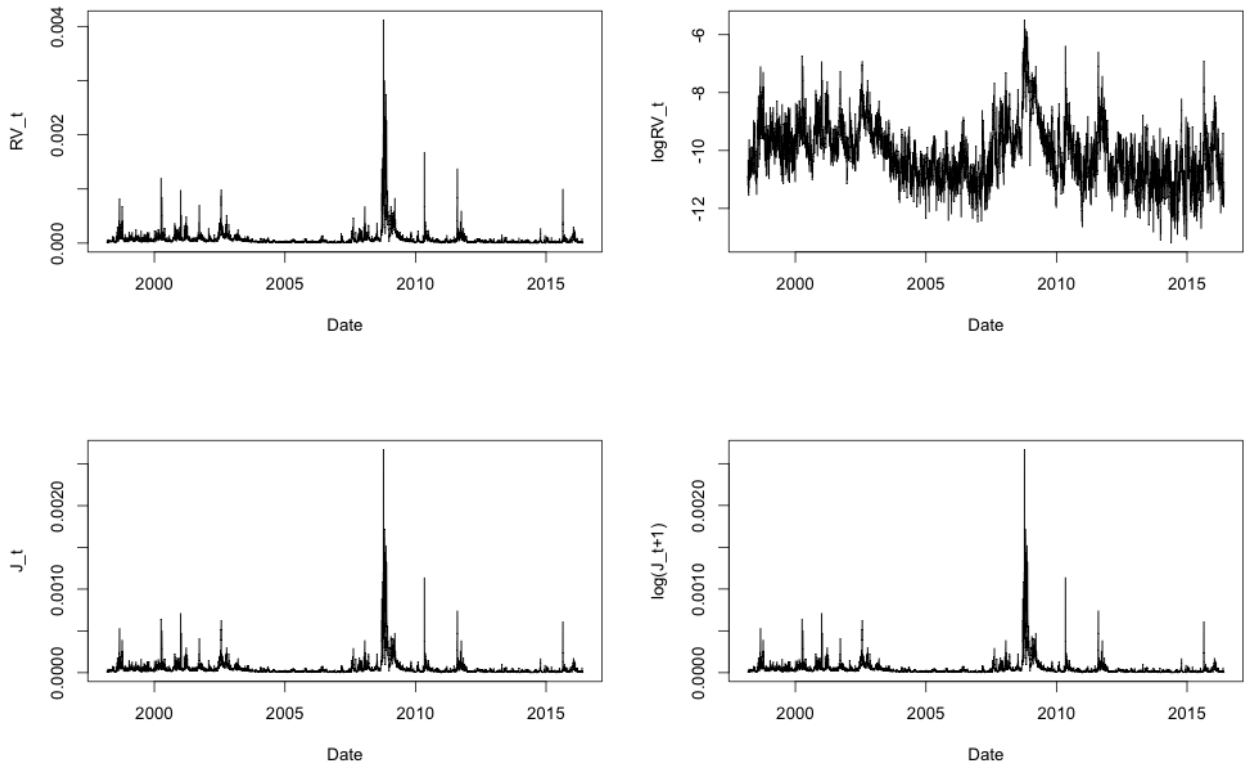
5.1 Statistical Errors

As a first indication with respect to the quality and robustness of the different HAR model implementations, we present and discuss the usual statistical errors like *Root Mean Squared Error* (RMSE), *Mean Absolute Error* (MAE), and *Mean Absolute Percent Error* (MAPE) for the different types of models computed, Hyndman and Athanasopoulos (2014).[2]

Each group of columns in the following Tables 2 and 3 represents a forecast with the given number of days (lookahead) described by the letter l. The first group of columns therefore stands for a 1-day forecast, the second column group for a 2-day forecast and the third column group for a 1-week (i.e., 5 business days) forecast. Indeed, while the original HAR model limits

[2]In Hyndman and Koehler (2006) (and references therein), there is an interesting discussion and comparison among different measures of accuracy of univariate time series forecasts.

Figure 2: Daily S&P500 Realized Volatility and Jumps



Note: The top panel shows daily realized volatility and its logarithmic transformation, RV_t and $log(RV_t)$, respectively. The lower panel graphs the jump components, J_t and $log(J_t + 1)$, respectively.

Table 1: Summary statistics for S&P500 index at 5-minute frequency

	RV_t	$RV_t^{1/2}$	$log(RV_t)$	J_t	$J_t^{1/2}$	$log(J_t + 1)$
Mean	0.7863	0.0074	-10.1053	0.4898	0.0059	0.4897
St.dev	0.0002	0.0049	1.0367	0.0001	0.0038	0.0001
Skewness	10.0472	3.3765	0.4676	10.2250	3.3436	10.2159
Kurtosis	153.3177	22.9232	3.5630	162.9341	22.7987	162.6297
Min	0.0186	0.0014	-13.1961	0.0113	0.0011	0.0113
Max	0.0041	0.0642	-5.4919	0.0027	0.0517	0.0027
LB_{10}	15188	22354	22391	14782	22086	14790

Note: The table summarizes the distributional properties of the daily realized volatility. $RV_t, RV_t^{1/2}$ and $log(RV_t)$ denote the daily realized volatility, realized standard deviation and realized logarithmic form respectively. $J_t, J_t^{1/2}$ and $log(J_t+1)$ denote the daily jump measures invariance, standard deviation, and logarithmic forms respectively. LB is the statistics of Ljung and Box (1978) Q test for up to tenth-order correlation. The mean and minimum values of RV, J, and $log(J + 1)$ are multiplied by 10^4.

the analysis to a 1-day forecast, there is no specific reason to assume, that this model type would be less suitable for forecasts several time steps into the future. For this reason we believe it is of interest to analyze model robustness for different forecast horizons which would allow to identify model types which might be suitable for use in different setups.

Each group of rows in the Tables 2 and 3, furthermore, represents a different amount of historical data used in the rolling window. Therefore, the first group of rows just makes use of the past 22 business days (around 1 month) of data, while the last group of rows includes 504 business days (around 2 years) of in-sample data for computing the out-of-sample forecast.

Within the groups of rows, we present the different model types one separate rows. The first row in the group shows the basic linear (or logarithmic) variant, while the second row presents the RNN version (using the same inputs). Finally the third row in each group shows the results of the hybrid model. To recall, the hybrid model is also an RNN that uses the previously mentioned four inputs, and, additionally, the linear model forecast as a fifth input. The two tables, finally, present grouped results for the basic linear version of the HAR-RV-J approach and for the logarithmic version.

As the different forecast horizons solve different forecasting problems, it makes sense to analyze and discuss the results depending on the forecast horizon. For the HAR base version and a 1-day ahead forecast, we notice that the RNN with just 22 days of in-sample data consistently produces the best results, when looking at the fit (out-of-sample RMSE). This is remarkable, because this best result is achieved with the least amount of data. The linear model manages to come close to the RNN results, but only when using 2 years (504 business) days of in-sample data. It appears, therefore, that the RNN is able to better extract information from a relatively small amount of data, than the linear model. This was rather expected as the RNN has the additional advantage of an implicit storage of state data, while the linear model only explicitly stores state as given by the different (three) lags of RV. Now, this may seem like a trade-off between a more complex model using less data and a simpler model using more data. However, for practical applications, we may be limited in the amount of intraday data available. Let alone that not everyone has an easy access to large historical database of intraday data. In practice, most paid-for data services might only offer a 3 months historical intraday database at an affordable price. Or conversely, the user may be tempted to collect tick data. In all these cases, the RNN provides a practical and robust solution, that, at the worst, takes 1-month of data collection of ramp-up time. On the other hand, having to potentially collect 2 years' worth of data before being able to use the linear model does not seem very practical.

When we further analyze the two time steps ahead forecast, we notice, again, that the single best fit is achieved by an RNN variant for a rolling-window length of 6 months (126 business days). The forecast quality for the short input interval of 1 month is also not bad, but is, indeed, just ever so slightly beaten by the linear model fit for a 2 years rolling-window length. Nevertheless, we cannot fail to notice that in all cases the RNN performs more robustly and more consistently.

Finally, for the 5-day ahead forecast, we again have the single best fit provided by the RNN with a rolling-window length of 22 days. The performance of the linear model for short rolling-window lengths is abysmal. However, when adding more data the linear model performance improves. It is almost on-par with the RNN results when considering the longest input intervals.

If we consider other error measures like MAE and MAPE we also notice, that the RNN solution consistently produces excellent results, even for the shortest rolling-window length. However, for MAE and MAPE the single best solution may in some cases also be produced by the linear model with the longest rolling-window. However, we notice that in all cases the RNN

solution has a robustly low error, while the linear model produces erratic results which does not inspire confidence in its robustness for the shorter intervals. Therefore, even when considering MAE and MAPE there is a good deal to be said in favor of the RNN model, because the result is among the best achievable results for all rolling-window lengths but only requires a small amount of training data.

We also notice, that the RNN model keeps its generally good forecasting performance, even for longer forecast horizons. Indeed, we expect the effect that forecasts for longer horizons are less good than forecasts for shorter horizons. But, the linear model worsens dramatically, while the RNN model does not change much. As a general recommendation, it could be argued that, if a suitable implementation is available, the RNN seems like a good choice for practical applications, because it produces excellent (often even best) results, for the shortest rolling window length. If, however, plenty of data is available, and/or a linear model is preferred for whatever reasons, then good results can also be expected in the linear model case, but a longer rolling-window interval should be used. As in Barunik and Krehlik (2016) for the energy market volatility, the hybrid model's quality is often between that of the RNN and linear version. This may seem disappointing at first sight. Indeed, the initial expectation would have been, that the RNN is able to make better use of the linear forecast. In theory, it is expected that the hybrid model to be at least as good as the basic RNN model. In the worst case, it can be argued that the hybrid RNN appears to ignore all the linear forecasts by setting the corresponding weights to zero in the training process. However, in practical applications, this is not quite so clear cut as the training process depends on many factors. As we see here, the linear forecast does not help in improving basic RNN forecast, although the performances are pretty close, see Table 2. Therefore, there does not seem to be any compelling reason for using a hybrid model in this specific case.

When using the log variant of the HAR model, Table 3, the results are less clear cut. Generally, we may find that the RNN still produces very good results, but they are on-par with the results of the linear model. Still, the feature remains, that for all forecast horizons and all error measures the RNN is the model that makes best use of a short dataset. With more data the linear model improves and, even beats the RNN for very long rolling windows, but the results are pretty close. Therefore, if data is scarce, the RNN model still provides good and robust results. However, when a very long dataset is available, then the linear model should be preferred.

5.2 Trading Efficiency

While modeling and forecasting RV is an interesting and instructive exercise in itself, the question arises, to which extent the results are useful in financial applications. Or, put in other words, can the models uncover an exploitable market inefficiency? For this reason it is useful to develop ideas as to how a forecast of RV could actually be traded on the financial markets. Two general strategies come to mind:

- An outright volatility trade.
- Trade the expected reaction of another asset with respect to volatility.

Since the advent of VIX futures in 2004 on the most important volatility index, a simple way for outright volatility trading is available. The VXX ETN that mimics a 30-day constant maturity future makes trading volatility even easier and puts it in reach of retail investors. The VXX ETN started trading in 2009. Our basic volatility trading strategy consists of very simple elements:

Table 2: Comparison of Models I

	RMSE	MAE	MAPE		RMSE	MAE	MAPE		RMSE	MAE	MAPE
$t = 22, l = 1$				$t = 22, l = 2$				$t = 22, l = 5$			
HAR-RV-J	0.1963	0.0523	100.8305		0.2847	0.0705	152.0512		0.6694	0.1153	205.5808
RNN	**0.1192**	**0.0398**	**53.9350**		0.1329	0.0451	58.1954		**0.1329**	**0.0479**	**64.7888**
Hybrid	**0.1303**	**0.0483**	**55.9974**		0.1571	0.0568	61.1080		0.1564	0.0594	65.9130
$t = 63, l = 1$				$t = 63, l = 2$				$t = 63, l = 5$			
HAR-RV-J	0.1563	0.0448	64.8351		0.1712	0.0539	88.7721		0.2544	0.0714	99.4946
RNN	0.1214	0.0433	60.5210		0.1288	0.0476	63.9215		0.1367	0.0507	69.0026
Hybrid	0.1479	0.0646	63.4334		**0.1401**	**0.0573**	**65.3191**		**0.1381**	**0.0551**	**70.8009**
$t = 126, l = 1$				$t = 126, l = 2$				$t = 126, l = 5$			
HAR-RV-J	0.1363	0.0415	48.7744		0.1502	0.0485	58.2050		0.1933	0.0601	75.6028
RNN	0.1228	0.0445	64.1628		**0.1273**	**0.0479**	**66.7512**		0.1417	0.0532	71.6463
Hybrid	0.1515	0.0701	67.4231		0.1448	0.0635	70.1290		0.1482	0.0605	73.1933
$t = 252, l = 1$				$t = 252, l = 2$				$t = 252, l = 5$			
HAR-RV-J	0.1239	0.0349	46.9789		0.1410	0.0455	52.2581		0.1651	0.0529	88.1153
RNN	0.1238	0.0471	66.9135		0.1350	0.0501	70.5627		0.1421	0.0541	73.2343
Hybrid	0.1448	0.0700	70.6262		0.1700	0.0680	72.9267		0.1548	0.0656	74.5489
$t = 504, l = 1$				$t = 504, l = 2$				$t = 504, l = 5$			
HAR-RV-J	**0.1194**	**0.0382**	**45.7595**		**0.1316**	**0.0431**	**50.8488**		**0.1522**	**0.0497**	**61.2953**
RNN	0.1309	0.0508	72.4723		0.1364	0.0534	75.5360		0.1488	0.0572	80.2657
Hybrid	0.1438	0.0727	75.4535		0.1552	0.0747	76.3257		0.1635	0.0745	80.9149

Note: The table presents the comparison of forecasting performance of HAR-RV-J, RNN and hybrid models in which the daily realized volatility, RV_t, is used. t indicates the rolling window length, e.g. $t = 22$ denotes the length of rolling window is 1 month. $l = 1, 2, 5$ indicates one step forecast, two steps forecast and five steps forecast, respectively. The best results considering RMSE, MAE and MAPE for those models for the different lengths of rolling windows and steps of forecast are highlighted in bold.

Table 3: Comparison of Models: II

	RMSE	MAE	MAPE		RMSE	MAE	MAPE		RMSE	MAE	MAPE
$t = 22, l = 1$				$t = 22, l = 2$				$t = 22, l = 5$			
HAR-RV-J LOG	0.0593	0.0281	5.6179		0.0633	0.0331	6.8792		0.1012	0.0456	9.2539
RNN LOG	**0.0325**	**0.0253**	**5.1552**		**0.0338**	**0.0263**	**5.3496**		**0.0362**	**0.0281**	**5.7234**
Hybrid LOG	**0.0365**	**0.0287**	**5.8164**		**0.0370**	**0.0289**	**5.8884**		**0.0398**	**0.0312**	**6.3415**
$t = 63, l = 1$				$t = 63, l = 2$				$t = 63, l = 5$			
HAR-RV-J LOG	0.0345	0.0228	4.6061		0.0356	0.0261	5.2888		0.0537	0.0321	6.5048
RNN LOG	0.0381	0.0297	6.0166		0.0387	0.0302	6.1148		0.0400	0.0311	6.3093
Hybrid LOG	0.0418	0.0330	6.6341		0.0417	0.0327	6.5987		0.0423	0.0332	6.7356
$t = 126, l = 1$				$t = 126, l = 2$				$t = 126, l = 5$			
HAR-RV-J LOG	0.0329	0.0246	4.4305		0.0328	0.0246	5.0191		0.0390	0.0284	5.7771
RNN LOG	0.0415	0.0323	6.5449		0.0419	0.0326	6.5984		0.0430	0.0334	6.7578
Hybrid LOG	0.0446	0.0352	7.0505		0.0444	0.0349	7.0295		0.0454	0.0354	7.1593
$t = 252, l = 1$				$t = 252, l = 2$				$t = 252, l = 5$			
HAR-RV-J LOG	0.0276	0.0213	4.3524		0.0310	0.0239	4.8576		**0.0254**	**0.0269**	**5.4719**
RNN LOG	0.0457	0.0355	7.1384		0.0460	0.0358	7.1962		0.0466	0.0361	7.2589
Hybrid LOG	0.0490	0.0384	7.5772		0.0484	0.0380	7.5772		0.0494	0.0384	7.7117
$t = 504, l = 1$				$t = 504, l = 2$				$t = 504, l = 5$			
HAR-RV-J LOG	**0.0274**	**0.0212**	**4.3370**		**0.0303**	**0.0235**	**4.8048**		0.0342	0.0264	5.3852
RNN LOG	0.0490	0.0382	7.6719		0.0388	0.0498	0.0388		0.0502	0.0390	7.8337
Hybrid LOG	0.0526	0.0414	8.2160		0.0524	0.0411	8.1993		0.0529	0.0413	8.2777

Note: The table presents the comparison of forecasting performance of HAR-RV-J, RNN and hybrid models in which the logarithmic transformation of daily realized volatility, $log(RV_t)$, is used. t indicates the rolling window length, e.g. $t = 22$ denotes the length of rolling window is 1 month. $l = 1, 2, 5$ indicates 1 step, 2 steps, and 5 steps forecast, respectively. The best results considering RMSE, MAE and MAPE for those models for the different lengths of rolling windows and steps of forecast are highlighted in bold.

- If $RV_{t+k} > RV_t$ then go (and stay) long volatility for k time steps,
- else go (and stay) short volatility for k time steps.

In the above, RV_{t+k} denotes the k time steps ahead RV forecast at time t, while RV_t simply denotes today's RV. In the case $k > 1$, we split our investing capital equally among the number of concurrent trades. In our case, we invest 50% of the capital on two potentially differing trades for 2-day ahead forecasts, and 20% of our capital for 5-day ahead forecasts. Note, that this may also lead to two trades cancelling each other out. That means, if we have a signal for a long volatility trade today (and a 2-day ahead forecast) and a signal for a short volatility trade the next day, then the net position would be zero, because the long and the short trade cancel each other out. In the above strategy, even in the special case that $RV_{t+k} = RV_t$ we go short volatility, because going short volatility is a long term winning strategy (although with mind boggling drawdowns, therefore surely not advised for practical implementation). This is due to the observation that volatility tends to go up in short (but large) spikes, and then continuously fall again.

However, we may want to limit trading in this strategy and only trade when our forecast predicts a significant change in RV. If we assume a threshold p expressed as a percentage range, for example $p = 0.01$ or $p = 1\%$, then we can modify the above strategy to:

- If $RV_{t+k} > RV_t * (1 + p)$ then go (and stay) long volatility for k time steps,
- if $RV_{t+k} < RV_t * (1 - p)$ then go (and stay) short volatility for k time steps,
- else stay out of the market (flat).

Again, investment capital is allocated equally, however still depending on the forecast horizon. When we speak of trading volatility, we have to be careful to note that RV is, in itself, not a tradable quantity. Also, the VIX index itself is not tradable. Strictly speaking, we can only trade volatility derivatives with a defined maturity. The threshold p has to be determined heuristically. However, it makes sense to set it in a way, that, at least, transaction costs are overcome if the forecast proves to be correct.

A second variant involves trading an asset, of which it is expected that it moves in close relation to volatility. Generally, asset returns and volatility are supposed to be negatively correlated. Indeed, in our sample period a quick analysis of the S&P500 Index and the VIX index reveals that in around 80% of the cases, where the S&P500 Index is up, the VIX index is down and the other way round. We would therefore expect that a trading strategy that uses a volatility forecast to trade the S&P500 index ETF would be a sensible approach. Therefore, we propose the following base strategy:

- If $RV_{t+k} > RV_t$ then go (and stay) short the base index ETF for k time steps,
- else go (and stay) long the base index ETF for k time steps.

Here, again, our default case for $RV_{t+k} = RV_t$ is to go long the base index ETF, because, on average, and on a very long time frame, asset prices tend to go up. This argument is, of course, debatable, but it only concerns a very minor edge case.

In a similar way, we may want to implement a threshold in order to avoid overtrading and engaging in trades that potentially do not cover the transaction cost. Analogously this allows us to define the following trading strategy:

Table 4: Out-of-sample rolling window trading results for the Linear Model using a lookback of 252 days

Strategy	Ann. Ret.	Ann. Vol.	Sharpe
$h = 1$			
Index	5.68%	20.49%	0.28
Vol.	55.99%	108.58%	0.52
$h = 2$			
Index	5.18%	13.95%	0.37
Vol.	58.86%	73.54%	0.80
$h = 5$			
Index	5.57%	8.30%	0.67
Vol.	56.98%	42.60%	1.34

- If $RV_{t+k} > RV_t * (1 + p)$ then go (and stay) short the base index ETF for k time steps,
- if $RV_{t+k} < RV_t * (1 - p)$ then go (and stay) long the base index ETF for k time steps,
- else stay out of the market (flat).

The advantage of the direct volatility trade is that the asset traded (a volatility derivative) may correspond more closely, to what is actually forecast. On the other hand, the base index strategy can be implemented, even if there is no volatility derivative available. However, we have to be careful to first carry out an analysis whether the implied correlation between asset returns and changes in volatility is indeed present in the index that we want to trade. This significantly broadens the universe of assets within reach of a volatility forecasting model.

In the above strategies the threshold approach seems appealing to limit trades and whipsaw but introduces a new meta-parameter into the system. To avoid too much interference with the base system, we estimate the threshold that it removes the 10% smallest trades (by absolute forecast difference) in the lookback interval. Of course, it could now be argued that removing the bottom x% trades also represents a meta-parameter and that is true. However, it makes the meta-parameter adaptable and dependent on the actual history.

In the following, we present trading results for the linear and RNN model. We only present results for the lookbacks which seem most suitable according to the statistical evaluation. Table 4 shows results for the linear model with a lookback of 252 days. For this lookback, performance seems to stabilize in the statistical evaluation. Table 5 presents trading results for the RNN model using the short lookback of 22 days in the rolling window forecast.

For each model type (linear and RNN), we present trading results for trading the index and the volatility derivative. As a risk adjusted measure of return, we opt for the Sharpe ratio. We are very well aware of the inherent limitations of the Sharpe ratio, but we still use it, as it is a good basis for comparison. And, despite, all the critics of the Sharpe ratio, mostly only artificially constructed return series may exhibit good Sharpe ratios but otherwise disappointing risk measures. For realistic return series, we may expect good Sharpe ratios to generally lead to also otherwise attractive capital curves.

Table 5: Out-of-sample rolling window trading results for the Recurrent Neural Network (RNN) using a lookback of 22 days

Strategy	Ann. Ret.	Ann. Vol.	Sharpe
$h = 1$			
Index	0.36%	1.75%	0.20
Vol.	13.28%	14.61%	0.95
$h = 2$			
Index	5.79%	13.96%	0.41
Vol.	55.44%	73.56%	0.75
$h = 5$			
Index	5.72%	8.30%	0.69
Vol.	55.69%	42.63%	1.31

As it is typical, the volatility derivative trading strategies show high returns. However, volatility is in itself a very volatile asset class, and the optically high returns are mitigated by correspondingly high volatility of the trade returns. Therefore, we would only advise allocating a small portion of the portfolio to a volatility strategy. It is not suitable as the only trading strategy for any but the most risk-loving portfolios.

As expected the results between the linear and RNN model do not differ much, as the chosen parameter sets have very similar statistical results. While the RNN model had overall better statistical error measures (and, definitely, was able to produce these good results with a modest amount of data) this does not translate to a clearly better trading strategy. For all forecast horizons h, the results are pretty similar when comparing linear and RNN model. There is no clearly dominant strategy.

We may want to put the trading results into the context of the overall Sharpe ratio for the S&P500 index, that is often used as benchmark. For our trading period this is 0.29. We therefore notice that trading the index with either model improves slightly on this benchmark for forecast horizons $h = 2$ and $h = 5$. For $h = 1$ performance is not attractive. We, generally, get more attractive Sharpe ratios for trading a volatility derivative. This is expected as, in this latter case, we are actually trading something very similar to what is forecast and not using an indirect correlation. Also, we notice that Sharpe ratios tend to improve for the longer forecast horizon. Here, we use the effect to our advantage, that a potentially wrong position might be corrected the next day by the correct forecast. It is a very crude way of diversification in time.

While Sharpe ratios of more than 1.3 for the longest horizon volatility strategies with either model seem attractive, it is necessary to notice, that we did not yet carry out an analysis for different time periods. The effect steems stable, but, as volatility gets more and more attention, it seems probably that any potential inefficiency will fade quickly.

As an overall observation, we conclude that the given inputs seem to be able to produce attractive Sharpe ratios for either an index or volatility derivative trade for a forecast horizon of 1 week, $h = 5$. This applies to both the linear and RNN models. As there does not seem to be any systematic bias towards one model type or the other the availability of data versus ease of implementation can be the basis for the choice of which strategy to adopt. In both cases

5-minute intraday data is necessary to operate the models. Both models train very quickly even on a standard laptop or personal computer. Therefore, training and evaluation time are of no concern, typically. If only a short amount of data is available the RNN model seems preferable. If a longer amount of data is available, then, the choice is pretty much up to the availability of an implementation of one or the other model.

6 Conclusion

This paper analyzes the potential of a heterogeneous autoregressive model including jumps to forecast realized volatility (RV). For this approach we computed RV based on a 20-year history of 5-minutes intraday data for the S&P500 index. Our results show that the base HAR-RV-J model is indeed able to provide a satisfactory forecast of RV. This outlined not only by the statistical error measures, but also by an analysis of trading efficiency based on the SPY ETF, the VIX index and the VXX ETN. Using our approaches attractive Sharpe Ratios can be obtained that outperform a common benchmark.

Our analysis also includes a Recurrent Neural Network (RRN) that uses the same inputs as the linear base HAR-RV-J model for a comparison of performance. These inputs are daily, weekly, and monthly RVs, plus the jumps. Finally, we build a hybrid model that additionally feeds the linear forecast to an RNN.

The results of all three model types are of similar quality. However, we notice that the RNN models are able to achieve these results with a shorter input time frame. That means, that when historical data is scarce, we can rely on an RNN to still deliver robust performance. Additionally, the RNN errors are uniformly low, while the errors for the linear model only reduce, once an input time frame of 1 or 2 years is used. Finally, we observe that the results between HAR-RV-J and RNNs do not differ too much, and attractive Sharpe ratios are obtained for trading a volatility derivative. The present work is, in our view, just a starting point to analyze the trading efficiency of intraday RV models.

Acknowledgement: The authors would like to acknowledge also the gracious supports of this work by the EPSRC and ESRC Centre for Doctoral Training on Quantification and Management of Risk & Uncertainty in Complex Systems & Environments (EP/L015927/1). We would like to thank participants at the Forecasting Financial Markets (FFM 2016), Quantitative Finance and Risk Analysis (QFRA2016) conferences, and at seminar talks in the University of Liverpool and Leibniz Universität Hannover. Remaining errors are ours.

References

Aït-Sahalia, Y. and Jacod, J., Testing for jumps in a discretely observed process. *Ann Stat* **37**(1), 184–222.

Alexander, C., Kapraun, J. and Korovilas, D., Trading and investing in volatility products. *Financ Markets, Inst Instruments*, 2015, **24**(4), 313–347.

Andersen, T.G. and Bollerslev, T., Answering the skeptics: Yes, standard volatility models do provide accurate forecast. *Int Econ Rev*, 1998 **39**(4), 885–905.

Andersen, T.G., Bollerslev, T., Diebold, F.X. and Labys, P., The distribution of realized exchange rate volatility. *J Am Stat Assoc*, 2001, **96**, 42–55.

Andersen, T.G., Bollerslev, T., Diebold, F.X. and Labys, P., Modeling and forecasting realized volatility. *Econometrica*, 2003, **71**(2), 579–625.

Andersen, T.G., Bollerslev, T., Christoffersen, P.F. and Diebold, F.X., Volatility and correlation forecasting. *Handbook Econ Forecasting*, 2006, **1**, 777–878.

Andersen, T.G., Bollerslev, T. and Diebold, F.X., Roughing it up: Including jump components in the measurement, modelling, and forecasting of return volatility. *Rev Econ Statistics*, 2007, **89**(4), 701–720.

Andersen T.G., Bollerslev T. and Dobrev D., No-arbitrage semimartingale restrictions for continuous-time volatility models subject to leverage effects, jumps and i.i.d. noise: Theory and testable distributional implications. *J. Econometrics*, 2007b, **138**(1), 125–180.

Andersen, T.G., Dobrev, D. and Schaumburg, E., Jump-robust volatility estimation using nearest neighbour truncation. *J Econometrics*, 2012, **169**(1), 75–93.

Andreou, E. and Ghysels, E., Rolling-sample volatility estimators: Some new theoretical simulation, and empirical results. *J Bus Econ Stat*, 2002, **20**(3), 363–376.

Back, K., Asset Pricing for General Processes, *J Math Econ*, 1991, **20**, 317–395.

Bajgrowicz, P., Scaillet, O. and Treccani, A., Jumps in high-frequency data: Spurious detections, dynamics, and news. *Manag Sci*, 2016, **62**(8), 2198–2217.

Barndorff-Nielsen, E.E. and Shephard, N., Non-Gaussian Ornstein-Uhlenbeck based models and some of their uses in financial economics. *J R Statist Soc B*, 2001, **63**(2), 167–241.

Barndorff-Nielsen, E.E. and Shephard, N., Econometric analysis of realized volatility and its use in estimating stochastic volatility models. *J R Statist Soc B*, 2002a, **64**(2), 253–280.

Barndorff-Nielsen, E.E. and Shephard, N., Estimating quadratic variation using realized variance. *J Appl Econom*, 2002b, **17**(5), 457–477.

Barndorff-Nielsen, O. and Shephard, N., Power and bipower variation with stochastic volatility and jumps (with discussion). *J. Financ. Economet*, 2004, **2**, 1–48.

Barndorff-Nielsen O.E and Shephard N., Econometrics of testing for jumps in financial economics using bipower variation. *J Financ Economet*, 2006, **4**(1), 1–30.

Barunik, J. and Krehlik, T., Combining high frequency data with non-linear models for forecasting energy market volatility. *Expert Sys Appl*, 2016, **55**, 222–242.

Bildirici, M. and Ersin, O.O., Improving forecasts of GARCH family models with the artificial neural networks: An application to the daily returns in Istanbul Stock Exchange. *Expert Syst Appl*, 2009, **36**(4), 7355-7362.

Bollerslev, T., Generalized autoregressive conditional heteroskedasticity. *J Econometrics*, 1986, **31**, 307–327.

Bollerslev, T., Engle, R.F. and Nelson, D.B., ARCH models, in Daniel Mcfadden and Robert F. Engle, eds., *Handbook of Econometrics*, 1994, **4**, Elsevier: Amsterdam.

Borovkova, S. and Mahakena, D., News, volatility and jumps: The case of natural gas futures. *Quant Finance*, 2015, **15**(7), 1217–1242.

Boudt, K. and Zhang, J., Jumps robust two time scale covariance estimation and realized volatility budgets. *Quant Finance*, 2015, **15**(6), 1041–1054.

Carr, P and Wu, L (2003) What type of process underlies options? A simple robust test. *J Finance*, 2003, **58**(6), 2581–2610.

Carr, P. and Lee, R., Volatility Derivatives. *Annu Rev Financial Econ*, 2009, **1**, 319–339.

Celik, S. and Ergin, H., Volatility forecasting using high frequency data: Evidence from stock markets. *Econ Model*, 2014, **36**, 176–190.

Corsi, F., A simple approximate long-memory model of realized volatility. *J Financ Economet*, 2009, **7**(2), 174–196.

Christensen, K., Oomen R.C.A. and Podolskij, M., Fact or friction: Jumps at ultra high frequency. *J Financial Econom*, 2014, **114**(3), 576–599.

Dunis, C.L. and Huang, X., Forecasting and trading currency volatility: An application of recurrent neural regression and model combination. *J Forecasting*, 2002, **21**(5), 317–354.

Elman, J.L., Finding structures in time. *Cognitive Sci*, 1990, **14**(2), 179–211.

Engle, R and Patton, S.J., What good is a volatility model? *Quant Finance*, 2001, **1**(2), 237–245.

Fan J, Wang Y, Multi-scale jump and volatility analysis for high-frequency financial data. *J Am Stat Assoc*, 2007, 102, 1349–1362.

Fassas, A. and Siriopoulos, C., The efficiency of VIX futures market A panel data approach. *J. Alternative Inv*, 2012, **14**(3), 55–65.

Hajizadeh, E., Seifi, A., Zarandi, M.F. and Turksen, I., A hybrid modeling approach for forecasting the volatility of S&P index return. *Expert Syst Appl*, 2012, **39**(1), 431–436.

Hansen, P.R. and Lunde, A., Realized variance and market microstructure noise. *J Bus Econ Stat*, 2006, **24**(2), 127–161.

Hansen, P.R., Huang, Z. and Shek, H.H., Realized GARCH: A Joint Model for Returns and Realized Measures of Volatility. *J Appl Economet*, 2012, 27(6), 877–906.

Harvey, A., Ruiz, E. and Shephard, N., Multivariate stochastic variance models. *Rev Econ Studies*, 1994, **61**(2), 247–264 .

Haykin, S., *Neural networks: a comprehensive foundation*. Prentice Hall Englewood Cliffs, 2007, NJ, USA.

Huang, X. and Tauchen, G. (2005) The relative contribution of jumps to total price variance. J Finan Economet, 2005 **3**(4), 456–499.

Hyndman, R.J. and Koehler, A.B., Another look at measures of forecast accuracy, *Int J Forecasting*, 2006, 22(4), 679-688.

Hyndman, R.J. and Athanasopoulos, G., *Forecasting: principles and practice*, OTexts.com (online open access), 2014.

Jammazi, R. and Aloui, C., Crude oil price forecasting: Experimental evidence from wavelet decomposition and neural network modelling. *Energ Econ*, 2012, **34**(3), 828–841.

Jiang, G.J. and Oomen, R., Testing for jumps when asset prices are observed with noise - A swap variance approach. *J. Econometrics*, 2008, **144**(2), 352–370.

Kotkatvuori-Ornberg, J., Measuring actual daily volatility from high frequency intraday returns of the S&P futures and index observations. *Expert Syst Appl*, 2016, **43**, 213–222.

Kristjanpoller, W., Fadic, A. and Minutolo, M., Volatility forecast using hybrid Neural Network models. *Expert Syst Appl*, 2014, **41**(5), 2437–2442.

Kristjanpoller, W. and Minutolo, M., Gold price volatility: A forecasting approach using the Artificial Neural Network-GARCH model. *Expert Syst Appl*, 2015, **42**(20), 7245–7251.

Lee, S.S., Jumps and information flow in financial markets. *Rev Fin Stud*, 2012, **25**(2), 439–479.

Lee, S.S. and Hannig, J., Detecting jumps from Levy jump diffusion processes. *J Fin Econ*, 2010, *96*(2), 271–290.

Lee, S.S. and Mykland, P.A., Jumps in financial markets: A new nonparametric test and jump dynamics. *Rev Financial Stud* **21**(6), 2535–2563.

Ljung, G.M. and Box, G.E.P., On a measure of lack of fit in time series models, *Biometrika*, 1978, **65**(2), 297-303.

Lux, T. and Moreles-Arias, L., Relative forecasting performance of volatility models: Monte Carlo evidence. *Quant Finance*, 2013, **13**, 1375–1394.

Panella, M., Barcellona, F. and D'Ecclesia, R.L., Forecasting energy commodity prices using neural networks. *Adv Decision Sci*, 2012, Article ID 289810, 26 pages.

Papavassiliou V.G., Allowing for jump measurement in volatility: A high-frequency financial data analysis of individual stocks. *B Econ Res*, 2016, **68**(2), 124-132.

Papadimitriou, T., Gogas, P. and Stathakis, E., Forecasting energy markets using support vector machines. *Energ Econ*, 2014, **44**, 135–142.

Prokopczuk, M., Symeonidis, L. and Wese Simen, C., Do jumps matter for volatility forecasting? evidence from markets. *J Futures Markets*, 2016, **36**(8), 758-792.

Sevi, B., Forecasting the volatility of crude oil futures using intraday data. *Eur J Oper Res*, 2014, **235**(3), 643–659.

Taylor, S.J., *Modeling Financial Time Series*. Chichester, 1986, Wiley, UK.

Zhang, L., Mykland, P. and Ait-Sahalia, Y., A tale of two time scales: Determining integrated volatility with noisy high frequency data. *J Am Stat Assoc*, 2005, **100**, 1394–1411.

Zhang, J.E., Shu, J. and Brenner, M. The new market for volatility trading, *J Future Markets*, 2010, **30**(9), 809-833.

Zhou, B., High frequency data and volatility in foreign exchange rates. *J Bus Econ Stat*, 1996, **14**(1), 45–52.