Volatility Forecasting with Artificial Neural Networks: Can we trust them?

Carl Oscar Dannström & Axel Broang

Stockholm Business School

Master’s Degree Thesis 30 HE credits

Subject: Asset Pricing - Finance

Program: Master's Programme in Banking and Finance 120 HE credits

Spring semester 2023

Supervisor: Associate Professor Håkan Jankensgård
Acknowledgments

We would first like to thank our thesis supervisor, Associate Professor Håkan Jankensgård of Stockholm Business School at Stockholm University. Prof. Jankensgård came with many valuable insights to the thesis that we could not have managed without and was very supportive throughout the entire writing process. Lastly, we would like to take this opportunity to express our sincere gratitude to all participants in our seminar group for their active participation and thoughtful feedback throughout the semester, considerably contributing to the improvement of our thesis.

Authors

Carl Oscar Dannström & Axel Broang
Abstract

This thesis investigates how two types of artificial neural network models (ANN), feedforward neural networks (FNN) and long short-term memory (LSTM), used for realized volatility (RV) forecasting, perform during high and low volatility regimes in comparison to the heterogeneous autoregressive (HAR) model. This is done for 23 stocks, constituents of the Swedish index OMXS30, between the 8th of February 2010 and the 31st of January 2022 using ten exogenous and three endogenous input variables. We find the ANNs generally superior to the HAR model, but also a lack of robustness when investigating ANNs performance in different volatility regimes. The study shows that HAR and ANN models have differing forecasting performances across the volatility range and that the variation is dependent on the regularization regime in place. Where lower regularization supports enhanced accuracy during high-volatility days while higher regularization promotes performance during low-volatility days. In addition, the existence of a trade-off between model complexity and performance during high versus low volatility for LSTM models are confirmed, and it is concluded that this relation is conditioned upon the regularization.
# Table of Content

1. **Introduction**  
2. **Literature Review and Theoretical Approach**  
   2.1 Volatility Prediction  
   2.2 Artificial Neural Networks  
   2.3 Model Tuning  
      2.3.1 Optimization  
      2.3.2 Regularization  
   2.4 Realized Volatility Forecasting with ANNs  
   2.5 Hypothesis  
3. **Empirical Setting & Research Design**  
   3.1 Data  
      3.1.1 Sample Selection  
      3.1.2 Calculations and Transformation  
   3.2 Research Design  
      3.2.1 HAR  
      3.2.2 HAR-X  
      3.2.3 Training and Out-Of-Sample Prediction for ANN Models  
      3.2.4 FNN Models  
      3.2.5 LSTM Models  
   3.3 Data Analysis Method  
      3.3.1 Comparing Performance  
      3.3.2 Sample Split Comparison  
      3.3.3 Complexity to Accuracy Trade-Off  
   3.4 Philosophy of Social Science and Research Ethics  
4. **Findings & Analysis**  
   4.1 Out-of-Sample Analysis – FNN  
      4.1.1 FNN with Low Regularization  
      4.1.2 FNN with High Regularization  
   4.2 Out-of-Sample Analysis – LSTM  
      4.2.1 LSTM with Low Regularization  
      4.2.2 LSTM with High Regularization  
   4.3 Main Findings  
5. **Discussion**  
   5.1 Discussion of Findings  
   5.2 Limitations  
6. **Conclusion**  
References
1. Introduction

Advancements in machine learning and AI have been rapid over the last decade. The recent release of the autoregressive pre-trained neural network natural language model, ChatGPT, has amazed the world with its capabilities and accuracy.

An artificial neural network is an algorithm inspired by the biological neural networks making up the brains of living organisms. Artificial neural networks are made out of layers of artificial neurons connected in a network. Each neuron in an artificial neural network receives input signals from other neurons or an external source, processes the input signals using a mathematical function and then passes on the output signal to other neurons in the network. The connections between the neurons have weights that determine the strength of the signal passing through them. The weights are estimated or “learned” from a training data set by minimizing a pre-specified loss function. Hence, the algorithm can be seen as a functional summary of the training data. The artificial neural network's capacity to represent the training data, referred to as the network's complexity, can be adjusted by increasing or decreasing the number of neurons. An artificial neural network with enough neurons can represent the data by approximating almost any continuous functional form, while networks utilizing few neurons are limited to fewer functions.

Artificial neural network models have been successfully applied in many fields, and research in financial time series prediction is no exception. These findings are somewhat puzzling in the traditional econometric framework since one of its cornerstones is the concept of parsimony (Box et al., 2015). However, artificial neural network models are far from parsimonious, often containing hundreds of parameters to estimate. Critics further claim that artificial neural networks are problematic since they lack an apparent theoretical motivation for use in finance, interpretability, and transparency, and have a tendency of overfitting (Brooks, 2019). Advocates instead point out the nonlinear relations in financial time series data and the complex interchangeability between variables, which are not captured by the more traditional linear models (Dixon et al., 2020; Fernandes et al., 2014).

Still, several contemporary studies have formed a convincing picture of artificial neural networks' superiority on subjects ranging from default prediction (Ciampi & Gordini, 2013; Kvamme et al., 2018) to asset pricing (Gu et al., 2020; Gu et al., 2021), stock return forecasting (Jiang et al., 2020) and implied volatility directional forecasting (Vrontos et al., 2021).
Utilizing off-the-shelf artificial neural network models has also emerged as a successful tool for improving forecasts of realized volatility, outperforming more traditional econometric methods, as shown by Liu (2019), Bucci (2020), Christensen et al. (2022), Lu et al. (2022) and Rahimikia and Poon (2020b) amongst others. This relatively new literature seems promising, and forecasting improvements are critical to financial institutions, risk managers, and investors.

However, recent studies have presented concerning results regarding volatility predictions utilizing artificial neural networks in the extremes, which is of importance for risk management. Rahimikia and Poon (2020b) find the superior volatility prediction accuracy of artificial neural network models to be driven by outperformance during low volatility, while linear models outperform during high volatility. More nuanced results are reported by Christensen et al. (2022) suggesting that artificial neural network volatility forecasts underperform during extremely low and extremely high volatility while outperforming during medium-to-high volatility, as compared to a classic econometric benchmark model. These findings indicate a severe disadvantage of artificial neural network models since volatility forecasting could be regarded as of greater importance during high volatility conditions compared to moderate volatility conditions. This behavior could have a logical explanation based on the standard practice of fitting artificial neural networks to training data by adding an extra penalty when minimizing the mean squared error, called regularization. Since only a small fraction of the total sample consists of high-volatility days, the algorithm could prioritize high accuracy in most samples when low volatility exceeds the performance during high-volatility days. This could lead to the exclusion of the ability to predict outliers. Christensen et al. (2022) propose another explanation, that a lower forecasting performance during high volatility is due to overfitting when volatility rises. This would suggest that a better performance can be achieved during high volatility if the regularization of the model is increased.

Notably, the property of artificial neural networks to underperform when volatility is high has not been consistently reported in previous studies. Contradicting evidence is for example presented by Lu et al. (2022), finding artificial neural networks to outperform linear models during both high and low volatility, with the greatest performance in high volatility. Hence, it is still an open question under which conditions artificial neural networks generate reliable predictions, and if this can be adjusted for by adjusting the regularization.
Further interesting findings by Rahimikia and Poon (2020b) suggest that artificial neural network models with more complex structures perform better during high volatility than less complex ones. They also find that models with modestly complex structures perform better during normal volatility days, indicating a trade-off between model complexity on one hand and accuracy in high versus low volatility on the other. These findings get support from Christensen et al. (2022), as they suggest that nonlinearity intensifies with increasing volatility, necessitating the use of more complex models with greater capacity during such periods. Additionally, Christensen et al. (2022) state that the presence of nonlinearity is noncrucial during periods of low volatility. Consequently, it is suggested that more complex models perform suboptimally during low volatility, as they introduce additional variability into the predictions (Goodfellow et al., 2016).

Based on the previous research, we have identified two gaps; Firstly, contradicting results from previous studies regarding the relative performance between traditional linear models and artificial neural network models during high compared to low-volatility days. Secondly, the lack of research on the relationship between the complexity of artificial neural network models and their ability to forecast volatility under different volatility levels. In light of this, we propose the following research questions:

(i) Do traditional linear and artificial neural network models have differing forecasting performances across various levels of volatility?

(ii) Is there a correlation between the complexity of artificial neural network models and their predictive accuracy during high versus low levels of volatility?

(iii) Are the two phenomena described in (i) and (ii) conditioned on the regularization regime applied when estimating the models?

Our contribution to the current literature is to shed light on how the structure and regularization regime affect forecasting outcomes of artificial neural networks when predicting realized volatility.

To investigate these questions, we sample data from a NASDAQ HFT data set provided by the Swedish House of Finance, including 23 OMX Stockholm 30 (OMXS30) stock index constituents with continuous trading between February 8, 2010, and January 28, 2022. Following the methodology of Andersen et al. (2001), we calculate the realized volatility for the popular Heterogeneous AutoRegressive Realized Volatility (HAR) model first described
by Corsi (2009), which will represent the linear models. In addition, we collect and calculate several financial variables using data gathered from Refinitiv Eikon. We construct nine different structures of artificial neural network models, inspired by the economic literature, such as Christensen et al. (2022) and Rahimikia and Poon (2020b). Four of these models come from a family of contemporaneous artificial neural networks and five from a sequential one. We apply two distinct regularization regimes to each family of models. All artificial network models are constructed and estimated using Keras, a deep learning library written in Python, and executed on top of the TensorFlow API (Keras, n.d.; TensorFlow, n.d.). We followed Christensen et al. (2022) and compared the out-of-sample mean squared errors across all models. The out-of-sample data is partitioned into high- and low-volatility days and into ten deciles with increasing volatility to facilitate comparison across the sample range.

Our study confirms that artificial neural networks outperform linear econometric models in forecasting daily realized volatility. We further conclude that artificial neural networks do vary in their forecasting ability depending on the level of volatility and that they in general perform best during high-volatility days, while the magnitude depends on the model family.

However, this relationship is reduced when increasing the regularization, suggesting that the increased performance during high volatility is due to an increase of nonlinearities with rising volatility. This is further confirmed by the detection of a correlation between model complexity and forecasting accuracy on high and low-volatility days. Nevertheless, this correlation is in turn conditional on the regularization regime in place.

Our findings suggest that artificial neural networks can be customized to specialize in specific volatility conditions, catering to the needs and objectives of different stakeholders and paving the way for future research on the subject.

The thesis is structured as follows: In section two the theoretical framework and the current state of the literature are reviewed. Section three describes the data used, the calculations conducted and the methodology applied. The fourth section presents our empirical results and main findings, while section five contains a discussion of our findings in the context of current literature. Lastly, we end the thesis with our conclusion and suggested topics for future research.
2. Literature Review and Theoretical Approach

2.1 Volatility Prediction

Predicting asset price volatility is critical for investors, traders, and financial institutions due to its close connection to financial risk, and modeling. Forecasting volatility poses several particular challenges due to the nonlinear nature of volatility data (Brooks, 2019). Engle (1982) overcame the challenges of volatility clustering and leptokurtosis with the AutoRegressive Conditionally Heteroscedastic (ARCH) model, using the lagged error terms from a mean equation as independent variables. Due to difficulties in specifying the lag length in the ARCH model, Bollerslev (1986) and Taylor (1986) independently proposed the Generalized ARCH (GARCH) model, with the contribution of including previous lags of the variance, effectively comprising infinite past lag values of the squared error from the mean equation. However, the GARCH model assumes symmetric return variance, while observations show that negative returns impact future variance more than positive. This phenomenon was captured by modifications such as the Exponential GARCH (EGARCH) proposed by Nelson (1991) and the GJR model suggested by Glosten et al. (1993).

Alongside the ARCH-type models, several stochastic volatility models have been developed like those proposed by Hull and White (1987), Chesney and Scott (1989), and Heston (1993). Taylor (1994) recognized these different types of models and showed that both types yielded equally satisfactory results. Andersen et al. (2001) argued that both of these two types of models were misspecified, as only one model could be rightly specified at a time, hence both had to be misspecified. This conclusion led Andersen et al. (2001) to propose an estimate of daily volatility, the Realized Volatility (RV), model-free and approximately free from estimation errors. They showed that the RV estimate makes it possible to make volatility forecasts using standard statistical models without making assumptions about the underlying return-generating process. Andersen et al. (2001) also showed RV to be highly serially correlated at a monthly level, contradicting the general belief at the time (Christoffersen & Diebold, 2000).

Müller et al. (1993) computed 20-minute autocorrelation of absolute returns for the USD/DEM FX rate, using Mandelbrot's (1963) fractal modeling, finding anomalies in one-day and one-week lags. These findings supported their Heterogeneous Market Hypothesis (HMH), which states that market participants differ in investment horizon and thus differ in their reaction time...
to news. Therefore volatility consists of short-term, medium-term, and long-term components. Inspired by Andersen et al. (2001) and the HMH, Corsi (2009) formulated the basis for the Heterogeneous AutoRegressive Realized Volatility (HAR) model. The formulation was simple and parsimonious, including only three terms; daily RV, weekly moving average of daily RV, and monthly moving average of daily RV. Corsi (2009) examined the USD/CHF FX rate return volatility, comparing GARCH and fractal models to HAR, concluding that HAR had superior performance. The main driver of the HAR model was the ability to model both short and long dependencies on past values. Numerous variations of the HAR model have been proposed, and it is arguably one of the most widely used models for volatility forecasting (Bollerslev et al., 2016; Corsi et al., 2010; Patton & Sheppard, 2015).

In addition to autoregressive dependencies on lags, several studies have proposed exogenous factors to affect future volatility. Implied volatility is a widely used proxy for future volatility, a measure of expected volatility backed out from option prices (Poon & Granger, 2005). Limit order book variables, backed by market structure theories like Glosten and Milgrom (1985), have been shown empirically to play a part in volatility forecasting (Bollerslev & Melvin, 1994; Haugom et al., 2014). Further, volatility spillover effects between equity markets are determined to be significant (Liu & Pan, 1997; Xiao & Dhesi, 2010) and hence might have properties to influence future volatility, as well as other asset-specific variables, like the momentum factor, which seem to have predictive properties (Huang et al., 2013).

### 2.2 Artificial Neural Networks

In its simplest form, an Artificial Neural Network (ANN) is a mathematical algorithm taking inputs and transforming them into outputs. It is often referred to as a simplified model of the neural networks which are the building blocks of a biological brain (Haykin, 2009). A schematic illustration of the building blocks, the artificial neurons often called nodes, is displayed in Figure 1. The algorithm takes inputs, multiplies them with a weight, and sums the results with a constant term called the bias. This sum is then used as input in the activation function which delivers the output from the node.
Figure 1: Artificial neuron

![Figure 1: Artificial neuron](image)

Schematic illustration of an artificial neuron where $x_i$ represents the input variables, $w_{kj}$ the weights, $b_k$ the bias (or constant), $\Sigma$ the summing junction, $\varphi(\cdot)$ the activation function and $y_k$ the output variable. The figure is inspired by Haykin (2009).

The complete ANN is constructed by combining a set of input units that take input values, referred to as the input layer, and feeding the resulting output through a set of hidden neurons, called hidden layers. The processed values are then delivered to the output units, which return the output values. A schematic illustration of a simple ANN is depicted in Figure 2. This kind of ANN is called a Feedforward Neural Network (FNN), as the calculations start in the input layer and feed-forward toward the output.

Figure 2: Feedforward neural network

![Figure 2: Feedforward neural network](image)

Notes: Schematic illustration of a two-layered artificial neural network where $x_i$ represents the input variables, $y_k$ the output variable, and each circle an artificial neuron, depicted in Figure 1. The figure is inspired by Christensen et al. (2022).
The ANN architecture is built by combining an input layer which takes a number of inputs (i), adding hidden layers (h) with a specified number of nodes in each layer, and attaching an output layer (o) with a certain number of outputs. This structure can be written (i, h, o). Hence a neural network described (6, 8, 4, 1) has six input variables, a first hidden layer with eight nodes, a second hidden layer with four nodes, and an output layer with one output value.

A complex enough ANN, using a high enough number of nodes, is sufficient for approximating any given continuous function (Haykin, 2009), following Cybenko's (1989) universal approximation theorem. The approximation is estimated by changing the weights and the bias until a satisfactory approximation is reached. This is done by iterative optimization using in-sample data, often called training. The goal is to find the optimal weights and biases that produce the smallest prediction error (Dixon et al., 2020).

When training ANNs, this is typically done using the local optimization method gradient descent (Goodfellow et al., 2016), first described by Cauchy (1847). The ANN can then be seen as a statistical model constructed through iterative parameter optimization, hence a functional summary of the training data (Swingler, 1996).

An important aspect when optimizing ANNs using gradient descent or other local optimization methods is that there is no guarantee of finding the true global optimization solution. Moreover, the initial guess or starting value is critical and can significantly influence the local solution obtained (Boyd & Vandenberghe, 2004). These factors must be taken into account during the optimization of an ANN model.

While the feedforward network (FNN) uses only contemporaneous variables, financial volatility prediction often uses time series data or what we can refer to as sequential data. Another class of ANNs, Recurrent Neural Networks (RNN), was constructed to address this issue. To construct the RNN form from the FNN structure, parameter sharing is necessary (Goodfellow et al., 2016). This follows as a particular piece of information appears at every time step as the sequence moves forward. This stands in contrast to the FNN where the parameters are position-specific and a specific input feature always appears at the same position, and therefore uses the same parameter.

One of the most widely used RNNs is the Long Short-Term Memory (LSTM) model introduced by Hochreiter and Schmidhuber (1997) and further developed by Gers and Schmidhuber (2001). The model includes internal recurrent self-loops called memory cells, that enable the
LSTM to store information between sequences and also delete information when deemed irrelevant.

2.3 Model Tuning

Constructing and training ANNs requires careful consideration of various aspects, and several parameters must be set in advance (Goodfellow et al., 2016). These parameters are contained in the model tuning, optimization, and regularization processes described below.

2.3.1 Optimization.

The optimization of ANNs is the process of through iterations to find the function best fitting the training data, which means finding the bias and weights that minimize the specified loss function. This is a demanding task since ANNs have many parameters, and the optimal value for every parameter has to be found. Due to the nature of the iterative local optimization function gradient descent, the global minimum of the loss function is seldom found. Hence a low local minimum is often accepted. Because of this, the same neural network can find different “optimal” solutions in the same data. As described by Boyd and Vandenberghe (2004) the local optimization process can therefore partly be seen as an art rather than a science. Several optimization techniques have been developed to overcome these issues, with the most prominent presented below.

**Adaptive Learning Rate.** The learning rate is the step size of the gradient descent. Adjusting the learning rate during training can reduce the risk of the algorithm finding a too-high local minimum (Goodfellow et al., 2016). Several optimization algorithms utilize adaptive learning rates, such as AdaGrad (Duchi et al., 2011), RMSProp (Graves, 2013), and Adam (Kingma & Ba, 2014). A deeper exposition of adaptive learning algorithms is out of the scope of this paper and interested readers are referred to the sources above.

**Initialization.** As the use of gradient descent requires initial values for weights and biases to be chosen pre-training, the initial values do in part pre-determine the final parameter estimation (Boyd & Vandenberghe, 2004). One basic but important property of the initial parameters is the need to “break symmetry” between the nodes, or else the gradient descent will update all weights the same way. This leads to the network losing its complexity and not finding a solution sufficiently close to the approximated global one (Goodfellow et al., 2016). To break symmetry the initialization is normally done by choosing parameters randomly from a specified distribution (Glorot & Bengio, 2010).
2.3.2 Regularization.

All machine learning prediction methodologies, such as artificial neural networks, face the challenge of performing well on new unseen data. This means that a model with high data fitting capacity will fit the function to the training data very well; this holds for all observations, including those contaminated with measurement error or noise. This is referred to as overfitting (Haykin, 2009). The result is poor out-of-sample performance as the model fits both signal and noise. Regularization can be seen as a method for lowering a model’s overfitting capacity (Goodfellow et al., 2016) or as a way to smoothen the function between input and output mappings (Haykin, 2009). The key goal is minimizing the prediction error on the generalized unseen data while having only limited training data available. This can be done by lowering the model complexity (Franses & Van Dijk, 2000) or by applying different regularization techniques, some of which are described below, as well as by splitting the data set to evaluate tuning, performance, and validation (Kuhn & Johnson, 2013).

Parameter norm regularization. The basic principle of norm regularization is penalizing the objective function with a weight of the specific norm chosen. The two most frequently used norms in this context are $l_2$-norm and $l_1$-norm. The $l_2$-norm is defined as

$$||x||_2 = \sqrt{\sum_{i=1}^{n} w_i^2},$$

where $w_i$ is the estimated weight in the neural network. The square $l_2$-norm is usually multiplied by a penalty term $\lambda$ and then added to the objective function, as the sum of squared error in the case of an OLS regression. When done to a linear regression this is referred to as a Ridge regression, pushing all weights towards zero as the penalty term increases (Hoerl & Kennard, 1970). When the $l_1$-norm is applied to a linear regression it is called Least Absolute Shrinkage and Selection Operator (LASSO). While the definition looks similar to the $l_2$-norm,

$$||x||_1 = \sum_{i=1}^{n} |w_i|,$$

the regularizing effect is different. Instead of shrinking all weights towards zero, LASSO shrinks weights associated with noninformative input variables to zero (Tibshirani, 1996). Both methods inhibit the variance of predictions from the trained model, and the $l_1$-norm also functions as a simultaneous variable selection device, rejecting variables by reducing their coefficients to zero (Kuhn & Johnson, 2013).
**Ensemble (Model Averaging).** Model averaging is an ensemble technique where several models are trained and the prediction used is the average of all predictions. As neural networks are complex and dependent on the stochastic initialization of weights among other parameters, the same model often comes to different solutions when trained with the same dataset. Hence it benefits from model averaging even when the model and dataset are the same (Perrone & Cooper, 1995; Tumer & Ghosh, 1996). According to Goodfellow et al. (2016), this leads ensembles to in general generate lower prediction errors than any individual neural network it is composed of.

**Dropout.** According to Srivastava et al. (2014), who first proposed the dropout method, the idea comes from reproduction in nature, where only half of the genes come from one part. The method can be seen as regularization by adding noise to the hidden units. It is implemented by randomly multiplying the output or input from any layer with zero, at a predetermined probability. This process is named thinning and makes every iteration of training a different structured neural network. Thus, using dropout can be seen as training an ensemble of different models with different structures, still sharing parameters throughout the network (Goodfellow et al., 2016). Srivastava et al. (2014) show that the use of dropouts is more effective and less computationally expensive than other competing regularizers.

**Early Stopping.** As described, the training set is used when fitting the model, adjusting the model parameters for every training iteration by minimizing the loss function with respect to the training errors. When training a model of great capacity, the training errors typically decrease continuously for every iteration until, in theory, the model fits the training data perfectly. At this point, the loss function has reached its minimum and this is hence extreme overfitting. However, the generalization error is typically only decreasing to a certain point until it starts to increase. Early stopping handles this phenomenon by evaluating errors from a validation data set as a proxy for the generalization error at every iteration (Goodfellow et al., 2016). When the error from the validation set starts increasing, training is halted and the parameters with the lowest validation error are returned. In this way, a model with greater generalization ability is achieved (Wang et al., 1993).

### 2.4 Realized Volatility Forecasting with ANNs

Early studies from the late 1990s by Gonzalez Miranda and Burgess (1997), Schittenkopf et al. (1998), Ormoneit and Neuneier (1996), and Donaldson and Kamstra (1997) showed promising results for using artificial neural networks in volatility forecasting, compared to the...
contemporary linear models. Hamid and Iqbal (2004) acknowledged that these studies mainly were published in nonfinancial journals, reasoning that the black box perception of the models deterred financial and economic academics from using this category of models.

A quarter of a century later, the use of ANNs for volatility forecasting is still uncommon in financial academia compared to linear forecasting. However, the amount of research in the field is accelerating and the understanding evolving.

For example; Bucci (2020) investigated single-layered FNNs for predicting the index S&P 500 one month ahead and included exogenous factors. This resulted in the FNNs outperforming the traditional econometric models.

More recently, Christensen et al. (2022) constructed four differing FNN configurations employing exogenous variables in addition to the daily, weekly, and monthly RV as input. They compared the resulting out-of-sample prediction performance to the classic HAR model and the more advanced HAR-X, which was defined as a multivariate linear regression including the same independent variables as the FNN models. This was done for 29 Dow Jones Industrial Average constituents, producing one-day ahead forecasts for each stock's RV, including up to twelve independent variables. Christensen et al. (2022) concluded the FNNs to be superior, both when only employing endogenous variables in the shape of lagged RV, as well as when employing exogenous variables. One of the most significant contributions of this study was providing suggestions for the optimal FNN architecture for use in RV forecasting. The FNNs were configured with different numbers of hidden layers and nodes: NN1(i, 4, 1), NN2(i, 8, 4, 1), NN3(i, 16, 8, 4, 1), and NN4(i, 32, 16, 8, 4, 1), inspired by Masters (1993) geometric pyramid. Christensen et al. (2022) concluded that a three-layered FNN was optimal in their setting, with the four-layered close behind. They also showed that even the single-layered FNN outperformed the HAR and HAR-X in predicting RV, taking the results from Bucci (2020) one step further. However, they also employed the logarithm of RV, as this implicitly imposes a non-linearity to the linear model and is known to improve the HAR model forecasting ability. When comparing all models using the logarithm of RV, the NN3 model was still superior to the HAR model and the NN1 model had performance comparable to HAR-X, but the performance gap between the FNNs and the linear HAR models was reduced.

As volatility forecasting is a time series problem, due to future values being determined by past values, researchers have suggested modeling predictions through the recurrent neural network structure LSTM, which takes data sequences as inputs. Liu (2019) investigated forecasting the
squared returns of the daily S&P 500 index and an individual stock by comparing the GARCH(1,1) and a single-layered LSTM. The study used log returns and logged squared returns as explanatory variables using two lags and a one-step-ahead forecasting horizon, resulting in the LSTM model outperforming the traditional GARCH(1,1) model.

Bucci (2020) pursued a similar exercise with monthly data when comparing traditional econometric models to a single-layered LSTM model utilizing 50 nodes in the hidden layer, and a sequence of three-time lags and exogenous variables as inputs. The results again showed that the LSTM models outperformed the traditional econometric models. Confirming results were also obtained from a more recent paper by Lu et al. (2022), where the authors compare the forecasts given by LSTM and linear models on the monthly RV of oil futures. Once again the LSTM model was shown to outperform the traditional econometric models. Rahimikia and Poon (2020b) came to a similar conclusion when comparing the forecasting results by HAR models and single-layered LSTMs on returns from 23 Nasdaq stocks between the 27th of July 2007 and the 27th of January 2022. The authors also investigated how the number of nodes affected forecasting accuracy, by estimating five different LSTMs using 5 to 25 nodes with a five-node interval. The results indicated that the structure using the highest number of nodes, 25, performed best. This is in line with Liu (2019), who further investigated how the configuration of a single-layered LSTM affects forecasting ability by successively increasing the number of nodes from 10 to 100, finding 30 nodes to be optimal.

The abovementioned papers are examples of FNN and LSTM models outperforming traditional econometric models. However, Rahimikia and Poon (2020b) report findings regarding the consistency of forecasts generated by LSTM models. They divide the out-of-sample data into two groups, representing the 10% of the days with the highest volatility and the other 90% as days with normal volatility. When comparing the prediction mean squared errors between the two groups, the LSTM models outperform HAR during normal days, but more importantly, they find the LSTM models to perform worse than HAR during high-volatility days. Hence, Rahimikia and Poon (2020b) conclude that the seemingly superior accuracy for the full out-of-sample predictions is driven by the outperformance during normal volatility days. Their proposed explanation for this phenomenon is that the LSTM algorithm optimizes the forecasting accuracy for 90% of the full out-of-sample predictions, thus as a result of regularization, it tends to disregard the highest 10% of volatility.
The findings of Rahimikia and Poon (2020b) contradict previous research and a contrasting conclusion was made by Bucci (2020) when studying the highly volatile period of the Great Recession of 2008. Bucci concludes that the LSTM and FNN models outperform the traditional econometric models during these high-volatility conditions. Lu et al. (2022) confirms the results of Bucci (2020) when conducting a more thorough investigation, dividing the conditions into six groups: economic expansion, recession, non-crisis, crisis as well as high and low volatility. Lu et al. (2022) classified their sample, where months with higher or equal to the out-of-sample mean volatility were classified as high volatility months, and months with volatility lower than the mean were classified as low. They found that both the LSTM and FNN models outperformed the linear benchmark model during high and low volatility, with a greater outperformance during high volatility. These results contradict Rahimikia and Poon (2020b) who found an opposite relative relationship, with the LSTM performing worse than the benchmark model during high volatility.

Nevertheless, more nuanced studies have been conducted on the subject. Christensen et al. (2022) study one-day-ahead forecasts using FNNs in different levels of volatility. To explore the accuracy's dependency on volatility, they divide the test data into deciles based on actual RV levels each day and calculate the MSE HAR ratios for each decile. The results reveal that their best-performing FNN model, structured NN3(i, 16, 8, 4, 1), struggled during extreme volatility conditions. The worst performance was shown to occur during days when RV was in the bottom 10%, and the second worst in the top 10%. Hence, Christensen et al. (2022) conclude that the main driver of ANNs’ superiority over HAR is the predictions in the majority of the sample, where volatility is relatively modest, aligning with Rahimikia and Poon (2020b). Although the model performs worse in extreme conditions, Christensen et al. (2022) find the ANN model to outperform the HAR model regardless of volatility conditions. Christensen et al. (2022) further compare the unregularized linear HAR-X model to their best-performing FNN model at all volatility decile levels. They found that starting from the lowest decile, the FNN model had an increasing relative forecasting accuracy as the volatility level increased, except for the top decile where the performance was leveled.

According to their assertion, Christensen et al (2021) concluded that in a low-volatility environment, regularization and nonlinearities are deemed unnecessary, whereas they become crucial when volatility increases to medium and moderately high. Consequently, Christensen et al. (2022) argue that the superiority of ANNs is twofold, firstly they point to the importance of regularization during higher volatility regimes, and hence underperformance during
medium-to-high volatility ought to be an overfitting problem. Secondly, they suggest that the importance of nonlinearity in the forecasting model increases when volatility increases to the range of medium-to-high volatility.

However, Christensen et al. (2022) do not give any suggestion as to how these two effects interact or a specification of their mutual relationship. Going back to the neural network literature suggests an antagonistic relationship between the capacity of a model to capture a highly nonlinear relation and the regularization (Goodfellow et al., 2016). Hence, it is unclear what aspect of ANN models that are mainly driving their superiority, the regularization or the nonlinearity.

Further, Rahimikia and Poon (2020b) studied how different models with different complexity performed during the most volatile decile and the remaining data. The results showed that a modestly complex structure was preferred during normal volatility, while the most complex model performed best during high volatility. Hence, their results point towards a trade-off between ANNs structure complexity and the forecasting accuracy during different volatility regimes. Where more complex models outperform during high volatility conditions and less complex models outperform during low volatility.

As Franses and Van Dijk (2000) point out, model complexity and regularization are closely related since both determine an artificial neural network's capacity to approximate the mapping function of the training data. Hence, the trade-off relation found by Rahimikia and Poon (2020b) supports the argument of Christensen et al. (2022), that an increasing ability to capture nonlinearities is needed when volatility rises. This also confirms the suggestions of Christensen et al. (2022), that high nonlinearities are unnecessary when volatility is low.

To the best of our knowledge, the identification of this trade-off relationship represents a novel finding that has not been previously reported or examined further in existing research. Considering the existing literature and discussions regarding the impact of regularization on forecasting accuracy during high versus low volatility periods, a natural question arises: Could the trade-off relationship reported by Rahimikia and Poon (2020b) also depend on the specific regularization regime employed?

As described, there is contradicting evidence in the current literature on artificial neural networks' abilities to predict daily realized volatility under different volatility conditions. It is also unclear how the regularization regime applied affects the forecasting accuracy during these
different volatility conditions. In addition, there is recent evidence of a trade-off relation between model complexity and forecasting accuracy during high versus low volatility periods, which is not extensively explored in the current literature.

2.5 Hypothesis

As previous studies on daily forecasting of realized volatility using LSTM and FNN models by Christensen et al. (2022) and Rahimikia and Poon (2020b) show artificial neural networks to perform worse during high than during moderate volatility conditions, as compared to the HAR model, our first hypothesis is:

Hypothesis 1a: Artificial neural networks are less accurate during high-volatility days than during low-volatility days compared to the HAR model, when forecasting daily realized volatility.

Since the research presented by Bucci (2020) and Lu et al. (2022) contradicts hypothesis 1a, our alternative hypothesis is:

Hypothesis 1b: Artificial neural networks are more accurate during high-volatility days than during low-volatility days compared to the HAR model, when forecasting the daily realized volatility.

Christensen et al. (2022) proposed that the main driver of the increased forecasting performance of artificial neural networks compared to the linear HAR model during medium-to-high volatility conditions is the additional use of regularizers. Hence, mitigating the overfitting problem that occurs during increasing volatility by increasing the regularization would lead to better forecasting performance during high volatility. We would therefore expect;

Hypothesis 2a: Increasing the regularization of artificial neural networks improves their out-of-sample forecasting ability during high-volatility days compared to the linear HAR model.

However, if there is not a severe overfitting problem during high volatility, and hence the main driver of increased forecasting accuracy during these conditions instead is the increased nonlinearity, then the forecasting accuracy during high-volatility days is expected to decrease with increasing regularization. The reason is that increasing the regularization will disregard the smaller sub-sample with high volatility as outliers and focus on a close fit to the majority of the sample, leading to an increase in accuracy during low-volatility days. Therefore our third hypothesis is:
Hypothesis 2b: *Increasing the regularization of artificial neural networks inhibits their out-of-sample forecasting ability during high-volatility days compared to the linear HAR model while supporting their accuracy during low-volatility days.*

Rahimikia and Poon (2020b) report a trade-off relationship between the complexity in the structure of an artificial neural network and the ability to predict realized volatility during high versus low-volatility days. This would be in line with Christensen et al. (2022) suggesting that the nonlinearity increases with volatility and hence a more complex model with higher capacity is needed during these times. This trade-off relation is further supported by Christensen et al. (2022) stating that the nonlinear capacity is unnecessary during low volatility; this suggests more complex models perform worse during low volatility since these models introduce more variability into the predictions (Goodfellow et al., 2016). Our last hypothesis is therefore;

Hypothesis 3a: *There is a trade-off between the artificial neural network model’s complexity and the ability to predict realized volatility during high versus low-volatility days.*

Since there is to our knowledge no previous theoretical support for nonlinear relations to accelerate during rising volatility conditions we expect that;

Hypothesis 3b: *There is NOT a trade-off between the artificial neural network model’s complexity and the ability to predict realized volatility during high versus low-volatility days.*

3. **Empirical Setting & Research Design**

3.1 **Data**

Under this topic, we outline the data gathering process, data selection, and data sources. We also explain and motivate why some data has been discarded as well as the transformation and cleaning processes.

3.1.1 **Sample Selection.**

For this study the empirical setting is the Swedish Stock Market. We gather limit order book data from the constituents of OMX Stockholm 30 (OMXS30) as of the 1st of March 2023. The constituents of OMXS30 were chosen due to the stocks’ relatively high liquidity. High-frequency data for each stock was gathered from the Swedish House of Finance (SHoF), using the NASDAQ HFT reconstructed order book, at a one-minute frequency. The high-frequency data needed for our analysis was available between the 8th of February 2010 and
the 28th of January 2022, further referred to as the full sample period. From this set, all stocks with continuous trading data for the full sample period were chosen; resulting in the removal of four stocks; Evolution, SBB B, Sinch, and Essity B, reducing the dataset to consist of 26 stocks.

The exogenous data set was collected from Refinitiv Eikon. OMXS30 100% Moneyness 1 month implied volatility is utilized as a proxy for VIX used by Christensen et al. (2022). In addition, the 100% Moneyness 1 month implied volatility from the option market was gathered for each stock. 100% Moneyness one-month implied volatility data were missing for two stocks; Nibe Industrier and Atlas Copco B. Large parts of this data were also missing for Hexagon, and therefore these three stocks were removed from the data set, leaving us with a complete time series for 23 stocks. Additional daily data gathered from Refinitiv Eikon was the S&P Sweden Investment Grade Corporate Bond Index for the grades AAA and A, the traded volume of each stock, the closing price of each stock, the price of the S&P 500 index, the price of the DAX index and the price of the SIX Return index.

### Table 1

List of variables used in HAR-X and ANN estimations.

<table>
<thead>
<tr>
<th>Variables</th>
<th>Transformation</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>Daily RV&lt;sup&gt;t&lt;/sup&gt;</td>
<td>Log</td>
<td>SHoF</td>
</tr>
<tr>
<td>Weekly RV&lt;sup&gt;t&lt;/sup&gt;</td>
<td>Log</td>
<td>SHoF</td>
</tr>
<tr>
<td>Monthly RV&lt;sup&gt;t&lt;/sup&gt;</td>
<td>Log</td>
<td>SHoF</td>
</tr>
<tr>
<td>Relative Quoted Spread&lt;sup&gt;t&lt;/sup&gt;</td>
<td>Log Difference</td>
<td>SHoF</td>
</tr>
<tr>
<td>Mean Depth at BBO&lt;sup&gt;t&lt;/sup&gt;</td>
<td>Log Difference</td>
<td>SHoF</td>
</tr>
<tr>
<td>Daily Traded Volume&lt;sup&gt;t&lt;/sup&gt;</td>
<td>Log Difference</td>
<td>Datastream</td>
</tr>
<tr>
<td>OMXS30 Implied Volatility</td>
<td>Log</td>
<td>Datastream</td>
</tr>
<tr>
<td>Stock Implied Volatility&lt;sup&gt;t&lt;/sup&gt;</td>
<td>Log</td>
<td>Datastream</td>
</tr>
<tr>
<td>S&amp;P 500 Index</td>
<td>Log Return</td>
<td>Datastream</td>
</tr>
<tr>
<td>DAX Index</td>
<td>Log Return</td>
<td>Datastream</td>
</tr>
<tr>
<td>SIX Return Index</td>
<td>Log Return</td>
<td>Datastream</td>
</tr>
<tr>
<td>Default Spread</td>
<td>None</td>
<td>Datastream</td>
</tr>
<tr>
<td>One Week Momentum&lt;sup&gt;t&lt;/sup&gt;</td>
<td>None</td>
<td>Datastream</td>
</tr>
</tbody>
</table>

**Notes:** The table includes the entire variable set used in estimating the HAR-X and ANN models. The variables are the final calculated ones, the source specified is where the data for the calculation is gathered from and the transformation shows how the variable has been transformed. All stock-specific variables are marked with a <sup>t</sup>.

### 3.1.2 Calculations and Transformation.

In addition to realized volatility measures, exogenous variables were used. The exogenous variables, as well as the implementation, are to the greatest extent inspired by Christensen et
al. (2022), with minor adjustments related to the choice of market, such as changes in the specific stock indices. The variable set, as well as the source and the transformation, is depicted in Table 1.

**Realized Volatility.** Following the method of Andersen et al. (2001), the one-minute LOB price data was converted to a five-minute frequency to avoid microstructure biases. The first and last five minutes of the trading day, after the opening auction and before the closing auction respectively, were removed due to the risk of excessive noninformative noise during these periods. Hence the data was collected from 9:05 to 17:25 CET.

**Figure 3: Effect of log transformation for RV of Nordea**

![Histogram of RV and log RV for Nordea stock](image)

*Notes: The figure shows a histogram of the RV for the Nordea stock, where the non-transformed RV is depicted to the left and the log-transformed RV is depicted to the right, the gray dotted line is the corresponding normal distribution.*

The daily realized volatility was then calculated following Andersen et al. (2001) and Corsi (2009) as in Equation 1. $M$ is the number of observations, $\Delta$ is $\frac{1}{M} \frac{\text{day}}{\text{day}}$, $j$ is the observation order that day, and $r_{t-j\Delta}$ is the continuously compounded return at frequency $\Delta$.

$$RV_t^{(d)} = \sqrt{\sum_{j=1}^{M} r_{t-j\Delta}^2}$$

(1)

The weekly realized volatility, Equation 2, is calculated as the rolling window of $RV_t^{(d)}$ with the number of weekly trading days set to five.

$$RV_t^{(w)} = \frac{1}{5} \sum_{i=0}^{4} RV_{t-i}^{(d)}$$

(2)
The monthly realized volatility, Equation 3, is calculated in the same fashion, using 22 monthly trading days.

\[
RV_t^{(m)} = \frac{1}{22} \sum_{i=0}^{21} RV_t^{(d)}
\]  

(3)

Since the \(RV_t^{(m)}\) variable is used for every model estimation the first 22 observations are excluded and hence the sample used becomes 2010-03-09 to 2022-01-28.

The natural logarithm transformation of Andersen et al. (2001) was used to make the RV distribution closer to a Gaussian distribution, depicted in Figure 3. Christensen et al. (2022) showed that this transformation improves the performance of HAR models, and Swingler (1996) points at its positive effects for scaling purposes when training ANNs.

**Limit Order Book Variables.** Limit order book (LOB) variables have been shown to have predictive power for RV forecasting using both HAR and ANN models (Rahimikia & Poon, 2020a, 2020b), which is also backed by theories like those by Glosten and Milgrom (1985).

Two common market quality measures derived from the 1-minute LOB are used as the daily mean; the relative bid-ask spread calculated as in Foucault et al. (2013) and mean depth at best-bid-and-offer (BBO) as calculated in Goldstein and Kavajecz (2000). The measures are defined as below.

Relative Quoted Spread  
\[
\text{Relative Quoted Spread} = \frac{1}{N} \sum_{t=1}^{N} \frac{\text{Ask}_{it} - \text{Bid}_{it}}{\text{Mid}_{it}}
\]  

(4)

Where \(\text{Ask}_{it}\) is the best ask price for stock \(i\) at the discrete time \(t\) and \(\text{Bid}_{it}\) is the best bid price for stock \(i\) at the discrete time \(t\). \(\text{Mid}_{it}\) is the mid-price and is calculated as the mean of the best ask and bid price. \(N\) represents the total discrete 1-minute time steps during a trading day.

Mean Depth at BBO  
\[
\text{Mean Depth at BBO} = \frac{1}{N} \sum_{t=1}^{N} \frac{\text{AskSize}_{it} + \text{BidSize}_{it}}{2}
\]  

(5)

Where \(\text{AskSize}_{it}\) is the number of shares at the best ask price for stock \(i\) at the discrete time \(t\) and \(\text{BidSize}_{it}\) is the number of shares at the best bid price for stock \(i\) at the discrete time \(t\). \(N\) again represents the total discrete 1-minute time steps during a trading day.

Since the absolute level of these measures is not necessarily stationary we transform the measures to the log difference.
Daily Traded Volume. Since the daily traded volume of a stock has been shown to be correlated with volatility and have predictive power (Christensen et al., 2022), we have included the variable in our data set. To facilitate stationarity we have transformed it into a daily log difference.

Implied Volatility Measures. We use the 100% Moneyness 1-month implied volatility from the option market as an approximation for implied volatility both for individual stocks and for the OMXS30 index, as we consider these variables to be the best approximations at our disposal. As implied volatility exhibits the same non-normal distribution and excess skewness as RV the data was log-transformed.

Stock Market Indices. Per Christensen et al. (2022) and backed by the findings of Liu and Pan (1997) and Xiao and Dhesi (2010) we include the two foreign stock market indices to capture spillover effects; the American S&P 500 and the German DAX, The SIX Return Index is used to capture possible correlation with the broader Swedish stock market, a commonly used approximation for the Swedish market return (Aytug et al., 2020). The stock market indices are transformed into stock market log returns, a standard practice among researchers, ensuring stationary and giving the data additive properties.

Default Spread. In coherence with Bucci (2020), corporate default spread is used as an approximation for credit risk. The default spread is calculated using daily observations from S&P Sweden Investment Grade Corporate Bond Index for grades AAA and A per Equation 6, where DefaultSpread$_t$ is the default spread at time $t$, $A_t$ is the single A bond index at time $t$ and AAA$_t$ is the ditto for triple-A.

$$DefaultSpread_t = A_t - AAA_t$$ (6)

One-Week Momentum Factor. Like Christensen et al. (2022) the one-week momentum factor is included in the exogenous dataset, calculated as per Equation (7), where OneWeekMomentum$_{it}$ is the momentum factor for stock $i$ at time $t$, Price$_{it}$ is the stock price of stock $i$ at time $t$ and Price$_{it-5}$ is the stock price of stock $i$ at five time-steps prior to time $t$.

$$OneWeekMomentum_{it} = \frac{Price_{it} - Price_{it-5}}{Price_{it}}$$ (7)
3.2 Research Design

3.2.1 HAR.

The HAR model is chosen as a benchmark, representing the traditional linear models against the different ANN models. The model, first proposed by Corsi (2009) has proven to be one of the most accurate and has been used for benchmarking in several earlier studies (Christensen et al., 2022; Fernandes et al., 2014). The model is specified per Equation 8, where $c$ is a constant, $\varepsilon$ is an error term, and $\beta(d)$, $\beta(w)$, and $\beta(m)$ are the coefficients assigned to daily, weekly, and monthly realized volatility respectively.

$$RV_{(t+1d)} = c + \beta(d)RV_{(t)} + \beta(w)RV_{(t)} + \beta(m)RV_{(t)} + \varepsilon_{t+1d} \tag{8}$$

For estimation and out-of-sample predictions, a 1000-day rolling window is used as in Corsi (2009) and Christensen et al. (2022), and for regression estimation the Ordinary Least Squares (OLS) method is used.

3.2.2 HAR-X.

To check robustness and enhance comparability with the ANN models, the HAR-X model is included as described by Christensen et al. (2022). The model is estimated like the HAR model with the addition of the full dataset presented in Table 1, resulting in a multivariate regression model with thirteen independent variables estimated by OLS.

3.2.3 Training and Out-Of-Sample Prediction for ANN Models.

All ANNs described below are trained using Keras, a Python deep-learning library, running on top of the TensorFlow API (Keras, n.d.; TensorFlow, n.d.). Following the method of Christensen et al. (2022), the data set is split into three subsets; the first 70% selected for training, 10% for validation, and 20% for out-of-sample prediction. The in-sample data is hence from the 9th of March, 2010, to the 26th of July, 2019, and the out-of-sample 29th of July, 2019, to the 28th of January, 2022. All ANNs are trained the same way, using the training set for parameter adjustment and employing the validation set for monitoring the training progress as described in the 2.3.2 subsection Early Stopping. All input variables are standardized before training, as proposed by Swingler (1996). Standardization is referred to as scaling between zero and one, assigning the lowest value zero and the highest one, with the remainder linearly scaled in between.
While Christensen et al. (2022) train 400 independent ANNs to impede the effect of random initiation we limit the number to 100 due to a lack of computational resources, as every ANN takes approximately two days for a personal computer to estimate. These independent models are then ranked by the MSE of the validation set. From this ranking, we form three different ensembles by grouping the ten, five, and single best ANNs.

### 3.2.4 FNN Models

Four configurations of FNNs are constructed to enable prediction comparison across model complexity levels in the FNN space; All FNN models were inspired by Christensen et al. (2022). The architectures chosen are FNN(i, 8, 4, o), FNN(i, 16, 8, 4, o), FNN(i, 32, 16, 8, 4, o) and FNN(i, 64, 32, 16, 8, 4, o), we will refer to these as FNN2, FNN3, FNN4 and FNN5. Following Christensen et al. (2022) all FNNs are assembled using learning rate shrinkage, drop-out, early stopping with 500 epochs and patience of 100, the Glorot normal initializer, the optimizer Adam, and the Leaky-ReLU activation function.

The regularization regimes are altered by shifting the drop-out rate, with low regularization represented by a 0.2 drop-out rate and for high a 0.5. The drop-out rates are chosen as these values are given as upper and lower boundaries in the machine learning literature (Goodfellow et al., 2016).

### 3.2.5 LSTM Models

Inspired by Liu (2019) and Rahimikia and Poon (2020b) we construct five different one-layered LSTM models with an increasing number of nodes at five, ten, fifteen, twenty, and thirty. Hence with the model structures LSTM(i, 5, o), LSTM(i, 10, o), LSTM(i, 15, o), LSTM(i, 20, o), LSTM(i, 30, o). These models utilize learning rate shrinkage, early stopping with 500 epochs and patience of 100, the Glorot normal initializer, the optimizer Adam as well as the regularizers $l_2$-norm and $l_1$-norm. Since Rahimikia and Poon (2020b) concluded that 21 lags of all independent variables were optimal in their setting, the same number of lags is used.

As a robustness check of the dependent lag value, the Schwarz Information Criterion (SIC) proposed by Schwarz (1978) is used to find the optimal lags of daily RV, as this information criterion is known to choose a low number of lags compared to other criteria. The SIC gave us the average optimal lag number of 7.4783, hence we use 7 lag values in the second round of LSTM model estimation.
We conduct a small grid search in the decadic logarithm space, applying both the kernel $l_2$-norm and $l_1$-norm using the same penalty value for both regularizers, to find the upper and lower boundary for effective regularization. The first regime, called “low regularization”, uses both $l_2$-norm and $l_1$-norm set to $10^{-6}$. The second regime, referred to as “high regularization”, uses both $l_2$-norm and $l_1$-norm set to $10^{-4}$.

3.3 Data Analysis Method

3.3.1 Comparing Performance.

We have chosen to measure forecasting performance using MSE since it is easy to apply, understand, and utilize in the estimations of all investigated models. According to Patton (2011), the loss function MSE is both robust and homogeneous for forecasting evaluation and comparison between models, specified as Equation 9. $N$ is the sample size of the evaluated period, $RV_t$ is the actual outcome of realized volatility, and $\widehat{RV}_t$ is the realized volatility prediction of the model evaluated.

$$Mean\ Squared\ Error = \frac{1}{N} \sum_{t=1}^{N} (RV_t - \widehat{RV}_t)^2$$  \hspace{1cm} (9)

To enable a standardized comparison between models and model complexity, a HAR MSE ratio is constructed as in Christensen et al. (2022). This is done stock by stock and then averaged over all estimates. We do not report ratios for the HAR model since it is standardized to one.

To check the statistical significance of the mean MSE ratio we conduct a two-sided t-test with the null hypothesis that the model mean is equal to one, i.e. equal to the HAR model.

3.3.2 Sample Split Comparison.

To enable a comparison between days with high and low volatility levels, we divide the data using the method outlined by Lu et al. (2022). Hence, the definitions of low and high volatility are as given below.

$$High\ Volatility\ Days: RV_t \geq \overline{RV}_t$$ \hspace{1cm} (10)

$$Low\ Volatility\ Days: RV_t < \overline{RV}_t$$ \hspace{1cm} (11)
This division of the data resulted in 31% of the observations in the test data set being classified as high-volatility days and 69% as low. We calculate the MSE ratio for each period, model, and stock resulting in two series with 23 observations each. We hence report the average MSE ratio across the models for high and low days respectively.

A two-sided t-test is then conducted, testing if the two series are significantly different from each other. In addition, we perform a two-sided t-test to check if the estimated mean for the high and low-volatility days are different from one, i.e. deferring from the HAR model.

To closely analyze the models’ different volatility level accuracy, we follow Christensen et al. (2022) and split the out-of-sample data into volatility deciles, with the first decile being the 10% of the days with the lowest volatility and so forth, reporting the MSE ratio for each decile.

As a robustness check, we divide the out-of-sample into two periods. Period 1 consists of the first half of the test data, from the 29th of July, 2019, to the 21st of September, 2020, Period 2 consists of the rest of the data. We thereafter perform the decile analysis for the two periods.

3.3.3 Complexity to Accuracy Trade-Off.

To examine the existence of a trade-off relationship between model complexity and out-of-sample forecasting accuracy during high and low-volatility days we construct a regression analysis. The independent variable is a constructed binary dummy variable for each level of model complexity. Model complexity is defined as the number of FNN layers or LSTM nodes and the dependent variable is the average MSE ratio. To avoid multicollinearity and simplify interpretation the smallest model of each structure, i.e. FNN2 and LSTM5, is used as the base in the analysis and hence no coefficients for these models are reported. We use OLS regressions to estimate all regressions.

3.4 Philosophy of Social Science and Research Ethics

This study is quantitative in its nature, employing a deductive view of the relationship between theory and research. Hence, we first elaborate with the theories and results from previous research, formulate hypotheses and finally analyze the quantitative data to answer our hypothesis, as outlined by Bryan and Bell (2009).

We declare that the data has been used in accordance with the guidelines of the data providers and in line with the report Good Research Practice issued by the Swedish Research Council
Further, since only data on public companies has been used, no harm has been caused to participants and there is no breakage of the principle of informed consent.

Further, the authors report no conflicting interests.

4. Findings & Analysis

4.1 Out-of-Sample Analysis – FNN

In this section, we analyze the feedforward neural network (FNN) performance in the out-of-sample forecasting task and compare the results under high and low regularization.

4.1.1 FNN with Low Regularization.

We start by estimating FNN models using daily, weekly, and monthly realized volatility, to test if the ANNs extract the same information as HAR, or if additional nonlinear relations between these three variables are present which the ANNs can explore.

The smaller FNN models have an average MSE ratio close to one, this indicates that FNNs can capture the same relation as the linear HAR model from the three RV variables. It also suggests that the smaller FNNs are unable to extract further information from these three variables and hence do not capture any nonlinear relation which the linear HAR model is unable to exploit. However, the largest model, FNN5, shows a significant improvement over HAR as depicted in Table 2. These results are in line with Christensen et al. (2022) finding only a slight increase in prediction accuracy when employing only RV variables.

<table>
<thead>
<tr>
<th>Ensemble</th>
<th>FNN2</th>
<th>FNN3</th>
<th>FNN4</th>
<th>FNN5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.0073</td>
<td>0.9905</td>
<td>0.9862**</td>
<td>0.9756***</td>
</tr>
<tr>
<td>5</td>
<td>1.0012</td>
<td>0.9917*</td>
<td>0.9897**</td>
<td>0.9793***</td>
</tr>
<tr>
<td>10</td>
<td>0.9988</td>
<td>0.9936</td>
<td>0.9935</td>
<td>0.9815***</td>
</tr>
</tbody>
</table>

*Notes: Prediction MSE HAR ratio per FNN models and ensemble using only the three RV variables; daily, weekly, and monthly. The significance levels of the tests are indicated by: * 10%, ** 5%, and *** 1%, and tests if the estimates are significantly different from the HAR model’s predictions. The ensembles 1, 5, and 10 refer to the average prediction of the respective top models.*

HAR-X represents a linear regression model utilizing the full 13 input variable set. As seen in Table 3, the HAR-X model outperforms the traditional HAR model, albeit only by a statistical
When incorporating exogenous variables into the input variable set, all FNN models for all ensembles outperform the HAR benchmark model, with a statistical significance level of 1%, as presented in Table 3. The 10th ensemble of the FNN3 model with three layers demonstrates the highest improvement, a 9.93% lower out-of-sample MSE. This shows that a great improvement in forecasting accuracy is accomplished when introducing more variables to the input set for all FNN structures, as seen in Table 3. Again this is in line with the findings of Christensen et al. (2022) who also found an increase in out-of-sample prediction performance when adding additional variables to the input data. We further confirm Christensen et al. (2022) by showing that the FNN3 model structure is optimal in our setting.

Table 3

<table>
<thead>
<tr>
<th>Ensemble</th>
<th>HAR-X</th>
<th>FNN2</th>
<th>FNN3</th>
<th>FNN4</th>
<th>FNN5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.9859**</td>
<td>0.9282***</td>
<td>0.9297***</td>
<td>0.9165***</td>
<td>0.9279***</td>
</tr>
<tr>
<td>5</td>
<td>-</td>
<td>0.9069***</td>
<td>0.9011***</td>
<td>0.9060***</td>
<td>0.9272***</td>
</tr>
<tr>
<td>10</td>
<td>-</td>
<td>0.9051***</td>
<td>0.9007***</td>
<td>0.9069***</td>
<td>0.9320***</td>
</tr>
</tbody>
</table>

Notes: Prediction MSE ratio per model and ensemble. Where HAR-X is the HAR estimated using all exogenous variables in the data set. The significance levels of the tests are indicated by: * 10%, ** 5%, and *** 1%, and tests if the estimates are significantly different from the HAR model’s predictions. The ensembles 1, 5, and 10 refer to the average prediction of the respective top models.

To analyze the model’s performance during different volatility levels we break up the out-of-sample predictions into high-volatility days and low-volatility days. The results show that the series are significantly different from each other for all FNN models at the 1% level, however, this is not true for the HAR-X model where no significance is shown. Table 4 in turn shows what significance level the series in each subsample differs from the HAR models predictions. From Table 4 we find that the HAR-X model beats the classic HAR model by 2.5% during low volatility at a 1% significance level. However, during high volatility the HAR-X model does not show a significantly different performance from the classic HAR model. The FNN models show a different relationship where all estimated models have lower MSE during high volatility compared to the benchmarking HAR model at the 1% significance level. In our out-of-sample prediction, this improvement during high-volatility days ranged from 12.76% for the FNN4 ensemble 10 to 10.64% for FNN2 ensemble 1. However, during the low-volatility days the
FNN models as a group struggles to show any significant improvement as compared to the HAR model. Comparing all FNNs, only the FNN3 ensemble 10 model shows higher forecasting accuracy at a 1% significance level during low volatility with an estimated improvement over the HAR model of 5.77%. This also shows that the FNN3 model is the only FNN with significantly better forecasting performance at the 1% level for both high and low-volatility days.

The overall results depicted in Table 4 indicate that FNNs perform better during high-volatility days compared to low-volatility days confirming the evidence from Lu et al. (2022) and contradicting those of Christensen et al. (2022).

Table 4
Out-of-sample MSE HAR ratio for feedforward neural networks and the HAR-X model prediction during high as well as low volatility days using the full input variable set as well as drop-out set to 0.2.

<table>
<thead>
<tr>
<th>Ensemble</th>
<th>FNN2</th>
<th>FNN3</th>
<th>FNN4</th>
<th>FNN5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>High days</td>
<td>Low days</td>
<td>High days</td>
<td>Low days</td>
</tr>
<tr>
<td>1</td>
<td>0.9914</td>
<td>0.9749***</td>
<td>0.8936***</td>
<td>0.9898</td>
</tr>
<tr>
<td>5</td>
<td>-</td>
<td>-</td>
<td>0.8722***</td>
<td>0.9690</td>
</tr>
<tr>
<td>10</td>
<td>-</td>
<td>-</td>
<td>0.8730***</td>
<td>0.9621*</td>
</tr>
</tbody>
</table>

Notes: Prediction MSE ratio per FNN and ensemble comparing high versus low-volatility days. The t-test examines if the series of MSE ratios during high and low-volatility days are significantly different from the HAR model’s predictions. The significance levels of the tests are indicated by: * 10%, ** 5%, and *** 1%. Additionally, all estimates for high days are significantly different from those during low days at the 1% level for FNN models. The two series for the HAR-X model are not significantly different from each other.

To further examine the relationship between forecasting accuracy and volatility levels, we conduct a comprehensive analysis by dividing and evaluating the predictions of the FNNs and HAR-X models across different deciles of volatility levels. Figure 4 provides a graphical representation of these findings, while Table 5 presents the corresponding results in tabular format.

Interestingly, the HAR-X model demonstrates relatively stable predictive performance across all deciles, consistently achieving a value close to one within each sub-sample. In contrast, the collective performance of the FNN models displays distinct patterns. The FNN models exhibit their highest performance during the first to fourth deciles of volatility, indicating superior accuracy in periods of low volatility. Additionally, they also demonstrate notable performance during the top decile, suggesting effectiveness during periods of exceptionally high volatility. This finding contrasts with the results reported by Christensen et al. (2022), who observed that the FNN3 model underperformed in these more atypical circumstances, thus indicating a reversed relationship.
Notes: MSE ratios per volatility level grouped by decile, for FNNs and HAR-X. The red dashed line at the level where the MSE ratio is one represents the performance of the HAR model.

Table 5

Out-of-sample MSE HAR ratio reported per volatility decile for feedforward neural networks with drop-out set to 0.2 and HAR-X model predictions using the full input variable set.

<table>
<thead>
<tr>
<th>Decile</th>
<th>HAR-X</th>
<th>FNN2</th>
<th>FNN3</th>
<th>FNN4</th>
<th>FNN5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st</td>
<td>0.9848</td>
<td>1.0288</td>
<td>0.9616</td>
<td>0.9661</td>
<td>0.9839</td>
</tr>
<tr>
<td>2nd</td>
<td>0.9554</td>
<td>0.9766</td>
<td>0.9104</td>
<td>0.9309</td>
<td>0.9713</td>
</tr>
<tr>
<td>3rd</td>
<td>0.9652</td>
<td>0.9327</td>
<td>0.8852</td>
<td>0.9184</td>
<td>0.965</td>
</tr>
<tr>
<td>4th</td>
<td>0.9327</td>
<td>0.8695</td>
<td>0.8624</td>
<td>0.9328</td>
<td>1.0149</td>
</tr>
<tr>
<td>5th</td>
<td>0.9931</td>
<td>0.9159</td>
<td>0.9844</td>
<td>1.0701</td>
<td>1.1643</td>
</tr>
<tr>
<td>6th</td>
<td>0.9771</td>
<td>0.924</td>
<td>1.0216</td>
<td>1.0885</td>
<td>1.1806</td>
</tr>
<tr>
<td>7th</td>
<td>1.0098</td>
<td>0.9399</td>
<td>1.0256</td>
<td>1.0692</td>
<td>1.151</td>
</tr>
<tr>
<td>8th</td>
<td>1.0206</td>
<td>0.988</td>
<td>1.0589</td>
<td>1.1088</td>
<td>1.1753</td>
</tr>
<tr>
<td>9th</td>
<td>0.9953</td>
<td>0.9746</td>
<td>1.0087</td>
<td>1.0176</td>
<td>1.0644</td>
</tr>
<tr>
<td>10th</td>
<td>0.9874</td>
<td>0.836</td>
<td>0.8211</td>
<td>0.804</td>
<td>0.8009</td>
</tr>
</tbody>
</table>

Notes: Prediction MSE ratio per model and decile for ensemble 10 using exogenous data.

To test the robustness of our findings and eliminate potential biases resulting from the sample distribution or idiosyncrasies, we conducted a robustness check and divided the sample into period 1 and 2. By performing the same analysis on a volatility decile level within each of these sub-samples, we aimed to examine the validity and stability of our results across different periods.

We focused our investigation on the best-performing model, namely the FNN3 model with ensemble 10. The outcomes of this analysis are presented in Figure 5, revealing distinct relationships between volatility levels and prediction accuracy in the two periods. In Period 1, there is a tendency for prediction accuracy to increase as the volatility level rises. However, in contrast, during Period 2, prediction accuracy demonstrates a decrease as the volatility level
increases. These divergent patterns indicate that the results obtained from the entire sample might be influenced by the specific distribution and characteristics of our dataset, and thus cannot be generalized beyond these specific conditions.

Based on these findings, we conclude that, within our specific setting, it is not possible to establish a general relationship between volatility levels and prediction accuracy for the FNN models.

**Figure 5**

![MSE ratio per decile comparing FNN3 with low regularization predictions Period 1 versus period 2](image)

*Notes: Comparison of Period 1 and Period 2 of the prediction MSE ratio per decile for model FNN3 ensemble 10. The red dashed line at the level where the MSE ratio is one represents the performance of the HAR model.*

As described in section 3.3.3 we conduct a regression analysis. To identify a trade-off relationship we have to observe significantly different signs of all coefficients from the high and low-volatility days. In addition, we need to see the negative coefficients decrease with model complexity and the positive coefficients increase with model size, or vice versa. Meeting these conditions we would be able to establish a trade-off relationship. The result from the regression analysis for both high-volatility days and low-volatility days is presented in Table 6.

The coefficients for each model during high-volatility days show no statistical significance, indicating that increasing the number of hidden layers beyond two does not alter the forecasting accuracy during high-volatility days, the results are presented in Table 6A. During low-volatility days a more dynamic picture is presented, where increasing the number of hidden layers from two to three, as in FNN3, leads to improvements in the out-of-sample forecasting accuracy as shown by the negative sign of the coefficient, which is significant at the 1% level. Increasing the model complexity further, to four layers, the regression coefficient is not statistically significant indicating no different performance compared to the two-layered model. The effect of adding one more layer to the model results in a statistically significant coefficient at the 1% level with a positive sign, suggesting the FNN5 model performs worse compared to the FNN2 model.
The regression results described for low-volatility days are intuitive. It shows that the model forecasting accuracy increases with complexity only to the point where the optimal model is reached, and then declines as the complexity increases further. The regression analysis results for low-volatility days are depicted in Table 6B.

As shown above, our results do not indicate any trade-off relation between model complexity and forecasting accuracy during high versus low-volatility days for FNN models. However, we find that the main differentiating factor between the FNN models is their performance during low-volatility days since no significant difference could be shown for high-volatility days.

Table 6

<table>
<thead>
<tr>
<th>Model</th>
<th>Coefficient Std. error</th>
<th>t-value</th>
<th>L. conf.</th>
<th>U. conf.</th>
<th>Coefficient Std. error</th>
<th>t-value</th>
<th>L. conf.</th>
<th>U. conf.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cons.</td>
<td>0.8919*** 0.0842</td>
<td>10.564</td>
<td>0.2246</td>
<td>0.5591</td>
<td>Cons.</td>
<td>0.4237*** 0.0688</td>
<td>6.4565</td>
<td>0.2869</td>
</tr>
<tr>
<td>FNN3</td>
<td>0.1643 0.1190</td>
<td>1.3806</td>
<td>-0.0722</td>
<td>0.4008</td>
<td>FNN3</td>
<td>-0.2810*** 0.0973</td>
<td>-2.8933</td>
<td>-0.4750</td>
</tr>
<tr>
<td>FNN4</td>
<td>0.0177 0.1190</td>
<td>0.1487</td>
<td>-0.2188</td>
<td>0.2542</td>
<td>FNN4</td>
<td>-0.0391 0.0973</td>
<td>-0.4013</td>
<td>-0.2525</td>
</tr>
<tr>
<td>FNN5</td>
<td>0.1877 0.1190</td>
<td>1.5775</td>
<td>-0.0488</td>
<td>0.4242</td>
<td>FNN5</td>
<td>0.3662*** 0.0973</td>
<td>3.7026</td>
<td>0.1728</td>
</tr>
</tbody>
</table>

Notes: OLS regression results when using model complexity as a dummy variable during high volatility days for feedforward neural networks with dropout set to 0.2.

4.1.2 FNN with High Regularization.

To investigate the relationship between our findings concerning distinct out-of-sample forecasting accuracy during high and low volatility periods and the level of regularization, as well as to determine if the previously discussed complexity-accuracy trade-off persists under higher regularization regimes, we perform the same procedure outlined earlier. However, in this case, we modify the methodology by increasing the dropout rate to 0.5 for all models, we will refer to this as high regularization.

As the regularization of the FNNs is increased through shifting the drop-out rate from 0.2 to 0.5 the less complex models struggle to beat the benchmarking HAR model, depicted in Table 7. The optimal model size in this higher regularized setting is the FNN4 model, decreasing in accuracy by 1.52% in units compared to the optimal model during low regularization.
When turning to the comparative analysis between high and low-volatility days for the highly regularized FNNs the statistical difference between the two series, as was observed for the less regularized FNNs, has vanished. This is true for all models except for the FNN5 ensemble 10, which manages to show a significant difference between high and low volatility at the 5% level. These findings suggest that by reducing the model capacity through regularization, the previously observed pattern of higher forecasting accuracy during high-volatility days diminishes. Interestingly, the most complex model, FNN5, with the greatest capacity and thus affected the least by the extra regularization still shows a statistical difference between prediction from high and low-volatility days. This further indicates a link between the observed pattern and regularization.

Table 7
Out-of-sample MSE HAR ratio for feedforward neural networks predictions using the full input variable set and drop-out set to 0.5.

<table>
<thead>
<tr>
<th>Ensemble</th>
<th>FNN2</th>
<th>FNN3</th>
<th>FNN4</th>
<th>FNN5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.9871</td>
<td>0.9669</td>
<td>0.9334***</td>
<td>0.9405***</td>
</tr>
<tr>
<td>5</td>
<td>0.9643**</td>
<td>0.9346***</td>
<td>0.9165***</td>
<td>0.9215***</td>
</tr>
<tr>
<td>10</td>
<td>0.9591**</td>
<td>0.9350***</td>
<td>0.9159***</td>
<td>0.9212***</td>
</tr>
</tbody>
</table>

Notes: Prediction MSE ratio per model and ensemble for FNNs estimated with a drop-out rate set to 0.5. The significance levels of the tests are indicated by: * 10%, ** 5%, and *** 1%, and tests if the estimates are significantly different from the HAR model’s predictions. The ensembles 1, 5, and 10 refer to the average prediction of the respective top models.

As would be expected, the highly regularized FNNs out-of-sample forecasts exhibit a lower MSE ratio variation than the lower regularized, as can be seen in Figure 6 and Table 9 where the MSE ratio is presented by volatility decile. It is also evident that the predictions are in general closer to the HAR models.
Figure 6

MSE ratio per decile for FNN models with high regularization

Notes: The MSE ratios per volatility level grouped by decile, for highly regularized FNNs. The red dashed line at the level where the MSE ratio is one represents the performance of the HAR model.

Table 9

<table>
<thead>
<tr>
<th>Decile</th>
<th>FNN2</th>
<th>FNN3</th>
<th>FNN4</th>
<th>FNN5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st</td>
<td>1.0982</td>
<td>0.9836</td>
<td>0.9532</td>
<td>0.9391</td>
</tr>
<tr>
<td>2nd</td>
<td>1.0037</td>
<td>0.8962</td>
<td>0.8955</td>
<td>0.9003</td>
</tr>
<tr>
<td>3rd</td>
<td>0.9126</td>
<td>0.8473</td>
<td>0.8770</td>
<td>0.9048</td>
</tr>
<tr>
<td>4th</td>
<td>0.7884</td>
<td>0.7800</td>
<td>0.8554</td>
<td>0.9306</td>
</tr>
<tr>
<td>5th</td>
<td>0.8168</td>
<td>0.8823</td>
<td>0.9681</td>
<td>1.0790</td>
</tr>
<tr>
<td>6th</td>
<td>0.8180</td>
<td>0.9242</td>
<td>1.0192</td>
<td>1.1370</td>
</tr>
<tr>
<td>7th</td>
<td>0.8679</td>
<td>0.9602</td>
<td>1.0167</td>
<td>1.1189</td>
</tr>
<tr>
<td>8th</td>
<td>0.9404</td>
<td>1.0223</td>
<td>1.0617</td>
<td>1.1290</td>
</tr>
<tr>
<td>9th</td>
<td>0.9996</td>
<td>1.0300</td>
<td>1.0073</td>
<td>1.0415</td>
</tr>
<tr>
<td>10th</td>
<td>0.9590</td>
<td>0.9160</td>
<td>0.8649</td>
<td>0.8326</td>
</tr>
</tbody>
</table>

Notes: Prediction MSE ratio per model and decile for ensemble 10 using exogenous data.

Turning to the regression analysis concerning the aforementioned trade-off relation between model complexity and forecasting ability at different volatility levels, the results are shown in Table 10. Observing the results in Table 10A from the more regularized FNNs for high-volatility days we find that all coefficients for the dummy variables are negative, instead of positive as for the less regularized. In addition, the coefficients are decreasing in value with model complexity, indicating that the forecasting ability increases with model complexity. However, we do not find a trade-off relation since there is no uniform pattern for the low-volatility days, as seen in Table 10B. Hence our results show no sign of a trade-off relation for FNNs in volatility prediction, neither under low regularization nor high regularization.
Table 10

A. OLS regression results using model complexity as a dummy variable during high volatility days for feedforward neural networks using a drop-out rate of 0.5.

<table>
<thead>
<tr>
<th>Model</th>
<th>Coefficient</th>
<th>Std. error</th>
<th>t-value</th>
<th>L. conf.</th>
<th>U. conf.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cons.</td>
<td>0.7023***</td>
<td>0.0740</td>
<td>9.4852</td>
<td>0.5552</td>
<td>0.8494</td>
</tr>
<tr>
<td>FNN2</td>
<td>-0.0614</td>
<td>0.1047</td>
<td>-0.5867</td>
<td>-0.2985</td>
<td>0.1467</td>
</tr>
<tr>
<td>FNN4</td>
<td>-0.3575***</td>
<td>0.1047</td>
<td>-3.4308</td>
<td>-0.5860</td>
<td>-0.1498</td>
</tr>
<tr>
<td>FNN8</td>
<td>-0.4576***</td>
<td>0.1047</td>
<td>-3.6941</td>
<td>-0.6051</td>
<td>-0.2489</td>
</tr>
</tbody>
</table>

B. OLS regression results using model complexity as a dummy variable during low volatility days for feedforward neural networks using a drop-out rate of 0.5.

<table>
<thead>
<tr>
<th>Model</th>
<th>Coefficient</th>
<th>Std. error</th>
<th>t-value</th>
<th>L. conf.</th>
<th>U. conf.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cons.</td>
<td>0.5824***</td>
<td>0.0806</td>
<td>7.2285</td>
<td>0.4223</td>
<td>0.7425</td>
</tr>
<tr>
<td>FNN2</td>
<td>-0.3253***</td>
<td>0.1139</td>
<td>-2.8347</td>
<td>-0.5317</td>
<td>-0.0988</td>
</tr>
<tr>
<td>FNN4</td>
<td>-0.1919*</td>
<td>0.1139</td>
<td>-1.8845</td>
<td>-0.4184</td>
<td>0.0345</td>
</tr>
<tr>
<td>FNN8</td>
<td>0.0944</td>
<td>0.1139</td>
<td>0.8289</td>
<td>-0.1320</td>
<td>0.3209</td>
</tr>
</tbody>
</table>

Notes: OLS regression results when using model complexity as an independent variable and MSE ratio as a dependent variable. The results are shown for FNN models with high regularization, during high and low volatility respectively. The significance levels of the tests are indicated by: * 10%, ** 5%, and *** 1%.

4.2 Out-of-Sample Analysis – LSTM

In this section, we analyze the long short-term memory (LSTM) model’s performance in the out-of-sample forecasting task and compare the results under high and low regularization.

4.2.1 LSTM with Low Regularization.

In the following analysis of the long short-term memory model, we begin by presenting the results from the models estimated with a low regularization regime with both $l_2$-norm and $l_1$-norm set to $10^{-6}$. As described in Section 3.2.5 we follow Rahimikia and Poon (2020b) and estimate the set of LSTM models utilizing 21 lags of each of the 13 input variables. In this setting, the LSTM model utilizing 10 nodes, LSTM10, gives the best estimated prediction performance in all three ensembles. In addition, the results show that the LSTM model using 21 lags outperforms the benchmark HAR model at a 1% significance level at all levels and in all ensembles, as shown in Table 11A.

As a robustness check, we also conduct an additional estimation of the LSTM models utilizing only 7 lags, which is the optimal lags for RV in our sample suggested by the SIC as described in Section 3.2.5, this is to check if Rahimikia and Poon (2020b) results, to use up to 21 lags of each input variable, are optimal in our setting.

The results when using 7 lag and a low regularizer are very similar to those when utilizing 21 lags and the same regularization regime, with some minor differences. The LSTM model utilizing 15 nodes, LSTM15, gives the best estimated prediction performance in all three ensembles in this setting. Further, when comparing the performance of the best LSTM model using 7 lags to the LSTM model using 21 lags, the best LSTM model with 7 lags has a lower estimated MSE ratio, as presented in Table 11B. Hence, this contradicts the findings of Rahimikia and Poon (2020b) and suggests including 21 lags is of no benefit as compared to restricting the lags to 7.
The LSTM model in this sample with the estimated highest out-of-sample forecasting accuracy is the LSTM15 using 7 lags presenting an 8.87% higher accuracy compared to the HAR model and hence approximately 1% lower accuracy in units compared to the best FNN model. This difference is small and not statistically significant, indicating a similar performance between the two model structures.

Table 11

A. 
Out-of-sample MSE HAR ratio for long short-term memory neural networks predictions using the full input variable set and utilizing 21 lags and \( l_2\)-norm as well as \( l_1\)-norm equal to \( 10^{-6} \).

<table>
<thead>
<tr>
<th>Ensemble</th>
<th>LSTM5</th>
<th>LSTM10</th>
<th>LSTM15</th>
<th>LSTM20</th>
<th>LSTM30</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.9450***</td>
<td>0.9209***</td>
<td>0.9267***</td>
<td>0.9243***</td>
<td>0.9374***</td>
</tr>
<tr>
<td>5</td>
<td>0.9235***</td>
<td>0.9141***</td>
<td>0.9174***</td>
<td>0.9155***</td>
<td>0.9281***</td>
</tr>
<tr>
<td>10</td>
<td>0.9250***</td>
<td>0.9137***</td>
<td>0.9177***</td>
<td>0.9210***</td>
<td>0.9304***</td>
</tr>
</tbody>
</table>

B. 
Out-of-sample MSE HAR ratio for long short-term memory neural networks predictions using the full input variable set and utilizing 7 lags and \( l_2\)-norm as well as \( l_1\)-norm equal to \( 10^{-6} \).

<table>
<thead>
<tr>
<th>Ensemble</th>
<th>LSTM5</th>
<th>LSTM10</th>
<th>LSTM15</th>
<th>LSTM20</th>
<th>LSTM30</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.9450***</td>
<td>0.9244***</td>
<td>0.9071***</td>
<td>0.9292***</td>
<td>0.9166***</td>
</tr>
<tr>
<td>5</td>
<td>0.9154***</td>
<td>0.9108***</td>
<td>0.9083***</td>
<td>0.9110***</td>
<td>0.9188***</td>
</tr>
<tr>
<td>10</td>
<td>0.9176***</td>
<td>0.9118***</td>
<td>0.9113***</td>
<td>0.9175***</td>
<td>0.9240***</td>
</tr>
</tbody>
</table>

*Notes: The prediction MSE ratio for the LSTM with low regularization. The significance levels of the tests are indicated by: * 10%, ** 5%, and *** 1%, and tests if the estimates are significantly different from the HAR model’s predictions. The ensembles 1, 5, and 10 refer to the average prediction of the respective top models.*

Rahimikia and Poon (2020b) further found that the LSTM model outperformed their benchmarking HAR model during low volatility but underperformed during high volatility. This again contradicts our findings for all five estimated LSTM models with low regularization, as is presented in Table 12. The results show that the MSE ratio is lower during high volatility conditions compared to low volatility conditions, this holds for estimations using 21 lags as well as 7 lags, and all estimates are significantly different from the HAR model at a 1% significance level. Hence, this suggests a superior out-of-sample performance of all models compared to the benchmark HAR model during high volatility. These findings instead confirm the results and analysis of LSTM conducted by Lu et al. (2022).

However, for low-volatility days the MSE ratio for all estimated LSTM models with low regularization is greater than one, indicating an underperformance compared to the HAR model. All these estimates are in addition significant at the 1% level and hence the findings
suggest that the LSTM model is inferior to the HAR model during almost 70% of the out-of-sample observation, i.e. when volatility is lower than average. This is a striking result and shows the opposite relationship to the one presented by Rahimikia and Poon (2020b).

Table 12

A. Out-of-sample MSE HAR ratio for long short-term memory neural networks prediction during high and low volatility days utilizing 21 lags of all independent variables and regularized with $l_2$-norm as well as $l_1$-norm equal to $10^{-6}$.

<table>
<thead>
<tr>
<th>Ensemble</th>
<th>LSTM5</th>
<th>LSTM10</th>
<th>LSTM15</th>
<th>LSTM20</th>
<th>LSTM30</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.1160***</td>
<td>1.1248***</td>
<td>0.7867***</td>
<td>1.1433***</td>
<td>0.7876***</td>
</tr>
<tr>
<td>5</td>
<td>1.1000***</td>
<td>1.1216***</td>
<td>0.7846***</td>
<td>1.1366***</td>
<td>0.7780***</td>
</tr>
<tr>
<td>10</td>
<td>1.1071***</td>
<td>0.7824***</td>
<td>1.1310***</td>
<td>0.7808***</td>
<td>1.1447***</td>
</tr>
</tbody>
</table>

Notes: Prediction MSE ratio per model and ensemble comparing high versus low volatility days. The t-test examines if the series of MSE ratios during high and low volatility days are significantly different from the HAR model’s predictions. The significance levels of the tests are indicated by: * 10%, ** 5%, and *** 1%. Additionally, all estimates for high days are significantly different from those during low days at the 1% level.

When looking at how the LSTM models with low regularization perform in different volatility regimes at the decile level, as depicted in Figure 7 and in tabular format in Table 13, a pattern emerges. While all models perform better at the first and second decile compared to the third, after the third decile the LSTM model performance consistently increases with increasing decile number. These findings hold for both the LSTM models utilizing 21 lags as well as those utilizing only 7 lags. Hence the findings from the previous comparison between high and low-volatility days are reinforced by the volatility decile analysis.

Another remarkable pattern that can be observed is that this relationship is exacerbated when increasing the model size, indicating a trade-off between model complexity and the forecasting accuracy for changing volatility levels.
Notes: MSE ratios per volatility level grouped by decile. The red dashed line at the level where the MSE ratio is 1 represents the performance of the HAR model.

Table 13

A. Out-of-sample MSE HAR ratio reported per volatility decile for long short-term memory neural networks predictions using the full input variable set, utilizing 21 lags and $l_2$-norm as well as $l_1$-norm equal to 10^6.

<table>
<thead>
<tr>
<th>Decile</th>
<th>LSTM5</th>
<th>LSTM10</th>
<th>LSTM15</th>
<th>LSTM20</th>
<th>LSTM30</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st</td>
<td>1.1596</td>
<td>1.1757</td>
<td>1.1885</td>
<td>1.1893</td>
<td>1.1213</td>
</tr>
<tr>
<td>2nd</td>
<td>1.2197</td>
<td>1.2426</td>
<td>1.2552</td>
<td>1.2694</td>
<td>1.2885</td>
</tr>
<tr>
<td>3rd</td>
<td>1.248</td>
<td>1.2749</td>
<td>1.2808</td>
<td>1.3014</td>
<td>1.3286</td>
</tr>
<tr>
<td>4th</td>
<td>1.1875</td>
<td>1.2239</td>
<td>1.2461</td>
<td>1.2670</td>
<td>1.3021</td>
</tr>
<tr>
<td>5th</td>
<td>1.0654</td>
<td>1.0990</td>
<td>1.1155</td>
<td>1.1371</td>
<td>1.1638</td>
</tr>
<tr>
<td>6th</td>
<td>0.9687</td>
<td>0.9979</td>
<td>1.0138</td>
<td>1.0391</td>
<td>1.0417</td>
</tr>
<tr>
<td>7th</td>
<td>0.8738</td>
<td>0.8970</td>
<td>0.9200</td>
<td>0.9428</td>
<td>0.9538</td>
</tr>
<tr>
<td>8th</td>
<td>0.8522</td>
<td>0.8592</td>
<td>0.8998</td>
<td>0.9248</td>
<td>0.9270</td>
</tr>
<tr>
<td>9th</td>
<td>0.8415</td>
<td>0.8425</td>
<td>0.8685</td>
<td>0.8801</td>
<td>0.8875</td>
</tr>
<tr>
<td>10th</td>
<td>0.8023</td>
<td>0.7563</td>
<td>0.7417</td>
<td>0.7308</td>
<td>0.7324</td>
</tr>
</tbody>
</table>

B. Out-of-sample MSE HAR ratio reported per volatility decile for long short-term memory neural networks predictions using the full input variable set, utilizing 7 lags and $l_2$-norm as well as $l_1$-norm equal to 10^6.

<table>
<thead>
<tr>
<th>Decile</th>
<th>LSTM5</th>
<th>LSTM10</th>
<th>LSTM15</th>
<th>LSTM20</th>
<th>LSTM30</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st</td>
<td>1.1532</td>
<td>1.1776</td>
<td>1.1834</td>
<td>1.2039</td>
<td>1.2083</td>
</tr>
<tr>
<td>2nd</td>
<td>1.2233</td>
<td>1.2412</td>
<td>1.2555</td>
<td>1.2809</td>
<td>1.3005</td>
</tr>
<tr>
<td>3rd</td>
<td>1.2579</td>
<td>1.2738</td>
<td>1.2977</td>
<td>1.3157</td>
<td>1.3454</td>
</tr>
<tr>
<td>4th</td>
<td>1.1819</td>
<td>1.2161</td>
<td>1.2212</td>
<td>1.2296</td>
<td>1.3060</td>
</tr>
<tr>
<td>5th</td>
<td>1.0946</td>
<td>1.1068</td>
<td>1.1304</td>
<td>1.1659</td>
<td>1.1943</td>
</tr>
<tr>
<td>6th</td>
<td>0.9874</td>
<td>0.9842</td>
<td>1.0013</td>
<td>1.0143</td>
<td>1.0463</td>
</tr>
<tr>
<td>7th</td>
<td>0.8742</td>
<td>0.8716</td>
<td>0.8953</td>
<td>0.9196</td>
<td>0.9390</td>
</tr>
<tr>
<td>8th</td>
<td>0.8573</td>
<td>0.8661</td>
<td>0.8982</td>
<td>0.9189</td>
<td>0.9422</td>
</tr>
<tr>
<td>9th</td>
<td>0.8554</td>
<td>0.8474</td>
<td>0.8669</td>
<td>0.8845</td>
<td>0.8953</td>
</tr>
<tr>
<td>10th</td>
<td>0.7943</td>
<td>0.7675</td>
<td>0.7455</td>
<td>0.7338</td>
<td>0.7266</td>
</tr>
</tbody>
</table>

Notes: The prediction MSE ratio per model and decile for ensemble 10 using exogenous data.

As with the FNN models, we do the robustness check for periods 1 and 2. The result for the best-performing model, LSTM10 utilizing 21 lags, is depicted in Figure 8A and the ditto for the models utilizing 7 lags, LSTM15, is depicted in Figure 8B. All the samples from the first to the fifth volatility decile show higher MSE ratio levels compared to the remaining six to ten.
Hence, we can conclude that for the LSTM model, there is a relationship where higher model complexity gives higher prediction accuracy during high-volatility days.

Figure 8
A. MSE ratio per decile comparing LSTM10 with low regularization and utilizing 21 lags predictions Period 1 versus Period 2

B. MSE ratio per decile comparing LSTM15 with low regularization and utilizing 7 lags predictions Period 1 versus Period 2

Notes: A: Comparison of Period 1 and Period 2 of the prediction MSE ratio per decile for model LSTM10. B: Comparison of Period 1 and Period 2 of the prediction MSE ratio per decile for model LSTM15. The red dashed line at the level where the MSE ratio is one represents the performance of the HAR model.

To further examine the trade-off between model complexity and forecasting accuracy between high versus low-volatility days we conduct the regression analysis. The results for the high-volatility days for the LSTMs utilizing 21 lags and the ones utilizing 7 lags are presented in Tables 14A and 14C respectively and ditto for low-volatility days are presented in Tables 14B and 14D.

Comparing the coefficients for all models during high volatility with the ones during low volatility we find that the estimated coefficients for high-volatility days are all negative and the coefficients for low-volatility days are all positive. Further, all negative coefficients for the high volatility sample are statistically significant at the 1% level. In addition, except for the transition from LSTM20 to LSTM30 when estimated including 21 lags, all coefficients during high-volatility days are decreasing in value with an increase in model size.

The reverse relationship is found for the coefficients during low-volatility days where all estimates are increasing with model size. However, the coefficients for LSTM10 using 7 lags are not significant and LSTM15 utilizing the same number of lags are only statistically
significant at the 10% level. Except for these two models, all other coefficients for the low volatility period are significant at the 1% level.

Notably, these results suggest a trade-off relation where the accuracy increases with model complexity during high-volatility days and decreases with model complexity during low-volatility days. Hence these findings confirm the results of Rahimikia and Poon (2020b) that the LSTM model exhibits this trade-off property when used to predict realized volatility.

Table 14

<table>
<thead>
<tr>
<th>Model</th>
<th>Coefficient Std. error</th>
<th>t-value</th>
<th>L. conf.</th>
<th>U. conf.</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSTM0</td>
<td>-0.4354*** 0.0149</td>
<td>0.0270</td>
<td>-0.9235 0.2543</td>
<td></td>
</tr>
<tr>
<td>LSTM15</td>
<td>-0.5170*** 0.0052</td>
<td>0.0259</td>
<td>-0.9780 0.4970</td>
<td></td>
</tr>
<tr>
<td>LSTM20</td>
<td>-0.6447*** 0.0092</td>
<td>0.0230</td>
<td>-0.9850 0.4950</td>
<td></td>
</tr>
<tr>
<td>LSTM25</td>
<td>-0.6969*** 0.0062</td>
<td>0.0230</td>
<td>-0.9850 0.4950</td>
<td></td>
</tr>
</tbody>
</table>

Notes: OLS regression results when using model complexity as a dummy variable during high volatility days for long short-term memory neural networks utilizing 21 lags and $l_2$-norm as well as $l_1$-norm equal to $10^{-6}$.

4.2.2 LSTM with High Regularization.

We replicate the aforementioned estimation procedure for the LSTM model. However, in this case, we elevate both the $l_2$-norm and $l_1$-norm to a value of $10^{-4}$, which we will refer to as high regularization.

The forecasted MSE ratio for LSTMs estimated using high regularization and the full sample for models utilizing 21 lags as well as 7 lags are presented in Table 15A and 15B respectively. The tables show a marginal improvement compared to the estimates under lower regularization and the lowest MSE ratio is achieved by LSTM20 ensemble 10 for both estimations using 21 lags and 7 lags. The best of these models has an estimated out-of-sample improvement over the benchmark HAR model of 9.67%, this is a marginally lower compared to the best-performing FNN model by only 0.26% in units.
Table 15

A. Out-of-sample MSE HAR ratio for long short-term memory neural networks predictions using the full input variable set, utilizing 21 lags and $L_2$-norm as well as $L_1$-norm equal to $10^{-4}$.

<table>
<thead>
<tr>
<th>Ensemble</th>
<th>LSTM5</th>
<th>LSTM10</th>
<th>LSTM15</th>
<th>LSTM20</th>
<th>LSTM30</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.9433***</td>
<td>0.9271***</td>
<td>0.9206***</td>
<td>0.9083***</td>
<td>0.9045***</td>
</tr>
<tr>
<td>5</td>
<td>0.9266***</td>
<td>0.9116***</td>
<td>0.9127***</td>
<td>0.9094***</td>
<td>0.9083***</td>
</tr>
<tr>
<td>10</td>
<td>0.9279***</td>
<td>0.9139***</td>
<td>0.9139***</td>
<td>0.9094***</td>
<td>0.9121***</td>
</tr>
</tbody>
</table>

B. Out-of-sample MSE HAR ratio for long short-term memory neural networks predictions using the full input variable set, utilizing 7 lags and $L_2$-norm as well as $L_1$-norm equal to $10^{-4}$.

<table>
<thead>
<tr>
<th>Ensemble</th>
<th>LSTM5</th>
<th>LSTM10</th>
<th>LSTM15</th>
<th>LSTM20</th>
<th>LSTM30</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.9246***</td>
<td>0.9075***</td>
<td>0.8976***</td>
<td>0.9058***</td>
<td>0.9044***</td>
</tr>
<tr>
<td>5</td>
<td>0.9178***</td>
<td>0.9090***</td>
<td>0.9034***</td>
<td>0.9035***</td>
<td>0.9056***</td>
</tr>
<tr>
<td>10</td>
<td>0.9217***</td>
<td>0.9130***</td>
<td>0.9061***</td>
<td>0.9033***</td>
<td>0.9075***</td>
</tr>
</tbody>
</table>

Notes: The prediction MSE ratio for the LSTM model with 21 lags. The significance levels of the tests are indicated by: * 10%, ** 5%, and *** 1%, and tests if the estimates are significantly different from the HAR model’s predictions. The ensembles 1, 5, and 10 refer to the average prediction of the respective top models.

When comparing high-volatility days forecasting accuracy to low-volatility days, estimated by the LSTM model with high regularization, we find the same relation as with low regularization. That is, higher accuracy during high-volatility days and lower accuracy during low-volatility days, as can be seen in Table 16.

Interestingly, the results show a uniform improvement of MSE ratio during low-volatility days for all LSTM models with high regularization compared to the same models with low regularization, however, the high regularized model still has significantly inferior performance compared to the HAR model indicated by an average MSE ratio above one and a significance at the 1% level.

As with the FNN models we observe a small deterioration of forecasting accuracy during high volatility when the regularization is increased for the LSTM model, this holds for all model complexities. Notably, even though we observe this decrease in accuracy during high-volatility days the performance overall is higher for the highly regularized LSTM models. Hence, the improved accuracy during low-volatility days more than compensate for the loss in accuracy under high-volatility conditions.
These findings reinforce the results presented for the FNN models, that increasing regularization reduces the out-of-sample forecasting accuracy during high-volatility days.

Table 16

A.

Out-of-sample MSE HAR ratio for long short-term memory neural networks prediction during high and low volatility days utilizing 21 lags of all independent variables and regularized with $l_2$-norm as well as $l_1$-norm equal to $10^4$.

<table>
<thead>
<tr>
<th>Ensemble</th>
<th>LSTM5</th>
<th>LSTM10</th>
<th>LSTM15</th>
<th>LSTM20</th>
<th>LSTM30</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>High days</td>
<td>Low days</td>
<td>High days</td>
<td>Low days</td>
<td>High days</td>
</tr>
<tr>
<td>1</td>
<td>0.8412***</td>
<td>1.1117***</td>
<td>0.8230***</td>
<td>1.1032***</td>
<td>0.8062***</td>
</tr>
<tr>
<td>5</td>
<td>0.8208***</td>
<td>1.0975***</td>
<td>0.8036***</td>
<td>1.0888***</td>
<td>0.8001***</td>
</tr>
<tr>
<td>10</td>
<td>0.8247***</td>
<td>1.0946***</td>
<td>0.8059***</td>
<td>1.0902***</td>
<td>0.8006***</td>
</tr>
</tbody>
</table>

Notes: Prediction MSE ratio per model and ensemble comparing high versus low volatility days. The $t$-test examines if the series of MSE ratios during high and low volatility days are significantly different from the HAR model’s predictions. The significance levels of the tests are indicated by: * 10%, ** 5%, and *** 1%. Additionally, all estimates for high days are significantly different from those during low days at the 1% level.

The results from the regression analysis to detect the aforementioned complexity-accuracy trade-off for high-volatility days, show that all coefficients are negative and significant at the 1% level, meaning that all the other models exhibit greater out-of-sample performance than the LSTM5 model, as shown in Tables 17A and 17C. There are differences compared to the coefficients estimated for the LSTM models with low regularization during high-volatility days. Notably, the forecasting accuracy increases with model complexity up until the best-performing model, LSTM20, then it decreases again for the greater model. This is hence intuitive, the best-performing model in the full sample is also the best-performing model during high-volatility days.

During low-volatility days, the highly regularized LSTM models show further differences compared to the LSTM models estimated with low regularization. As shown in Tables 17B and 17D none of the regression coefficients are significant, indicating no improvement in out-of-sample forecasting accuracy during low-volatility days when increasing the model complexity beyond five nodes.

Hence, the trade-off relation between model complexity and forecasting accuracy during high versus low-volatility days as we observed for low regularized LSTMs and as reported by
Rahimikia and Poon (2020b) vanishes in our setting when increasing the regularization from $10^{-6}$ to $10^{-4}$.

### Table 17

A. OLS regression result using model complexity as dummy variable during high volatility days for long short-term memory neural networks utilizing 21 lags and $l_{-\infty}$-norm as well as $l_{-\infty}$-norm equal to $10^{6}$.

<table>
<thead>
<tr>
<th>Model</th>
<th>Coefficient</th>
<th>Std. error</th>
<th>t-value</th>
<th>L. conf.</th>
<th>U. conf.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conv.</td>
<td>0.8667***</td>
<td>0.0299</td>
<td>28.7774</td>
<td>0.7421</td>
<td>0.9814</td>
</tr>
<tr>
<td>LSTM0</td>
<td>-0.3079***</td>
<td>0.0049</td>
<td>-4.0239</td>
<td>-0.5342</td>
<td>-0.1817</td>
</tr>
<tr>
<td>LSTM15</td>
<td>-0.5322***</td>
<td>0.0049</td>
<td>-5.9939</td>
<td>-0.7094</td>
<td>-0.3569</td>
</tr>
<tr>
<td>LSTM20</td>
<td>-0.7108***</td>
<td>0.0049</td>
<td>-7.9903</td>
<td>-0.8870</td>
<td>-0.5345</td>
</tr>
<tr>
<td>LSTM25</td>
<td>-0.6007***</td>
<td>0.0049</td>
<td>-6.7536</td>
<td>-0.7770</td>
<td>-0.4245</td>
</tr>
</tbody>
</table>

B. OLS regression result using model complexity as dummy variable during low volatility days for long short-term memory neural networks utilizing 21 lags and $l_{-\infty}$-norm as well as $l_{-\infty}$-norm equal to $10^{4}$.

<table>
<thead>
<tr>
<th>Model</th>
<th>Coefficient</th>
<th>Std. error</th>
<th>t-value</th>
<th>L. conf.</th>
<th>U. conf.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conv.</td>
<td>0.5197***</td>
<td>0.0796</td>
<td>6.5299</td>
<td>0.3620</td>
<td>0.6774</td>
</tr>
<tr>
<td>LSTM0</td>
<td>-0.1002</td>
<td>0.1126</td>
<td>-0.9434</td>
<td>-0.3292</td>
<td>0.5169</td>
</tr>
<tr>
<td>LSTM15</td>
<td>0.0654</td>
<td>0.1126</td>
<td>0.5807</td>
<td>-0.1577</td>
<td>0.2884</td>
</tr>
<tr>
<td>LSTM20</td>
<td>-0.0291</td>
<td>0.1126</td>
<td>-0.2584</td>
<td>-0.2523</td>
<td>0.2940</td>
</tr>
<tr>
<td>LSTM25</td>
<td>0.0495</td>
<td>0.1126</td>
<td>0.4394</td>
<td>-0.1736</td>
<td>0.2725</td>
</tr>
</tbody>
</table>

C. OLS regression result using model complexity as dummy variable during high volatility days for long short-term memory neural networks utilizing 7 lags and $l_{-\infty}$-norm as well as $l_{-\infty}$-norm equal to $10^{6}$.

<table>
<thead>
<tr>
<th>Model</th>
<th>Coefficient</th>
<th>Std. error</th>
<th>t-value</th>
<th>L. conf.</th>
<th>U. conf.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conv.</td>
<td>0.8485***</td>
<td>0.0113</td>
<td>13.7147</td>
<td>0.7189</td>
<td>0.9618</td>
</tr>
<tr>
<td>LSTM0</td>
<td>-0.3282***</td>
<td>0.0087</td>
<td>-3.7876</td>
<td>-0.4999</td>
<td>-0.1265</td>
</tr>
<tr>
<td>LSTM15</td>
<td>-0.5040***</td>
<td>0.0087</td>
<td>-5.1048</td>
<td>-0.6758</td>
<td>-0.3523</td>
</tr>
<tr>
<td>LSTM20</td>
<td>-0.7025***</td>
<td>0.0087</td>
<td>-8.1070</td>
<td>-0.8742</td>
<td>-0.5368</td>
</tr>
<tr>
<td>LSTM25</td>
<td>-0.4784***</td>
<td>0.0087</td>
<td>-5.5207</td>
<td>-0.6591</td>
<td>-0.3067</td>
</tr>
</tbody>
</table>

D. OLS regression result using model complexity as dummy variable during low volatility days for long short-term memory neural networks utilizing 7 lags and $l_{-\infty}$-norm as well as $l_{-\infty}$-norm equal to $10^{4}$.

<table>
<thead>
<tr>
<th>Model</th>
<th>Coefficient</th>
<th>Std. error</th>
<th>t-value</th>
<th>L. conf.</th>
<th>U. conf.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conv.</td>
<td>0.4603***</td>
<td>0.0812</td>
<td>5.6088</td>
<td>0.2994</td>
<td>0.6213</td>
</tr>
<tr>
<td>LSTM0</td>
<td>0.0904</td>
<td>0.1148</td>
<td>0.7870</td>
<td>-0.1372</td>
<td>0.3180</td>
</tr>
<tr>
<td>LSTM15</td>
<td>0.0049</td>
<td>0.1148</td>
<td>0.0427</td>
<td>-0.2227</td>
<td>0.2225</td>
</tr>
<tr>
<td>LSTM20</td>
<td>0.0965</td>
<td>0.1148</td>
<td>0.8402</td>
<td>-0.1311</td>
<td>0.3241</td>
</tr>
<tr>
<td>LSTM25</td>
<td>-0.0073</td>
<td>0.1148</td>
<td>-0.0652</td>
<td>-0.2148</td>
<td>0.2203</td>
</tr>
</tbody>
</table>

Notes: OLS regression results when using model complexity as an independent variable and MSE ratio as a dependent variable. The results are shown for LSTM models with high regularization during high and low volatility respectively. The significance levels of the tests are indicated by: * 10%, ** 5%, and *** 1%.

### 4.3 Main Findings

Our findings indicate that the ANN models in general outperform the linear HAR model in the full sample when using the full dataset. Our results also suggest that ANN models in general have better performance on high-volatility days than on low-volatility days, compared to the benchmark HAR model. Hence, we reject hypothesis 1a in favor of hypothesis 1b, and we conclude that in our setting artificial neural networks are more accurate during high-volatility days than during low-volatility days compared to the HAR model, when forecasting daily realized volatility.

Further, we find the best-performing FNN models to outperform the benchmark during both high and low volatility. In addition, our results point to the best LSTM models significantly outperforming the benchmark HAR model during days when volatility is higher than average while, importantly, underperforming during days when volatility is lower than average. These results hold for both low and high regularization regimes.

However, when moving from the low regularization regimes to the high, we found that the forecasting accuracy for high-volatility days decreased while the forecasting accuracy increased during low-volatility days, compared to the HAR model. This was evident for both...
FNNs and LSTMs. This leads us to the rejection of hypothesis 2a and instead accept hypothesis 2b which states that *increasing the regularization of artificial neural networks inhibits their out-of-sample forecasting ability during high-volatility days compared to the linear HAR model while supporting their accuracy during low-volatility days.*

When conducting the regression analysis to detect any trade-off between the artificial neural network model complexity and the ability to predict realized volatility during high versus low-volatility days the outcome gave us mixed results. For FNNs, we could not find any indication of such trade-off relations, neither under low nor high regularization. Nevertheless, the trade-off was obvious for the LSTM model under low regularization. However, this phenomenon dissipated when increasing the regularization. Hence, we accept hypothesis 3a, *there is a trade-off between the artificial neural network model’s complexity and the ability to predict realized volatility during high versus low-volatility days* for less regularized LSTM models. However, we reject it for highly regularized ones and for FNN models and thus instead accept hypothesis 3b, *there is NOT a trade-off between the artificial neural network model’s complexity and the ability to predict realized volatility during high versus low-volatility days,* for those models.

Notably, our findings show that the two best-performing ANN models in our study, LSTM20 with high regularization and FNN3 with low regularization, show virtually the same performance in terms of MSE ratio for the full out-of-sample predictions. However, when comparing the two models' performance during high and low volatility we observe the LSTM model to outperform the HAR model by an estimated 7.97% in units during high volatility while during low-volatility days the FNN outperformed by 15.36% in units.
5. Discussion

5.1 Discussion of Findings

Our findings follow the broader literature and suggest that artificial neural networks are superior to traditional linear econometric models such as HAR in forecasting daily realized volatility. This confirms previous observations of a nonlinear relation between different financial variables and volatility.

Our results indicate that ANNs in general, and especially LSTM models, have higher forecasting performance during higher-than-average volatility conditions compared to lower-than-average volatility conditions. This is in line with Lu et al. (2022) findings, who reported this relation for both FNNs and LSTMs. We also find the ANN model's best performance to be at the top decile of volatility, this in contradiction to Rahimikia and Poon (2020b) and Christensen et al. (2022). Therefore, our findings are a contribution to the current contradicting literature in the field and previous studies have not managed to deliver concrete universal explanations for the contradicting results. Christensen et al. (2022) suggested that the differing accuracy between deciles could be related to the regularization of the ANN model. As we investigated this relation further the findings show that the gap between the forecasts produced for high and low-volatility days narrows with a reasonable increase in regularization. However, this effect is not large enough to alter the mutual relationship, and increasing the regularizers further would inhibit the forecasting performance to a level where the action is not feasible. Hence, we cannot contribute with an explanation for the contradiction more than suggesting it does not solely lay in the regularization of artificial neural networks.

Christensen et al. (2022) proposed that the main driver of the increased forecasting performance of artificial neural networks compared to the linear HAR model during medium-to-high volatility conditions is the additional use of regularizers. Hence, alluding to the conclusion that underperformance by linear models in comparison to ANN models during medium-to-high-volatility days is due to an increasing overfitting problem when volatility rises. However, our results show that increasing the regularization of ANNs inhibits their ability to forecast realized volatility during high-volatility days while at the same time increasing the performance during low-volatility days.

This showcases that the main driver of superior performance during medium-to-high volatility compared to the linear model does not seem to be the more rigorous regularization for these
models. Instead, our results indicate towards the other thesis of Christensen et al. (2022), that the increasing nonlinearity during higher volatility is the cause of the improved performance shown by ANNs during medium-to-high volatility conditions. It further confirms that the reasoning behind increasing the regularization will disregard the smaller sub-sample with high volatility as outliers and focus on a close fit to the majority of the sample, leading to an increase in accuracy during low-volatility days.

Rahimikia and Poon (2020b) report a trade-off relationship between the complexity in the structure of an artificial neural network and the ability to predict realized volatility during high versus low-volatility days. We only find weak support for the trade-off relation since this was only confirmed for the sequential LSTM model with low regularization. Interestingly, when analyzing the same model with higher regularization, the trade-off relation was eliminated. This points to the intrinsic relation between model complexity and regularization inherent in the tradition of using artificial neural networks, where model tuning and optimization is a major part of the practitioner’s tasks as pointed out by Goodfellow et al. (2016).

Furthermore, this aligns with the observation that the two most successful ANN models in our study, namely LSTM20 with high regularization and FNN3 with low regularization, exhibit virtually identical performance in terms of MSE ratio for the complete out-of-sample predictions. However, when evaluating the performance of these models during high and low volatility periods, we note that the LSTM model outperforms the FNN model during high-volatility days, while the FNN model outperforms the LSTM model during low-volatility days. Hence, this indicates that ANNs can be adjusted to specialize in specific volatility conditions and hence be customized to different stakeholders’ needs and objectives. This could be a future advantage of the ANN models in forecasting realized volatility, enabling tailor-made models where investors or institutions concerned with tail risk would tune their model in that direction while other market participants, more interested in fine-tuning volatility predictions during calm markets, could go down that route.

In addition, our finding further problematizes and indicates the weakness of the current financial literature's use of, what is called “off-the-shelf” neural network models or applying standard values for regularizers from “the subject literature”. These practices, as shown in our study, can have a severe effect on the performance of the model, not only on the overall performance but on the variations among sub-samples. These results suggest a more explicit
discussion of the specific regularizers employed and greater attention to their effect on the resulting outcome in the financial literature.

5.2 Limitations

This study was conducted in the realm of a Master’s thesis, resulting in limitations in the scope and depth of the investigation due to time and resource constraints. The data available to us was limited to the Nordic region and from that sample we chose a limited number of 23 stocks on the basis of inclusion in the OMXS30 index. This was due to mitigating the effect of uninformative volatility as an effect of illiquidity linked to individual stocks. However, the Swedish market is a modern and effective market and this sample selection should not introduce any biases. Lastly, when building and tuning artificial neural networks there is an infinite amount of possible parameter values that need to be pre-specified before training. We have approached this with caution and used standard practice from the literature and previous studies on the subject to decide upon these starting settings.
6. Conclusion

In this study we investigate how two types of artificial neural network models (ANN), feedforward neural networks (FNN) and long short-term memory models (LSTM) behave when predicting daily realized volatility (RV), as compared to a benchmark HAR model during differing volatility regimes, as well as investigating the relationship between model complexity, different regularization regimes and accuracy during high and low volatility. This is done for 23 stocks, constituents of the stock index OMX Stockholm 30 over the period 8th of February 2010 to 31st of January 2022, by utilizing ten exogenous and three endogenous input variables, earlier used in the literature. As a robustness check, the RV is predicted by using HAR and by the ANNs, using only the HAR variables. The full data set, including the exogenous variables, is then used for predicting the RV using HAR-X as well as the ANNs.

Our study contributes to the ongoing debate regarding the superiority of artificial neural networks (ANNs) over traditional linear econometric models in forecasting daily realized volatility. In conclusion, our findings support the existing literature, indicating that artificial neural networks (ANNs) outperform traditional linear econometric models, such as HAR, in forecasting daily realized volatility. This confirms the nonlinear relationship observed between financial variables and volatility in previous studies.

Artificial neural networks do differ in accuracy between varying volatility levels, both compared to linear models and between neural network structures. We also conclude that the volatility forecasting accuracy during high versus low volatility conditions is dependent on the level of regularization of the model. Where lower regularization supports enhanced accuracy during high-volatility days while higher regularization promotes performance during low-volatility days. This suggests that the improved performance of ANNs during high volatility conditions is likely attributed to the increasing nonlinearity with rising volatility.

Moreover, we find support for the trade-off relationship between model complexity and the ability to predict realized volatility during high and low-volatility days. This trade-off relation is however only confirmed for the sequential LSTM model with low regularization and is eliminated when increasing the regularization. This highlights the intrinsic connection between model complexity and regularization in ANN models, where fine-tuning and optimization play significant roles.
Furthermore, our study revealed that models demonstrating identical forecasting performance in the full sample can perform widely differently in sub-sections based on volatility level. This suggests that ANNs can be tailored to specific volatility conditions, catering to the needs and objectives of different stakeholders by consciously adjusting the model structure and regularization regime.

Hence we suggest that future research in volatility forecasting with artificial neural networks focus on how and why the regularization regime affects forecast and aim to build a theoretical foundation for this topic. As the contemporaneous feedforward neural network structure is shown to learn a different process compared to the long short-term memory model from the training data, further research suggestions are to investigate the theoretical implications for using the two models.
References


on artificial intelligence and statistics (pp. 249-256). JMLR Workshop and Conference Proceedings.


