Artificial neural network based non-linear transformation of high-frequency returns for volatility forecasting

Christian Mücher ^{1,2,*}

[1] *Chair of Statistics and Econometrics, University of Freiburg, Freiburg, Germany*
[2] *Graduate School of Decision Sciences, University of Konstanz, Kontanz, Germany*

Correspondence*:
Chair of Statistics and Econometrics, University of Freiburg, Rempartstraße 10,
79098, Freiburg, Germany
christian.muecher@vwl.uni-freiburg.de

ABSTRACT

This paper uses Long Short Term Memory Recurrent Neural Networks to extract information from the intraday high-frequency returns to forecast daily volatility. Applied to the IBM stock, we find significant improvements in the forecasting performance of models that use this extracted information compared to the forecasts of models that omit the extracted information and some of the most popular alternative models. Furthermore, we find that extracting the information through Long Short Term Memory Recurrent Neural Networks is superior to two Mixed Data Sampling alternatives.

Keywords: Neural networks, forecasting, high-frequency data, realized volatility, mixed data sampling, long short term memory

1 INTRODUCTION

The volatility, the time-varying centered second moment of a financial asset, is crucial to measure, forecast, and evaluate financial risk. There exist many ways of modeling and forecasting volatility separable into two main groups: Models based on (squared) daily returns and models based on the Realized Volatility (RV) estimator. In the first group, volatility is treated as a latent variable and estimated from the model. Famous examples in this regard are, on the one hand, (G)ARCH models (Engle, 1982; Bollerslev, 1986) and their various extensions that treat volatility as conditionally observable. On the other hand, Stochastic Volatility models (Taylor, 1986; Ruiz, 1994) treat conditional volatility as random variables and rely on filtering techniques for estimation and forecasting. While the vast models in this group capture stylized properties of financial data such as volatility clustering, long-memory, and asymmetric reactions of volatility to positive and negative shocks, they generally perform worse in forecasting volatility compared to the models of the second group (Andersen et al., 2004; Sizova, 2011).

The availability of high-frequency (HF), intraday returns and the introduction of RV as an estimator of integrated volatility over a day (Andersen et al. (2001a), Andersen et al. (2001b), Barndorff-Nielsen and Shephard (2002a), Barndorff-Nielsen and Shephard (2002b)) lead to the second group of models. Since the RV ex-post gives a consistent volatility estimate, the main focus of models in the second group is forecasting. Andersen et al. (2001a), Andersen et al. (2001b) and Barndorff-Nielsen and Shephard (2002a) find that RV and the logarithm of RV exhibit long memory. Their autocorrelation functions show a hyperbolic decay, meaning that past shocks have a very long influence on the system of RV. Therefore the authors propose forecasting volatility via fractional integrated autoregressive moving average (ARFIMA) models to account for the long memory. The most prominent alternative to ARFIMA models for forecasting volatility based on RV is the Heterogeneous Autoregressive Model (HAR) by Corsi (2009). The HAR model approximates the long memory in the data through RV's daily, weekly, and monthly averages. These averages are used in a linear model as explanatory variables to predict volatility. Corsi (2009) finds that the HAR model performs better than the ARFIMA models in forecasting volatility. The HAR model is popular because of its good performance and ease of implementation (the HAR can be estimated by simple OLS regression). There exist many extensions of the HAR model in the literature, such as the HAR with jumps model of Andersen et al. (2007), the Semivariance HAR of Patton and Sheppard (2015), or the HARQ of Bollerslev et al. (2016). However, the standard HAR model, both for the level and the logarithm of RV, still is a challenging benchmark to beat in applications on real financial data.

Artificial Neural Networks (ANNs) have become more and more popular over the last decade, and various fields apply them for classification, prediction, and modeling tasks. Cybenko (1989) and Hornik et al. (1989) show the capability of Feed Forward Neural Networks (FNNs), fully connected ANNs with one hidden layer, to approximate any continuous function on a compact set arbitrarily well. Furthermore, Schäfer and Zimmermann (2006) show that Recurrent Neural Networks (RNNs) can approximate any open, dynamic system arbitrarily well. The popularity of ANNs is, on the one hand, due to these theoretical results. On the other hand, ANNs have been among the winning algorithms for various classification and forecasting competitions over the past years. RNNs combine the ability of ANNs to capture complex non-linear dependencies in the data with capturing temporal relationships. Long Short Term Memory (LSTM) RNNs (Hochreiter and Schmidhuber, 1997) are a type of RNN specifically designed to capture long memory in data. Their capacity to capture non-linear, long-term dependencies in the data make them the perfect candidates for modeling volatility.

This paper aims to use LSTMs to non-linearly transform the HF returns of a financial asset, observed within a day, into a daily, scalar variable and to use this variable to forecast volatility. Non-linear transformations of the HF returns are not novel since the RV estimator (the sum of the squared HF returns of a day) is also a non-linear transformation, but a particular one. We investigate whether volatility forecasts solely constructed from the ANN-based transformation of the HF returns are different from forecasts obtained through the past RVs. While the ANN transformation is very flexible in the functional form, the resulting sequence might not capture the long persistence in the volatility, as the RV estimator does. However, the flexibility of the functional form might capture other information that is useful to predict volatility and that the RV estimator does not take into account. Examples of such information are the sign of the HF returns or patterns of HF returns occurring over a day. We thus combine the two approaches and investigate whether the resulting model exhibits a superior forecasting performance compared to the models that rely on each measure alone.

An alternative approach to transforming the HF returns is the Mixed Data Sampling (MIDAS) approach of Ghysels et al. (2004). In MIDAS, the transformation happens through a weighted sum of the HF returns. The weights are obtained non-linearly, e.g., by an Almon or a Beta Lag Polynomial (Ghysels et al., 2004). We introduce a novel type of MIDAS model that obtains those weights through an LSTM cell. In MIDAS applications, however, the construction of the transformed HF measure is linear.

Though, as mentioned earlier, the RV estimator is also a transformation of the HF returns, throughout the paper, we will use the term transformed HF returns or transformed measure to refer to the scalar variable obtained through either the ANN transformation or the MIDAS transformation.

We compare the forecasting performance of models that use either one of the transformed measures to forecast volatility with each other and with models that construct the forecasts relying solely on information from past RV, such as the HAR model. We can thus answer whether the transformation can extract at least the same information as the past RV. We further compare these models' forecasts with those obtained from models that combine the RV information with the transformed measures, allowing us to investigate whether the transformed measures contain information supplementary to the RV. Lastly, we can compare the different transformation methods to determine whether the non-linearity introduced through ANNs performs differently from the MIDAS approaches.

The remainder of this paper is structured as follows: Section two gives an overview of the literature in volatility forecasting with ANNs. Section three introduces the LSTM RNN, and section four explains the different transformations of the HF returns. It first describes the non-linear transformation through LSTMs and then shows the two MIDAS approaches. Section five elaborates on using the transformed HF returns to generate volatility forecasts. We further introduce the benchmark models to which we compare our proposed methodology. Finally, we present the results of our empirical application in section six, and section seven concludes.

2 LITERATURE REVIEW

A vast area of finance applies ANNs. For example, White (1988), among others, uses ANNs to predict stock returns while Gu et al. (2020) use ANNs for asset pricing and Sadhwani et al. (2021) apply ANNs for mortgage risk evaluation. ANNs are further applied to model and forecast financial risk. The literature in this field reflects the two main branches of volatility modeling and forecasting mentioned in the introduction: models based on daily (squared) returns and models based on realized measures estimated from the HF returns. An early contribution to the literature of volatility modeling and forecasting through daily squared returns is Donaldson and Kamstra (1997). The authors introduce a semi nonparametric non-linear GARCH model based on ANNs and show superior performance to other GARCH type alternatives. Franke and Diagne (2006) show that ANNs yield nonparametric estimators of the conditional variance function of an asset when trained with daily returns as inputs and squared returns as targets. Their results have been applied by Giordano et al. (2012) and generalized for the Multi- Layer-Perceptron (MLP), fully connected ANNs with multiple hidden layers, by Franke et al. (2019). Arnerić et al. (2014) exploit the non-linear Autoregressive Moving Average (ARMA) structure of a Jordan type RNN (Jordan, 1997) and the ARMA representation of the GARCH model to introduce the Jordan GARCH(1,1) model. Their model shows superior performance in out-of-sample root mean squared error (RMSE). Alternative approaches use the output of GARCH models, potentially combined with other explanatory variables, as inputs to an MLP see, e.g., Hajizadeh et al. (2012) and Kristjanpoller et al. (2014).

The literature on forecasting volatility via ANNs through realized measures consists of two main fields. The first field uses ANNs to relax the linearity of the HAR model by feeding the lagged daily, weekly, and monthly averages of RV to MLPs. The evidence in this branch is mixed. Rosa et al. (2014) find improvements in the forecasting performance of the non-linear HAR model, while (Vortelinos, 2017) concludes that the ANN HAR model is not predicting volatility better than the linear HAR. He argues that the MLP cannot capture the long-term dependencies in the RV. Baruník and Křehlík (2016) find mixed evidence of the ANN HAR model for the volatility of energy market prices. Their ANN-based model produces more accurate forecasts than the linear model for some forecasting horizons and some markets. Arnerić et al. (2018) find that an MLP fitted to the HAR inputs can outperform the linear benchmark. In addition, they find that including jump measures in the analysis further improves the forecasting performance. Christensen et al. (2021) find superior forecasting performance of their MLP HAR model over the linear HAR. Further, they find that the model's performance improves when additional firm-specific and macroeconomic indicators are added. Li and Tang (2020) apply an MLP to a large set of variables such as realized and MIDAS measures and option Implied Variances. They find that the resulting model outperforms the linear benchmark. The performance improves further through an ensemble learning algorithm that combines the outputs of other linear and non-linear machine learning techniques, such as penalized regression and random forests, with the output from the ANN model.

The second field in the literature utilizes RNNs to capture, in addition to non-linearity, long-term dependencies in the data. Miura et al. (2019) examine the volatility of cryptocurrencies finding that a ridge regression yields the best out of sample forecasting results, followed by LSTM RNNs. Baştürk et al. (2021) apply LSTM RNNs to the past measure of RV and the negative part of past daily returns to jointly forecast the volatility and the Value at Risk (VaR) of a financial asset. The authors find superior forecasting performance of the LSTM network for the VaR forecasts. However, their approach cannot produce improved volatility forecasts compared to the linear alternatives.

A recent contribution to both branches of this literature is Bucci (2020), who compares the forecasting performance of various ANN structures to standard benchmarks from the financial econometrics literature such as the HAR model and ARFIMA models. He further investigates how adding macroeconomic and financial indicators as exogenous explanatory variables improves the model's forecasting performance. The target variable in his analysis is the monthly log square root of the RV. He finds that the long memory type ANNs such as the LSTM network outperform the financial econometrics literature's classical models. Furthermore, these models outperform the ANNs that do not account for long memory in the data. This result holds for various forecasting horizons.

Finally, Rahimikia and Poon (2020a) and Rahimikia and Poon (2020b) propose a HAR model augmented by an ANN applied to HF limited order book information and news sentiment data. In both papers, the authors find a superior forecasting performance of their model compared to the HAR benchmark. Their approach of augmenting the HAR model by transformed HF data is similar to the idea of this paper. In parts of our application, we augment models for LF measures such as the HAR with transformed HF information. The difference is that we consider the HF returns and not other auxiliary HF information. Further, we also consider models that use only the information from the transformed HF returns for the forecast.

3 LONG SHORT TERM MEMORY

LSTM RNNs are a specific type of RNN structures that overcome the problem that classical RNNs face. Specifically, the limited capacity of such networks to learn long-term relationships due to vanishing or exponentially increasing gradients (Hochreiter, 1991; Bengio et al., 1994). The cornerstone of LSTMs is the long memory cell denoted by C_τ. A candidate value of which, $\tilde{C_\tau}$ is a non-linear transformation (using the hyperbolic tangent activation function $tanh$[1]) of a linear combination of the τ-th periods' input vector values v_τ and the previous periods' output value $y_{\tau-1}$ plus an intercept

$$\tilde{C_\tau} = tanh(\Theta_C[y_{\tau-1}, v_\tau] + c_C). \tag{1}$$

Next, the values of the forget f_τ and the input i_τ gate are computed. These are obtained by applying the sigmoid activation function $\sigma(\cdot)$[2] to a linear combination of the input vector values v_τ and the previous

[1] The hyperbolic tangent function applied to value x is $tanh(x) = [\exp(x) - \exp(-x)]/[\exp(x) + \exp(-x)]$. It is a sigmoid function rescaled to the interval $(-1, 1)$.

[2] The sigmoid function applied to value x is defined as $\sigma(x) = 1/(1 - \exp(-x))$.

periods' output value $y_{\tau-1}$ plus an intercept.

$$f_\tau = \sigma(\Theta_f[y_{\tau-1}, v_\tau] + c_f) \tag{2}$$

$$i_\tau = \sigma(\Theta_i[y_{\tau-1}, v_\tau] + c_i) \tag{3}$$

$$\tag{4}$$

The memory cell value is computed by

$$C_\tau = f_\tau C_{\tau-1} + i_\tau \tilde{C}_\tau, \tag{5}$$

i.e., it combines the previous periods' cell value and the current periods' candidate cell value. Since the sigmoid function returns values on the interval $(0, 1)$, f_τ denotes the share to be "forgotten" from the previous cell state and i_τ the share of the proposal state to be added to the new cell state. The output of the LSTM cell y_τ is generated by applying the $tanh$ function to the memory cell values and multiplying the result by the value of the output gate o_τ

$$y_\tau = o_\tau \psi(C_\tau), \tag{6}$$

where the latter is obtained in the same manner as the values of f_τ and i_τ

$$o_\tau = \sigma(\Theta_o[y_{\tau-1}, v_\tau] + c_o). \tag{7}$$

The output gate gives the share of the activated cell values to return as the output of the memory cell. LSTM cells thus are dynamic systems wherein interacting layers drive the hidden state dynamics. This interaction enables the LSTM cell to account for a high degree of non-linearity and to capture long-term dependencies in the data.

4 TRANSFORMATION OF THE HIGH-FREQUENCY RETURNS

Denote by $r_{t,j}$ the t-th days' j-th log-return. We have $j = 1, \dots, M$ equidistantly sampled returns within day t. The increments between two intraday returns determine the number of intraday observations. For returns sampled every 5 minutes within a normal trading day at the New York Stock Exchange, we obtain 78 intraday high-frequency returns. We will denote the vector of the intraday returns on day t by $r_{t,1:M}$. We aim to apply a transformation to $r_{t,1:M}$ that returns a scalar value $\tilde{x}_t(\theta^{HF})$ which we will refer to as the transformed measure. The transformation depends on parameter vector θ^{HF}. We present three different methods to obtain the transformed measure in the following.

4.1 Non-Linear Transformation

The LSTM architecture described earlier can be used for a non-linear transformation of the HF returns. In its' simplest form, we use the sequence of the M intraday returns as input to the LSTM cell, and the output of the cell at time M, $y_{t,M}$, as the transformed value. We thus iterate through the LSTM equations over the $j = 1, \dots, M$ intraday returns at day t

$$\tilde{C}_{t,j} = tanh(\Theta_C[y_{t,j-1}, v_{t,j}] + c_C) \tag{8}$$

$$f_{t,j} = \sigma(\Theta_f[y_{t,j-1}, v_{t,j}] + c_f) \tag{9}$$

$$i_{t,j} = \sigma(\Theta_i[y_{t,j-1}, v_{t,j}] + c_i) \tag{10}$$

$$C_{t,j} = f_{t,j}C_{t,j-1} + i_{t,j}\tilde{C}_{t,j} \tag{11}$$

$$o_{t,j} = \sigma(\Theta_o[y_{t,j-1}, v_{t,j}] + c_o) \tag{12}$$

$$y_{t,j} = o_{t,j}\psi(C_{t,j}), \tag{13}$$

and set $\tilde{x}_t(\theta^{HF}) = y_{t,M}$, where θ^{HF} contains the LSTM cell weights and intercepts. Through the interaction of the three gates and the non-linear activation of the proposal state and the actual state, the LSTM cell allows for a high degree of non-linearity while also capturing long memory in the data. Both the cell input ($v_{t,j}$) and output ($y_{t,j}$) at within day lag j are scalars. The parameter vector θ^{HF} of the model using one LSTM cell thus contains 12 parameters: Four 2×1 weight vectors and four intercepts. To increase the degree of non-linearity, we further use a network that consists of one hidden layer of LSTM cells and use the outputs of these cells as inputs to another LSTM cell returning a scalar value.

4.2 MIDAS Transformations

Denote by $\omega_j(\theta^{HF})$ the weights associated with the j-th intraday return on day t. The weights are determined by the elements of θ^{HF}. While the weights may be obtained in a non-linear manner, the resulting transformed measure

$$\tilde{x}_t(\theta^{HF}) = \sum_{j=1}^{M} w_j(\theta^{HF})r_{t,j} \tag{14}$$

is a weighted sum and thus linear.

In a Beta Lag MIDAS model (labeled Beta MIDAS hereafter), the weight associated with the j-th lag is obtained by

$$\omega_j(\varphi_1, \varphi_2) = \frac{B\left(\frac{j}{M}, \varphi_1, \varphi_2\right)}{\sum\limits_{j=1}^{M} B\left(\frac{j}{M}, \varphi_1, \varphi_2\right)} \tag{15}$$

where $B(\cdot, \varphi_1, \varphi_2)$ is the probability density function (pdf) of the Beta distribution. In this case, the parameter vector $\theta^{HF} = (\varphi_1, \varphi_2)'$ contains the Beta distribution parameters. While only depending on two parameters, the weights obtained from the normalized Beta pdf are capable to capture complex non-linear functional forms.

An alternative way to obtain the weights associated with the j-th observation is to use the lag values as inputs to an LSTM cell. In this case, the input to the LSTM cell is $v_j = j$ with $j = 1, 2, \ldots, M$. The corresponding output (y_j) lies on the interval between $(-1, 1)$. Note that in this case, v_j and y_j do not depend on t since they only vary within the day but not over the days. To transform the output at within day lag j (y_j) into a weight, we apply the $exponential$ function and normalize the values, i.e.

$$w_j = \frac{\exp{(y_j)}}{\sum\limits_{j=1}^{M} \exp{(y_j)}}, \tag{16}$$

yielding weights that lie in the interval $(0, 1)$ and sum up to one. This transformation is similar to the Beta MIDAS, where the Beta pdf values associated with j/M are normalized such that they sum up to one. Same as above, the parameter vector θ^{HF} of the LSTM MIDAS model contains 12 parameters.

5 VOLATILITY FORECASTING

This paper aims to forecast the daily volatility of a financial asset. Consider the price process of a financial asset P_t, determined by the stochastic differential equation

$$d\ln(P_t) = \mu_t dt + \sigma_t dW_t \tag{17}$$

where μ_t and σ_t denote the drift and the instantaneous or spot volatility process, respectively, and W_t is a standard Brownian motion. The integrated variance from day $t-1$ to t is then defined as

$$IV_t = \int\limits_{t-1}^{t} \sigma_s^2 ds. \tag{18}$$

The integrated variance yields a direct measure of the discrete time return volatility (Andersen et al., 2004), but the series is latent, and we can not observe it directly. However, we can estimate the integrated variance ex-post through the RV estimator defined as

$$RV_t = \sum_{j=1}^{M} r_{t,j}^2, \tag{19}$$

i.e., the sum of the M squared intraday HF returns. Our goal is to assess how the information obtained from applying the different transformations of the HF returns explained earlier helps predict one step ahead volatility. To answer this, we consider different scenarios.

First, we vary the input variables used to predict volatility, considering three different settings. We start by combining the transformed measure $\tilde{x}_t$ (for readability, we omit the dependence of the transformed measure on θ^{HF} from here on) with past information on the RV. Next, we assess how this combination fares compared to using each stream of information on itself, i.e., using only the information obtained from the transformation and using only the past information on RV. Finally, when using the information on the transformed measure, we again differentiate between two settings: In the first, we only use the most recent (the past days) value of the transformed measure. In the second, we account for dynamics in the transformed measure and use the values of multiple past days. The contributions of (Andersen et al., 2001a) and (Andersen et al., 2001b) as well as models like the HAR and the work by, e.g., Audrino and Knaus (2016), show that it is necessary to account for the long memory in the volatility. In the setting where we use multiple past values of the transformed measure, we therefore apply an LSTM cell to the sequence of transformed measures. This means that for $\tau = 1, \ldots, t$, we iterate over

$$\tilde{C}_\tau = tanh(\Theta_C[y_{\tau-1}, \tilde{x}_\tau] + c_C) \tag{20}$$

$$f_\tau = \sigma(\Theta_f[y_{\tau-1}, \tilde{x}_\tau] + c_f) \tag{21}$$

$$i_\tau = \sigma(\Theta_i[y_{\tau-1}, \tilde{x}_\tau] + c_i) \tag{22}$$

$$C_\tau = f_\tau C_{\tau-1} + i_\tau \tilde{C}_\tau \tag{23}$$

$$o_\tau = \sigma(\Theta_o[y_{\tau-1}, \tilde{x}_\tau] + c_o) \tag{24}$$

$$y_\tau = o_\tau \psi(C_\tau). \tag{25}$$

and set $\tilde{y}_t(\tilde{\theta}^{LF}) = y_t$, where $\tilde{\theta}^{LF}$ contains the corresponding LSTM cell weights and intercepts. Using an LSTM cell circumvents the problem of lag order selection through either information criteria or shrinkage methods. The LSTM cell takes into account the whole sequence of $\tilde{x}_{1:t}$ by storing the necessary information in the memory cell. Figure 1 depicts the underlying idea.

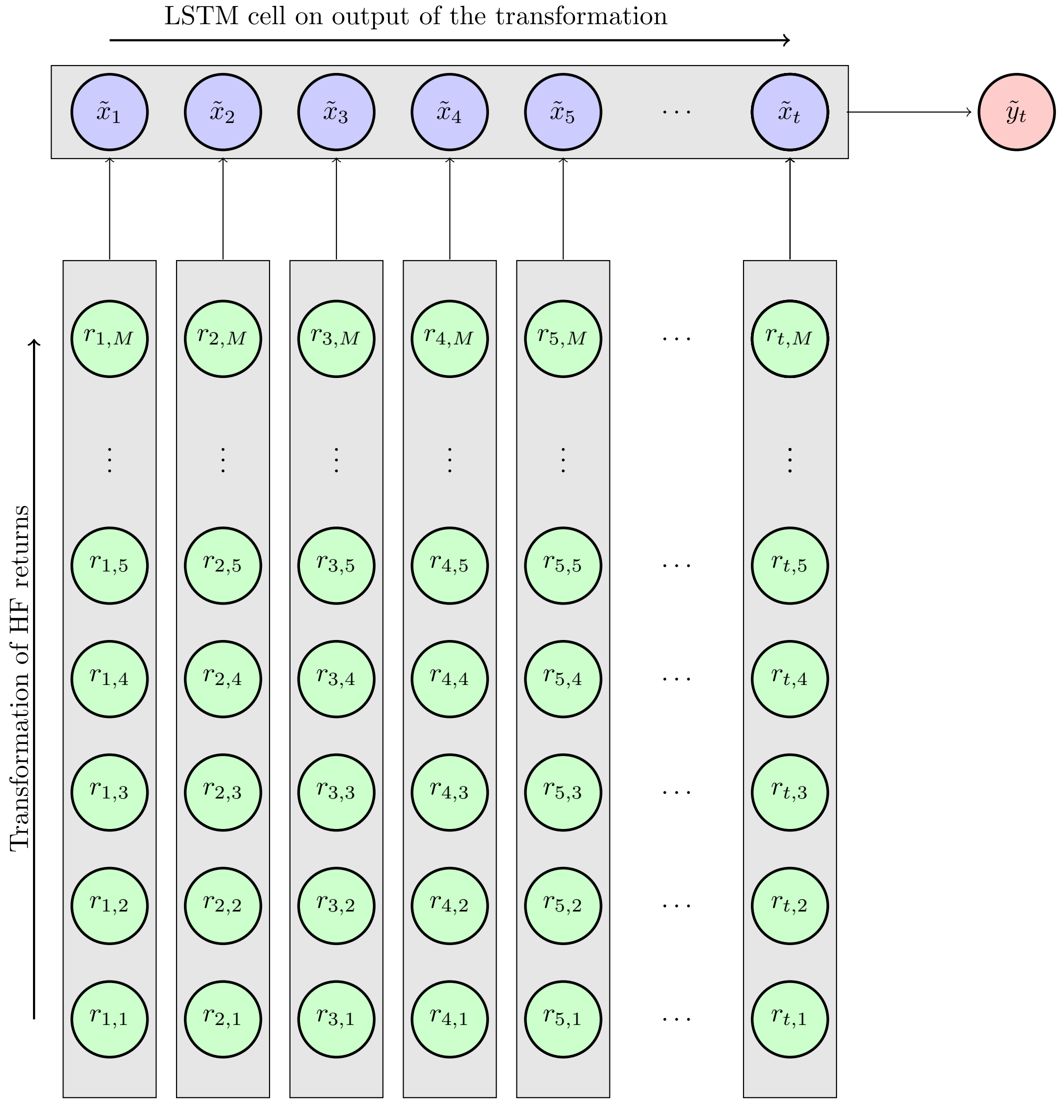

LSTM cell on output of the transformation
$\tilde{x}_1$
$\tilde{x}_2$
$\tilde{x}_3$
$\tilde{x}_4$
$\tilde{x}_5$
$\cdots$
$\tilde{x}_t$
$\tilde{y}_t$
Transformation of HF returns
$r_{1,M}$ $r_{2,M}$ $r_{3,M}$ $r_{4,M}$ $r_{5,M}$ $\cdots$ $r_{t,M}$
$r_{1,5}$ $r_{2,5}$ $r_{3,5}$ $r_{4,5}$ $r_{5,5}$ $\cdots$ $r_{t,5}$
$r_{1,4}$ $r_{2,4}$ $r_{3,4}$ $r_{4,4}$ $r_{5,4}$ $\cdots$ $r_{t,4}$
$r_{1,3}$ $r_{2,3}$ $r_{3,3}$ $r_{4,3}$ $r_{5,3}$ $\cdots$ $r_{t,3}$
$r_{1,2}$ $r_{2,2}$ $r_{3,2}$ $r_{4,2}$ $r_{5,2}$ $\cdots$ $r_{t,2}$
$r_{1,1}$ $r_{2,1}$ $r_{3,1}$ $r_{4,1}$ $r_{5,1}$ $\cdots$ $r_{t,1}$

Figure 1. Method depicted

The HF returns are transformed in the vertical direction, meaning that for each day $1, \ldots, t$ the same type of transformation is applied to the HF returns of that day. The result is a sequence of $\tau = 1, \ldots, t$ transformed measures $\tilde{x}_\tau$ to which an LSTM cell is horizontally applied.

We then linearly combine the output $\tilde{y}_t$ (for readability, we omit the dependence of $\tilde{y}_t$ on $\tilde{\theta}^{LF}$ from here on) with the other LF measures under consideration. This results in the following, case dependent, transformed HF information input variable

$$\nu_t^{HF} = \begin{cases} \tilde{x}_t & \text{only recent HF information} \\ \tilde{y}_t & \text{all HF information.} \end{cases} \tag{26}$$

We consider two settings, where we linearly combine the information from the past RV with that of the transformed HF returns and add an intercept. Herein, in resemblance to the classical HAR model, we first use past, daily, weekly, and monthly averages of the natural logarithm of RV (referred to as log RV hereafter). Denote the logarithm of RV at day t by $\ln RV_t$ i.e.

$$\ln RV_t = \ln\left(RV_t\right). \tag{27}$$

Weekly and monthly averages of log RV are then defined by

$$\overline{\ln RV}_t^w = \frac{1}{5}\sum_{i=1}^{5} \ln RV_{t-(i-1)} \tag{28}$$

and

$$\overline{\ln RV}_t^m = \frac{1}{22}\sum_{i=1}^{22} \ln RV_{t-(i-1)}. \tag{29}$$

Second, we use the output of an LSTM cell applied to the sequence of the past log RVs.

This results in the, case dependent, low frequency information input variable ν_t^{LF} and

$$\nu_t^{LF} = \begin{cases} (\ln RV_t, \overline{\ln RV}_t^w, \overline{\ln RV}_t^m)' \\ LSTM(\ln RV_{1:t}, \theta^{LF}) \end{cases}. \tag{30}$$

The HAR model is the most commonly used benchmark in volatility forecasting. However, its' implicit lag order selection (it is a restricted AR(22) model) is not necessarily validated in real data applications (Audrino and Knaus, 2016). As mentioned earlier, we circumvent the trouble of lag order selection since we apply an LSTM cell to the LF inputs. The LSTM cell can capture the long-term dynamics. Alternatively, one could fit an autoregressive model of order p on the RV, add the lags of the transformed measure as additional explanatory variables, and perform lag order selection via Information Criteria or shrinkage methods. However, we leave these two alternatives for further research.

We take the exponential of these linear combinations to guarantee the positiveness of the generated forecast. The output of the model thus is generated by

$$y_t(\theta) = \exp\left(c + (\nu_t^{LF})'\beta^{LF} + \beta^{HF}\nu_t^{HF}\right) \tag{31}$$

where θ is a vector collecting all parameters. β^{LF} contains either the parameters associated with the daily, weekly, and monthly averages of log RV or the parameters associated with the output of the LSTM cell applied to the sequence of log RV. β^{HF} is the parameter of the scalar measure obtained from the transformation of the HF returns, and c is an intercept. The model that only uses the transformed HF returns for the forecast corresponds to restricting $\beta^{LF} = 0$. This comparison allows for a very detailed analysis of the source of potential gains in the forecasting performance:

1. We can assess whether there are significant differences in the forecasting performances of the models that only use the transformed measure as inputs to those that combine them with the LF variables. It is thus possible to inspect whether or not the sequence of transformed HF returns captures the information included in the past RV.

2. We can investigate whether it is necessary to consider the entire information in the transformed measure or whether the most recent information suffices.
3. We can compare the different transformation methods, assessing the differences between the linear MIDAS type transformations and the non-linear transformations.
4. We can examine whether using the classical HAR inputs with a fixed lag order of 22 is enough or whether using an LSTM cell on the past RV values, which is less restrictive in terms of the lag order selection, is fruitful.

5.1 Benchmark Models

We apply a variety of benchmark models, four models of the HAR family and an ARFIMA(p,d,q) model. Our proposed methodology ensures the positiveness of the volatility predictions by construction (see equation 31). However, when fitting the benchmark models to the level of RV, the forecasts are not guaranteed to be positive. We thus implement each benchmark model once for the level of RV and once for the log of RV to allow for a fair comparison. In the latter case, the forecasts are bias corrected (Granger and Newbold, 1976), i.e.,

$$\widehat{RV}_{t+1} = \exp\left(\widehat{\ln RV}_{t+1} + \frac{1}{2}\sigma_{\varepsilon}^2\right) \tag{32}$$

where σ_{ε}^2 is the forecast error variance estimated from the residuals.

Following the suggestion of (Andersen et al., 2003), we start by fitting an ARFIMA model

$$(1-\Phi(L))(1-L)^d x_t = \Theta(L)\varepsilon_t, \quad \text{with} \quad 0 < |d| < 0.5 \tag{33}$$

for $x_t = RV_t$ and $x_t = \ln RV_t$, where ε_t is a Gaussian white noise with zero mean and variance σ^2. $\Phi(L)$ and $\Theta(L)$ are lag polynomials of degrees p and q respectively whose roots lie outside the unit circle.

Next, we implement benchmark models from the HAR family, starting with the classical HAR model (Corsi, 2009) in levels

$$RV_{t+1} = \beta_0 + \beta_1 RV_t + \beta_2 \overline{RV}_t^w + \beta_3 \overline{RV}_t^m + \varepsilon_{t+1} \tag{34}$$

and in logs

$$\ln RV_{t+1} = \beta_0 + \beta_1 \ln RV_t + \beta_2 \overline{\ln RV}_t^w + \beta_3 \overline{\ln RV}_t^m + \varepsilon_{t+1}. \tag{35}$$

For all HAR family models, the error term ε_t is assumed to be a white noise process with $\mathbb{E}[\varepsilon_t] = 0$ and $\mathbb{V}[\varepsilon_t] = \sigma_{\varepsilon}^2$. Following Andersen et al. (2007), we include the CHAR model as the second benchmark. The CHAR model is based on the jump robust Bi-Power Variation (BPV) measure of Barndorff-Nielsen and Shephard (2004), defined as

$$BPV_t = \frac{1}{\mu_1} \sum_{j=1}^{M-1} |r_{t,j}|\,|r_{t,j+1}| \tag{36}$$

where $\mu_1 = \sqrt{2/\pi}$ is the expectation of the absolute value of a standard normal random variable. The CHAR model then replaces the daily, weekly, and monthly averages of RV on the right hand side of the HAR model with the corresponding averages of BPV, i.e for levels

$$RV_{t+1} = \beta_0 + \beta_1 BPV_t + \beta_2 \overline{BPV}_t^w + \beta_3 \overline{BPV}_t^m + \varepsilon_{t+1} \tag{37}$$

and for logs

$$\ln RV_{t+1} = \beta_0 + \beta_1 \ln BPV_t + \beta_2 \overline{\ln BPV}_t^w + \beta_3 \overline{\ln BPV}_t^m + \varepsilon_{t+1}. \tag{38}$$

An alternative model that accounts for jumps is the HAR with jumps (HAR-J) model (Andersen et al., 2007). The HAR-J model adds the jump measure $J_t = \max(RV_t - BPV_t, 0)$ or, when modeling log RV, $\ln(1 + J_t)$, as an additional explanatory variable to the HAR model. However, in our application, the HAR-J model in levels produces negative volatility predictions in two cases. For the log case the average losses of the HAR-J model are very similar to those from the HAR model. We thus omit the results from the HAR-J model, though the differences are statistically significant. They are available from the authors upon request. The next benchmark model is the Semivariance-HAR (SHAR) model by Patton and Sheppard (2015), which builds on the semi-variation measure of Barndorff-Nielsen et al. (2010) differntiating between variation associated with positive and negative intraday returns. The estimators are defined as

$$RS_t^+ = \sum_{j=1}^{M} r_{t,j}^2 \mathbb{I}_{r_{t,j}>0} \tag{39}$$

and

$$RS_t^- = \sum_{j=1}^{M} r_{t,j}^2 \mathbb{I}_{r_{t,j}<0}, \tag{40}$$

where $\mathbb{I}$ is the indicator function and $RV_t = RS_t^+ + RS_t^-$. The SHAR model uses this decomposition of RV_t such that for levels

$$RV_{t+1} = \beta_0 + \beta_1^+ RS_t^+ + \beta_1^- RS_t^- + \beta_2 \overline{RV}_t^m + \beta_3 \overline{RV}_t^m + \varepsilon_{t+1} \tag{41}$$

and for logs

$$\ln RV_{t+1} = \beta_0 + \beta_1^+ \ln RS_t^+ + \beta_1^- \ln RS_t^- + \beta_2 \overline{\ln RV}_t^w + \beta_3 \overline{\ln RV}_t^m + \varepsilon_{t+1}. \tag{42}$$

The last benchmark from the HAR family is the HARQ model of Bollerslev et al. (2016). The HARQ model uses the Realized Quarticity (RQ) estimator of (Barndorff-Nielsen and Shephard, 2002a) to correct for measurement error in the RV estimator. The HARQ model for levels is

$$RV_{t+1} = \beta_0 + \beta_1 RV_t + \beta_{1Q} RQ_t^{1/2} RV_t + \beta_2 \overline{RV}_t^m + \beta_3 \overline{RV}_t^m + \varepsilon_{t+1} \tag{43}$$

and for logs

$$\ln RV_{t+1} = \beta_0 + \beta_1 \ln RV_t + \beta_{1Q} \ln RQ_t \ln RV_t + \beta_2 \overline{\ln RV}_t^w + \beta_3 \overline{\ln RV}_t^m + \varepsilon_{t+1}. \tag{44}$$

6 APPLICATION

We use the 5-minute log-returns ($M = 78$ intraday observations per trading day) of IBM from January, 02, 2001 till December 28, 2018 ($T = 4482$ days). We use the first 80% of the data (till May 27, 2015) as the in-sample data and the last 20% as the out-of-sample data. In order to obtain forecasts from each model introduced earlier, the QLIKE loss (Patton, 2011) between the forecast $y_t(\theta)$ and the next periods

RV, RV_{t+1} is minimized, i.e., the objective is to find

$$\hat{\theta} = \operatorname{argmin}_{\theta} \text{QLIKE}\left(y_t(\theta), RV_{t+1}\right), \tag{45}$$

where the QLIKE loss function is defined as

$$\text{QLIKE}(RV_{t+1}, y_t(\theta)) = \frac{RV_{t+1}}{y_t(\theta)} - \ln\left(\frac{RV_{t+1}}{y_t(\theta)}\right) - 1. \tag{46}$$

The QLIKE is a better choice when forecasting volatility than the mean squared error since it considers that the variable of interest is positive. We implement all models (except the benchmark models) in $Python$ using $Keras$ (Chollet et al., 2015) with the $TensorFlow$ (Abadi et al., 2015) backend. This workflow comes with a comprehensive set of functions, allowing custom types of neural networks. We implement, e.g., the Beta MIDAS model as a specific case of an MLP that takes a 2×1 vector of ones as inputs and has a diagonal weight matrix coinciding with the parameters φ_1 and φ_2 of the Beta pdf. The layer then returns an $M \times 1$ vector of weights associated with the standardized Beta pdf as described earlier. We estimate the parameters of all models under consideration (except the Benchmark models) by Stochastic Gradient Descent (SGD). Since SGD introduces an implicit regularization of the parameters (Soudry et al., 2018) this methodology should allow for a fair comparison of the forecasting results of the different models.

We train by Adaptive Moments SGD (ADAM (Kingma and Ba, 2014)) with a batch size (length of a randomly selected sample selected for one SGD parameter update) of 128. $Keras$ computes the gradient of RNNs by Truncated Back Propagation Through Time (Rumelhart et al., 1986); truncated in the manner that the computation of the gradient considers only a limited amount of past lags. The horizon of truncation is referred to as *lookback* and does not change the fact that the RNN considers the whole sequence of inputs when producing the forecast after training. We set the *lookback* equal to 128. We standardize the input data and divide the target data (the one step ahead RV) by its' standard deviation. We do not demean the target data to ensure positivity. We store the standard deviations to re-scale the resulting predictions in each forecasting step.

We use an expanding window scheme for forecasting: We start training for 1000 epochs (one epoch means the algorithm went through the whole sample once) on the first 80% of the data and use the trained network and the newly available information to make a one step ahead prediction. Then, the model is re-trained for another 100 epochs for each one step ahead prediction with the previous iterations parameter values as starting values, resulting in 897 out of sample forecasts. To make training more feasible, we employ early stopping criteria. These interrupt the training before the target number of epochs is hit, given that there was no improvement of the training error over several specified past epochs. The term $patience$ refers to this specified number of epochs. We set the minimum, absolute change of the training loss to be considered an improvement to 10^{-6}, the initial training step $patience$ to 500, and the $patience$ in the re-training steps to 50. The code runs on an *NVIDIA Tesla V100* GPU on the bwHPC Cluster.

We estimate the HAR family benchmark models by OLS and the ARFIMA models using R's $fracdiff$ (Maechler, 2020) and $forecast$ (Hyndman and Khandakar, 2008) packages. On the in-sample data, an ARFIMA(5,d,2) model provides the best fit for the level of RV and an ARFIMA (0,d,1) for the logarithm of RV.

6.1 Forecast Evaluation

We compare the forecasting performance of our presented model with varying inputs and the benchmark models for both levels and logs using different loss measures. First, we compare the average QLIKE loss of the different models. Next, we report the square root of the average squared error loss $(RV_t - \widehat{RV}_t)^2$ (the RMSE). We further compute Value at Risk (VaR) and Expected Shortfall (ES) forecasts based on the volatility forecasts. The VaR is the p-th quantile of the return distribution and the ES is the expected value of the return, given that the return is smaller than the VaR. We compute the daily log returns r_t as the sum of the intraday returns of day t, which is equivalent to the log return based on the difference between the log closing and opening prices. Table 1 reports descriptive statistics of the daily returns (r_t), the RV estimated from 5-minute log-returns (RV_t), and the standardized returns $z_t = r_t/\sqrt{RV_t}$ for the whole sample, the in and the out-of-sample period.

Table 1. Descriptive statistics.

	Min	Max	Mean	Median	Std	Skewness	Kurtosis
				Whole-sample period			
r_t	-11.1695	11.6993	0.0053	0.0120	1.4997	0.1136	10.5653
RV_t	0.1219	130.5922	2.3877	1.0188	5.7425	10.0287	153.5745
z_t	-2.9571	3.2444	0.0308	0.0142	0.9554	0.0833	2.6061
				In-sample period			
	Min	Max	Mean	Median	Std	Skewness	Kurtosis
r_t	-11.1695	11.6993	0.0236	0.0148	1.5545	0.2314	10.1870
RV_t	0.1325	130.5922	2.6083	1.1267	6.0041	9.9336	151.2284
z_t	-2.9571	3.2444	0.0408	0.0147	0.9514	0.0910	2.6362
				Out-of-sample period			
	Min	Max	Mean	Median	Std	Skewness	Kurtosis
r_t	-7.9331	8.5542	-0.0679	0.0000	1.2548	-0.8824	11.5016
RV_t	0.1219	63.4346	1.5063	0.6800	4.4410	9.7437	113.8447
z_t	-2.5287	2.7506	-0.0092	0.0000	0.9703	0.0594	2.4857

Note: For the descriptive statistics, r_t is scaled by 10^2 and RV_t by 10^4.

After standardizing the daily returns, their skewness and kurtosis are close to those of a standard normal distribution. For the out-of-sample period, we can not reject the H_0 of a Kolmogorov-Smirnov test that the standardized returns are standard normally distributed (p-value = 0.646). For the whole sample (p-value = 0.018) and the in-sample period (p-value = 0.011) we reject this hypothesis at the 5% level. These results are reasonable since the out-of-sample period does not contain the financial crisis. We thus use the normal distribution to compute forecasts of VaR and ES. We also compute forecasts of VaR and ES using the standardized Student-t distribution, where, similar to Brownlees and Gallo (2010), for each iteration in the expanding window, we estimate the degrees of freedom based on the information available up to time t. All estimated degrees of freedom are larger than 100, indicating no need to account for fat tails. Further, the statistical analysis results and the ranking of the models do not change compared to the case of the normal distribution. We thus do not report the Student-t distribution results here. They are available from the authors on request.

To evaluate the performance of the models in forecasting VaR and ES, we use the *asymmetric piece-wise linear* loss function of Gneiting (2011) for the VaR and the *zero-homogeneous* loss function of Fissler and Ziegel (2016) for the VaR and the ES jointly[3]. Using the short notation $r = r_t$, $\widehat{VaR} = \widehat{VaR}_{t,p}$ and $\widehat{ES} = \widehat{ES}_{t,p}$, these loss functions are

$$S_p^{VaR}(\widehat{VaR}, r) = \left(r - \widehat{VaR}\right)\left(p - \mathbb{I}_{\{r \leq \widehat{VaR}\}}\right) \tag{47}$$

and

$$S_p^{VaRES}(\widehat{VaR}, \widehat{ES}, r) = -\frac{(\widehat{VaR} - r)\mathbb{I}_{\{r \leq \widehat{VaR}\}}}{p\widehat{ES}} + \frac{\widehat{VaR}}{\widehat{ES}} + \ln(-\widehat{ES}) - 1. \tag{48}$$

6.2 Results

Table 2 reports the results of the out-of-sample losses introduced above for the different models. It further shows which models are in the Model Confidence Set (MCS) of Hansen et al. (2011) at the 10% level. We use the *arch* library of Sheppard et al. (2021) to compute the MCS p-values. In addition, we report the results of Binomial tests[4], where we test each model against the other models in figures 6 to 11 in the appendix. The table consists of two main blocks, again consisting of multiple blocks as indicated by the horizontal lines. The first main block contains the results for the ANN models, where the first two rows show the results for the models that do not use the transformed measure as additional input (indicated by the superscript O), i.e., the models corresponding to the restriction $\beta^{HF} = 0$. The first model is a non-linear HAR estimated by SGD. Non-linear since we use the exponential of a linear combination of daily, weekly, and monthly averages of log RV on the right-hand side. The second row shows the results of modeling the long memory in RV not via the restricted AR(22) character of the HAR model but an LSTM cell applied to the log RV.

Next, follow the models that use the information from the transformed measure. The superscript indicates the type of transformation: The superscript O indicates no transformation, the superscripts M-B and M-L indicate the Beta and LSTM MIDAS transformation, respectively, and the superscript LSTM plus a number indicates the non-linear, LSTM based transformation. The number indicates the number of LSTM cells in the hidden layer in this case.

The model's name indicates the type of low-frequency information: HAR refers to the daily, weekly, and monthly averages, and LSTM refers to an LSTM cell applied to the sequence of log RV. They reflect choosing $\nu_t^{LF} = (\ln RV_t, \overline{\ln RV}_t^w, \overline{\ln RV}_t^m)'$ and $\nu_t^{LF} = LSTM(\ln RV_{1:t}, \theta^{LF})$ respectively. The name O refers to only using the information from the transformed measure, i.e., it corresponds to the restriction $\beta^{LF} = 0$. Here we have two blocks again. The first reporting models that apply an LSTM cell to the sequence of the transformed measure. These models thus take into account the full information in the sequence of the transformed measure, indicated by -F in the model name. The second block refers to models that only use the most recent value of the transformed measure. The two blocks thus correspond to the choice of $\nu_t^{HF} = \tilde{x}_t$ and $\nu_t^{HF} = \tilde{y}_t$, respectively. The last block shows the results of the benchmark models, where we differentiate between them being applied for the level of RV and to the logarithm of RV (indicated by -ln in the model name). We transform the forecasts using the bias correction mentioned earlier for the latter case.

[3] There exists no strictly consistent loss function for the ES alone (Gneiting, 2011).

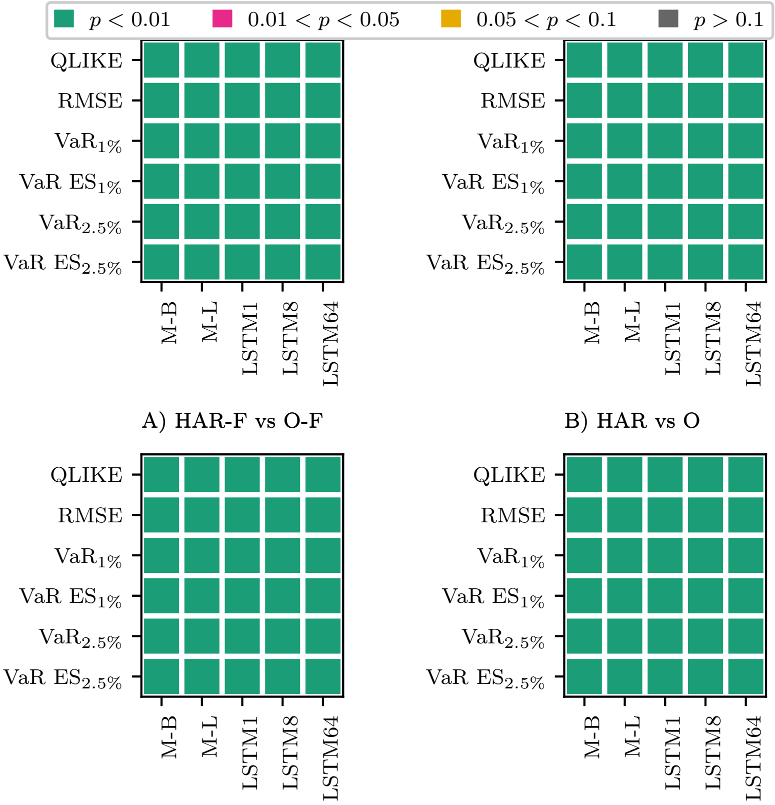
[4] The Binomial test tests whether positive and negative sign changes in the loss differential of two models are equally likely. It is also known as the Diebold Mariano Sign test.

Table 2. Out of sample losses

	Model	QLIKE	RMSE	VaR$_{1\%}$	VaR ES$_{1\%}$	VaR$_{2.5\%}$	VaR ES$_{2.5\%}$
ANN Models	HAR$^{\text{O}}$	$0.6012_{\dagger}$	0.4401	$0.6598_{\dagger}$	$-2.2635_{\dagger}$	$1.0444_{\dagger}$	$-3.0502_{\dagger}$
	LSTM$^{\text{O}}$	$0.5683_{\dagger}$	$0.4356_{\dagger}$	$0.6687_{\dagger}$	$-2.1732_{\dagger}$	$1.0497_{\dagger}$	$-3.0029_{\dagger}$
	HAR$^{\text{M-B}}$-F	$0.5995_{\dagger}$	$0.4407_{\dagger}$	$0.6580_{\dagger}$	$-2.2793_{\dagger}$	$1.0374_{\dagger}$	$-3.0671_{\dagger}$
	HAR$^{\text{M-L}}$-F	$0.5655_{\dagger}{}^{*}$	$0.4380_{\dagger}{}^{*}$	$0.6397_{\dagger}$	$-2.3329_{\dagger}$	$1.0167_{\dagger}$	$-3.1032_{\dagger}$
	HAR$^{\text{LSTM1}}$-F	$0.5990_{\dagger}{}^{*}$	$0.4347_{\dagger}{}^{*}$	$0.6446_{\dagger}{}^{*}$	$-2.2779_{\dagger}{}^{*}$	$1.0278_{\dagger}{}^{*}$	$-3.0626_{\dagger}{}^{*}$
	HAR$^{\text{LSTM8}}$-F	$0.5976_{\dagger}$	0.4400	$0.6576_{\dagger}$	$-2.2711_{\dagger}$	$1.0420_{\dagger}$	$-3.0545_{\dagger}$
	HAR$^{\text{LSTM64}}$-F	$0.5992_{\dagger}$	0.4400	$0.6579_{\dagger}$	$-2.2731_{\dagger}$	$1.0420_{\dagger}$	$-3.0559_{\dagger}$
	LSTM$^{\text{M-B}}$-F	$0.5813_{\dagger}$	$0.4361_{\dagger}$	$0.6616_{\dagger}{}^{*}$	$-2.2074_{\dagger}{}^{*}$	$1.0462_{\dagger}{}^{*}$	$-3.0209_{\dagger}{}^{*}$
	LSTM$^{\text{M-L}}$-F	$0.5458_{\dagger}$	$0.4354_{\dagger}$	$0.6617_{\dagger}$	$-2.2360_{\dagger}$	$1.0378_{\dagger}$	$-3.0464_{\dagger}$
	LSTM$^{\text{LSTM1}}$-F	$0.5779_{\dagger}$	$0.4406_{\dagger}$	$0.6668_{\dagger}{}^{*}$	$-2.1577_{\dagger}{}^{*}$	$1.0513_{\dagger}{}^{*}$	$-2.9970_{\dagger}{}^{*}$
	LSTM$^{\text{LSTM8}}$-F	$0.5701_{\dagger}$	$0.4356_{\dagger}$	$0.6688_{\dagger}$	$-2.1669_{\dagger}$	$1.0508_{\dagger}{}^{*}$	$-3.0011_{\dagger}{}^{*}$
	LSTM$^{\text{LSTM64}}$-F	$0.5693_{\dagger}$	$0.4356_{\dagger}$	$0.6626_{\dagger}{}^{*}$	$-2.2115_{\dagger}{}^{*}$	$1.0479_{\dagger}{}^{*}$	$-3.0200_{\dagger}{}^{*}$
	O$^{\text{M-B}}$-F	0.7923^{*}	$0.5205_{\dagger}{}^{*}$	$0.6666_{\dagger}{}^{*}$	$-2.5151_{\dagger}{}^{*}$	$1.1599_{\dagger}{}^{*}$	$-3.0053_{\dagger}{}^{*}$
	O$^{\text{M-L}}$-F	$0.7035_{\dagger}{}^{*}$	$0.4499_{\dagger}{}^{*}$	$0.7301_{\dagger}{}^{*}$	$-2.1101_{\dagger}{}^{*}$	$1.1650_{\dagger}{}^{*}$	$-2.8801_{\dagger}{}^{*}$
	O$^{\text{LSTM1}}$-F	0.7468^{*}	0.4514^{*}	$0.6772_{\dagger}{}^{*}$	$-2.4570_{\dagger}{}^{*}$	1.1580^{*}	$-2.9845_{\dagger}{}^{*}$
	O$^{\text{LSTM8}}$-F	0.7536^{*}	0.4517^{*}	$0.6695_{\dagger}{}^{*}$	$-2.4596_{\dagger}{}^{*}$	1.1496^{*}	$-2.9898_{\dagger}{}^{*}$
	O$^{\text{LSTM64}}$-F	0.7458^{*}	0.4516^{*}	$0.6525_{\dagger}{}^{*}$	$-2.5659_{\dagger}{}^{*}$	$1.1448_{\dagger}{}^{*}$	$-3.0239_{\dagger}{}^{*}$
	HAR$^{\text{M-B}}$	0.6108	0.4403	$0.6613_{\dagger}$	$-2.2538_{\dagger}$	$1.0485_{\dagger}$	$-3.0407_{\dagger}$
	HAR$^{\text{M-L}}$	$0.5764_{\dagger}$	$0.4406_{\dagger}$	$0.6620_{\dagger}$	$-2.2263_{\dagger}$	$1.0456_{\dagger}$	$-3.0356_{\dagger}$
	HAR$^{\text{LSTM1}}$	$0.5453_{\dagger}{}^{*}$	$0.4339_{\dagger}{}^{*}$	$0.6224_{\dagger}{}^{*}$	$-2.4520_{\dagger}{}^{*}$	$\mathbf{0.9901}_{\dagger}{}^{*}$	$\mathbf{-3.1643}_{\dagger}{}^{*}$
	HAR$^{\text{LSTM8}}$	0.6029	0.4400	$0.6576_{\dagger}$	$-2.2768_{\dagger}$	$1.0401_{\dagger}$	$-3.0606_{\dagger}$
	HAR$^{\text{LSTM64}}$	$0.5989_{\dagger}$	0.4400	$0.6594_{\dagger}$	$-2.2576_{\dagger}$	$1.0448_{\dagger}$	$-3.0472_{\dagger}$
	LSTM$^{\text{M-B}}$	$0.5764_{\dagger}{}^{*}$	$0.4356_{\dagger}{}^{*}$	$0.6627_{\dagger}{}^{*}$	$-2.2073_{\dagger}{}^{*}$	$1.0508_{\dagger}{}^{*}$	-3.0143^{*}
	LSTM$^{\text{M-L}}$	$0.5567_{\dagger}$	$0.4350_{\dagger}$	$0.6648_{\dagger}$	$-2.1684_{\dagger}$	$1.0413_{\dagger}$	$-3.0117_{\dagger}$
	LSTM$^{\text{LSTM1}}$	$\mathbf{0.5371}_{\dagger}{}^{*}$	$\mathbf{0.4316}_{\dagger}{}^{*}$	$\mathbf{0.6193}_{\dagger}{}^{*}$	$-2.4187_{\dagger}{}^{*}$	$0.9993_{\dagger}{}^{*}$	$-3.1350_{\dagger}{}^{*}$
	LSTM$^{\text{LSTM8}}$	$0.5755_{\dagger}$	$0.4352_{\dagger}$	$0.6631_{\dagger}$	$-2.1964_{\dagger}$	$1.0445_{\dagger}$	$-3.0176_{\dagger}$
	LSTM$^{\text{LSTM64}}$	$0.5712_{\dagger}{}^{*}$	$0.4361_{\dagger}{}^{*}$	$0.6656_{\dagger}{}^{*}$	$-2.2058_{\dagger}{}^{*}$	$1.0504_{\dagger}{}^{*}$	$-3.0169_{\dagger}{}^{*}$
	O$^{\text{M-B}}$	0.7463^{*}	0.4525^{*}	$0.6494_{\dagger}{}^{*}$	$-2.5692_{\dagger}{}^{*}$	$1.1381_{\dagger}{}^{*}$	$-3.0315_{\dagger}{}^{*}$
	O$^{\text{M-L}}$	0.7480^{*}	0.4529^{*}	$0.6515_{\dagger}{}^{*}$	$-2.5576_{\dagger}{}^{*}$	$1.1389_{\dagger}{}^{*}$	$-3.0276_{\dagger}{}^{*}$
	O$^{\text{LSTM1}}$	$0.6701_{\dagger}{}^{*}$	0.4466^{*}	$0.6333_{\dagger}{}^{*}$	$\mathbf{-2.5970}_{\dagger}{}^{*}$	$1.0973_{\dagger}{}^{*}$	$-3.0812_{\dagger}{}^{*}$
	O$^{\text{LSTM8}}$	0.7413^{*}	0.4517^{*}	$0.6519_{\dagger}{}^{*}$	$-2.5700_{\dagger}{}^{*}$	1.1396^{*}	$-3.0323_{\dagger}{}^{*}$
	O$^{\text{LSTM64}}$	0.7457^{*}	0.4516^{*}	$0.6525_{\dagger}{}^{*}$	$-2.5681_{\dagger}{}^{*}$	$1.1456_{\dagger}{}^{*}$	$-3.0240_{\dagger}{}^{*}$
Benchmark Models	ARFIMA	0.6947^{*}	0.4538^{*}	$0.6906_{\dagger}{}^{*}$	$-2.1262_{\dagger}{}^{*}$	$1.0833_{\dagger}{}^{*}$	$-2.9677_{\dagger}{}^{*}$
	HAR	0.6758^{*}	0.4532^{*}	$0.6975_{\dagger}{}^{*}$	-2.1380^{*}	$1.0925_{\dagger}{}^{*}$	$-2.9620_{\dagger}{}^{*}$
	CHAR	$0.5665_{\dagger}{}^{*}$	$0.4384_{\dagger}{}^{*}$	$0.6462_{\dagger}{}^{*}$	$-2.3974_{\dagger}{}^{*}$	$1.0455_{\dagger}{}^{*}$	$-3.0830_{\dagger}{}^{*}$
	SHAR	0.6785^{*}	0.4548^{*}	$0.6976_{\dagger}{}^{*}$	-2.1348^{*}	$1.0928_{\dagger}{}^{*}$	$-2.9613_{\dagger}{}^{*}$
	HARQ	0.6632^{*}	0.4525^{*}	$0.6827_{\dagger}{}^{*}$	-2.1610^{*}	$1.0736_{\dagger}{}^{*}$	$-2.9881_{\dagger}{}^{*}$
	ARFIMA-ln	0.6656^{*}	$0.4394_{\dagger}{}^{*}$	$0.7095_{\dagger}{}^{*}$	-1.9656^{*}	$1.0812_{\dagger}{}^{*}$	$-2.9242_{\dagger}{}^{*}$
	HAR-ln	0.6751^{*}	$0.4396_{\dagger}{}^{*}$	$0.7020_{\dagger}{}^{*}$	$-2.0047_{\dagger}{}^{*}$	$1.0703_{\dagger}{}^{*}$	$-2.9496_{\dagger}{}^{*}$
	CHAR-ln	0.6141^{*}	$0.4373_{\dagger}{}^{*}$	$0.6773_{\dagger}{}^{*}$	$-2.1527_{\dagger}{}^{*}$	$1.0503_{\dagger}{}^{*}$	$-3.0169_{\dagger}{}^{*}$
	SHAR-ln	0.6549^{*}	$0.4389_{\dagger}{}^{*}$	$0.6922_{\dagger}{}^{*}$	$-2.0547_{\dagger}{}^{*}$	$1.0615_{\dagger}{}^{*}$	$-2.9743_{\dagger}{}^{*}$
	HARQ-ln	0.6413^{*}	$0.4403_{\dagger}{}^{*}$	$0.6930_{\dagger}{}^{*}$	$-2.0982_{\dagger}{}^{*}$	$1.0660_{\dagger}{}^{*}$	$-2.9873_{\dagger}{}^{*}$

Note: The RMSE and VaR losses are scaled by 10^3. *Bold face numbers indicate the lowest out of sample loss.* **denotes models for which the* H_0 *of equal forecasting performance of a Binomial test with HAR*^{O} *model as benchmark is rejected at the 5% level and* $_{\dagger}$ *denotes models that are in the Model Confidence set at the 10% level.*

We first consider whether using only the transformed measure for forecasting volatility is fruitful. Table 2 clearly shows that the models that only rely on the transformed measure (labeled O plus the superscript corresponding to the transformation used) are the worst-performing models within their respective blocks in terms of the QLIKE and the squared error loss. These models perform comparably or worse than the alternatives for the VaR loss and the joint loss of VaR and ES. The only exception is when jointly evaluating forecasts of VaR and ES at p = 1%. In this case, these models are the best performing ones, and among them, the model that uses the non-linear transformation via one LSTM cell performs best. The differences in the forecasting performance of the only transformed measure models to those that also use the information on past RV (HAR and LSTM plus superscript) are significant in terms of a binomial test for equal forecasting performance at the 1% level, as Figure 2 shows. Panel A) of the figure displays the test decision when comparing the models that combine the HAR inputs with $\tilde{y}_t$ against the model that only uses $\tilde{y}_t$ (O plus superscript). The x-axis labels specify the type of transformation used to obtain the transformed measure. Panel B) shows the results for only using the most recent information in the transformed measure, i.e., combining the HAR inputs with $\tilde{x}_t$ vs. solely using $\tilde{x}_t$. Panels C) and D) show the results for the case where the HAR inputs are replaced by the output of an LSTM cell applied to the sequence of log RV. All p-values are smaller than 0.01 in all cases. For the QLIKE and the squared error loss, we can conclude that none of the transformations can extract enough information from the HF returns to replace the information on past RV for forecasting volatility. When forecasting the VaR and ES, these models yield results comparable to those of the other models. They outperform the alternative models only for the joint evaluation of the VaR and the ES at the 1% level.



A) HAR-F vs O-F

B) HAR vs O

C) LSTM-F vs O-F

D) LSTM vs O

Figure 2. Results for a Binomial test of equal forecasting performance between the models that use only the transformed measure and their counterpart that use it in combination.

Next, we address whether the -F models (the models that use the output of an LSTM cell applied to the sequence of the transformed measure) yield any differences in the forecasting performance compared to the models that only use the most recent information from the transformation. Figure 3 shows the testing results for differences between a model that only uses the most recent information in the transformed measure against its' -F counterpart. At the 5% level, regardless of the LF input they are combined with, we see no significant differences in the forecasting performance of the MIDAS Beta, the LSTM8, and the LSTM64 transformation models compared to their -F counterparts. However, these differences are significant for the LSTM MIDAS and the LSTM1 transformation. When we only use the transformed measure, the LSTM MIDAS transformation model with full information produces lower average OLIKE and squared error losses. In contrast, the model that only uses the most recent information produces lower average losses when jointly evaluating VaR and ES. For the LSTM1 transformation, the model that only uses the most recent information yields the lower average loss for all loss functions. Combining the transformed measure with other LF information yields the following pattern: For the LSTM MIDAS transformation, where the differences are significant, the -F model gives the lower average losses. For the LSTM1 transformation, using only the most recent information yields lower average losses.

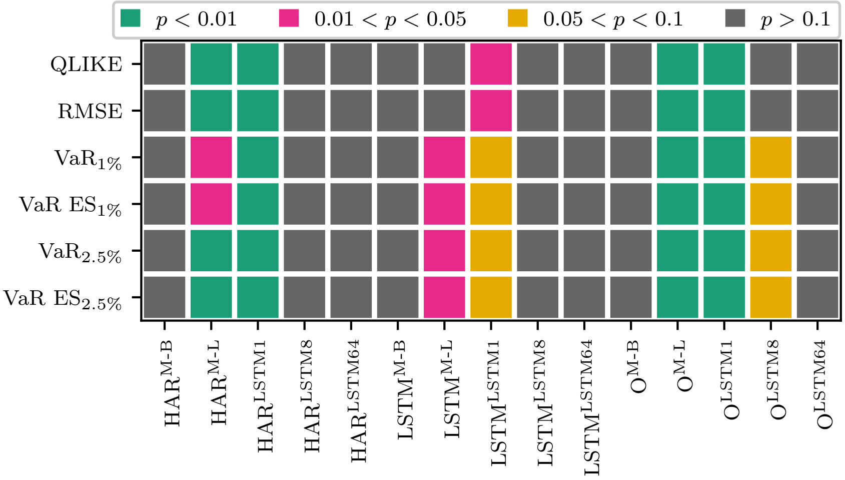


Figure 3. Results for a Binomial test of equal forecasting performance between the full information ($\tilde{y}_t$) and the recent information ($\tilde{x}_t$) models.

Whether there are significant differences in the forecasting performance between the models that use $\tilde{y}_t$ and the models that use $\tilde{x}_t$ is thus case dependent. There are no significant differences for most models and transformations. For the LSTM MIDAS, it depends on whether it is used alone or in combination. In the former case, the -F models produce lower losses when the differences are significant. Using the -F models yields the lower QLIKE and squared error in the latter case. However, using $\tilde{x}_t$ produces lower VaR and ES losses. For the non-linear transformation through one LSTM cell, only applying the transformation to the most recent HF returns yields lower average losses. It seems that, in this case, the more distant information in the HF returns gets accounted for by the RV. However, the most recent HF returns contain information that the lagged RV does not yet capture.

When we use the LSTM MIDAS transformation, it is necessary to use the sequential information in the transformed measure. An alternative explanation for this could be that the -F model introduces additional non-linearity into the transformed measure by applying an LSTM cell to its' sequence. While $\tilde{x}_t$ in the

LSTM MIDAS case is constructed linearly as a weighted sum, $\tilde{y}_t$ is a non-linear transformation of the sequence of that linear measure. So the better forecasting performance of the model that uses $\tilde{y}_t$ for the LSTM MIDAS transformation could be due to that non-linearity. However, the transformation of the HF returns through one LSTM cell is already the output of a non-linear function. Since only the most recent transformed measure is informative for this transformation, it appears that there are no gains from introducing more non-linearity through an LSTM cell on the sequence of transformed measures. Comparing these two against each other, we see that models that use only the most recent non-linear transformed measure produce lower losses than the models that use the LSTM cell applied to the LSTM MIDAS transformation. These differences are significant at the 1% level for all losses (see figures 6 to 11). Thus, the non-linearity within the transformed measure seems to produce more helpful information for forecasting volatility than introducing non-linearity to the transformed measure obtained from the linear method.

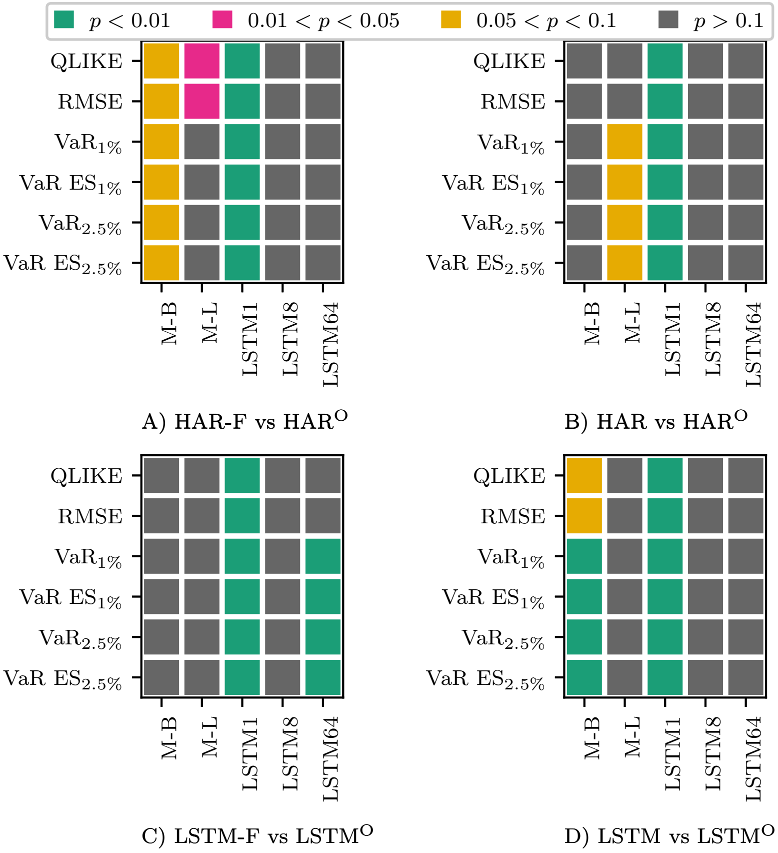


Figure 4. Results for a Binomial test of equal forecasting performance between the models combine the transformed measure and their counterpart that does not use the transformed measure.

Next, we address whether combining the transformed measure with the other LF inputs yields significant gains in forecasting compared to only using the LF inputs. Figure 4 displays the test results. The upper panels, A) and B), show the test results for combining the HAR model inputs with the transformed measures against the model that does not use the transformation. The x-axis labels again indicate the type of transformation. Panel A) shows the results for the -F models and panel B) for the models that only use the most recent information. The lower part of the figure, panels C) and D), display the corresponding results when replacing the HAR inputs with the output of an LSTM cell applied to the sequence of log RV. Combining any of the LF inputs with the LSTM1 transformation yields statistically different forecasts to the models that omit the transformed measure, in any case, and for all losses. In the case of the HAR model inputs, the combined model yields lower losses in both cases. For the LSTM input, the -F model performs worse, whereas the model that only uses the most recent HF information yields lower losses. For the other transformations, the results are case-dependent. When combined with the HAR inputs, the LSTM MIDAS model yields significantly different QLIKE and RMSE losses in the full information case. The HAR$^{\text{M-L}}$-F model yields the lower QLIKE and RMSE losses in these two cases. The differences are not statistically different at the 5% level in the remaining cases. The non-linear transformation with 64 LSTM cells yields statically different results for all losses but the QLIKE and the squared error loss in the full information case. Its' losses are lower than the comparison model in these cases. When using the recent information only the non-linear transformation with 64 LSTM cells does not deliver significantly different results. However, in the full information case, the Beta MIDAS transformation for the VaR and ES for $p = 1\%$ and $p = 2.5\%$ yields losses significantly different from the comparison model's (at the 5% level). In these cases, the Beta MIDAS transformation model yields slightly better results.

Next, we consider whether there are differences in the forecasting performances of the models depending on whether we use the HAR inputs or the output of an LSTM cell applied to past log RV. Table 2 reports a rejection of the H$_0$ of the Binomial test for equal forecasting performance concerning the HAR$^{\text{O}}$ model at the 5% level with an asterisk. From the table, we see that for the LSTM$^{\text{O}}$ model, we can not reject the H$_0$ for any of the loss measures. Thus there are no significant differences between the ANN model that only uses the HAR model inputs and the ANN model that uses an LSTM cell on past log RVs. One difference is that the LSTM$^{\text{O}}$ model is in the MCS at the 10% level for all losses, whereas the HAR$^{\text{O}}$ model is not in the 10% MCS for the squared error loss. For the remaining models that use the transformed measure, the test results are displayed in Figure 5.

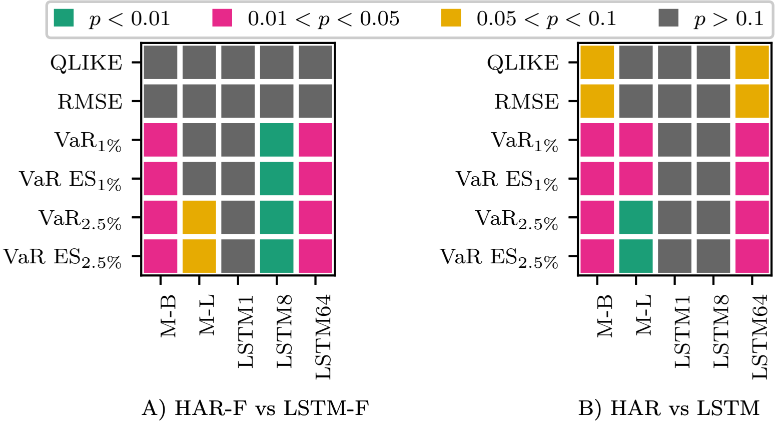


Figure 5. Results for a Binomial test of equal forecasting performance between the models that combine the transformed measure with the HAR inputs and their counterpart that uses the LSTM cell applied to log RV.

According to the figure, we can not reject the H$_0$ at the 5% level for the QLIKE and the squared error loss. For the VaR and ES losses, we reject the H$_0$ at the 5% level for the Beta MIDAS, the LSTM8, and the LSTM64 transformations when using $\tilde{y}_t$. In this case, the models that use the HAR inputs perform better than those using the LSTM input. When using $\tilde{x}_t$, we reject for the Beta and LSTM MIDAS models and the LSTM64 models considering the VaR and ES losses. In all except one case, the models that use the HAR inputs yield the lower out of sample loss. The only exception is the LSTM MIDAS model for the VaR$_{2.5\%}$. In this case, the LSTM inputs are performing marginally better. Overall, it appears that the

HAR model inputs can approximate the long memory in the data to an extent comparable to that of an LSTM cell. However, we want to stress that we did not hunt for an optimal LSTM network architecture for this task. The purpose of the LSTM cell in this application is simply to circumvent the implicit lag order selection of the HAR model. A network of LSTM cells applied to the sequence of log RV as in Bucci (2020) might yield more consistent improvements in the forecasting performance than the HAR model inputs. It is interesting to see that the daily, weekly, and monthly averages used in the HAR model are not only comparable to ARFIMA models in the extent they account for long memory (Corsi, 2009), but also to an LSTM cell.

Among the benchmark models, the CHAR model is performing best. It produces the lowest out of sample loss among the benchmark models for the level and the log of RV. Furthermore, it produces lower losses for all except the RMSE loss in levels than in logs. It is the only benchmark model in the 10% MCS for all losses and it produces lower QLIKE and RMSE losses than the HAR^{O} model, i.e., the HAR model estimated by SGD. These differences are significant at the 5% level. Also, for the other losses except for the $\text{VaR}_{2.5\%}$, it yields lower losses than the HAR^{O}. This is in line with Rahimikia and Poon (2020a), who also find that the CHAR is performing best among the HAR family models. Apart from the CHAR model, the remaining benchmark models cannot perform better than any of the ANN models except those that only use the transformed measure.

We come to a short intermediate conclusion:

1. We found that using only the transformed measure to forecast RV results in higher out-of-sample forecast losses than models that combine the transformed measure with information on past log RV. This holds especially true for the QLIKE and the RMSE error loss. The only exception is the loss of jointly evaluating the VaR and ES at p = 1%.
2. We found that when using linear means to construct the transformed measure, it is crucial to consider the sequential information in the transformed measure. However, this might be due to non-linearity induced through the LSTM cell that we apply to the transformed measure. Therefore, it is sufficient only to use the most recent information when constructing the transformed measure non-linearly. In most cases, this yields better forecasting performance.
3. The non-linear transformation through one LSTM cell seems superior to the other transformations throughout the statistical analysis. The models performing best are those that use this transformation. Further, we have the most statistical evidence for differences in the forecasting performance for these models. We will further investigate this in the following.
4. For the QLIKE and the RMSE loss, there are no statistical differences in the performance of the models that use the HAR inputs and the models that use an LSTM cell applied to log RV. The daily, weekly, and monthly averages of log RV appear to be sufficient to account for the long memory in the data. Especially when combined with the LSTM1 transformed measure, this also holds for all other losses.

This short wrap-up leads to two hypotheses. First, the non-linear transformation through one LSTM cell is superior to all other transformations. Second, the models that combine the transformed measure from such a non-linear transformation with the information on past log RV perform better than all other models. These two models are the two best ranked models for each loss measure, except the joint evaluation of $\text{VaR}_{1\%}$ and $\text{ES}_{1\%}$. We cannot reject that these two models perform equally well for any of the losses (see Figures 6 to 11 in the appendix).

Investigating these hypotheses results in non-pairwise comparisons of the models. Further, the hypotheses are uni-directional, i.e., we are interested in whether these models perform better than the competitors. Thus we can not use a Binomial Test for equal forecasting performance but instead use the test for superior predictive ability (SPA test) of Hansen (2005). We use the *arch* library of Sheppard et al. (2021) to perform the SPA test. When computing the p-values, we use a block bootstrap with the number of bootstrap resamplings set to 1000 and the block length set to 5. The results are not sensitive to the choice of these two values. We also computed the p-values with resamplings set to $3000, 5000, 7000, 9000$ and block lengths of $10, 15, 20, \dots, 95, 100$. The results did not change by much. The SPA test tests whether the expected loss difference between the loss of a candidate and a set of alternative models is smaller or equal to zero. A rejection of the null hypothesis thus means that there is a model among the alternatives performing significantly better than the candidate model.

We start by reporting the p-values of a sequence of SPA tests where we use the LSTM1 transformation models as candidates against the models that use the other transformations. The p-values displayed in table 3 show that, at the 5% level, we can not reject the H_0 of the SPA test in any case. Thus, at the 5% level, the non-linear transformation by one LSTM cell gives forecasting losses smaller or equal to those of all alternative transformations used. This holds for any loss function. At the more conservative 10% level, for the models that use the full information on the transformed measure (upper part of the table) and the joint loss of VaR and ES at 2.5%, we reject the H_0. Thus, for this loss, at least one transformation works better. Overall, however, this evidence supports the first hypothesis of the non-linear transformation through one LSTM cell performing best.

Table 3. p-values of SPA tests for the LSTM1 against the alternative transformations.

	QLIKE	RMSE	$VaR_{1\%}$	VaR $ES_{1\%}$	$VaR_{2.5\%}$	VaR $ES_{2.5\%}$
HAR^{LSTM1}-F	0.182	0.870	0.542	0.385	0.365	0.232
$LSTM^{LSTM1}$-F	0.124	0.135	0.456	0.156	0.162	0.086
O^{LSTM1}-F	0.233	0.724	0.405	0.431	0.603	0.517
HAR^{LSTM1}	0.823	0.937	0.580	0.567	0.543	0.546
$LSTM^{LSTM1}$	0.707	0.967	0.607	0.578	0.548	0.591
O^{LSTM1}	0.543	0.593	0.906	0.789	0.601	0.975

To assess the second hypothesis, we use all models excluding the HAR^{LSTM1} and $LSTM^{LSTM1}$ as the set of alternatives. We then apply the SPA test for each of these two models as candidates. Table 4 displays the p-values of those tests. Again, we see that the null hypothesis that no alternative model performs better than any of the two models under consideration can not be rejected for any loss function. Among the considered models, including the benchmarks for logs and levels, no model performs significantly better than the HAR^{LSTM1} and the $LSTM^{LSTM1}$.

Table 4. p-values of SPA tests for HAR^{LSTM1} and $LSTM^{LSTM1}$ against the remaining models.

	QLIKE	RMSE	$VaR_{1\%}$	VaR $ES_{1\%}$	$VaR_{2.5\%}$	VaR $ES_{2.5\%}$
HAR^{LSTM1}	0.726	0.952	0.975	0.527	0.955	0.996
$LSTM^{LSTM1}$	0.794	0.993	0.970	0.408	0.928	0.973

7 CONCLUSION

This paper aims to forecast the daily volatility utilizing information extracted from the intraday high-frequency (HF) returns through Long Short Term Memory (LSTM) Recurrent Neural Networks (RNN). These structures are flexible in the degree of non-linearity they allow for and capture long persistence in the data. Our method extracts a non-linear, scalar transformation of the HF returns (referred to as transformed HF measure). We use this measure to make one step ahead predictions of the daily volatility. We vary the degree of non-linearity by testing different numbers of LSTM cells in the RNN and find no merits in using more than one LSTM cell for the non-linear transformation. For comparison, we implement two Mixed Data Sampling (MIDAS) approaches to construct the transformation of the HF returns. The MIDAS models obtain weights associated with the HF return and build the transformation as a weighted sum. The first MIDAS model generates weights associated with the lag of an intraday return through an LSTM cell (LSTM MIDAS). The second is an Artificial Neural Network (ANN) implementation of the Beta Lag Polynomial MIDAS (Beta MIDAS) (Ghysels et al., 2004).

To account for dynamics and long memory in the volatility series, we apply an LSTM cell to the sequence of transformed measures. However, we also compare settings where we only use the most recent information from the transformed measure. The reason is that the information from the HF returns might only be ”new” for a short time. Further in the past, it is probably incorporated by the RV estimator. We compare the forecasting performance of models solely based on the transformed HF measure to those of models that only use the information from the past Realized Volatility (RV). Namely, the HAR model and a model that applies an LSTM cell to the sequence of past RVs. The HAR model is one of the most popular models to approximate long memory in the volatility series. LSTM RNNs can account for complex non-linear dependencies in the data and capture long-term dependencies. Our comparison assesses whether the proposed transformation can extract the same or more information from the HF returns than the RV estimator. Finally, we combine the information from the transformed measure and the information from the RV for the forecast. We can thus investigate whether our proposed transformations extract information from the HF returns that is supplementary to the RV information when forecasting volatility.

In an expanding window forecasting exercise on data on the IBM stock, we compare the performance of the models in forecasting out-of-sample volatility. We further compute Value at Risk (VaR) and Expected Shortfall (ES) forecasts based on the volatility forecast. We perform a thorough statistical analysis to identify the source of the improved forecasting performance. Our results on the data set under consideration are fourfold:

First, they show that making volatility forecasts based solely on the transformed HF measure is not fruitful. Neither of the transformations can produce a measure that accounts for the long persistence in the volatility. This result is independent of whether we account for dynamics in the transformed measure or only take the most recent value for the forecast. Interestingly, when jointly evaluating the $\text{VaR}_{1\%}$ and $\text{ES}_{1\%}$ forecasts based on the volatility forecasts, those models perform better than the alternatives. However, for the 2.5% VaR and ES, their performance is again worse or comparable to the alternatives. When forecasting volatility, the transformations we propose are thus unable to extract the same information from the high-frequency returns as the RV estimator. Since the RV estimator ex-post is a consistent estimator of the volatility of a day, it is crucial to take this information into account for the forecasting task. Maybe more complex non-linear ANN structures could extract the same amount of information from the HF returns. However, in our eyes, it is more fruitful to facilitate the forecasting task for the method by using the RV information.

Second, there is no difference in using the sequence of the transformed measure or only the most recent value for most cases. There are significant differences only for the LSTM MIDAS transformation and the non-linear transformation based on one LSTM cell. The LSTM MIDAS transformation excels when we account for dynamics in the transformed measure. In contrast, the non-linear transformation excels when only using the most recent information. Though puzzling at first, this finding is quite intuitive. The LSTM MIDAS builds the transformed measure as a weighted sum. The transformation is thus linear. However, the linearity is insufficient to extract additional information from the HF returns. Therefore the model that only uses the most recent transformed measure, in this case, performs no different than the model that does not use the information. However, we account for dynamics in the sequence of transformed measures by applying an LSTM cell to it. While this circumvents the trouble of lag order selection, it introduces non-linearity in the transformed measure, which likely results in better models' better performance. When we use an LSTM cell to transform the HF returns non-linearly, there are no additional gains from accounting for dynamics in the measure. Accounting for dynamics leads to worse forecasting performance in some cases. We thus conclude that the transformed measure must be non-linear for the transformation to extract additional information from the HF returns. However, allowing for dynamics in the non-linearly obtained transformed measure does not add any additional gains. This coincides with our previous findings indicating that the additional information in the HF returns gets picked up by the RV estimator further in the past. In the short run, though, this information is helpful for the prediction of volatility.

Third, we add to the literature by finding another prove for the improved forecasting performance of ANN models compared to the linear HAR model benchmark. Our models that do not include the transformed measure, i.e., only use either the HAR model inputs or apply an LSTM cell to the sequence of RV, perform significantly differently from the classical HAR model, both estimated in logs and levels. The simple non-linearity we induce through modeling the exponential of the linear combination of past daily, weekly, and monthly averages of the logarithm of RV already is sufficient to outperform the classical linear HAR for both logs and levels. Our results thus add to the evidence provided by, e.g.(Rosa et al., 2014) and (Arnerić et al., 2018). We also apply an LSTM cell to the sequence of the logarithm of RV as an alternative to the HAR inputs. The LSTM cell allows for a high degree of non-linearity, and it captures long memory in the data. We find no significant differences between the LSTM and the HAR input models when predicting volatility in most cases. Our findings thus indicate that, for the simple structures we use, the HAR inputs capture the long persistence in the volatility series equally well as the LSTM cell on the data set under consideration. To some extent, this contradicts the findings of (Bucci, 2020) who finds that gated recurrent ANNs such as LSTM RNNs outperform ANNs that do not account for long memory in the data. However, the author forecasts the logarithm of the square root of monthly RV and not, as in this case, the level of daily RV. When constructing VaR and ES forecasts based on the volatility forecasts, we find significant differences in the performance of the HAR and the LSTM input models, where for these quantities, the HAR input models show better performance.

Fourth, the statistical analysis of the forecasting results pointed towards two hypotheses. First, the non-linear transformation through one LSTM cell is superior to all alternative transformations that we suggest, especially when only accounting for the most recent information in the transformed measure. Through a sequence of tests for superior predictive ability (SPA tests), we find that the non-linear transformation through one LSTM cell outperforms the alternatives. When only considering the most recent HF information, this result holds under conservative choices for the significance level. However, this result only holds for less conservative choices for the significance level (5%) for the setting where we account for dynamics in the transformed measure. So the non-linear transformation through one LSTM cell outperforms the MIDAS alternatives and the alternatives that allow for higher degrees of non-linearity by using a network of LSTM cells. This is very convenient since it circumvents the challenging task of finding the optimal network architecture for the transformation. Second, combining this transformed measure with the information on the past RV yields superior forecasting performance to all other models under consideration. Another sequence of SPA tests shows that the models that augment the information from the log RV with the most recent transformation from one LSTM cell significantly outperform all alternative models, including the benchmarks. When augmented by the most recent transformation from one LSTM cell, there are no significant differences between the model that uses an LSTM on past log RV and the model that uses the HAR. So also in this case, the HAR models' lagged daily, weekly, and monthly averages are approximating the long persistence in the volatility equally well as the LSTM cell.

Our analysis thus directs to a new type of HAR model that augments the classical HAR by a non-linear transformation of the HF returns within a day. These results are in line with the findings of Rahimikia and Poon (2020a), who also find that their proposed HAR model augmented by HF limited order book and news sentiment data shows superior forecasting performance. However, the information we utilize for the augmentation does not stem from an auxiliary source such as news feeds but from the same information used to construct the RV estimator. Our resulting models can outperform some of the most popular benchmark models in the literature, such as ARFIMA models, the HAR, the CHAR, and the HARQ model. A natural extension of the presented work would be to use Bi-Power Variation and Realized Quarticity measures as additional inputs for the forecasting task. One could then assess, whether in this case, there are also gains in the forecasting performance through augmenting this model with the non-linear transformation of the HF returns through one LSTM cell.

Acknowledgements

We want to thank Gerhard Fechteler, Eric Ghysels, Lyudmila Grigoryeva, Roxana Halbleib, Ekaterina Kazak, Ingmar Nolte, Winfried Pohlmeier, the members of the Chair of Econometrics at the Department of Economics at the University of Konstanz, Germany, and two anonymous referees, for helpful comments. All remaining errors are ours. We acknowledge support by the state of Baden-Württemberg through bwHPC. Christian Mücher acknowledges financial support from the German federal state of Baden-Württemberg through a Landesgraduiertenstipendium.

REFERENCES

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Andersen, T. G., Bollerslev, T., and Diebold, F. X. (2007). Roughing It Up: Including Jump Components in the Measurement, Modeling, and Forecasting of Return Volatility. *The Review of Economics and Statistics*, 89(4):701–720.

Andersen, T. G., Bollerslev, T., Diebold, F. X., and Ebens, H. (2001a). The distribution of realized stock return volatility. *Journal of financial economics*, 61(1):43–76.

Andersen, T. G., Bollerslev, T., Diebold, F. X., and Labys, P. (2001b). The distribution of realized exchange rate volatility. *Journal of the American statistical association*, 96(453):42–55.

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

Andersen, T. G., Bollerslev, T., and Meddahi, N. (2004). Analytical evaluation of volatility forecasts. *International Economic Review*, 45(4):1079–1110.

Arnerić, J., Poklepović, T., and Aljinović, Z. (2014). Garch based artificial neural networks in forecasting conditional variance of stock returns. *Croatian Operational Research Review*, pages 329–343.

Arnerić, J., Poklepović, T., and Teai, J. W. (2018). Neural network approach in forecasting realized variance using high-frequency data. *Business Systems Research: International journal of the Society for Advancing Innovation and Research in Economy*, 9(2):18–34.

Audrino, F. and Knaus, S. D. (2016). Lassoing the har model: A model selection perspective on realized volatility dynamics. *Econometric Reviews*, 35(8-10):1485–1521.

Baştürk, N., Schotman, P. C., and Schyns, H. (2021). A neural network with shared dynamics for multi-step prediction of value-at-risk and volatility. *Available at SSRN 3871096*.

Barndorff-Nielsen, O. E., Kinnebrock, S., and Shephard, N. (2010). Measuring Downside Risk - Realized Semivariance. In Bollerslev, T., Russel, J., and Watson, M., editors, *Volatility and Time Series Econometrics: Essays in Honor of Robert F. Engle*, pages 117–136. Oxford University Press.

Barndorff-Nielsen, O. E. and Shephard, N. (2002a). Econometric analysis of realized volatility and its use in estimating stochastic volatility models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 64(2):253–280.

Barndorff-Nielsen, O. E. and Shephard, N. (2002b). Estimating quadratic variation using realized variance. *Journal of Applied econometrics*, 17(5):457–477.

Barndorff-Nielsen, O. E. and Shephard, N. (2004). Power and Bipower Variation with Stochastic Volatility and Jumps. *Journal of Financial Econometrics*, 2(1):1–37.

Baruník, J. and Křehlík, T. (2016). Combining high frequency data with non-linear models for forecasting energy market volatility. *Expert Systems With Applications*, 55:222–242.

Bengio, Y., Simard, P., Frasconi, P., et al. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166.

Bollerslev, T. (1986). Generalized Autoregressive Conditional Heteroscedasticity. *Journal of Econometrics*, 31.

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

Brownlees, C. T. and Gallo, G. M. (2010). Comparison of volatility measures: a risk management perspective. *Journal of Financial Econometrics*, 8(1):29–56.

Bucci, A. (2020). Realized volatility forecasting with neural networks. *Journal of Financial Econometrics*, 18(3):502–531.

Chollet, F. et al. (2015). Keras. `https://keras.io`.

Christensen, K., Siggaard, M., and Veliyev, B. (2021). A machine learning approach to volatility forecasting. *CREATES Research Paper 2021-03*, 3.

Corsi, F. (2009). A Simple Approximate Long-Memory Model of Realized Volatility. *Journal of Financial Econometrics*, 7(2):174–196.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314.

Donaldson, R. and Kamstra, M. (1997). An artificial neural network-garch model for international stock return volatility. *Journal of Empirical Finance*, 4(1):17–46.

Engle, R. (1982). Autoregressive Conditional Heteroskedasticity with Estimates of the Variance of United Kingdom Inflation. *Econometrica*, 50:987–1007.

Fissler, T. and Ziegel, J. F. (2016). Higher order elicitability and osband's principle. *The Annals of Statistics*, 44(4):1680–1707.

Franke, J. and Diagne, M. (2006). Estimating market risk with neural networks. *Statistics & decisions*, 24(2):233–253.

Franke, J., Härdle, W. K., and Hafner, C. M. (2019). *Neural Networks and Deep Learning*, pages 459–495. Springer International Publishing, Cham.

Ghysels, E., Santa-Clara, P., and Valkanov, R. (2004). The midas touch: Mixed data sampling regression models.

Giordano, F., La Rocca, M., and Perna, C. (2012). *Nonparametric estimation of volatility functions: Some experimental evidences*, pages 229–236. Springer Milan, Milano.

Gneiting, T. (2011). Making and evaluating point forecasts. *Journal of the American Statistical Association*, 106(494):746–762.

Granger, C. W. and Newbold, P. (1976). Forecasting transformed series. *Journal of the Royal Statistical Society: Series B (Methodological)*, 38(2):189–203.

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

Hajizadeh, E., Seifi, A., Zarandi, M. F., and Turksen, I. (2012). A hybrid modeling approach for forecasting the volatility of s&p 500 index return. *Expert Systems with Applications*, 39(1):431–436.

Hansen, P. R. (2005). A test for superior predictive ability. *Journal of Business & Economic Statistics*, 23(4):365–380.

Hansen, P. R., Lunde, A., and Nason, J. M. (2011). The Model Confidence Set. *Econometrica*, 79(2):453–497.

Hochreiter, S. (1991). Untersuchungen zu dynamischen neuronalen netzen. *Diploma, Technische Universität München*, 91(1).

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

Hornik, K., Stinchcombe, M., and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):551–560.

Hyndman, R. J. and Khandakar, Y. (2008). Automatic time series forecasting: the forecast package for R. *Journal of Statistical Software*, 26(3):1–22.

Jordan, M. I. (1997). Chapter 25 - serial order: A parallel distributed processing approach. In Donahoe, J. W. and Dorsel, V. P., editors, *Neural-Network Models of Cognition*, volume 121 of *Advances in Psychology*, pages 471 – 495. North-Holland.

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

Kristjanpoller, W., Fadic, A., and Minutolo, M. C. (2014). Volatility forecast using hybrid neural network models. *Expert Systems with Applications*, 41(5):2437–2442.

Li, S. Z. and Tang, Y. (2020). Forecasting realized volatility: An automatic system using many features and many machine learning algorithms. *Available at SSRN*.

Maechler, M. (2020). *fracdiff: Fractionally Differenced ARIMA aka ARFIMA(P,d,q) Models*. R package version 1.5-1.

Miura, R., Pichl, L., and Kaizoji, T. (2019). Artificial neural networks for realized volatility prediction in cryptocurrency time series. In Lu, H., Tang, H., and Wang, Z., editors, *Advances in Neural Networks – ISNN 2019*, pages 165–172, Cham. Springer International Publishing.

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

Patton, A. J. and Sheppard, K. (2015). Good Volatility, Bad Volatility: Signed Jumps and The Persistence of Volatility. *The Review of Economics and Statistics*, 97(3):683–697.

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

Rahimikia, E. and Poon, S.-H. (2020b). Machine learning for realised volatility forecasting. *Available at SSRN 3707796*.

Rosa, R., Maciel, L., Gomide, F., and Ballini, R. (2014). Evolving hybrid neural fuzzy network for realized volatility forecasting with jumps. In *2014 IEEE Conference on Computational Intelligence for Financial Engineering & Economics (CIFEr)*, pages 481–488. IEEE.

Ruiz, E. (1994). Quasi-Maximum Likelihood Estimation of Stochastic Volatility Modles. *Journal of Econometrics*, 63:289–306.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, 323(6088):533–536.

Sadhwani, A., Giesecke, K., and Sirignano, J. (2021). Deep learning for mortgage risk. *Journal of Financial Econometrics*, 19(2):313–368.

Schäfer, A. M. and Zimmermann, H. G. (2006). Recurrent neural networks are universal approximators. In Kollias, S. D., Stafylopatis, A., Duch, W., and Oja, E., editors, *Artificial Neural Networks – ICANN 2006*, pages 632–640, Berlin, Heidelberg. Springer Berlin Heidelberg.

Sheppard, K., Khrapov, S., Lipták, G., mikedeltalima, Capellini, R., alejandro cermeno, Hugle, esvhd, Fortin, A., JPN, Judell, M., Li, W., bot, S., Adams, A., jbrockmendel, Rabba, M., Rose, M. E., Tretyak, N., Rochette, T., Leo, U., RENE-CORAIL, X., Du, X., and syncoding (2021). bashtage/arch: Release 5.1.0.

Sizova, N. (2011). Integrated variance forecasting: Model based vs. reduced form. *Journal of Econometrics*, 162(2):294–311.

Soudry, D., Hoffer, E., Nacson, M. S., Gunasekar, S., and Srebro, N. (2018). The implicit bias of gradient descent on separable data. *The Journal of Machine Learning Research*, 19(1):2822–2878.

Taylor, S. J. (1986). *Modeling Financial Time Series*. Wiley. Reprinted in World Scientific, (2008).

Vortelinos, D. I. (2017). Forecasting realized volatility: Har against principal components combining, neural networks and garch. *Research in international business and finance*, 39:824–839.

White, H. (1988). Economic prediction using neural networks: The case of ibm daily stock returns. In *ICNN*, volume 2, pages 451–458.

APPENDIX

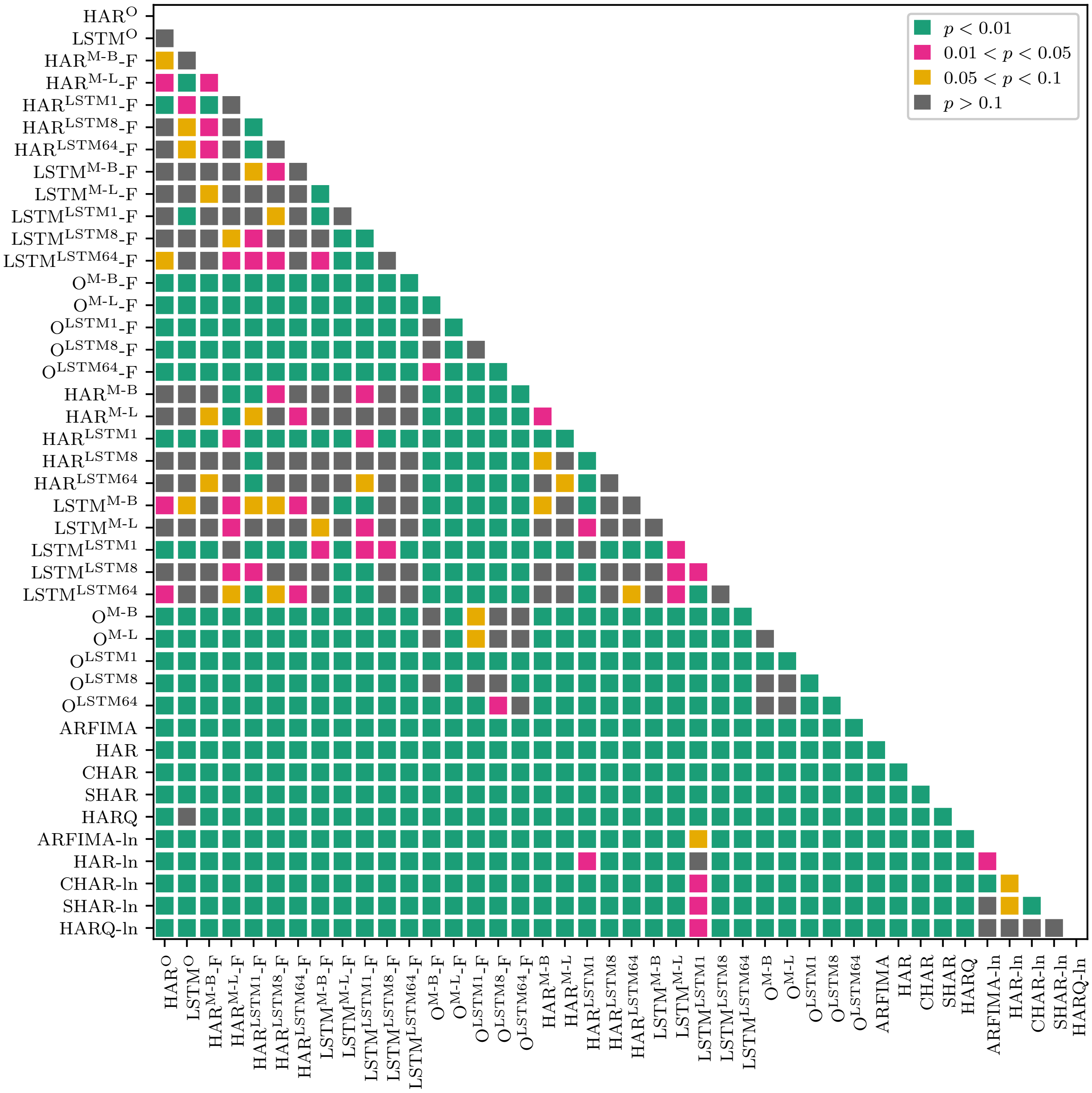


Figure 6. p-values of a Binomial test for equal forecasting performance for the QLIKE loss depicted.

We test the models in the rows against the model specified by the column. Since the test result is symmetric, we report the lower triangle. A rejection of the H_0 indicates significant differences in the forecasting performance of the two models.

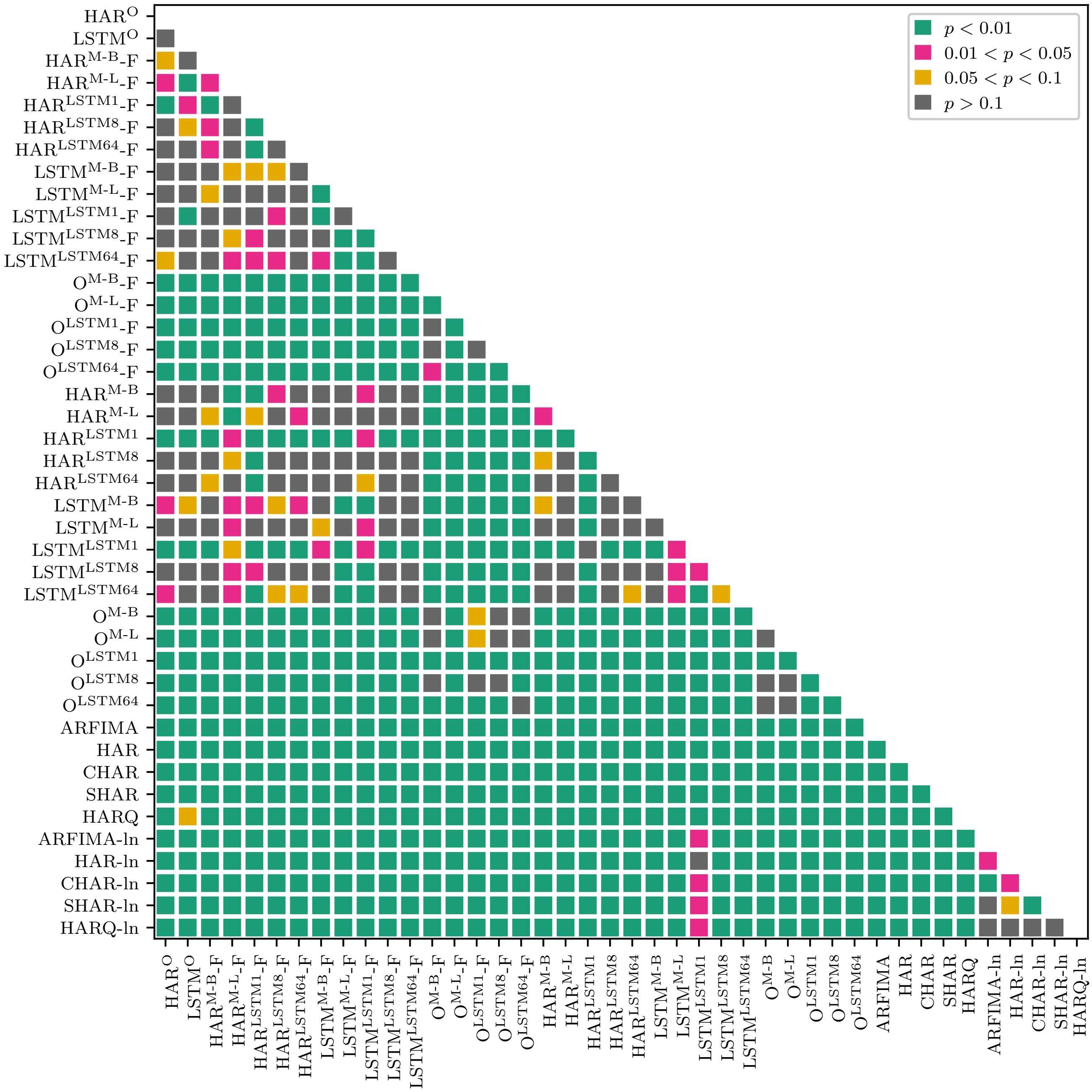


Figure 7. p-values of a Binomial test for equal forecasting performance for the squared error loss depicted.

We test the models in the rows against the model specified by the column. Since the test result is symmetric, we report the lower triangle. A rejection of the H_0 indicates significant differences in the forecasting performance of the two models.

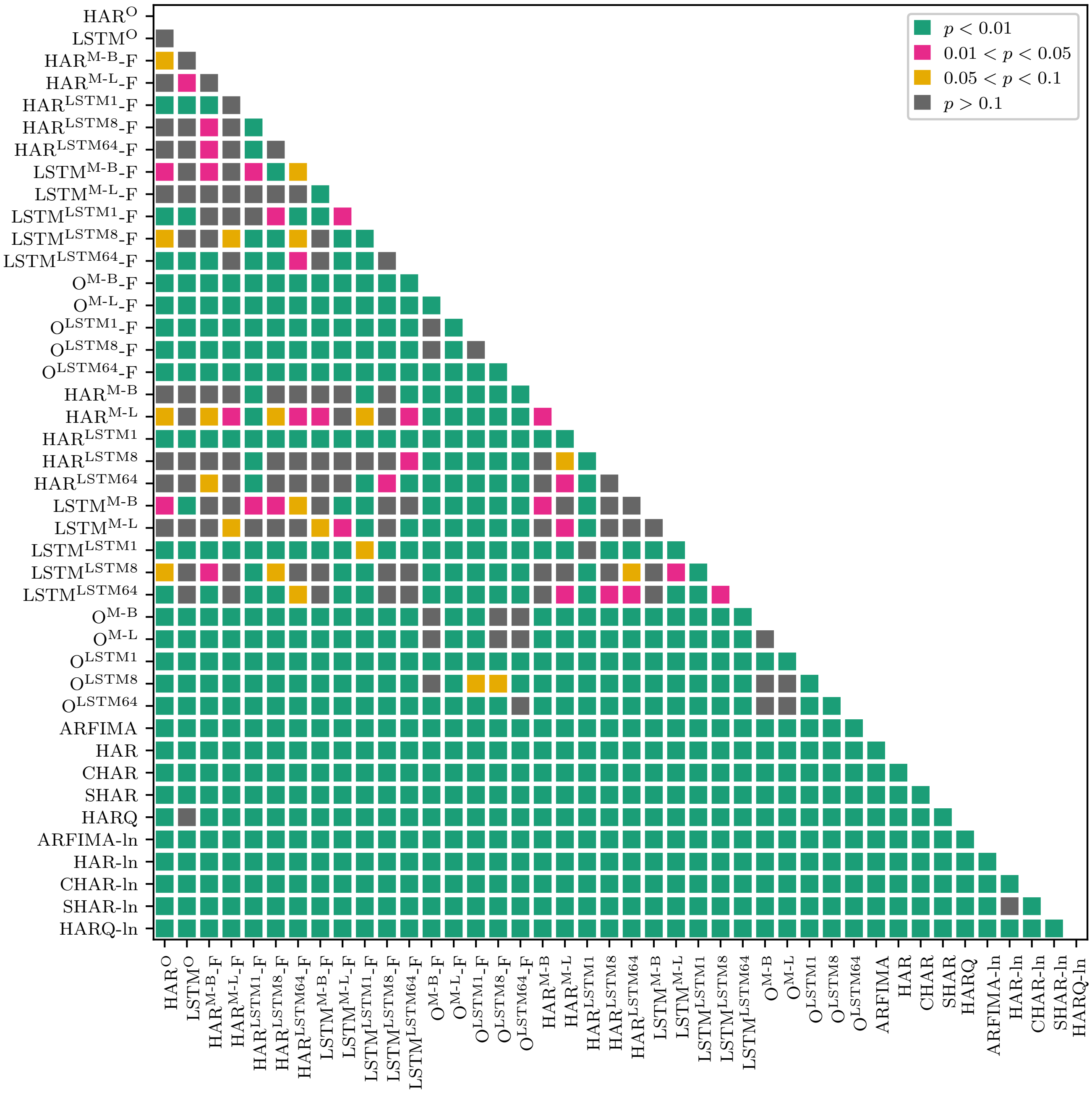


Figure 8. p-values of a Binomial test for equal forecasting performance for the VaR$_{1\%}$ score loss depicted.

We test the models in the rows against the model specified by the column. Since the test result is symmetric, we report the lower triangle. A rejection of the H_0 indicates significant differences in the forecasting performance of the two models.

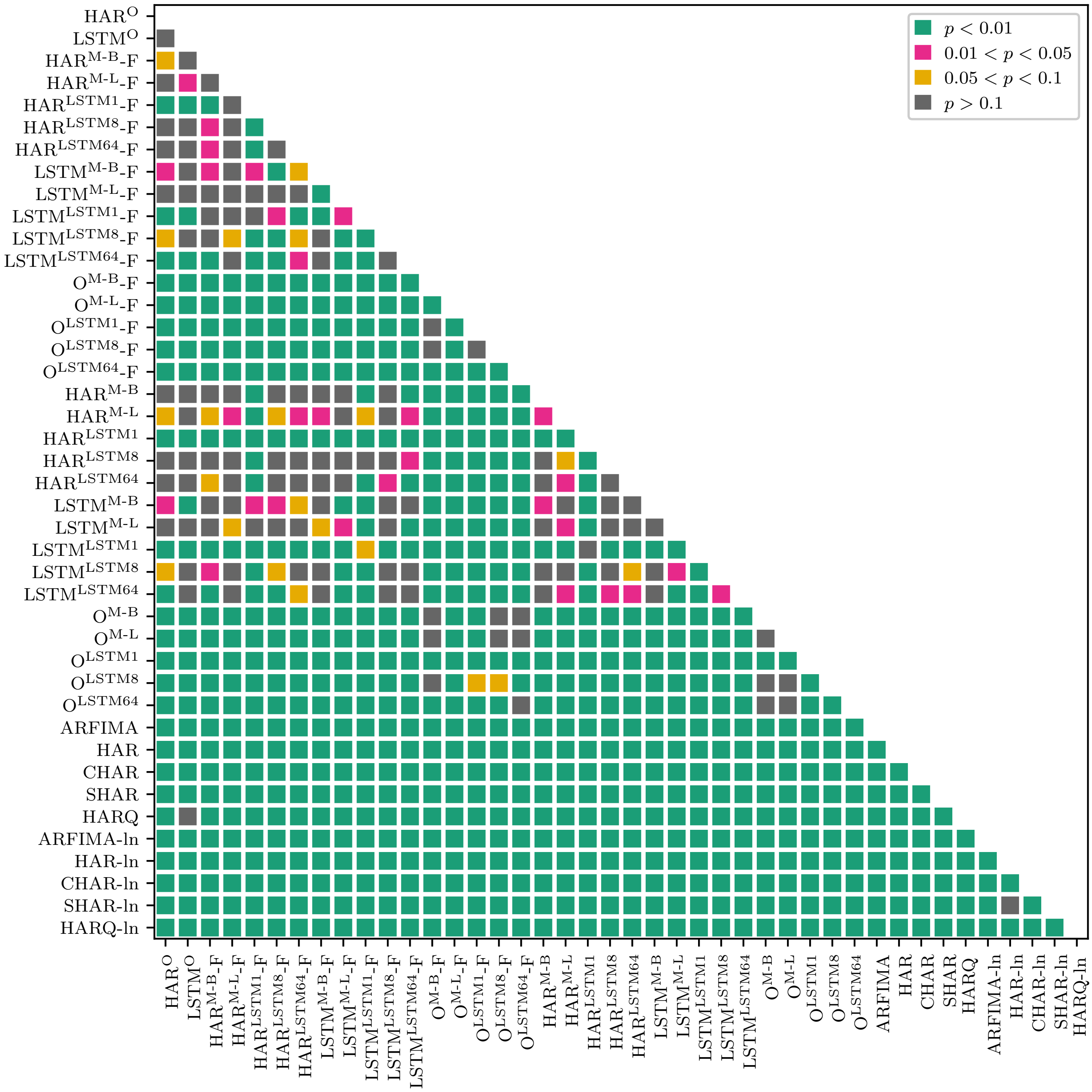


Figure 9. p-values of a Binomial test for equal forecasting performance for the VaR ES$_{1\%}$ score loss depicted.

We test the models in the rows against the model specified by the column. Since the test result is symmetric, we report the lower triangle. A rejection of the H_0 indicates significant differences in the forecasting performance of the two models.

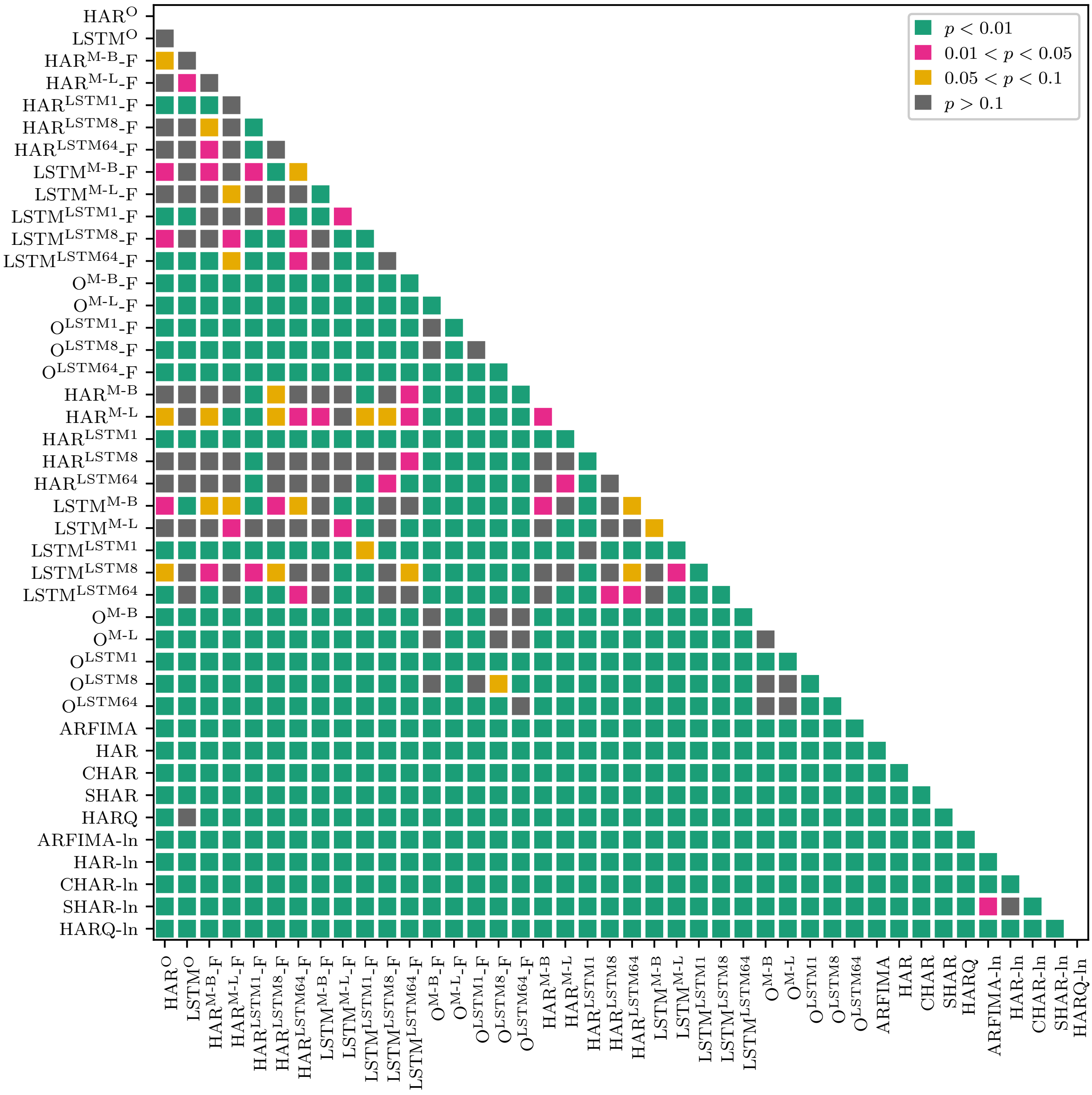


Figure 10. p-values of a Binomial test for equal forecasting performance for the VaR$_{2.5\%}$ score loss depicted.

We test the models in the rows against the model specified by the column. Since the test result is symmetric, we report the lower triangle. A rejection of the H_0 indicates significant differences in the forecasting performance of the two models.

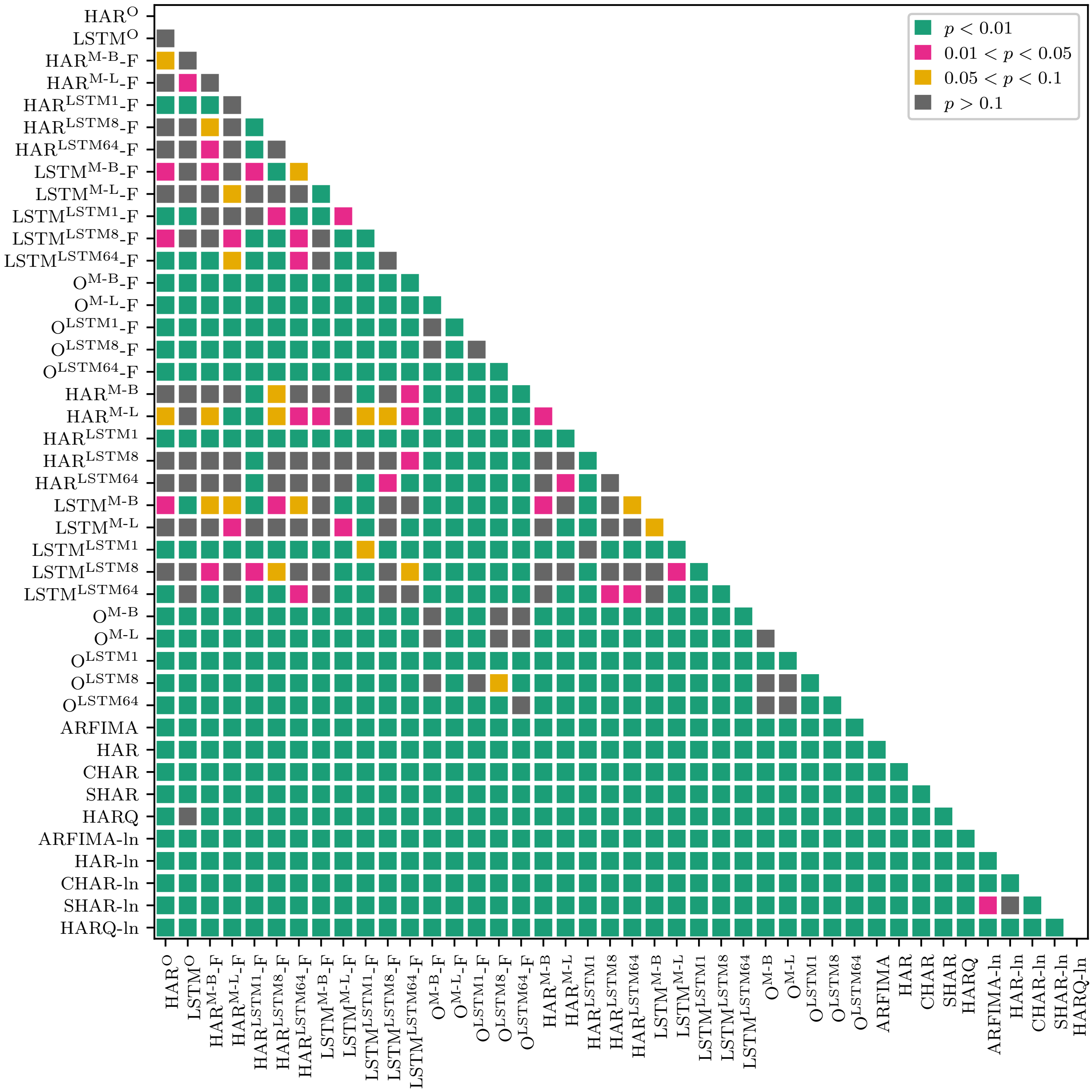


Figure 11. p-values of a Binomial test for equal forecasting performance for the VaR $ES_{2.5\%}$ score loss depicted.

We test the models in the rows against the model specified by the column. Since the test result is symmetric, we report the lower triangle. A rejection of the H_0 indicates significant differences in the forecasting performance of the two models.