Soft Computing
https://doi.org/10.1007/s00500-022-07161-1

DATA ANALYTICS AND MACHINE LEARNING

# Deep learning for volatility forecasting in asset management

**Alessio Petrozziello[1] · Luigi Troiano[2] · Angela Serra[3] · Ivan Jordanov[4] · Giuseppe Storti[5] · Roberto Tagliaferri[2] · Michele La Rocca[5]**

Accepted: 6 April 2022


**Abstract**
Predicting volatility is a critical activity for taking risk- adjusted decisions in asset trading and allocation. In order to provide effective decision-making support, in this paper we investigate the profitability of a deep Long Short-Term Memory (LSTM) Neural Network for forecasting daily stock market volatility using a panel of 28 assets representative of the Dow Jones Industrial Average index combined with the market factor proxied by the SPY and, separately, a panel of 92 assets belonging to the NASDAQ 100 index. The Dow Jones plus SPY data are from January 2002 to August 2008, while the NASDAQ 100 is from December 2012 to November 2017. If, on the one hand, we expect that this evolutionary behavior can be effectively captured adaptively through the use of Artificial Intelligence (AI) flexible methods, on the other, in this setting, standard parametric approaches could fail to provide optimal predictions. We compared the volatility forecasts generated by the LSTM approach to those obtained through use of widely recognized benchmarks models in this field, in particular, univariate parametric models such as the Realized Generalized Autoregressive Conditionally Heteroskedastic (R-GARCH) and the Glosten–Jagannathan–Runkle Multiplicative Error Models (GJR-MEM). The results demonstrate the superiority of the LSTM over the widely popular R-GARCH and GJR-MEM univariate parametric methods, when forecasting in condition of high volatility, while still producing comparable predictions for more tranquil periods.

**Keywords** Deep learning · LSTM · Time series · Forecasting · Volatility

## 1 Introduction

Asset management decisions made by investors with global portfolios of equities, bonds and other asset management instruments are driven by the fundamental principle of maximizing the expected return on a given level of risk. These decisions include the allocation of assets through trading in asset management instruments subject to price changes in order to obtain the best combination of risk and reward (known as the risk-reward trade-off) that takes into account the investor's specific assets and objectives.

There is no doubt that the decision on how to invest heritage lies primarily in the ability of human judgment to recognize opportunities for return based on the economic, social and political context and the events that condition its evolution. The altered context translates into a change in the attractiveness of asset management assets, i.e., in their value and in turn in the price that can lead to gains or losses.

✉ Alessio Petrozziello
a.petrozziello@ucl.ac.uk

Luigi Troiano
ltroiano@unisa.it

Angela Serra
angela.serra@tuni.fi

Ivan Jordanov
ivan.jordanov@port.ac.uk

Giuseppe Storti
storti@unisa.it

Roberto Tagliaferri
robtag@unisa.it

Michele La Rocca
larocca@unisa.it

[1] Department of Computer Science, University College London, London, UK

[2] DISA-MIS, University of Salerno, Fisciano (SA), Italy

[3] Faculty of Medicine and Health Technology, Tampere University, Tampere, Finland

[4] School of Computing, University of Portsmouth, Portsmouth, UK

[5] DISES, University of Salerno, Fisciano (SA), Italy

Published online: 15 July 2022

Another element that helps to change the price is those factors that specifically affect a company, a sector or even a country. Prudent management relies on decisions to diversify investments between assets with a negative price correlation, i.e., assets whose price tends to move in opposite directions. In this way, any losses in value on certain assets can be offset at least in part by returns on other assets. Finally, trading activity and the additional element contribute significantly to the price fluctuations of an asset. It reflects the ability to cross supply and demand on the asset management markets.

Therefore, risk, return and asset correlation are the key measures used in asset management models. Quantifying the potential loss of assets is a major part of risk management, trading in asset management markets and asset allocation. To be able to measure these losses and make informed investment decisions, investors need to estimate risks (Blume 1971). Since volatility has some well-known statistical regularities that make it inherently forecastable, it is among one of the most accepted and used measures of risk in the financial market. These regularities include the volatility clustering effect, leading to positive and persistent auto-correlations of volatility measures, the leverage effect, which is related to the negative correlation between past returns and current volatility values, and the dynamic cross-correlation between the volatilities of different assets that give rise to the well-known phenomenon of volatility spillovers. In addition, it is worth remembering that volatility is a key ingredient for computing more refined risk measures such as the Value at Risk or the Expected Shortfall.

Predicting volatility is challenging task, where modern artificial intelligence can come at rescue. Conventional techniques, mostly based on GARCH modeling and its variant, are not able to consider a market as a whole, thus volatility spillovers. In this paper, we aim to show that deep learning can help to build models of volatility forecasting, discussing properties and experimental results. In particular, the paper focuses on LSTM as an effective tool for the purpose of estimating the forecasting volatility.

In the evaluation of volatility forecasts, identifying the underlying market regime is of fundamental importance, as generating accurate predictions is more important in periods of higher volatility of the stock markets rather than in more tranquil periods. In fact, in periods of greater price variability, such as those that accompany and follow periods of economic and financial crisis, the timing risk in trading operations exposes to a greater risk of large losses. It is, at this stage, important to be aware that the forecasting performance of different approaches to volatility forecasting can be highly dependent on the market regime, as identified in terms of the corresponding long-run volatility level. An obvious consequence is that the (ex-ante) identification of the best performing model could, and should, take into account the underlying volatility level.

Despite the significant progresses in the field of financial econometrics, see, e.g., Hamilton and Susmel (1994) and Kim and Kim (1996), and artificial intelligence, this task is still very complex and expert judgment still plays a fundamental role in recognizing the factors that anticipate and characterize moments of high uncertainty in the markets. This is due to the fact that, adequately supported by the outcome provided by quantitative forecasting models, human intervention is able to convey heterogeneous sources of information, of both quantitative and qualitative nature, that would be otherwise hardly embedded in any formalized quantitative forecasting approach.

In this paper we propose a model for volatility prediction that is based on LSTM. Comparison to existing approaches, namely R-GARCH and GJR-MEM, together with conventional RNN, is performed with respect to different volatility regimes, in order to point out how the identification of the market volatility regime is important for identification of the optimal forecasting model.

LSTM effectiveness is showcased by using a panel of 28 assets representative of the Dow Jones Industrial Average index combined with the market factor proxied by the SPY and, separately, a panel of 92 assets belonging to the NASDAQ 100 index. The Dow Jones plus SPY data are from January 2002 to August 2008, while the NASDAQ 100 is from December 2012 to November 2017. Both markets, in the periods considered, were affected by extremely critical events as the global financial crisis (GFC) of 2007–2008 and the consequences of the European debt crisis 2011–2012, which led to changes in the level and dynamics of market volatility during and after the crisis.

The remainder of the paper is structured as follows. An overview of related literature is given in Sect. 2. The data used in the empirical application and the investigated techniques are described in Sect. 3.1. The LSTM approach and the related experimental setup are briefly discussed in Sects. 3.2 and 3.3. The two parametric competitors used as benchmarks in our empirical analysis, i.e., the R-GARCH and GJR-MEM, are presented in Sects. 3.4 and 3.5, respectively, while Sect. 3.6 introduces the two loss functions used for training the models and the metrics for out-of-sample forecast evaluation. The empirical results of the out-of-sample forecasting comparison are presented and discussed in Sect. 4, followed by the comparison with Recurrent Neural Networks in Sect. 5. Conclusions are given in Sect. 6.

## 2 Related work

As the prediction of volatility is a major factor in risk analysis, many efforts have been made to implement parametric as well as nonparametric predictive methods for forecasting future volatility. Depending on the reference information

set, the proposed approaches can be classified into two broad categories. Of these, the first includes approaches fitted to time series of daily log-returns. In the parametric world, this class of methods includes the Generalized Autoregressive Conditionally Heteroskedastic (GARCH) models (Bollerslev 1986) and their numerous univariate and multivariate recent extensions (Bauwens et al. 2006; Gerlach and Wang 2016; Huang et al. 2017; Wang et al. 2018). In the nonparametric world, we recall a consistent number of papers applying nonparametric approximators of squared returns (such as smoothing splines Langrock et al. 2015; Zhang and Teo 2015 or neural networks to time series Fischer and Krauss 2018; Kourentzes et al. 2014; Wang et al. 2015) which provides an unbiased but noisy volatility proxy.

The second and more recent class of approaches for volatility forecasting replaces this noisy volatility proxy with more efficient *realized volatility* measures (Andersen and Teräsvirta 2009) built from time series of high-frequency asset prices. Notable examples of parametric models falling into this second class are the Multiplicative Error Models (MEM) Engle and Russell (1998) and the Realized GARCH (R-GARCH) Hansen et al. (2012) techniques. The main structural difference between R-GARCH and MEM models is that R-GARCH uses bivariate information on log-returns and realized volatility, whereas MEM are directly fitted to a univariate realized volatility series—log-returns are eventually used only as external regressors for capturing leverage effects. Similarly, in a nonparametric environment, neural networks (McAleer and Medeiros 2011) or other nonparametric filters (Chen et al. 2018) can be applied to time series of realized volatility measures to forecast future volatility. See also Han and Zhang (2012) nonparametric volatility modeling for non-stationary time series.

Several extensions have been proposed to parametric models. For instance, in Andersen et al. (2007) the authors suggest that the total return variation process can be separated in jumps and non-jumps movements, where almost all of the predictability in daily, weekly, and monthly return volatilities comes from the non-jump component. Building on this idea, Maciel et al. (2017) investigated evolving possibilistic fuzzy modeling to forecast realized volatility with jumps. Their possibilistic model improves robustness to noisy data and outliers, which is an essential requirement in financial markets volatility modeling and forecasting.

All above-discussed approaches are univariate. Nevertheless, the presence of phenomena such as common features and volatility spillovers make the analysis of multivariate volatility panels potentially very profitable. In a parametric setting, however, the number of required model parameters explodes rapidly as the cross-sectional dimension of the panel increases, making the estimation unfeasible even for moderately large dimensions, unless some heavy, and untested parametric restrictions are imposed.

Typically, it is often assumed that all the volatilities in the panel share the same dynamic dependence structure and volatility spillovers are not present (Pakel et al. 2011). Those assumptions are clearly unrealistic and they greatly reduce the ability of the parametric models, albeit multivariate, to describe the complexity of the dynamic structure which is observed in financial time series. These considerations contribute to the scarce attention that multivariate models for volatility panels have received in the literature on parametric modeling of financial time series.

Feed-forward neural networks are a favorite class of multivariate, nonparametric models used to study dependencies and trends in the data, e.g., using multiple inputs from the past to predict the future time step (Chakraborty et al. 1992). However, when using traditional neural network models, much effort is devoted to making sure that what is presented for training in the input layer is already in a format that allows the network to recognize the significant patterns ("feature engineering"). This process usually requires some ad-hoc procedures and soon becomes one of the most time-consuming parts of neural network modeling. In a deep learning framework instead, by adding more and more layers between input and output (hence "deep"), the model allows richer intermediate representations to be built and most of the feature engineering process can be achieved by the algorithm itself, in an almost automatic fashion. This latter point improves prediction accuracy and strongly widens the domains of applications. As a drawback, deep learning models require a large amount of data to outperform other approaches and are computationally expensive to train. However, in financial applications, a large amount of data can be quickly gathered, making deep learning applications appropriate and viable.

The main focus of this paper is on proving the effectiveness of using deep learning techniques for multivariate volatility forecasting. In particular, a deep Long Short-Term Memory (LSTM) Neural Network (Hochreiter and Schmidhuber 1997) is applied. LSTMs can have several advantages compared to the modeling approaches used so far in the literature. The application of this class of models to multivariate volatility forecasting is particularly appealing since it allows to overcome the curse of dimensionality typically limiting the application of complex multivariate parametric models.

Firstly, they can be seen as nonparametric statistical models, and consequently, they do not suffer from the misspecification problems which typically affect parametric modeling strategies. Secondly, they can overcome the curse of dimensionality problem which affects both standard nonparametric estimation techniques and several multivariate parametric models (Poggio et al. 2017). This makes the use of LSTMs feasible even for high dimensional temporal datasets. Moreover, they do not require any undesirable reduction of the parameter space through untestable, unrealistic restric-

tions, which, otherwise, might significantly reduce the ability of the model to reveal well-known stylized facts about multivariate financial time series (such as spillovers and complex dynamics). Finally, they can benefit from modeling complex nonlinear and long-term dependencies, leading to improved accurate predictions (Gers et al. 2002).

One of the main benefits of using a recurring network such as LSTM is the ability to make the prediction adaptive to time. The concept of internal state/memory translates into an implicit stretching or warping of the time axis. This detail is well known to those involved in speech analysis, since the correspondence between patterns must necessarily take into account an inevitable misalignment along the time axis. In the financial domain we are dealing with in this paper, this translates into the need to consider a misalignment of the time series of asset prices or volatility in a market, due to market inefficiencies or propagation between sectors. To solve this problem, one of the most popular techniques is dynamic time warping (DWT) (Itakura 1975). This aspect has been very recently taken up and generalized by Rivest and Kohar (2020). In their work, they propose squared timing error (STE) as a new cost function of timing error that incorporates the principles of DWT. Among the various experiments, they also consider a price prediction problem in finance. Their method focuses on binary time series. In this case, they try to determine when a price exceeds a certain threshold. In the experiment they assume the closing prices of the shares that make up the NASDAQ-100 index in the period from April 1, 2013 until March 31, 2014. They calculate the 14-day moving average, assuming as threshold 0.7 of historical volatility in the same period, i.e., the 14-day moving standard deviation.

However, the advantage of using a recurring network, whether LSTM or any other architecture, is precisely in incorporating propagation effects into hidden state units. The question of how to make the internal memory of a recurring network effective has been the focus of many recent developments. Yu et al. (2019) offer an extensive overview of them. Besides the pioneering work of Hochreiter and Schmidhuber (1997) that led to the definition of LSTM, it is worth to mention the contributions given by Cho et al. (2014) regarding the Gated Recurrent Unit (GRU), by Kalchbrenner et al. (2015) for Grid LSTM, by Shi et al. (2015) for Convolutional LSTM. A different approach is fostered by Graves et al. (2014) for their Neural Turin Machine, suggesting to make explicit into the architecture an external addressable memory. Following this idea, one of the latest developments in this area is offered by Quan et al. (2020), where they propose to equip a RNN with an external addressable working memory (EAWM). This makes short and long-term information explicit and directly manipulable, letting SGD to train the network end-to-end.

Following a similar approach, Nápoles et al. (2020) propose long-term cognitive network (LTCN) as a neural cognitive mapping technique able to store long-term dependencies between input and output sequences, especially in a context where values of several dependent variables have to be predicted. The approach consists in preserving the knowledge of the expert encoded in a matrix of weights, trying to optimize the nonlinear relationship offered by the activation function of each neuron. In this sense, the training algorithm is called non-synaptic back-propagation.

Most of the above work focuses on sequence analysis in natural language processing problems, in particular machine translation, text comprehension and text generation. Although LSTM is now a tool made available to finance (e.g., see Troiano et al. 2018), at the best of our knowledge the method proposed in this paper is novel as no attempt to specify and estimate a comprehensive nonparametric volatility forecasting model for the whole market has been made so far. In particular, nobody approached the problem using Deep Learning (i.e., LSTM).

## 3 Materials and methods

As mentioned above, the methodology proposed in this work is based on the application of Long Short-Term Memory (LSTM) Neural Network for the forecast of the volatility of the assets at a particular time, given its past values.

### 3.1 Data

Two datasets are used for the empirical experimentation. The first dataset used by Hansen et al. (2012), includes 28 assets from the Dow Jones Industrial Average (DJI 500) index plus one exchange-traded index fund SPY, that tracks the S&P 500 index. The sample spans the period from 1st January 2002 to 31st August 2008, including 1549 trading days. The second dataset is related to 92 stocks belonging to the NASDAQ 100 index within the period 1st December 2012 to 29th November 2017, for a total of 1256 trading days. Further detail on the two datasets can be found in Tables 1 and 2, respectively[1].

Each asset is represented by two different time series, the realized measure, namely volatility, and the related open–close return. The realized measure $v_t$ is given by a realized kernel estimator computed using the Parzen kernel function. This estimator is similar to the realized variance, and more importantly, it is robust to market micro-structure noise and is more accurate than the quadratic variation estimator. The implementation of the realized kernel follows the method

[1] The data are publicly available on the following Zenodo repository https://zenodo.org/record/2540818.

**Table 1** Dow Jones industrial average assets used as case study

| Symbol | Name | Sector | Capitalization (USD) |
|---|---|---|---|
| AA | Alcoa Corp | Materials | 6.89B |
| AIG | American International Group | Financials | 58.79B |
| AXP | American Express | Financials | 75.99B |
| BA | Boeing | Industrials | 140.50B |
| BAC | Bank of America | Financials | 245.97B |
| C | Citigroup | Financials | 187.94B |
| CAT | Caterpillar | Industrials | 67.58B |
| CVX | Chevron | Energy | 208.66B |
| DD | EI du Pont de Nemours | Materials | 71.17B |
| DIS | Disney | Consumer Discretionary | 168.52B |
| GE | General Electric | Industrials | 223.20B |
| GM | General Motors | Consumer Discretionary | 51.39B |
| HD | Home Depot | Consumer Discretionary | 182.62B |
| IBM | IBM | Information Technology | 135.28B |
| INTC | Intel | Information Technology | 170.57B |
| JNJ | Johnson & Johnson | Health Care | 357.45B |
| JPM | JPMorgan Chase | Financials | 329.58B |
| KO | Coca-Cola | Consumer Staples | 194.07B |
| MCD | McDonald's | Consumer Discretionary | 125.37B |
| MMM | 3M | Industrials | 123.92B |
| MRK | Merck | Health Care | 172.59B |
| MSFT | Microsoft | Information Technology | 559.80B |
| PG | Procter & Gamble | Consumer Staples | 231.51B |
| T | AT&T | Telecommunication Services | 235.96B |
| UTX | United Technologies | Industrials | 97.04B |
| VZ | Verizon | Telecommunication Services | 199.52B |
| WMT | Wal-Mart | Consumer Staples | 242.61B |
| XOM | Exxon Mobil | Energy | 339.86B |
| SPY | SPDR S&P500 ETF Trust | – | (Net Assets) 242,54B |

Capitalization is given with respect to 4th August 2017 values

proposed by Barndorff-Nielsen et al. (2011) that guarantees a positive estimate.

### 3.2 Long short-term memory network

Long Short-Term Memory (LSTM) Network (Gers et al. 1999; Hochreiter and Schmidhuber 1997) is a Recurrent Neural Network (RNN) architecture that acts as a Universal Turing Machine learner: given enough units to capture the state and a proper weighting matrix to control its evolution, the model can replicate the output of any computable function.

Because of this noticeable characteristic, LSTM is largely employed in tasks of sequence processing, e.g., in natural language processing (Sundermeyer et al. 2015; Tran et al. 2016; Yao et al. 2014), speech recognition (Graves et al. 2013; Han et al. 2017; Suwajanakorn et al. 2017), automatic control (Gers et al. 2002; Hirose and Tajima 2017), omics sciences (Lee et al. 2016; Leifert et al. 2016), and others. The LSTM networks are gaining increasing interest and popularity in time series modeling and prediction, as they can model long and short range dependencies (Bianchi et al. 2017; Zaytar and El Amrani 2016).

There are several variations (Graves 2013; Wang 2017; Wang and Niepert 2019) of the original model proposed by Hochreiter and Schmidhuber (1997). In this paper we adopt the model presented by Graves (2013) (Fig. 1a) which is governed by the set of equations given below:

$$i_t = \sigma(W_{xi}x_t + W_{li}l_{t-1} + W_{ci}c_{t-1} + b_i), \quad (1)$$

$$f_t = \sigma(W_{xf}x_t + W_{lf}l_{t-1} + W_{cf}c_{t-1} + b_f), \quad (2)$$

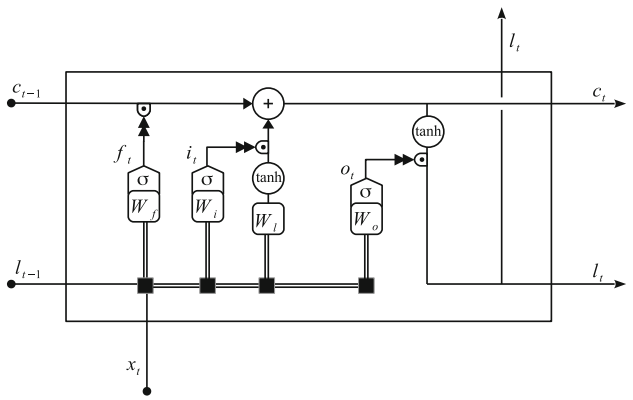
$$c_t = f_t c_{t-1} + i_t \tanh(W_{xc}x_t + W_{lc}l_{t-1} + b_c), \quad (3)$$

$$o_t = \sigma(W_{xo}x_t + W_{lo}l_{t-1} + W_{co}c_t + b_o), \quad (4)$$

$$l_t = o_t \tanh(c_t). \quad (5)$$

**Table 2** NASDAQ 100 assets used as case study

| Sector | Symbol |
|---|---|
| Capital Goods | ILMN, KLAC, PCAR, PCLN, TSLA |
| Consumer Non-Durables | CTAS, HAS, MDLZ, MNST |
| Consumer Services | AMZN, CHTR, CMCSA, COST, DISCA, DISCK, DISH, DLTR, EXPE, FAST, FOXA, FOX, LBTYA, LBTYK, LVNTA, NFLX, ORLY, PAYX, QVCA, ROST, SBUX, SIRI, TSCO, ULTA, VIAB, WYNN |
| Health Care | ALGN, ALXN, AMGN, BIIB, CELG, ESRX, GILD, HOLX, HSIC, IDXX, INCY, ISRG, MYL, SHPG, XRAY |
| Miscellaneous | AKAM, CTRP, EBAY, MELI, NTES |
| Public Utilities | VOD |
| Technology | AAPL, ADBE, ADI, ADP, ADSK, AMAT, ATVI, AVGO, BIDU, CA, CERN, CHKP, CSCO, CTSH, CTXS, EA, FB, FISV, GOOGL, INTC, INTU, LRCX, MCHP, MSFT, MU, MXIM, NVDA, QCOM, STX, SWKS, SYMC, TXN, VRSK, WDC, XLNX |
| Transportation | JBHT |




**(a)** LSTM cell

**(b)** unfolding of LSTM

**Fig. 1** LSTM architecture as presented by Graves (2013). **a** shows the internals of a specific cell, while **b** shows how the sequence is propagated through the LSTM cells

The core of the LSTM is represented by the $c_t$ which acts as a memory accumulator of the state information at time $t$. The state evolves according to Eq. (3), subject to two elements—the "forget gate" and the "input gate", represented at time $t$ by the variables $f_t$ and $i_t$, respectively. The role of $f_t$ is to erase the memory $c_{t-1}$ according to the current input $x_t$ (comprising open-close return and volatility), the state $l_{t-1}$ and the memory $c_{t-1}$ (Eq. (2)). The forget gate is counterbalanced by the input gate (Eq. (1)) that making use of the same information has instead the role of reinforcing or replacing the memory by activating a combination $x_t$ and $l_{t-1}$ (Eq. (3)). These last functions, as those governing the activation of $f_t$ and $i_t$ are learned as single-layer perceptrons using the logistic function $\sigma$ (Eq. (1) and Eq. (2)), or the tanh function (Eq. (3)) as activation, where $b_i$, $b_f$ and $b_c$ are the respective biases. Once the memory is recomputed at time $t$, the LSTM emits the output $o_t$ as a function of $x_t$, $l_{t-1}$ and the memory $c_t$ (Eq. (4)). This latter function is also learned as a single-layer perceptron and finally, the LSTM computes the state $l_t$ as given by Eq. (5). Figure 1b shows how a sequence is propagated through the LSTM.

The main advantage of this architecture is that the memory $c_t$ is refreshed under the control of gates so that the gradient is limited to the last stage (also known as constant error carousels Gers and Schmidhuber 2001; Gers et al. 1999) and prevented from vanishing too quickly. This latter issue is a critical one and a well-known limitation of RNN based on older architectures, such as the Elman's and Jordan's reference models (Jozefowicz et al. 2015; Pascanu et al. 2013).

Because the LSTM learning function can be decomposed into multiple intermediate steps, LSTMs can be "stacked" such that the information produced by one LSTM step becomes the input to another. This "stacked" architecture has been applied to many real-world sequence modeling problems (Sutskever et al. 2014; Xu et al. 2015).

### 3.3 Experimental setup

The proposed model is a 2-layer stacked LSTM made of $2n$ input and $n$ output units, respectively, with $n$ being the number of assets. This means 58/29 for DJI 500 and 184/92 for NASDAQ 100. The output of the top LSTM is given as input to a dense activation layer designed to provide the model's output (see Fig. 2 for a schematic description of the model). The hidden activation function is a hyperbolic tangent, while the recurrent activation is a hard sigmoid (default activation functions for LSTM, as advised in Hochreiter and Schmidhuber 1997). To avoid negative forecasts (the realized volatility is always continuous positive), a softplus function is used in the output layer.

Two topologies of LSTM are tested and evaluated: univariate and multivariate.

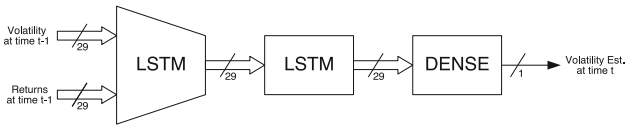



**Fig. 2** The proposed model for DJI 500 is a stack of two LSTM with 58 and 29 neurons each and a dense activation layer on the top. The size of the dense layer is one in the univariate approach (LSTM-1) and 29 in the multivariate one (LSTM-29)

**Table 3** Hyper-parameters for the LSTM

| Hyper-parameter | Range of optimization | Optimal value |
|---|---|---|
| Dropout | [0, 0.6] every 0.1 | 0.2 |
| Number of training epochs on pre-training data | [100, 400] every 50 | 300 |
| Number of training epochs on rolling window | [10, 30] every 5 | 20 |
| Look-back | [5, 100] every 5 | 20 |
| Loss function | MSE and QLIKE | QLIKE |

The optimal value of each hyper-parameter has been selected through a grid search on the defined range

LSTM-1 has a univariate architecture (one model independently trained for each asset) so takes as input only one asset at a time (open-close return and volatility for a chosen past window $(r_{t-k}, .., r_{t-1}, v_{t-k}, .., v_{t-1})$) and produces as output the one-step-ahead volatility $(v_t)$.

LSTM-$n$ has a multivariate architecture, where a single model is trained using all the assets. This version takes as input the daily returns and volatilities for a given past window $(r^i_{t-k}, .., r^i_{t-1}, v^i_{t-k}, .., v^i_{t-1}), i = 1, \ldots, n$ and outputs the $n$ one-step-ahead volatility $(v^i_t, i = 1, \ldots, n)$, where $n = 29$ for DJI 500 and $n = 92$ for NASDAQ 100.

The model is pre-trained by using the first 300 days of our dataset with a look-back window of 20 days. Subsequently, a rolling forecast is applied from day 301 onward. In particular, every time the one-step-ahead volatility is predicted, its observed features (realized return and volatility) are used to refine the network state, before moving the time window one step. This procedure allows having an up-to-date network, every time new information is available.

To avoid the look-ahead bias, the walk-forward testing technique (Ladyzynski et al. 2013) has been applied on the initial year of trading data and a grid search used to optimize the hyper-parameters of the network. The validation is performed by scanning the data using a sliding window of length $m$ (i.e., the look-back) on which the model is trained on, and subsequently predicting on the following $n$ samples (in our case, with one-step-ahead prediction, n = 1). When the end of this subset is reached the optimization window is shifted forward by $n$. The grid search boundaries and optimal values are reported in Table 3. The optimal number of training epochs on the initial data was found to be in line with the optimal number of days used for pre-training (i.e., 300 days), while the number of epochs for the rolling period had its best value consistent with the size of the look-back (i.e., 20 days, equivalent to four weeks of trading data). We observed that a smaller number of epochs would produce an under-fitted network (smooth forecast trending with the average of the last few data points), while a longer training would produce an over-fitting network (giving a lot of weight to the most recently shown data point). Furthermore, a larger look-back would impact the convergence (probably due to the vanishing gradient problem). In addition, dropout, a standard regularization technique used for deep learning models, was used to avoid over-fitting, and its best value was found to be around 0.20, as also suggested in Wan et al. (2013). Lastly, we employed two loss functions to train our model, which are described in Sect. 3.6.

#### 3.3.1 LSTM computational complexity

In this section we report the computational complexity of our proposed univariate and multivariate models.

The LSTM computational complexity can be estimated by calculating the number of operations and number of trainable parameters of the model.

Considering a network taking input vectors of size $m$ and giving output vectors of size $n$, the LSTM has a set of 2 matrices: $U$ of dimension $nm$ and $W$ of dimension $nn$ for each of the three gates and one set for updating the cell state. Lastly, there is a set of $n$ biases.

The memory complexity (number of trainable weights) of the LSTM can be calculated with the following formula: $4(nm + n^2 + n)$.

The time complexity (i.e., number of operations required to perform one training step) is calculated as follows. A matrix multiplication requires $mn$ multiplications and $m(n-1)$ additions, this needs to be done for $mn$ elements, so the complexity is $mn(n + (m - 1)) = n^2m + nm^2 - nm$

This operation is done for each gate (3), for the cell state, and for the number of time steps (k = 20), $4k(n^2m + nm^2 - nm)$.

While the number of operations required for one epoch grows with the size of the inputs and outputs of the model, the computational need is still negligible when compared to the available computational power when employing a GPU card. For instance, the biggest multivariate model trained in the empirical experimentation, requires roughly 744 million operations for one epoch, while the graphic card we used (i.e., NVIDIA GeForce 1070) has a computational power of 6.5 TFLOPS, which allows to run 9 epochs per second.

Table 4 reports the total number of weights and number of operations for each of the models trained in the empirical experimentation.

**Table 4** The following table reports for each LSTM the number of trainable weights and the number of operations required for one training epoch

| | LSTM-1 DJI 500 | LSTM-29 DJI 500 | LSTM-1 NASDAQ 100 | LSTM-92 NASDAQ 100 |
|---|---|---|---|---|
| First LSTM input ($m_1$) | 1 | 29 | 1 | 92 |
| First LSTM output ($n_1$) | 58 | 58 | 184 | 184 |
| Second LSTM input ($m_2$) | 58 | 58 | 184 | 184 |
| Second LSTM output ($m_2$) | 1 | 29 | 1 | 92 |
| Number of trained models | 29 | 1 | 92 | 1 |
| Number of weights | 410,640 | 30,624 | 12,662,880 | 305,808 |
| Number of operations | 15,608,960 | 23,144,320 | 498,360,320 | 744,832,000 |

Since in the univariate approach one model for each asset is trained individually, the number of weights and operations, is multiplied by the number of assets (29 for the DJI500 and 92 for the NASDAQ100)

#### 3.3.2 Data standardization

Before training the model, the data are standardized with 0-mean and 1-variance. Since our features are in $\mathbb{R}^+$, for each sample $(r_t^2, v_t)$, its negative $(-r_t^2, -v_t)$ is added to the data (to have a perfect bell-shaped distribution). The resulting distribution is already mean-centered, hence the values are divided by their standard deviation, and finally, the added negative values are dropped to restore the original set of observations.

### 3.4 Realized GARCH (R-GARCH)

The Realized GARCH introduced by Hansen et al. (2012) has extended the class of GARCH models by replacing, in the volatility dynamics, the squared returns with a much more efficient proxy such as a realized volatility measure. The structure of the R-GARCH(1, 1) in its linear formulation is given by:

$$r_t = \mu + \sqrt{h_t}\, z_t, \tag{6}$$

$$h_t = \omega + \beta\, h_{t-1} + \gamma\, v_{t-1}, \tag{7}$$

$$v_t = \xi + \varphi\, h_t + \tau(z_t) + u_t, \tag{8}$$

where $z_t \sim i.i.d.(0, 1)$ and $u_t \sim i.i.d.(0, \sigma_u^2)$ with $z_t$ and $u_t$ being mutually independent. The first two equations are the *return equation* and the *volatility equation* that define a class of GARCH-X models, including those estimated in Visser (2011), Engle (2002), and Barndorff-Nielsen and Shephard (2005). The GARCH-X acronym refers to the fact that $v_t$ is treated as an exogenous variable. It is worth noting that most variants of ARCH and GARCH models are nested in the R-GARCH framework. The measurement equation is justified by the fact that any consistent estimator of the Integrated Variance can be written as the sum of the conditional variance plus a random innovation, where the latter is captured by $\tau(z_t) + u_t$. The function $\tau(z_t)$ can accommodate leverage effects, because it captures the dependence between returns and future volatility. A common choice (Hansen et al. 2012) that has been found to be empirically satisfactory is to use the specification:

$$\tau(z_t) = \tau_1\, z_t + \tau_2(z_t^2 - 1). \tag{9}$$

Substituting the measurement equation into the volatility equation, it can be easily shown that the model implies an AR(1) representation of $h_t$:

$$h_t = (\omega + \xi\gamma) + (\beta + \varphi\gamma)h_{t-1} + \gamma\, w_{t-1}, \tag{10}$$

where $w_t = \tau(z_t) + u_t$. Furthermore, it is assumed that the expectation of $E(w_t) = 0$. The coefficient $(\beta + \varphi\gamma)$ reflects the persistence of volatility, whereas $\gamma$ summarizes the impact of the past realized measure on future volatility.

The general conditions required to ensure that the volatility process $h_t$ is stationary and the unconditional variance of $r_t$ is finite and positive are given by:

$$\omega + \xi\gamma > 0, \tag{11}$$

$$0 < \beta + \varphi\gamma < 1. \tag{12}$$

If the conditions in Eq. (11) are fulfilled, the unconditional variance of $r_t$, taking expectations of both sides in Eq. (10), can be easily shown to be equal to $(\omega + \xi\gamma)/[1 - (\beta + \varphi\gamma)]$. Finally, as for standard GARCH models, the positivity of $h_t$ $(\forall t)$ is achieved under the general condition that $\omega$, $\gamma$ and $\beta$ are all positive.

### 3.5 GJR-MEM

Multiplicative Error Models (MEM) were first proposed by Engle (2002) as a generalization to non-negative variables of the Autoregressive Conditional Duration (ACD) models of Engle and Russell (1998). Namely, let $v_t$ be a discrete time process on $[0, \infty)$ (e.g., a realized measure). A general formulation of the MEM is

$$v_t = \mu_t \epsilon_t, \tag{13}$$

$$\mu_t = \mu(\boldsymbol{\psi}_\mu, \mathcal{I}_{t-1}), \tag{14}$$

where $(\epsilon_t|\mathcal{I}_{t-1}) \overset{iid}{\sim} D^+(1,\sigma^2)$. It can be easily seen that

$$E[v_t|\mathcal{I}_{t-1}] = \mu_t, \tag{15}$$

$$\text{var}[v_t|\mathcal{I}_{t-1}] = \sigma^2\mu_t^2, \tag{16}$$

where the conditional expectation of the realized measure ($\mu_t$) provides an estimate of the latent conditional variance $h_t$.

The GJR-MEM model is obtained by borrowing from the GARCH literature (Engle 2002; Glosten et al. 1993) the following dynamic equation for $\mu_t$:

$$\mu_t = \omega + \alpha v_{t-1} + \beta\mu_{t-1} + \gamma v_{t-1} I(r_{t-1} < 0), \tag{17}$$

which allows the reproduction of volatility clustering as well as leverage effects.

Coming to the specification of the distribution of $\epsilon_t$, any unit mean distribution with positive support could be used. Possible choices include Gamma, Log-Normal, Weibull, Inverted-Gamma and mixtures of them. In this paper we consider the Gamma distribution which is a flexible choice able to fit a variety of empirical settings. If $(\epsilon_t|\mathcal{I}_{t-1}) \sim \Gamma(\theta,\phi)$, its density is given by

$$f(\epsilon_t|\mathcal{I}_{t-1}) = \frac{1}{\Gamma(\theta)\phi^\theta}\epsilon_t^{\theta-1}\exp\left(-\frac{\epsilon_t}{\phi}\right). \tag{18}$$

However, since $E(\epsilon_t|\mathcal{I}_{t-1}) = \theta\phi$, to ensure unit mean, it is needed to impose the constraint $\phi = 1/\theta$ giving rise to the following density

$$f(\epsilon_t|\mathcal{I}_{t-1}) = \frac{1}{\Gamma(\theta)}\theta^\theta\epsilon_t^{\theta-1}\exp(-\theta\epsilon_t). \tag{19}$$

Model parameters can then be estimated maximizing the likelihood function implied by the unit mean Gamma assumption. It is worth noting that these estimates have a quasi-maximum likelihood interpretation since it can be shown that, given that $\mu_t$ is correctly specified, they are still consistent and asymptotic normal even if the distribution of $\epsilon_t$ is misspecified.

### 3.6 Evaluation metrics

We have considered both online and offline evaluations. In the online evaluation case, two alternative loss functions have been used to train the LSTM models: the widely accepted Mean Squared Error (MSE) for regression and forecasting tasks; and the QLIKE function, particularly suitable for volatility forecasting (Patton 2011). For the offline evaluation case, we have considered a test data set (not used for training) for out-of-sample evaluation using MSE, QLIKE and the Pearson correlation index.

Given a vector $\hat{Y}$ of $N$ forecasts and the vector $Y$ of observed values, the MSE and QLIKE are defined as follows:

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(\hat{Y}_i - Y_i)^2, \tag{20}$$

$$\text{QLIKE} = \frac{1}{N}\sum_{i=1}^{N}(\log(\hat{Y}_i) + \frac{Y_i}{\hat{Y}_i}). \tag{21}$$

These measures are proposed in our evaluation framework since they are considered to be robust for assessing volatility forecast performance (Patton 2011). A robust measure must ensure that using a proxy for the volatility (the realized kernel in our case) gives the same ranking as using the true (unobservable) volatility of an asset.

Moreover, the Pearson correlation coefficient is computed between the forecast and realized volatility of each estimated model, to assess the models ability to follow the assets trends.

Also, statistical test, namely Diebold-Mariano (DM), is used to assess models' Conditional Predictive Ability (CPA). The one-tail DM (Diebold and Mariano 2002) is used with squared error, predictive horizon equal to 1 (for one step ahead forecast) and a significance threshold at 0.05, to test the following NULL hypothesis 'Model $M_i$ has better predictive ability than model $M_j$ with a size level equal to $\alpha = 0.05$'.

Lastly, the results are evaluated in terms of Value At Risk (VaR) and Expected Shortfall (ES) estimation which are widely adopted by practitioners and regulators as standard measures of market risk for financial assets. The VaR encapsulates in a point-wise fashion the potential market value loss of a financial asset over a time horizon $h$, at a significance or coverage level $\alpha_{VaR}$. In our case we consider $h = 1$ and $\alpha_{VaR} = 0.05$. Its performance is evaluated using two metrics: the violation ratio (VR) and the average square magnitude function (ASFM). The VR is the percentage occurrence of an actual loss that is greater than the estimated maximum loss in the VaR framework while the ASFM considers the amount of possible default measuring the average squared cost of exceptions. The ES (Acerbi and Tasche 2002) is often referred to as the conditional VaR (cVaR). The predictive performance of the models under comparison in forecasting the pair (VaR, ES) is assessed by computing the Asymmetric Laplace Score (ALS), as defined in Taylor (2019): more accurate models are expected to return lower values of the ALS criterion. The VR and ASFM metrics are defined as in Dunis et al. (2010) and Maciel et al. (2017).

## 4 Results and discussion

In this section, the proposed approach is implemented in two variants: univariate (LSTM-1) and multivariate (LSTM-29

**Table 5** Evaluation Metrics for the DJI 500 dataset: The MSE, QLIKE and Pearson measures are reported for each asset and for each compared model

| Asset | LSTM-1 | | | LSTM-29 | | | R-GARCH | | | GJR-MEM | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | MSE | QLIKE | Pearson | MSE | QLIKE | Pearson | MSE | QLIKE | Pearson | MSE | QLIKE | Pearson |
| AA | 4.77 | 1.9871 | 0.59 | **3.65** | 1.9753 | **0.69** | 4.83 | 2.0125 | 0.60 | 4.05 | **1.9702** | 0.65 |
| AIG | 6.71 | 1.4047 | 0.61 | **4.84** | **1.3502** | **0.72** | 6.40 | 1.4303 | 0.61 | 5.40 | 1.3534 | 0.70 |
| AXP | 3.19 | 1.1222 | 0.77 | **1.70** | **1.1057** | **0.87** | 2.11 | 1.1139 | 0.84 | 2.00 | 1.1060 | 0.85 |
| BA | 0.88 | 1.4089 | 0.56 | 0.81 | **1.4014** | **0.62** | 0.85 | 1.4102 | 0.59 | **0.80** | 1.4023 | **0.62** |
| BAC | 3.93 | 0.9167 | 0.73 | **2.86** | **0.8878** | **0.81** | 5.19 | 0.9652 | 0.71 | 3.35 | 0.8991 | 0.77 |
| C | 5.86 | 1.2183 | 0.74 | **2.48** | **1.1888** | **0.89** | 3.62 | 1.2208 | 0.83 | 2.92 | 1.1912 | 0.86 |
| CAT | 1.12 | 1.5686 | 0.65 | **1.03** | **1.5559** | **0.69** | 1.18 | 1.5688 | 0.62 | 1.10 | 1.5595 | 0.66 |
| CVX | 1.36 | 1.3297 | 0.69 | **1.05** | 1.3262 | **0.76** | 1.20 | **1.3210** | 0.72 | 1.07 | 1.3135 | 0.75 |
| DD | 1.56 | 1.3125 | 0.64 | **1.10** | 1.2951 | **0.76** | 1.33 | 1.3225 | 0.72 | 1.11 | **1.2906** | 0.75 |
| DIS | 1.01 | 1.3500 | 0.52 | 0.86 | 1.3308 | **0.63** | 0.88 | 1.3351 | 0.61 | **0.83** | **1.3230** | **0.63** |
| GE | 0.57 | 0.9274 | 0.66 | **0.50** | 0.9214 | **0.72** | 0.54 | 0.9328 | 0.69 | **0.50** | **0.9109** | 0.71 |
| GM | 11.44 | 2.1415 | 0.62 | **9.83** | 2.1035 | **0.68** | 11.45 | 2.1048 | 0.62 | 10.79 | **2.0971** | 0.64 |
| HD | 2.58 | 1.5623 | 0.68 | 1.83 | 1.5563 | **0.78** | 1.99 | 1.5530 | 0.76 | **1.81** | **1.5417** | 0.77 |
| IBM | 0.65 | 1.0042 | 0.64 | **0.51** | 0.9995 | **0.73** | 0.65 | 1.0165 | 0.64 | 0.54 | **0.9900** | 0.70 |
| INTC | 1.42 | **1.7701** | **0.62** | 1.52 | 1.7711 | **0.62** | 1.51 | 1.7707 | 0.59 | **1.40** | 1.7621 | **0.62** |
| JNJ | 0.33 | **0.6027** | 0.52 | **0.32** | 0.6031 | **0.54** | 0.35 | 0.6566 | 0.48 | 0.34 | 0.6032 | 0.51 |
| JPM | 4.92 | 1.3190 | 0.76 | 3.86 | 1.3097 | 0.80 | 4.27 | 1.3223 | 0.78 | **3.24** | **1.2999** | **0.84** |
| KO | 0.28 | 0.7644 | 0.59 | **0.27** | 0.7645 | **0.61** | 0.34 | 0.8399 | 0.56 | **0.27** | **0.7557** | 0.60 |
| MCD | **1.32** | **1.3551** | **0.49** | 1.40 | 1.3736 | 0.48 | **1.32** | 1.3630 | **0.49** | 1.41 | 1.3579 | 0.47 |
| MMM | 0.64 | 1.0530 | 0.56 | 0.55 | 1.0512 | **0.65** | 0.68 | 1.0991 | 0.55 | **0.54** | **1.0444** | 0.64 |
| MRK | 9.45 | 1.5524 | 0.25 | **8.63** | 1.5483 | **0.38** | 9.75 | 1.5783 | 0.26 | 10.30 | **1.5341** | 0.28 |
| MSFT | 0.77 | 1.2006 | 0.60 | 0.63 | 1.2030 | 0.69 | 0.63 | 1.2070 | 0.68 | **0.57** | **1.1838** | **0.70** |
| PG | **0.25** | 0.7635 | 0.56 | **0.25** | 0.7696 | **0.58** | 0.31 | 0.8072 | 0.52 | **0.25** | **0.7584** | 0.57 |
| SPY | 0.17 | 0.1620 | 0.72 | 0.17 | 0.1519 | 0.75 | 0.16 | 0.1369 | 0.75 | **0.14** | **0.1267** | **0.78** |
| T | 1.97 | 1.4403 | 0.63 | **1.64** | 1.4304 | **0.70** | 2.12 | 1.4644 | 0.63 | 1.72 | **1.4266** | 0.69 |
| UTX | 0.85 | 1.1702 | 0.50 | **0.69** | 1.1634 | **0.62** | 0.97 | 1.1875 | 0.47 | 0.74 | **1.1585** | 0.59 |
| VZ | 1.40 | 1.2661 | 0.54 | 1.05 | 1.2643 | **0.70** | 1.15 | 1.2907 | 0.67 | **1.02** | **1.2530** | **0.70** |
| WMT | 0.71 | 1.1262 | 0.64 | **0.67** | 1.1253 | **0.68** | 0.88 | 1.1604 | 0.60 | 0.70 | **1.1163** | 0.65 |
| XOM | 0.95 | 1.2453 | 0.71 | 0.89 | 1.2484 | **0.73** | 0.92 | 1.2385 | 0.71 | **0.87** | **1.2324** | **0.73** |

and LSTM-92). The method is compared with two state-of-the art methodologies, namely R-GARCH (Hansen et al. 2012) and GJR-MEM (Glosten et al. 1993). We discuss empirical results from an out-of-sample forecasting comparison, using returns and realized measures for the 28 Dow Jones Industrial Average stocks plus one exchange-traded index fund, SPY (which tracks the S&P 500 index) over a period of 1250 days and for the 92 stocks belonging to the NASDAQ 100 index over a period of 956 days.

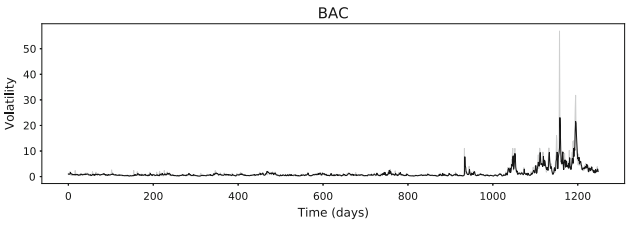



**Fig. 3** BAC one step ahead predictions. The observed time series are given in gray and the predicted volatility values in black. Data point 0 is the 18th March 2003; data point 600 is the 10th August 2005; and data point 1200 corresponds to the 8th August 2008

## 4.1 Dow Jones industrial average 500

A detailed comparison of the methods performance is given in Table 5 with respect to each asset. The LSTM-29 approach reaches the lowest MSE error for 18 out of the 29 assets when compared with LSTM-1, R-GARCH and GJR-MEM methods. In particular, the LSTM-29 has a lower error compared to our univariate model for 25 out of 29 assets, equal in 2 and worse in 2 cases. Compared to R-GARCH, the LSTM-29 is better again in 25 out of 29 cases, equal for 1, and worse for

**Table 6** Value At Risk analysis for the DJI 500 dataset: the VR, ALS and ASMF measures are reported for each asset and for each compared model

| Asset | Violation rate (%) | | | | Average square magnitude function | | | | Average Asymmetric Laplace Score | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | LSTM-1 | LSTM-29 | R-GARCH | GJR-MEM | LSTM-1 | LSTM-29 | R-GARCH | GJR-MEM | LSTM-1 | LSTM-29 | R-GARCH | GJR-MEM |
| AA | 4.88 | **4.96** | 6.00 | 4.64 | 1.59 | 1.16 | 1.46 | **1.08** | 0.85 | 0.95 | **0.11** | 0.56 |
| AIG | 4.32 | 4.56 | **5.20** | 3.60 | 3.62 | **3.16** | 3.25 | 3.98 | 0.26 | 0.47 | 0.74 | **0.02** |
| AXP | 6.08 | 6.33 | **5.04** | 5.44 | **0.49** | 0.54 | **0.49** | 0.54 | 0.09 | **0.04** | 0.94 | 0.48 |
| BA | 5.20 | 5.28 | **4.96** | 4.72 | 0.75 | **0.73** | 0.82 | 0.76 | 0.74 | **0.65** | 0.95 | **0.65** |
| BAC | **4.80** | 4.16 | 5.76 | 3.76 | 0.45 | **0.39** | 0.51 | 0.41 | 0.75 | 0.16 | 0.23 | **0.04** |
| C | **4.72** | 4.48 | 5.76 | 4.16 | 1.19 | **0.75** | 0.88 | 0.84 | 0.65 | 0.39 | 0.23 | **0.16** |
| CAT | **4.80** | 4.64 | 4.48 | 4.40 | 0.97 | **0.93** | 1.00 | 1.01 | 0.75 | 0.56 | 0.39 | **0.32** |
| CVX | 4.56 | **4.64** | 4.32 | 4.32 | 0.58 | 0.51 | 0.56 | **0.50** | 0.47 | 0.56 | **0.26** | **0.26** |
| DD | 3.60 | 3.68 | **4.80** | 3.28 | 0.58 | **0.45** | 0.53 | 0.52 | 0.02 | 0.03 | 0.75 | **0.00** |
| DIS | 4.24 | 4.64 | **4.72** | 3.92 | 0.60 | **0.48** | 0.57 | 0.56 | 0.21 | 0.56 | 0.65 | **0.07** |
| GE | **3.68** | 3.36 | 3.60 | 3.04 | 0.31 | 0.29 | 0.37 | 0.32 | 0.03 | **0.00** | 0.02 | **0.00** |
| GM | 8.09 | 8.09 | **5.36** | 6.16 | 2.53 | **2.16** | 2.82 | 2.61 | **0.00** | **0.00** | 0.56 | 0.07 |
| HD | **4.64** | 4.08 | 4.56 | 3.12 | **0.69** | 0.79 | 0.70 | 0.85 | 0.56 | 0.13 | 0.47 | **0.00** |
| IBM | **4.00** | 3.76 | 4.24 | 3.76 | 0.61 | 0.65 | 0.66 | **0.46** | 0.09 | **0.04** | 0.21 | **0.04** |
| INTC | 6.16 | 5.52 | 5.28 | **5.04** | 0.84 | 0.90 | **0.77** | 0.84 | **0.07** | 0.40 | 0.65 | 0.94 |
| JNJ | 2.64 | **3.28** | 3.76 | 2.48 | 0.49 | **0.38** | 0.44 | 0.50 | **0.00** | **0.00** | 0.04 | **0.00** |
| JPM | **4.72** | **4.72** | **4.72** | 3.92 | 0.63 | 0.61 | 0.61 | **0.59** | 0.65 | 0.65 | 0.65 | **0.07** |
| KO | 2.72 | 2.88 | **4.24** | 2.24 | 0.63 | **0.50** | 0.52 | 0.64 | **0.00** | **0.00** | 0.21 | **0.00** |
| MCD | 4.00 | 4.00 | **4.08** | 3.44 | 0.52 | 0.55 | 0.61 | **0.51** | 0.09 | 0.09 | 0.13 | **0.01** |
| MMM | 4.16 | 3.76 | **4.88** | 4.00 | 1.47 | 1.50 | **1.36** | 1.38 | 0.16 | **0.04** | 0.85 | 0.09 |
| MRK | 3.84 | 4.48 | **4.80** | 2.80 | 4.21 | 3.59 | **3.26** | 5.10 | 0.05 | 0.39 | 0.75 | **0.00** |
| MSFT | 3.76 | **4.00** | 3.92 | 3.20 | 0.66 | 0.54 | **0.53** | 0.54 | 0.04 | 0.09 | 0.07 | **0.00** |
| PG | 2.40 | 2.48 | **3.60** | 2.32 | 0.31 | 0.29 | **0.28** | **0.28** | **0.00** | **0.00** | 0.02 | **0.00** |
| SPY | 6.89 | 6.73 | **6.00** | 6.97 | 0.30 | 0.25 | **0.23** | 0.24 | **0.00** | 0.01 | 0.11 | **0.00** |
| T | 4.40 | 3.92 | **4.64** | 3.04 | **0.93** | 1.05 | 1.01 | 1.23 | 0.32 | 0.07 | 0.56 | **0.00** |
| UTX | 4.00 | **4.24** | **4.24** | 3.60 | 0.67 | **0.59** | 0.82 | 0.68 | 0.09 | 0.21 | 0.21 | **0.02** |
| VZ | 3.44 | 4.00 | **4.96** | 3.12 | 1.32 | 1.18 | **1.14** | 1.43 | 0.01 | 0.09 | 0.95 | **0.00** |
| WMT | 3.68 | 3.20 | **4.40** | 3.12 | **0.27** | 0.32 | 0.30 | 0.28 | 0.03 | **0.00** | 0.32 | **0.00** |
| XOM | 4.40 | **4.72** | 4.24 | 4.64 | 0.81 | 0.79 | 0.77 | **0.70** | 0.32 | 0.65 | **0.21** | 0.56 |

For the VR the closest value to 5% the better, while for the ALS and ASMF the lower the better

3 assets. Lastly, our proposed approach is better, equal and worse than GJR-MEM in 16, 3 and 10 cases, respectively. An example of the one step ahead prediction given by the LSTM-29 for the BAC asset is presented in Fig. 3.

Having a closer look at the MSE values from Table 5, the LSTM-29 is not better than the other benchmarks on assets with very low errors and hence volatility, in the considered period (e.g., JNJ, KO, PG and SPY).

Next, the forecast models are employed in risk management applications using Value At Risk (VaR). With the VaR estimates, the models are evaluated using the VR, CVR and the ASMF. Table 6 shows the values of VR, CVR and ASMF for VaR estimation using the four models for all DJI 500 assets. The LSTM-29 achieved better VR (values closer to our selected VaR confidence level—5%) compared to LSTM-1 in 15 out of 29 assets, equal for 3 and worse for 11; compared to the R-GARCH is better in 6 cases, equal in 2 and worse in 21, while compared to the GJR-MEM is better, equal and worse in 23, 1 and 5 cases, respectively.

For the CVR , the LSTM-29 achieved better results (the lower the value, the better) compared to LSTM-1 in 11 out of 29 assets, equal for 3 and worse for 15; compared to the R-GARCH is better in 21 cases, equal in 2 and worse in 6, while compared to the GJR-MEM is better, equal and worse in 5, 1 and 23 cases, respectively.

When considering the ASMF measure, the LSTM-29 is better than the LSTM-1, R-GARCH and GJR-MEM in 21, 17 and 17 cases; and worse in 8, 12 and 12 cases, respectively.

Figure 4 is a scatter plot illustration comparing for each asset, the performance of the LSTM-29 and the other models measured in terms of MSE difference (i.e., positive values representing smaller errors and better LSTM-29 performance) versus asset volatility in terms of its variance (i.e., higher values of variance representing stronger fluctuation in daily volatilities) over the out of sample period (1250 days).

As can be seen from Fig. 4, the LSTM-29 is generally comparable with the other models at lower volatility, while outperforming the LSTM-1 and the two state-of-the-art R-GARCH and GJR-MEM approaches in higher volatility regimes. This result is confirmed by the Pearson's correlation index with values 0.825 against LSTM-1, 0.800 against R-GARCH, and 0.608 against GJR-MEM over the 29 assets.

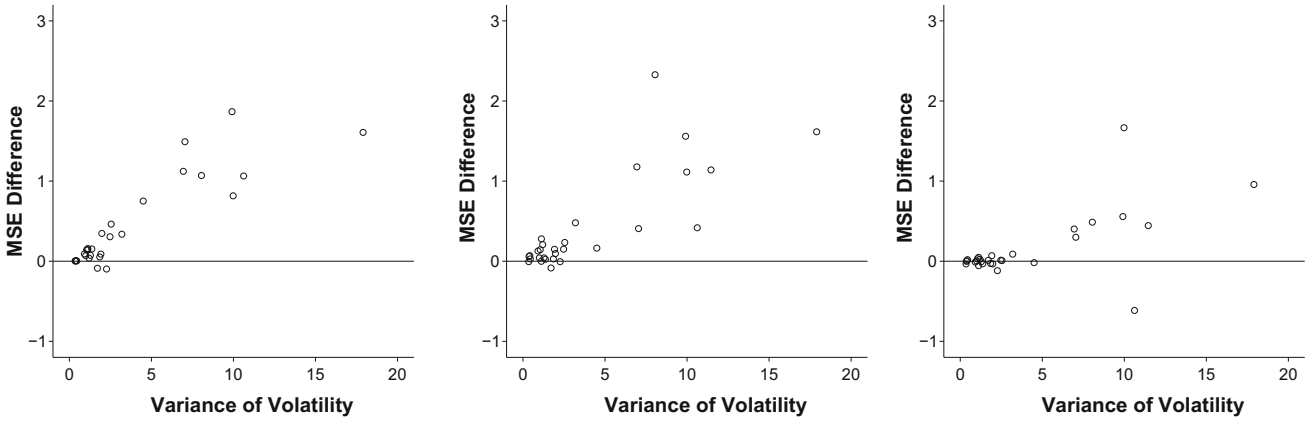
To verify whether the proposed approach has statistically superior predictive ability, a Diebold-Mariano test is performed using a predictive horizon equal to 1 (one-step-ahead forecast). As it can be observed from the results reported in Table 7, the LSTM-29 has a better predictive ability for 10 out of 29 assets compared to the LSTM-1, 16 over 29 against the R-GARCH, and 6 out of 29 assets for the GJR-MEM, when considering a $p$-value strictly lower than 0.05. It is also worth noticing that in the remaining cases LSTM-29 is never worse than the compared models.

Furthermore, to test the dependence of forecasting accuracy on volatility conditions, we evaluated the errors (mean, median, standard deviation (std) and median absolute deviation (MAD)) for four volatility clusters: very low (VL); low (L); high (H); and very high (VH) (Table 8). The clusters are calculated taking the 50, 75 and 95 percentiles of the smoothed volatility over time, using a 10-day centered moving average and moving variance of all the assets. Specifically, for the moving average, we consider the following ranges: 0 to 0.50 (up to 50%) for VL; 0.50 to 0.85 (up to 75%) for L; 0.85 to 2.80 (up to 95%) for H; and 2.80 to 13.92 (up to 100%) for VH. For the moving variance the ranges are: 0 to 1.18 (up to 25%) for VL; 1.18 to 1.79 (up to 75%) for L; 1.79 to 4.38 (up to 95%) for H; and 4.38 to 15.60 (up to 100%) for VH. As can be seen from the DM test results in Table 9, the LSTM-1 performs better than the multivariate counterpart for relatively low volatility periods, while having inferior performance for higher volatility ones. The LSTM-29 is never worse than the R-GARCH, slightly worse than the GJR-MEM for low volatilities and always statistically better in high volatility settings.

However, it is worth noticing that the difference between LSTM-1 and LSTM-29 seen from Table 8 is in practice negligible, valuing 0.010 (VL) and 0.024 (L) for the mean, 0.014 (VL) and 0.030 (L) for the median. The real impact is made by the LSTM-29 within the VH volatility regime, where the difference to LSTM-1, R-GARCH, GJR-MEM is, respectively, 10.61, 7.17, and 2.40 for the mean and 0.59, 1.35, and 0.92 for the median. Considering that the risk in trading assets is considerable at higher volatility, the VH cluster is also the most important to pay attention to.

In all regimes, we observed the tendency of LSTM-29 to provide larger values of volatility when compared to R-GARCH and GJR-MEM estimates, and to be more conservative from a risk management perspective.

Figure 5 outlines the cumulative MSE recorded by the considered models in the four different volatility regimes. These curves are plotted by sorting the errors in decreasing order so that larger errors come first, which is the reason of the up-sloped shapes of the curves: they outline the tendency of models to accumulate larger errors along the experimentation. One can observe that the LSTM-29 performance is always better than the other models in regimes of H and VH volatility. In VL and L volatility regimes, the LSTM-29 is a little worse than GJR-MEM, but still better than R-GARCH in all considered regimes. This is not surprising as the R-GARCH and GJR-MEM are econometric models of volatility, while the LSTM is unaware of the underlying stochastic process. Instead, the LSTM-1 achieved a better accuracy for VL and L regimes, but performed poorly for the VH volatility regime. For completeness, the LSTM-29 was also trained without the index fund SPY, showing consistent results.




**(a)** LSTM-1 **(b)** R-GARCH **(c)** GJR-MEM

**Fig. 4** Difference in MSE (y-axis) and variance of volatility (x-axis) for LSTM-29 vs: **a** LSTM-1; **b** R-GARCH; and **c** GJR-MEM. Each dot represents an asset. The LSTM-29 gets better in periods of high volatility

**Table 7** Diebold-Mariano statistic for the DJI 500 dataset (with a $p$-value given in brackets) for the LSTM-29 against LSTM-1, R-GARCH and GJR-MEM

| Asset | LSTM-29 vs. | | | | | |
|---|---|---|---|---|---|---|
| | LSTM-1 | | R-GARCH | | GJR-MEM | |
| AA | −2.42 | (0.008***) | −3.21 | (0.001***) | −2.53 | (0.006***) |
| AIG | −2.86 | (0.002***) | −2.35 | (0.009***) | −1.21 | (0.114) |
| AXP | −2.68 | (0.004***) | −2.76 | (0.003***) | −2.06 | (0.020**) |
| BA | −1.33 | (0.092*) | −1.92 | (0.028**) | 0.25 | (0.597) |
| BAC | −2.27 | (0.012**) | −3.11 | (0.001***) | −1.65 | (0.049**) |
| C | −3.12 | (0.001***) | −3.33 | (0.000***) | −2.17 | (0.015**) |
| CAT | −1.01 | (0.157) | −2.45 | (0.007***) | −1.28 | (0.100*) |
| CVX | −1.60 | (0.055*) | −1.54 | (0.061*) | −0.20 | (0.420) |
| DD | −1.77 | (0.039**) | −2.51 | (0.006***) | −0.18 | (0.429) |
| DIS | −1.55 | (0.060*) | −0.62 | (0.267) | 0.64 | (0.739) |
| GE | −0.85 | (0.198) | −0.85 | (0.197) | −0.10 | (0.460) |
| GM | −2.12 | (0.017**) | −2.91 | (0.002***) | −2.05 | (0.020**) |
| HD | −1.67 | (0.048**) | −1.03 | (0.153) | 0.14 | (0.556) |
| IBM | −1.81 | (0.035**) | −2.31 | (0.011**) | −0.89 | (0.187) |
| INTC | 0.64 | (0.740) | 0.05 | (0.520) | 1.27 | (0.898) |
| JNJ | −0.10 | (0.458) | −1.27 | (0.100*) | −1.69 | (0.046**) |
| JPM | −1.29 | (0.099*) | −1.37 | (0.085*) | 1.28 | (0.900) |
| KO | −0.28 | (0.389) | −2.9 | (0.002***) | −0.44 | (0.331) |
| MCD | 1.27 | (0.898) | 1.52 | (0.935) | −0.14 | (0.443) |
| MMM | −1.46 | (0.072*) | −2.36 | (0.009***) | 0.33 | (0.628) |
| MRK | −0.97 | (0.165) | −1.47 | (0.071**) | −1.58 | (0.057*) |
| MSFT | −0.93 | (0.176) | 0.02 | (0.509) | 1.38 | (0.916) |
| PG | −0.36 | (0.360) | −3.00 | (0.001***) | 0.00 | (0.501) |
| SPY | −0.09 | (0.463) | 0.33 | (0.629) | 1.92 | (0.972) |
| T | −1.76 | (0.040**) | −2.72 | (0.003***) | −1.20 | (0.116) |
| UTX | −0.87 | (0.192) | −1.72 | (0.043**) | −0.94 | (0.173) |
| VZ | −1.45 | (0.074*) | −1.02 | (0.153) | 0.58 | (0.718) |
| WMT | −0.74 | (0.230) | −2.26 | (0.012**) | −0.61 | (0.272) |
| XOM | −0.54 | (0.293) | −0.40 | (0.343) | 0.43 | (0.665) |

The $p$-values are marked with a * for 10% confidence level, with ** for 5% and with *** for 1%

**Table 8** Average with Std (in brackets) errors, Median with MAD (in brackets) errors for the four models at different volatility regimes (VL, L, H and VH) on the DJI 500 dataset

| | | LSTM-1 | | LSTM-29 | | R-GARCH | | GJR-MEM | |
|---|---|---|---|---|---|---|---|---|---|
| | | Mean (Std) | Median (MAD) | Mean (Std) | Median (MAD) | Mean (Std) | Median (MAD) | Mean (Std) | Median (MAD) |
| Moving Average | Very Low | **0.151** (0.657) | 0.037 (0.051) | 0.161 (0.677) | 0.038 (0.053) | 0.165 (0.709) | 0.036 (0.050) | 0.157 (0.636) | 0.045 (0.062) |
| | Low | **0.596** (3.290) | 0.136 (0.186) | 0.620 (3.234) | 0.139 (0.191) | 0.642 (3.367) | 0.135 (0.188) | 0.610 (3.232) | 0.151 (0.205) |
| | High | 2.867 (35.254) | 0.385 (0.535) | **2.796** (35.338) | 0.385 (0.539) | 3.034 (35.414) | 0.414 (0.580) | 2.941 (35.751) | 0.418 (0.579) |
| | Very High | 33.056 (144.510) | 3.638 (5.130) | **22.447** (104.469) | 3.043 (4.333) | 29.615 (125.571) | 4.394 (6.264) | 24.845 (105.169) | 3.966 (5.640) |
| Moving Variance | Very Low | **0.124** (0.259) | 0.038 (0.053) | 0.138 (0.292) | 0.039 (0.055) | 0.138 (0.305) | 0.038 (0.053) | 0.131 (0.260) | 0.045 (0.063) |
| | Low | **0.472** (0.971) | 0.142 (0.196) | 0.502 (1.021) | 0.143 (0.200) | 0.520 (1.091) | 0.140 (0.195) | 0.486 (0.951) | 0.162 (0.222) |
| | High | 2.253 (7.640) | 0.344 (0.486) | **2.126** (6.702) | 0.335 (0.476) | 2.423 (7.642) | 0.366 (0.523) | 2.160 (6.768) | 0.384 (0.538) |
| | Very High | 36.406 (159.551) | 2.685 (3.848) | **25.946** (125.006) | 2.331 (3.362) | 32.934 (142.879) | 3.615 (5.222) | 28.853 (125.875) | 3.382 (4.868) |

The four volatility regimes are determined using the 50, 75, and 95 percentiles over all the assets. The first comparison (rows 2 to 5) is using a centered moving average of the volatilities over time (5 time steps before and 5 time steps after the current one), while the second one (rows 6 to 9) is using a centered moving variance of the volatilities over time (5 time steps before and 5 time steps after the current one)

To investigate whether the proposed model is able to accurately predict volatility level throughout extreme market events, we also considered its predictive performance during the 2007–2008 crisis (in particular, focusing on the time span of 200 trading days starting from the 1st July 2007).

Table 10 shows the MSE scored by each model in the initial 1050 days (pre-crisis) and the last 200 days (in-crisis). As already shown in Fig. 5, the LSTM-1 and GJR-MEM are slightly better in the forecast within low volatility regimes (pre-crisis), closely followed by the LSTM-29. On the other hand, during the crisis period, the LSTM-29 performed better than LSTM-1 and R-GARCH in 28 out of 29 cases, and in 20 out of 29 cases when compared to GJR-MEM. Furthermore, R-GARCH was never able to achieve the best forecasting performance for any of the assets during both the pre-crisis and in-crisis periods.

In order to evaluate how much model A is better than model B, we compute their MSE ratio as:

$$\text{ratioMSE}(A, B) = \frac{\text{MSE}_A}{\text{MSE}_B}$$

which gives a value greater than 1, if model B has better accuracy than model A, and smaller than 1 otherwise. This metric has been applied in order to compare the four models performance during the pre-crisis and in-crisis periods.

As can be seen from Fig. 6a, the LSTM-29 is performing better than LSTM-1 for 28 out of 29 assets (i.e., all except for the MCD—the only point below the reference line) during the in-crisis period, with up to 1.8 MSE ratio.

When compared to the R-GARCH model (Fig. 6b), the LSTM-29 is showing similar performance. Again, the MCD is better predicted by the R-GARCH in both periods and three other assets are with worsened MSE ratio but are still better predicted by the LSTM-29. The remaining 25 assets showed an improved performance of LSTM-29 during the in-crisis period.

Lastly, the comparison with GJR-MEM (Fig. 6c), shows the LSTM-29 with increased accuracy on 15 assets during the 200 high risk days. Five assets are with slightly worsened prediction during the in-crisis period, five assets have close prediction accuracy (at near 1 ratio), and three assets (i.e., INTC, MSFT, and SPY) are better predicted by the GJR-MEM for both before and during the crisis.

Overall, the above discussed empirical results suggest that the use of the LSTM approach for volatility forecasting can be particularly profitable in turbulent periods where the economic pay-off derived from generation of more accurate volatility forecasts is potentially more substantial than those in more tranquil periods.

**Table 9** Diebold-Mariano statistic for the DJI 500 dataset (with a $p$-value given in brackets) for the LSTM-29 against LSTM-1, R-GARCH and GJR-MEM for the four volatility regimes

| | | LSTM-29 vs. | | |
|---|---|---|---|---|
| | | LSTM-1 | R-GARCH | GJR-MEM |
| Moving Average | Very Low | 6.65 (1.000) | – 2.12 (0.017) | 2.21 (0.987) |
| | Low | 2.56 (0.995) | – 2.30 (0.011) | 1.07 (0.859) |
| | High | – 1.79 (0.037) | – 5.23 (< 0.001) | – 1.91 (0.028) |
| | Very High | – 7.11 (< 0.001) | – 6.87 (< 0.001) | – 2.58 (0.005) |
| Moving Variance | Very Low | 10.64 (1.000) | 0.06 (0.524) | 4.83 (1.000) |
| | Low | 5.06 (1.000) | – 2.51 (0.006) | 2.55 (0.995) |
| | High | – 2.88 (0.002) | – 7.75 (< 0.001) | – 1.13 (0.13) |
| | Very High | – 7.02 (< 0.001) | – 6.66 (< 0.001) | – 3.00 (0.001) |

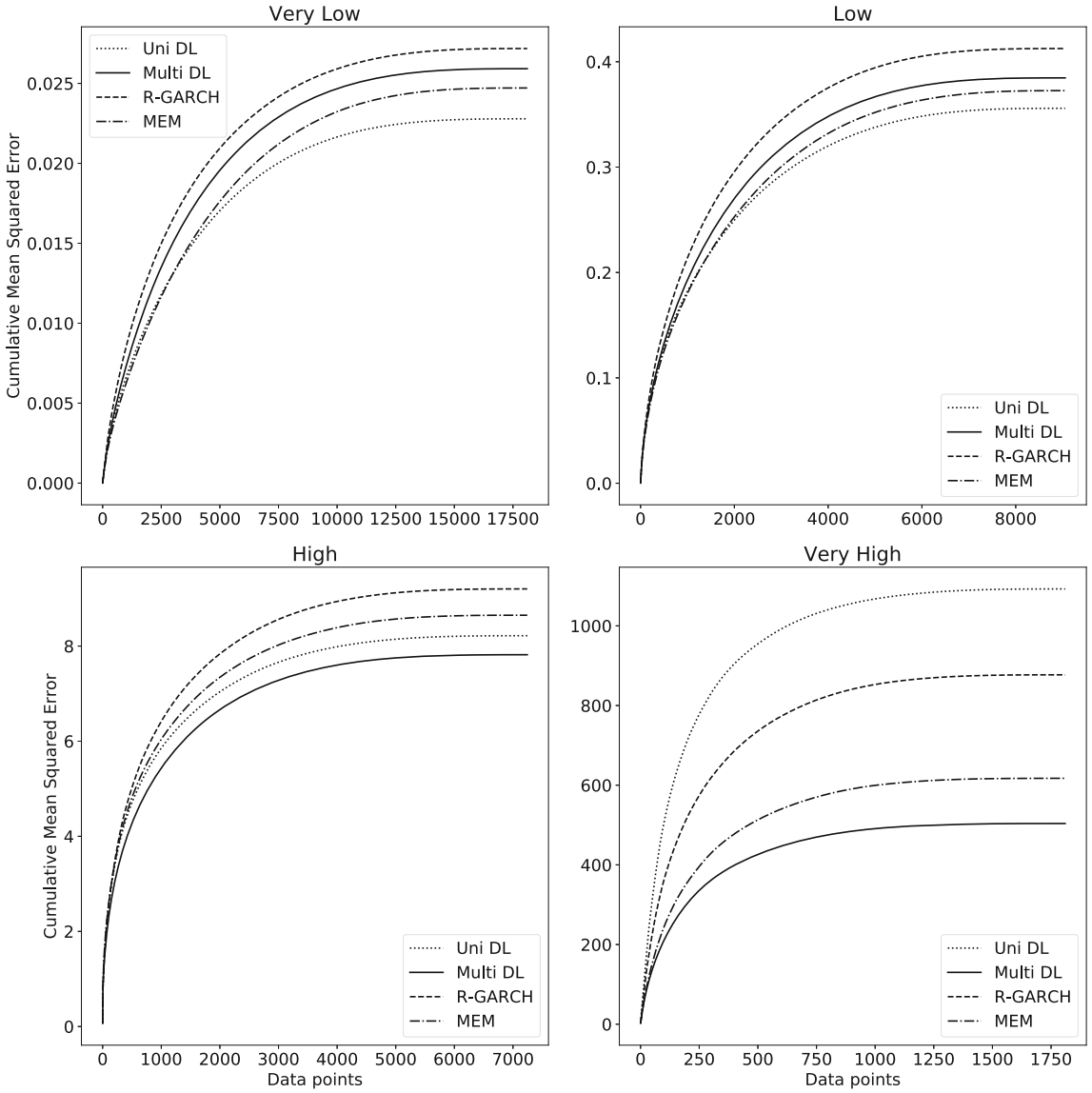
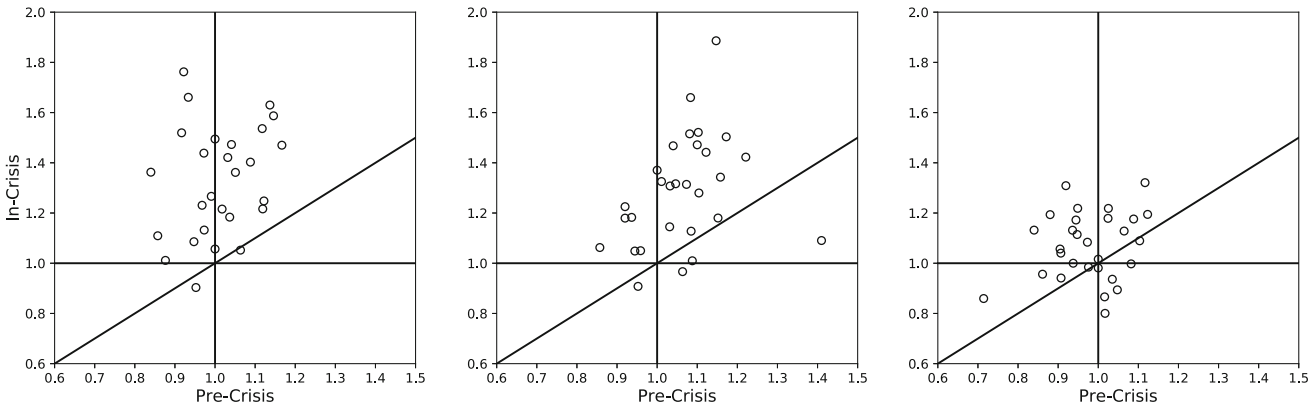



**Fig. 5** Cumulative MSE for the four models at different volatility regimes (VL, L, H, and VH). The four volatility levels are calculated using the 50, 75 and 95 percentiles over the 29 assets. The different scale on the y-axis is due to the magnitude of the error in the four volatility regimes

**Table 10** MSE before and during the crisis periods on the DJI 500 dataset for the four considered models (the best performing model is given in bold)

| Asset | Before crisis | | | | During crisis | | | |
|---|---|---|---|---|---|---|---|---|
| | LSTM-1 | LSTM-29 | R-GARCH | GJR-MEM | LSTM-1 | LSTM-29 | R-GARCH | GJR-MEM |
| AA | 2.66 | **2.38** | 2.79 | 2.44 | 15.90 | **10.35** | 15.56 | 12.61 |
| AIG | 3.24 | 2.85 | 3.48 | **2.62** | 25.07 | **15.38** | 21.88 | 20.13 |
| AXP | 0.71 | **0.65** | 0.68 | 0.73 | 16.35 | **7.30** | 9.61 | 8.72 |
| BA | 0.58 | **0.57** | 0.62 | 0.59 | 2.48 | **2.04** | 2.06 | 1.91 |
| BAC | 0.37 | **0.34** | 0.39 | 0.37 | 22.78 | **16.24** | 30.63 | 19.10 |
| C | 0.51 | 0.39 | 0.43 | **0.37** | 34.19 | **13.49** | 20.52 | 16.44 |
| CAT | 0.85 | **0.82** | 0.88 | 0.84 | 2.45 | **2.07** | 2.72 | 2.44 |
| CVX | 0.91 | 0.78 | 0.78 | **0.73** | 3.69 | **2.51** | 3.44 | 2.84 |
| DD | 0.55 | **0.48** | 0.53 | **0.48** | 6.92 | **4.36** | 5.58 | 4.43 |
| DIS | 0.65 | **0.63** | 0.67 | 0.64 | 2.97 | 2.09 | 2.02 | **1.81** |
| GE | **0.21** | 0.25 | 0.23 | **0.21** | 2.48 | **1.82** | 2.23 | 2.06 |
| GM | 7.15 | **6.39** | 7.36 | 7.05 | 34.18 | **28.10** | 33.16 | 30.61 |
| HD | 0.70 | 0.75 | 0.69 | **0.68** | 12.39 | **7.46** | 8.80 | 7.76 |
| IBM | 0.25 | 0.25 | 0.26 | **0.22** | 2.78 | **1.86** | 2.73 | 2.22 |
| INTC | **0.85** | 0.97 | 0.93 | 0.88 | 4.46 | 4.41 | 4.63 | **4.15** |
| JNJ | **0.30** | 0.31 | 0.32 | 0.33 | 0.48 | **0.39** | 0.51 | 0.44 |
| JPM | 0.62 | **0.59** | 0.64 | 0.60 | 25.71 | **18.88** | 21.29 | 15.11 |
| KO | **0.18** | 0.19 | 0.22 | **0.18** | 0.76 | **0.70** | 0.94 | 0.78 |
| MCD | **1.20** | 1.26 | 1.20 | 1.32 | 1.96 | 2.17 | 1.97 | **1.94** |
| MMM | 0.46 | 0.41 | 0.46 | **0.40** | 1.61 | 1.29 | 1.86 | **1.27** |
| MRK | **6.37** | 6.43 | 6.50 | 7.18 | 25.81 | **20.38** | 27.01 | 26.92 |
| MSFT | 0.33 | 0.36 | 0.34 | **0.31** | 3.13 | 2.06 | 2.16 | **1.97** |
| PG | **0.20** | **0.20** | 0.22 | **0.20** | 0.56 | 0.53 | 0.78 | **0.52** |
| SPY | 0.06 | 0.07 | 0.06 | **0.05** | 0.71 | 0.64 | 0.68 | **0.55** |
| T | 1.27 | **1.22** | 1.72 | 1.32 | 5.70 | 3.87 | 4.22 | **3.86** |
| UTX | 0.35 | 0.36 | 0.39 | **0.34** | 3.51 | **2.44** | 4.05 | 2.86 |
| VZ | **0.59** | 0.64 | 0.66 | 0.60 | 5.71 | **3.24** | 3.71 | **3.24** |
| WMT | **0.36** | 0.37 | 0.40 | **0.36** | 2.57 | **2.27** | 3.44 | 2.46 |
| XOM | 0.67 | 0.63 | 0.59 | **0.57** | 2.42 | **2.30** | 2.72 | 2.43 |




**(a)** LSTM-1 **(b)** R-GARCH **(c)** GJR-MEM

**Fig. 6** MSE rations of the LSTM-29 compared to: **a** the LSTM-1; **b** the R-GARCH; and **c** the GJR-MEM models. The x-axes represent the pre-crisis period (up to 1st July 2007) and the y-axes—the in-crisis one (after 1st July 2007). Each dot is an asset and the bisect line is shown as reference

### 4.2 NASDAQ 100

Results for the NASDAQ 100 dataset are reported in Tables 11, 12 and Fig. 7. The latter shows the models' cumulative MSE profile in the four volatility regimes for the NASDAQ 100 dataset. As already observed with the DJI 500 (Fig. 5), the proposed method generally achieves better accuracy when compared to the R-GARCH and GJR-MEM. In this experiment, the univariate LSTM-1 not only outperforms the state-of-the-art methods, but also the multivariate counterpart (LSTM-92) in all volatility regimes.

Table 11 reports the errors (mean, median, standard deviation (std) and median absolute deviation (MAD)) for the four volatility regimes. As it can be seen from Fig. 7, the LSTM-1 has smaller errors when compared to all other methods, for both moving average and moving variance volatilities over time. The mean/std are particularly high due to the difference in magnitudes across the 92 assets, which is also evident when using the more robust median/MAD metrics.

Furthermore, the DM test (Table 12) is used to statistically assess the difference in errors between the best model (LSTM-1) and the others (LSTM-92, R-GARCH and GJR-MEM). The DM test shows the LSTM-1 to be statistically better than the R-GARCH in all volatility regimes ($p$-value $< 0.05$), better than LSTM-92 in three volatility regimes (i.e., VL, H and VH) for moving variance, and in another three regimes (i.e., L, H and VH) for the moving variance. In the case of GJR-MEM, the LSTM-1 results are statistically better in two volatility settings (i.e., VL and H).

As it can be seen, the LSTM-1 outperforms the multivariate model for the NASDAQ data. This limitation could be due to the fact that while we are increasing the number of assets, the number of samples in the time series is shorter (i.e., curse of dimensionality). To allow the multivariate model to better learn the assets' interactions, there is the need for more data points.

## 5 Comparison with recurrent neural networks (RNN)

Eventually, we compare the proposed deep model with the classic Elman Network (also known as Simple Recurrent Network) (Elman 1990) on both DJI 500 and NASDAQ 100. The Elman RNN topology only stores the previous values of the hidden units, thus being only able to exploit information from the most recent past. This comparison is carried out to further justify the use of the more complex LSTM model. Table 13 presents the mean/std and median/MAD of the two methods (both univariate and multivariate) for the two datasets. As can be seen, the LSTMs performances are generally better than the RNN counterparts, achieving lower estimate errors for all analyzed volatility regimes and across

**Table 11** Average with Std (in brackets) errors, Median with MAD (in brackets) errors for the four models at different volatility regimes (VL, L, H and VH) on the NASDAQ 100 dataset

| | | LSTM-1 | | LSTM-92 | | R-GARCH | | GJR-MEM | |
|---|---|---|---|---|---|---|---|---|---|
| | | Mean (Std) | Median (MAD) | Mean (Std) | Median (MAD) | Mean (Std) | Median (MAD) | Mean (Std) | Median (MAD) |
| Moving average | Very low | **0.557** (3.311) | 0.082 (0.112) | 0.590 (2.960) | 0.084 (0.118) | 1.051 (2.765) | 0.566 (0.649) | 0.629 (2.675) | 0.183 (0.237) |
| | Low | **3.051** (20.954) | 0.312 (0.432) | 3.165 (15.148) | 0.355 (0.495) | 3.629 (13.814) | 1.105 (1.361) | 3.061 (14.050) | 0.606 (0.802) |
| | High | **15.233** (94.568) | 0.836 (1.181) | 16.201 (96.318) | 1.048 (1.481) | 17.127 (92.106) | 2.296 (3.089) | 15.933 (94.485) | 1.555 (2.111) |
| | Very high | **161.635** (1771.79) | 3.022 (4.330) | 167.151 (1771.46) | 3.714 (5.322) | 178.215 (1742.53) | 7.645 (10.896) | 186.709 (2023.93) | 5.054 (7.166) |
| Moving variance | Very low | **0.497** (2.376) | 0.090 (0.123) | 0.502 (1.355) | 0.093 (0.130) | 1.007 (1.923) | 0.569 (0.659) | 0.546 (1.147) | 0.193 (0.252) |
| | Low | **2.565** (9.518) | 0.339 (0.474) | 2.750 (7.390) | 0.385 (0.544) | 3.322 (6.759) | 1.199 (1.474) | 2.608 (6.313) | 0.639 (0.860) |
| | High | **12.955** (52.915) | 0.682 (0.976) | 13.964 (51.865) | 0.848 (1.220) | 15.120 (48.385) | 2.051 (2.803) | 13.207 (47.936) | 1.333 (1.848) |
| | Very high | **173.773** (1778.15) | 1.178 (1.697) | 179.051 (1778.06) | 1.555 (2.249) | 188.216 (1748.81) | 3.272 (4.662) | 200.715 (2029.43) | 2.652 (3.761) |

The four volatility regimes are determined using the 50, 75, and 95 percentiles over all the assets. The first comparison (rows 2 to 5) is using a centered moving average of the volatilities over time (5 time steps before and 5 time steps after the current one), while the second one (rows 6 to 9) is using a centered moving variance of the volatilities over time (5 time steps before and 5 time steps after the current one)

**Table 12** Diebold-Mariano statistic for the NASDAQ 100 dataset (with a $p$-value given in brackets) for the LSTM-1 against LSTM-92, R-GARCH and GJR-MEM for the four volatility regimes

| | | LSTM-1 vs. | | |
|---|---|---|---|---|
| | | LSTM-92 | R-GARCH | GJR-MEM |
| Moving average | Very low | – 3.35 (0.001) | – 40.06 (< 0.001) | – 7.44 (< 0.001) |
| | Low | – 1.13 (0.259) | – 5.54 (< 0.001) | – 0.1 (0.923) |
| | High | – 7.91 (< 0.001) | – 12.49 (< 0.001) | – 3.41 (0.001) |
| | Very high | – 4.42 (< 0.001) | – 5.27 (< 0.001) | – 1.59 (0.113) |
| Moving variance | Very Low | – 0.48 (0.632) | – 38.15 (< 0.001) | – 4.75 (< 0.001) |
| | Low | – 3.6 (< 0.001) | – 13.24 (< 0.001) | – 0.87 (0.383) |
| | High | – 5.85 (< 0.001) | – 10.53 (< 0.001) | – 1.42 (0.155) |
| | Very high | – 4.29 (< 0.001) | – 4.62 (< 0.001) | – 1.7 (0.089) |

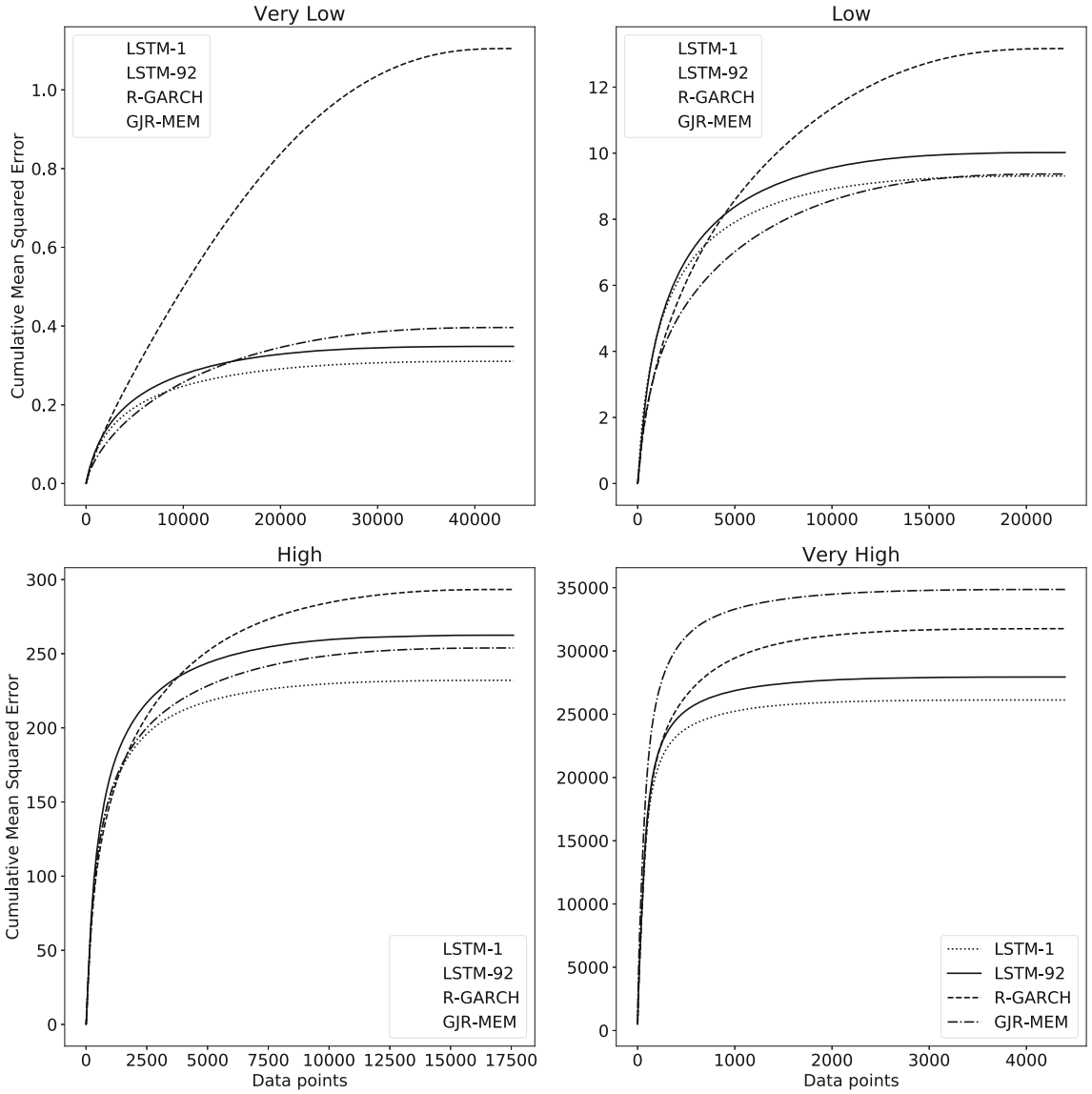
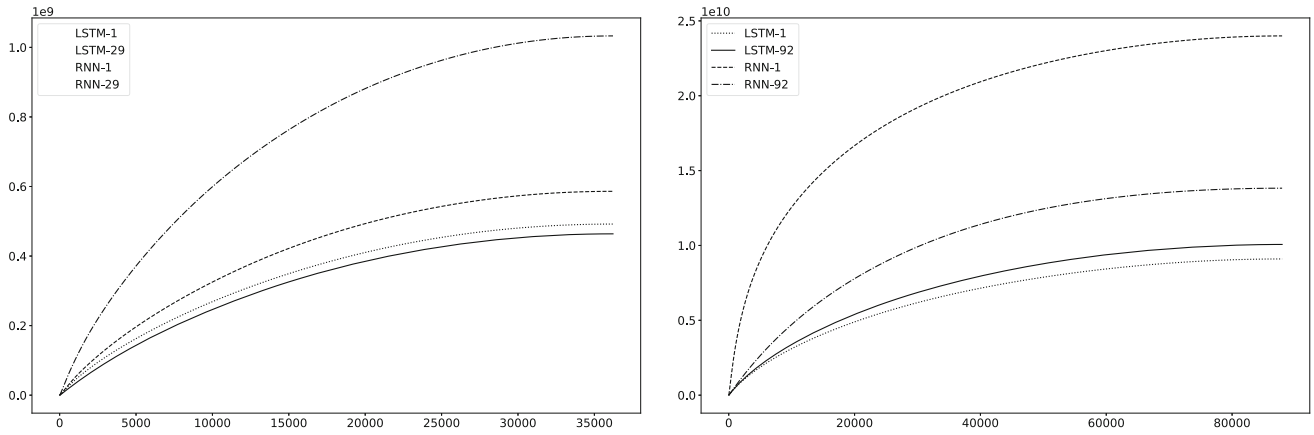



**Fig. 7** Cumulative MSE for the four models at different volatility regimes (VL, L, H and VH). The four volatility levels are calculated using the 50, 75 and 95 percentiles over the 92 assets. The different scale on the y-axis is due to the magnitude of the error in the four volatility regimes

**Table 13** Average with Std (in brackets) errors, Median with MAD (in brackets) errors for the four models at different volatility regimes (VL, L, H and VH) on the DJI 500 and NASDAQ 100 datasets. The four volatility regimes are determined using the 50, 75, and 95 percentiles over all the assets

| | | DJI 500 | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | LSTM-1 | | LSTM-29 | | RNN-1 | | RNN-29 | |
| | | Mean (Std) | Median (MAD) | Mean (Std) | Median (MAD) | Mean (Std) | Median (MAD) | Mean (Std) | Median (MAD) |
| Moving average | Very Low | **0.151** (0.657) | 0.037 (0.051) | 0.161 (0.677) | 0.038 (0.053) | 0.176 (0.687) | 0.04 (0.056) | 0.503 (7.577) | 0.061 (0.086) |
| | Low | **0.596** (3.29) | 0.136 (0.186) | 0.62 (3.234) | 0.139 (0.191) | 0.679 (3.289) | 0.155 (0.213) | 3.074 (82.297) | 0.222 (0.309) |
| | High | 2.867 (35.254) | 0.385 (0.535) | **2.796** (35.338) | 0.385 (0.539) | 3.714 (41.474) | 0.446 (0.629) | 8.303 (140.693) | 0.707 (0.995) |
| | Very High | 33.056 (144.510) | 3.638 (5.130) | **22.447** (104.469) | 3.043 (4.333) | 38.993 (231.107) | 4.405 (6.253) | 61.709 (312.965) | 6.707 (9.585) |
| Moving Variance | Very Low | 0.124 (0.259) | 0.038 (0.053) | 0.138 (0.292) | 0.039 (0.055) | 0.148 (0.307) | 0.041 (0.058) | **0.06** (0.634) | 0.015 (0.373) |
| | Low | 0.472 (0.971) | 0.142 (0.196) | 0.502 (1.021) | 0.143 (0.2) | 0.572 (1.156) | 0.165 (0.231) | **0.084** (1.165) | 0.02 (0.727) |
| | High | 2.253 (7.640) | 0.344 (0.486) | 2.126 (6.702) | 0.335 (0.476) | 2.796 (9.868) | 0.409 (0.582) | **0.234** (3.082) | 0.028 (1.182) |
| | Very High | 36.406 (159.551) | 2.685 (3.848) | 25.946 (125.006) | 2.332 (3.362) | 43.480 (244.150) | 3.4191 (4.942) | **0.647** (8.114) | 0.017 (3.328) |
| | | NASDAQ 100 | | | | | | | |
| | | LSTM-1 | | LSTM-92 | | RNN-1 | | RNN-92 | |
| | | Mean (Std) | Median (MAD) | Mean (Std) | Median (MAD) | Mean (Std) | Median (MAD) | Mean (Std) | Median (MAD) |
| Moving Average | Very Low | **0.557** (3.311) | 0.082 (0.112) | 0.590 (2.960) | 0.084 (0.118) | 2.427 (54.195) | 0.069 (0.097) | 0.928 (4.180) | 0.086 (0.125) |
| | Low | **3.051** (20.954) | 0.312 (0.432) | 3.165 (15.148) | 0.355 (0.495) | 16.100 (199.779) | 0.315 (0.445) | 4.349 (16.878) | 0.413 (0.601) |
| | High | **15.233** (94.568) | 0.836 (1.181) | 16.201 (96.318) | 1.048 (1.481) | 108.482 (1315.065) | 0.970 (1.395) | 19.953 (98.638) | 1.246 (1.822) |
| | Very High | **161.635** (1771.79) | 3.022 (4.330) | 167.151 (1771.46) | 3.714 (5.322) | 764.556 (6103.58) | 4.134 (6.019) | 177.407 (1761.22) | 4.342 (6.374) |
| Moving Variance | Very Low | **0.497** (2.376) | 0.090 (0.123) | 0.502 (1.355) | 0.093 (0.130) | 1.584 (32.389) | 0.077 (0.108) | 0.879 (3.226) | 0.098 (0.142) |
| | Low | **2.565** (9.518) | 0.339 (0.474) | 2.750 (7.390) | 0.385 (0.544) | 10.237 (121.438) | 0.341 (0.488) | 4.28 (12.877) | 0.449 (0.656) |
| | High | **12.955** (52.915) | 0.682 (0.976) | 13.964 (51.865) | 0.848 (1.220) | 78.006 (695.522) | 0.794 (1.155) | 18.125 (60.955) | 0.983 (1.444) |
| | Very High | **173.774** (1778.15) | 1.178 (1.697) | 179.051 (1778.06) | 1.555 (2.249) | 924.192 (6491.16) | 1.612 (2.356) | 185.557 (1767.47) | 1.896 (2.787) |

The first comparison (rows 2 to 5) is using a centered moving average of the volatilities over time (5 time steps before and 5 time steps after the current one), while the second one (rows 6 to 9) is using a centered moving variance of the volatilities over time (5 time steps before and 5 time steps after the current one)




**(a)** DJI 500 **(b)** NASDAQ 100

**Fig. 8** Cumulative MSE for the LSTM and RNN methods (both univariate and multivariate)

all metrics (with only few exceptions for RNN-29 with moving variance). Furthermore, Fig. 8 illustrates the cumulative errors for the DJI 500 (Fig. 8a) and NASDAQ 100 (Fig. 8b) datasets. As can be observed, the error profiles of both univariate and multivariate LSTM are better (lower cumulative error) than those achieved by the two compared RNN models. This result further acknowledges the ability of more complex time series models to exploit both short- and long-term dependencies in the available data.

## 6 Conclusion

In this paper, we investigated the profitability of using LSTM for forecasting daily stock market volatility in order to support decision making in risk management applications. We applied the model to a panel of 29 assets representative of the Dow Jones Industrial Average index over the period 2002–2008, in addition to the market factor proxied by the SPY, and to 92 assets belonging to the NASDAQ 100 index within the period December 2012 to November 2017.

Both periods entail different market regimes related the outrise of two extremely critical events: the global asset management crisis (2007–2008) and the European debt crisis (2011–2012).

Our findings confirmed the superiority of the LSTM over widely popular univariate parametric benchmarks, such as the R-GARCH and GJR-MEM, when forecasting in regimes of high volatility, while still producing comparable predictions for the low/medium volatility periods. These conclusions are result of performance evaluation, using the MSE, QLIKE and the Pearson correlation index in addition to the Diebold-Mariano statistical test.

An attractive feature of the LSTM is that it easily allows taking into account volatility spillover phenomena which are dynamic dependence relationships among the volatilities of different stocks. Such dependency is hard to identify with conventional parametric approaches, due to the need of large number of parameters to be handled by the models. Furthermore, even simple models such as standard vector auto-regressive techniques are easily affected by the curse of dimensionality. On the other hand, the LSTM (belonging to an emerging class of deep learning approaches) demonstrates yet again its capability to cope with complex and highly nonlinear dependencies among the considered variables and in particular, shows superior performance in predicting and forecasting especially in high turbulence and entropy conditions for the considered high volatility stock market periods.

Results of the experiments show the ability of deep learning models to capture cross-volatility dependencies using the whole market raw data, with no background knowledge of the distribution of values and the dependency across assets over time. Therefore, application of LSTM in this framework is beneficial, and, from the perspective of practitioners in Finance, it has the relevant advantage of being almost completely data driven and model-blind from a statistical point of view. Overall, it should be remarked that the degree of complexity of the relationships linking individual asset volatilities within a market is such to prevent the specification and estimation of feasible multivariate parametric models. In this perspective, deep learning models offer a highly valuable tool for making robust and accurate inference on complex phenomena such as volatility spillovers, contagion effects and volatility co-movements and, in general, for accurate risk and volatility forecasting.

**Author Contributions** AP contributed to software, investigation, validation and writing (original draft, and review and editing); LT contributed to methodology, validation and writing (review and editing); AS contributed to software and investigation; IJ performed supervision and writing (review and editing); GS performed formal analysis, validation and writing (original draft); RT performed conceptualization, methodol-

ogy and writing (original draft, and review and editing); MR performed conceptualization, methodology and writing (original draft).

**Fundings** The authors did not receive support from any organization for the submitted work.

## Declarations

**Conflict of interest** The authors have no conflicts of interest to declare that are relevant to the content of this article.

**Human and animal rights** The research did not involve human participants and/or animals.



## References

Acerbi C, Tasche D (2002) Expected shortfall: a natural coherent alternative to value at risk. Econ Notes 31(2):379–388 https://doi.org/10.1111/1468-0300.00091

Andersen TG, Bollerslev T, Diebold FX (2007) Roughing it up: Including jump components in the measurement, modeling, and forecasting of return volatility. Rev Econ Stat 89(4):701–720

Andersen TG, Teräsvirta T (2009) Realized volatility. In: Handbook of financial time series. Springer, pp 555–575

Barndorff-Nielsen OE, Hansen PR, Lunde A, Shephard N (2011) Multivariate realised kernels: consistent positive semi-definite estimators of the covariation of equity prices with noise and non-synchronous trading. J Econometr 162(2):149–169

Barndorff-Nielsen OE, Shephard N (2005) Variation, jumps, market frictions and high frequency data in financial econometrics. Nuffield College Economics Working Paper, vol 1, no 1, pp 1–5

Bauwens L, Laurent S, Rombouts JV (2006) Multivariate garch models: a survey. J Appl Economet 21(1):79–109

Bianchi FM, Maiorino E, Kampffmeyer MC, Rizzi A, Jenssen R (2017) Recurrent neural networks for short-term load forecasting: an overview and comparative analysis. Springer

Blume ME (1971) On the assessment of risk. J Financ 26(1):1–10

Bollerslev T (1986) Generalized autoregressive conditional heteroskedasticity. J Econometr 31:307–327

Chakraborty K, Mehrotra K, Mohan CK, Ranka S (1992) Forecasting the behavior of multivariate time series using neural networks. Neural Netw 5(6):961–970

Chen XB, Gao J, Li D, Silvapulle P (2018) Nonparametric estimation and forecasting for time-varying coefficient realized volatility models. J Business Econ Stat 36(1):88–100

Cho K, van Merrienboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y (2014) Learning phrase representations using rnn encoder-decoder for statistical machine translation

Diebold FX, Mariano RS (2002) Comparing predictive accuracy. J Business Econ Stat 20(1):134–144. https://doi.org/10.1198/073500102753410444

Dunis CL, Laws J, Sermpinis G (2010) Modelling commodity value at risk with higher order neural networks. Appl Finan Econ 20(7):585–600

Elman JL (1990) Finding structure in time. Cogn Sci 14(2):179–211

Engle R (2002) New frontiers for arch models. J Appl Econometr 17(5):425–446

Engle RF, Russell JR (1998) Autoregressive conditional duration: a new model for irregularly spaced transaction data. Econometrica 1127–1162

Fischer T, Krauss C (2018) Deep learning with long short-term memory networks for financial market predictions. Eur J Oper Res 270(2):654–669

Gerlach R, Wang C (2016) Forecasting risk via realized garch, incorporating the realized range. Quanti Finance 16(4):501–511

Gers FA, Eck D, Schmidhuber J (2002) Applying lstm to time series predictable through time-window approaches. In: Neural nets WIRN Vietri-01. Springer, pp 193–200

Gers FA, Schmidhuber J (2001) Long short-term memory learns context free and context sensitive languages. In: Artificial neural nets and genetic algorithms. Springer, pp 134–137

Gers FA, Schmidhuber J, Cummins F (1999) Learning to forget: Continual prediction with lstm. Neural Comput 850–855

Gers FA, Schraudolph NN, Schmidhuber J (2002) Learning precise timing with lstm recurrent networks. J Mach Learn Res 3(1):115–143

Glosten LR, Jagannathan R, Runkle DE (1993) On the relation between the expected value and the volatility of the nominal excess return on stocks. J Financ 48(5):1779–1801

Graves A (2013) Generating sequences with recurrent neural networks. arXiv preprint arXiv:1308.0850

Graves A, Jaitly N, Mohamed AR (2013) Hybrid speech recognition with deep bidirectional lstm. In: 2013 IEEE workshop on automatic speech recognition and understanding (ASRU). IEEE. pp 273–278. IEEE

Graves A, Wayne G, Danihelka I (2014) Neural turing machines

Hamilton J, Susmel R (1994) Autoregressive conditional heteroskedasticity and changes in regime. J Econometr 64(1):307–333. https://doi.org/10.1016/0304-4076(94)90067-1

Han H, Zhang S (2012) Non-stationary non-parametric volatility model. Economet J 15(2):204–225

Han S, Kang J, Mao H, Hu Y, Li X, Li Y, Xie D, Luo H, Yao S, Wang Y et al (2017) Ese: Efficient speech recognition engine with sparse lstm on fpga. In: Proceedings of the 2017 ACM/sigda international symposium on field-programmable gate arrays. ACMM, pp 75–84

Hansen PR, Huang Z, Shek HH (2012) Realized garch: a joint model for returns and realized measures of volatility. J Appl Economet 27(6):877–906

Hirose N, Tajima R (2017) Modeling of rolling friction by recurrent neural network using lstm. In: 2017 IEEE International conference on robotics and automation (ICRA). IEEE, pp. 471–6478

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

Huang Z, Wang T, Hansen PR (2017) Option pricing with the realized garch model: an analytical approximation approach. J Futur Mark 37(4):328–358

Itakura F (1975) Minimum prediction residual principle applied to speech recognition. IEEE Trans Acoust Speech Signal Process 23(1):67–72

Jozefowicz R, Zaremba W, Sutskever I (2015) An empirical exploration of recurrent network architectures. In: Proceedings of the 32nd international conference on machine learning (ICML-15), pp 2342–2350

Kalchbrenner N, Danihelka I, Graves A (2015) Grid long short-term memory

Kim CJ, Kim MJ (1996) Transient fads and the crash of '87. J Appl Economet 11(1):41–58. https://doi.org/10.1002/(SICI)1099-1255(199601)11:1<41::AID-JAE364>3.0.CO;2-R

Kourentzes N, Barrow DK, Crone SF (2014) Neural network ensemble operators for time series forecasting. Expert Syst Appl 41(9):4235–4244

Ladyzynski P, Zbikowski K, Grzegorzewski P (2013) Stock trading with random forests, trend detection tests and force index volume indicators. In: International conference on artificial intelligence and soft computing. Springer, pp 441–452

Langrock R, Michelot T, Sohn A, Kneib T (2015) Semiparametric stochastic volatility modelling using penalized splines. Comput Stat 30(2):517–537

Lee B, Baek J, Park S, Yoon S (2016) deeptarget: end-to-end learning framework for microrna target prediction using deep recurrent neural networks. In: Proceedings of the 7th ACM international conference on bioinformatics, computational biology, and health informatics. ACM, pp 434–442

Leifert G, Strauß T, Grüning T, Wustlich W, Labahn R (2016) Cells in multidimensional recurrent neural networks. J Mach Learn Res 17(1):3313–3349

Maciel L, Ballini R, Gomide F (2017) Evolving possibilistic fuzzy modeling for realized volatility forecasting with jumps. IEEE Trans Fuzzy Syst 25(2):302–314

McAleer M, Medeiros MC (2011) Forecasting realized volatility with linear and nonlinear univariate models. J Econ Surv 25(1):6–18

Nápoles G, Vanhoenshoven F, Falcon R, Vanhoof K (2020) Nonsynaptic error backpropagation in long-term cognitive networks. IEEE Trans Neural Netw Learn Syst 31(3):865–875

Pakel C, Shephard N, Sheppard K (2011) Nuisance parameters, composite likelihoods and a panel of garch models. Statistica Sinica, pp 307–329

Pascanu R, Mikolov T, Bengio Y (2013) On the difficulty of training recurrent neural networks. In: International conference on machine learning, pp 1310–1318

Patton AJ (2011) Volatility forecast comparison using imperfect volatility proxies. J Econometr 160(1):246–256

Poggio T, Mhaskar H, Rosasco L, Miranda B, Liao Q (2017) Why and when can deep-but not shallow-networks avoid the curse of dimensionality: a review. Int J Autom Comput 14(5):503–519

Quan Z, Zeng W, Li X, Liu Y, Yu Y, Yang W (2020) Recurrent neural networks with external addressable long-term and working memory for learning long-term dependences. IEEE Trans Neural Netw Learn Syst 31(3):813–826

Rivest F, Kohar R (2020) A new timing error cost function for binary time series prediction. IEEE Trans Neural Netw Learn Syst 31(1):174–185

Shi X, Chen Z, Wang H, Yeung DY, Wong Wk, Woo Wc (2015) Convolutional lstm network: a machine learning approach for precipitation nowcasting. In: Proceedings of the 28th international conference on neural information processing systems—volume 1, NIPS'15. MIT Press, Cambridge, MA, USA, pp 802–810

Sundermeyer M, Ney H, Schlüter R (2015) From feedforward to recurrent lstm neural networks for language modeling. IEEE Trans Audio Speech Lang Process 23(3):517–529

Sutskever I, Vinyals O, Le QV (2014) Sequence to sequence learning with neural networks. In: Advances in neural information processing systems, pp 3104–3112

Suwajanakorn S, Seitz SM, Kemelmacher-Shlizerman I (2017) Synthesizing obama: learning lip sync from audio. ACM Trans Graph (TOG) 36(4):95

Taylor JW (2019) Forecasting value at risk and expected shortfall using a semiparametric approach based on the asymmetric laplace distribution. J Business Econ Stat 37(1):121–133. https://doi.org/10.1080/07350015.2017.1281815

Tran NT, Luong VT, Nguyen NLT, Nghiem MQ (2016) Effective attention-based neural architectures for sentence compression with bidirectional long short-term memory. In: Proceedings of the seventh symposium on information and communication technology. ACM, pp 123–130

Troiano L, Villa E, Loia V (2018) Replicating a trading strategy by means of lstm for financial industry applications. IEEE Trans Industr Inf 14(7):3226–3234. https://doi.org/10.1109/TII.2018.2811377

Visser MP (2011) Garch parameter estimation using high-frequency data. J Financ Economet 9(1):162–197

Wan L, Zeiler M, Zhang S, Le Cun Y, Fergus R (2013) Regularization of neural networks using dropconnect. In: International conference on machine learning, pp 1058–1066

Wang C (2017) Rra: Recurrent residual attention for sequence learning. arXiv preprint arXiv:1709.03714

Wang C, Chen Q, Gerlach R (2018) Bayesian realized-garch models for financial tail risk forecasting incorporating the two-sided weibull distribution. Quant Finance 1–26

Wang C, Niepert M (2019) State-regularized recurrent neural networks. arXiv preprint arXiv:1901.08817

Wang L, Zeng Y, Chen T (2015) Back propagation neural network with adaptive differential evolution algorithm for time series forecasting. Expert Syst Appl 42(2):855–863

Xu K, Ba J, Kiros R, Cho K, Courville A, Salakhudinov R, Zemel R, Bengio Y (2015) Show, attend and tell: neural image caption generation with visual attention. In: International conference on machine learning, pp 2048–2057

Yao K, Peng B, Zhang Y, Yu D, Zweig G, Shi Y (2014) Spoken language understanding using long short-term memory neural networks. In: Spoken language technology workshop (SLT), 2014 IEEE. IEEE, pp 189–194

Yu Y, Si X, Hu C, Zhang J (2019) A review of recurrent neural networks: Lstm cells and network architectures. Neural Comput 31(7):1235–1270. https://doi.org/10.1162/neco_a_01199

Zaytar MA, El Amrani C (2016) Sequence to sequence weather forecasting with long short-term memory recurrent neural networks. Int J Comput Appl 143(11)

Zhang K, Teo KL (2015) A penalty-based method from reconstructing smooth local volatility surface from American options. J Ind Manag Optim 11:631–644