Neural Networks in Forecasting Financial Volatility

Wenbo Ge[1](✉), Pooia Lalbakhsh[2], Leigh Isai[3], and Hanna Suominen[1,4]

[1] School of Computing, The Australian National University, 145 Science Road,
Acton, Canberra, ACT 2601, Australia
wenbo.ge@anu.edu.au

[2] Department of Data Science and Artificial Intelligence, Monash University,
20 Exhibition Walk, Clayton, VIC 3168, Australia

[3] Euler Capital, 63-67 Cemetery Road, Drysdale, VIC 3222, Australia

[4] Department of Computing, University of Turku, 20014 Turku, Finland

Abstract. In 2020s, the state of the art (SOTA) in financial volatility forecasting is underpinned by deep learning (DL). Despite this, forecasting methods in practice tend to be dominated by their more traditional counterparts (e.g., Generalised Auto-Regressive Conditional Heteroscedasticity (GARCH) models) or relatively simple neural networks (NN), leaving much of DL unexplored. Hence, this study experimented the power of DL in forecasting financial volatility and expedited further progress in such multidisciplinary DL applications to quantitative finance by releasing open-source software and proposing a shared task. We compared the financial forecasting ability of the SOTA methods used to more recent DL work, proceeding from simpler or shallower to deeper and more complex models. Specifically, the volatility of five assets (i.e., S&P500, NASDAQ100, gold, silver, and oil) was forecast with the GARCH models, multi-layer perceptrons, recurrent NNs, temporal convolutional networks, and Temporal Fusion Transformer. The results indicated that in almost all cases, DL models forecast volatility with less error than the SOTA models in financial volatility research. These experiments were repeated and the difference between competing models was shown to be statistically significant, therefore encouraging their use in practice.

Keywords: Applications · Deep learning · Economics · Financial volatility · Multidisciplinary AI · Neural networks · Time series analysis

Supported by Euler Capital Pty Ltd, under APR Intern Agreement (INT - 0804). We would also like to thank the APR.Intern and *Australian National University* (ANU) for their support to this research through the Australian Postgraduate Research Internship Program (APR.Intern) program. We also gratefully acknowledge funding from the *Australian Government Research Training Program* (AGRTP) Domestic Scholarship for the first author's PhD studies.

T. Liu et al. (Eds.): AI 2023, LNAI 14471, pp. 178–189, 2024.
https://doi.org/10.1007/978-981-99-8388-9_15

1 Introduction

One of the most important tasks for a financial institution is to monitor the *volatility* of its portfolio and other market variables. However, there are many different ways to quantify this latent and unobservable variable, such as *historical volatility* (HV, a.k.a. close-to-close, the standard deviation of log-returns over a time window) [27][1], *realised volatility* (RV, the square root of the sum of squared log-returns over a time window) [1], *implied volatility* (IV, backwards calculated from options prices via an option pricing model, such as Black-Scholes) [20], and many more [26]. Because volatility is a key factor in security valuation, risk management, and options pricing, as well as affecting investment choice and valuation of public and corporate liabilities, sophisticated computational models are studied for financial volatility forecasting to support practitioners' judgment and decision-making in quantitative finance [6,19,27,30]. In the 2020s, such computer-assisted forecasting methods are dominated by *Generalised Auto-Regressive Conditional Heteroscedasticity* (GARCH) models and relatively simple *Neural Networks* (NN), leaving much of *Machine Learning* (ML) and *Deep Learning* (DL) unexplored [10].

Hence, this multidisciplinary paper will exemplify the power of ML/DL in forecasting financial volatility to practitioners in quantitative finance. We will compare the financial forecasting ability of a range of methods by proceeding from simpler or shallower models (i.e., the GARCH models and *Multi-Layer Perceptrons* (MLP)) to deeper and more complex NNs (i.e., the *Recurrent NNs* (RNN), *Convolutional NNs* (CNN), *Temporal Convolutional Networks* (TCN), and *Temporal Fusion Transformer* (TFT)). These performance evaluations and statistical analyses on five assets (i.e., S&P500, NASDAQ100, gold, silver, and oil), completed by releasing our Python code[2] under the MIT license should encourage practitioners to apply DL as a way to reduce error in forecasting financial volatility.

2 Related Work

A convenient property of financial price data is the efficient market hypothesis, which stipulates that all publicly available information is reflected in the market prices of assets at a given time point [30]. At the finest resolution, market prices are a list of prices of all the buy and sell orders that were matched, which can then be aggregated over time (e.g., 1-h or 1-day intervals) to create more coarse-grained views and can be further described by its highest, lowest, opening, and closing price for that interval, as well as the total number of assets traded, known as volume; however, introducing additional data tends to be helpful in

[1] Despite the name containing the word 'historical', it is not defined exclusively for historical data. This can still be forecast in the same way that realized volatility can forecast.

[2] All results, tables, figures, and analysis methods can be found at https://github.com/xyz, along with extended results.

this predictive modeling task [21,35]. In addition to the definition of volatility and financial price data, the volatility forecasting model should consider the time period for which the data is useful: If the goal was to forecast for the next 15 min, using data from the previous 50 years might be wasteful but with 1 week, information from past market regimes that could repeat might be missed. Moreover, the amount of information provided at inference time is important as it impacts the computation time, as well as may dilute the useful information; when inferring the volatility of the next 30 days, all data could be useful, but the most recent entries are likely to carry more insights than earlier ones. Finally, the timing of the data and the window of time that the volatility captures must also be considered, keeping in mind that the further into the future we are trying to forecast, the more uncertain any forecast will be. Although this part of modeling should depend on the reason for forecasting volatility[3], asking the model to forecast volatility in a wide range of time frames may be beneficial[4].

Of the many types of models that can be used to understand and forecast volatility, none are as widespread as the *auto-regressive* (AR) models: The seminal *Auto-Regressive Conditional Heteroscedasticity* (ARCH) models future volatility conditioned on previous observations [7] and its adaptation as the well-known GARCH model includes an *Auto-Regressive Moving Average* (ARMA) component [2]. Since these models from the 1980s, there have been many advancements that attempt to address the models' inability to capture several stylized facts of volatility [8][5]. Despite the countless variants of the GARCH model, several experiments have found that the simple ARCH(1) and GARCH(1, 1) forecasting models perform the best [11,24].

ML and DL models have also shown much success and are rising in popularity [4,5,12]. NN-based models are commonly used, and although they do not have the same theoretical underpinnings as the GARCH models, they are flexible, possessing the ability to learn any arbitrary mapping f from input $\mathbf{X}$ to output y; $y = f(\mathbf{X})$. In the context of time series analysis, a *Nonlinear Auto-Regressive* (NAR) framework is often adopted with the MLP, enforcing an AR property to the nonlinear mapping (e.g., $\widehat{y}_{t+1} = f([y_t, y_{t-1}, ..., y_{t-m}]^T)$ with t referring to a given time point) [15]. This can be extended into a NARX framework by including exogenous variables (such as those derived from several indices,

[3] E.g., market makers and day traders may want to monitor short-term volatility in the span of minutes to watch for entry/exit signals.

[4] E.g., (1) if using daily prices in forecasting financial volatility, some assets are not on the market every day, and as a result, when using multiple data streams, a mismatch between the date and time of each point in the time series is likely to be present, calling for interpolation to fill in the missing data between points, or (2) when using ML/DL methods for modeling, having multiple learning signals aggregated can keep the learning on track.

[5] E.g., *Exponential*, *Threshold*, and *Glosten-Jagannathan-Runkle* versions (EGARCH, TGARCH, GJR-GARCH) which allow for asymmetric dependencies in volatility, and the *Integrated* and *Fractionally Integrated* versions (IGARCH, FIGARCH) which address volatility persistence, where an observed shock in the volatility series seems to impact future volatility over a long horizon.

exchange rates, and outputs of GARCH models), thus providing more information to the model [3] which has been beneficial for forecasting performance [14]. Other NN architectures (e.g., RNNs, CNNs, and *Long Short Term Memory* (LSTM) models) have also been used in volatility forecasting. For instance, LSTM and GARCH models have been combined to forecast HV [13] and gold prices can be converted into a 3-channel RGB image and then processed with a pre-trained vgg16 (a well-known and high performing CNN model) [32].

Whilst RNNs, LSTMs, and CNNs are deep models, they are not considered as the *state-of-the-art* (SOTA) for time series processing in DL, and models used in financial volatility forecasting tend to be even shallower and simpler, a distinct gap highlighted in a recent systematic literature review [10]. This is reserved for recent models that have the extremely deep capacity and use complex models, often adapted from other fields such as TCNs, which have been successful in music generation, speech enhancement, and many other areas involving time series [16,23,25]. The TCN is a CNN adaptation, consisting of 1-dimensional convolutional blocks structured in a way that does not violate the temporal ordering of data (i.e., only past data can be seen when forecasting), known as a causal convolution [23]. In conjunction with a progressively increasing dilation size, the receptive field can be increased exponentially as layers increase, thus allowing the exploitation of long-term relationships. These blocks also often use residual connections, layer normalization, gradient clipping, and dropout, all of which have been shown to improve learning and performance [34]. Another recently developed SOTA model that handles sequential data well is the Transformer [31]. Its TFT variant deploys a gating mechanism to skip unused components of the network, variable selection networks to select relevant input variables at each time step, static co-variate encoders to provide context to the model, temporal processing to learn long and short-term relationships, and quantile predictions to forecast with a corresponding confidence [17].

3 Experimental Comparison of Forecasting Models

Our experimental study of forecasting models will next exemplify through comparative performance evaluation and statistical significance testing the power of DL in forecasting financial volatility. The volatility of five assets will be forecast with the SOTA methods; simpler or shallower DL models; and recent deeper and more complex models. The results will indicate that in almost all cases, DL models forecast volatility with less error than the SOTA models in financial volatility research. These experiments will be repeated to give evidence that the difference between competing models is statistically significant, therefore encouraging their use in practice and further study as a shared task.

3.1 Posing the Problem as a Shared Task

Volatility was forecast for five assets: S&P500, NASDAQ-100 (NDX), gold, silver, and oil. The data for each, as well as the corresponding volatility indices, were

Table 1. Description of data

Asset	Start date	End date
S&P500	22/Sep/2003	31/Dec/2018
NASDAQ100	01/Jan/2003	31/Dec/2018
Gold	03/Jun/2008	31/Dec/2018
Silver	16/Mar/2011	31/Dec/2018
WTI Crude Oil	10/May/2007	31/Dec/2018

retrieved from Global Financial Data[6] (Table 1). The proper permissions to use the data for the purposes of this study and its reporting were obtained from Global Financial Data[7]. The data consisted of the daily closing prices, as well as the open, high, and low prices for S&P500, NDX, gold, and oil. Volume was available only for S&P500 and NDX. Each asset was restricted to a starting date that corresponded with when the volatility index was introduced, except for S&P500 and NDX. This was because the volatility index for S&P500 was originally for the S&P100 and later changed on 22 September 2003 and because the volatility index for NDX began earlier than one of the exogenous variables used. Additionally, the ending date was restricted to 31 December 2018.

Exogenous variables were also retrieved from Global Financial Data, consisting of several other indices (SZSE, BSE SENSEX, FTSE100, and DJIA), exchange rates (US-YEN, US-EURO, and the US dollar trade weighted index), and United States fundamentals (Federal Reserve primary credit rate, mean and median duration of unemployment, consumer price index inflation rate, Government debt per *Gross Domestic Product* (GDP), gross Federal debt, and currency in circulation). All variables were date matched with the underlying assets by bringing forward the nearest historical value.

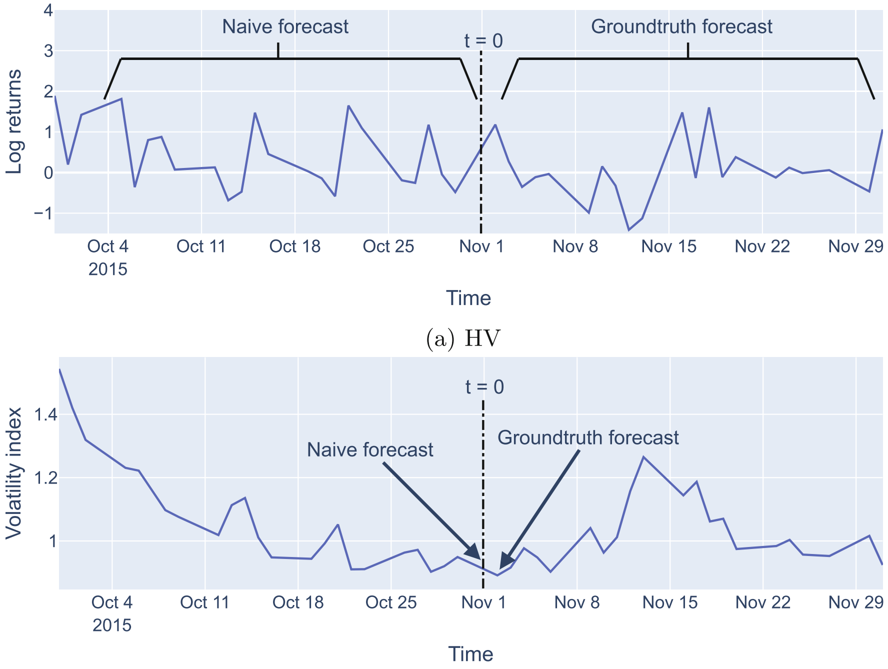
The task was to forecast the month-long HV and IV (Fig. 1), starting from 1 day ahead, for S&P500, NDX, gold, silver, and oil. The ground truth for HV was the standard deviation of log returns starting from 1 trading day ahead to 22 trading days ahead (= one calendar month). In other words, with t referring to the current time, we defined HV over a certain period $[\tau_1, \tau_2] = [t+1, t+22]$ as the standard deviation ($\mathrm{std}(\cdot)$) of log-returns as follows:

$$\mathrm{HV} = \sqrt{\frac{1}{N}\sum_{t=\tau_1}^{\tau_2}\left(r_t - \frac{1}{N}\sum_{t=\tau_1}^{\tau_2} r_t\right)^2} = \mathrm{std}\left(\begin{bmatrix} r_{\tau_2} \\ r_{\tau_2-1} \\ r_{\tau_2-2} \\ \vdots \\ r_{\tau_1} \end{bmatrix}\right) \tag{1}$$

where $N = \tau_2 - \tau_1 = 21$ is the number of samples between the time steps, P_t is price at time t, and $r_t = \log(P_t/P_{t-1}) \cdot 100$. For IV, this meant the ground

[6] https://globalfinancialdata.com/.

[7] Due to the underlying data use agreement, the data or their derivatives cannot be released as part of this paper.



(a) HV

(b) IV

Fig. 1. Groundtruth and naïve forecasts for HV (a) and IV (b).

truth was simply the value of the volatility index for the next trading day, as the volatility index was already defined for the next calendar month. The values of the volatility indices were also adjusted by a factor of $1/\sqrt{252}$, de-annualizing the value to be on the same scale as HV.

3.2 Methods

Five methods were used to represent the SOTA financial volatility forecasting performance, the combination of which will be the benchmark for comparison. These five methods were: a naïve model[8], a GARCH model, an MLP model, and two models from literature: ANN-GARCH [14], and CNN-LSTM [32].

Two models from DL were investigated to represent the experimental forecasting performance. The first model was the TCN, as well as the TCN with

[8] The naïve model simply repeated the most recent known value of volatility (Fig. 1). For IV, this was the value of the corresponding volatility index at the current time. For HV, this was the standard deviation of log returns for the current day and previous 20 days, that is, Eq. (1) with $\tau_1 = t - 20$ and $\tau_2 = t$. This assumed that at the time of forecasting, the current trading day is over and observed, an assumption maintained for all forecasting models.

several modifications. The first modification was to leverage the naïve model and forecast a residual, defined as either the difference $y_{\text{groundtruth},t} - y_{\text{naïve},t}$, or the log difference $log(y_{\text{groundtruth},t}/y_{\text{naïve},t})$. Another modification was to include multiple tasks to the network, introducing another loss function that will have a separate but related training effect [28]. The additional task was to predict either the direction of the forecast (up or down), or the change in direction (change or no change). The final modification was to include additional input channels, introducing new information to the network [33], such as descriptors of the underlying asset (log returns, naïve forecasts for HV and IV, and current direction of movement), and variables that describe the market (US dollar trade-weighted index, Federal Reserve primary credit rate, mean and median duration of unemployment, consumer price index inflation rate, Government debt per GDP, gross Federal debt, and currency in circulation) as there is literature to suggest that this may improve performance [14]. The second DL model explored was the TFT. Additional variables like descriptors of the asset (open, high, low, close, volume where possible, log returns, squared log returns, inverse price of the underlying asset, and naïve forecasts for both HV and IV), descriptors of the market (the US dollar trade-weighted index, Federal Reserve primary credit rate, mean and median duration of unemployment, consumer price index inflation rate, Government debt per GDP, gross Federal debt, and currency in circulation), and descriptors of time (day of the week, month, and a number of days since previous observation) were also included.

To engineer and evaluate the forecasting models using these five methods, a 70-15-15 train-validation-test split of the data was used because it did not violate the temporal aspect of the data[9]. All performances were quantified with the *Mean Squared Error* (MSE) with statistical significance testing to distinguish if competing models were statistically significantly different from each other. After the hyperparameters of a model were chosen and the testing phase was completed, the model was reinitialized with a random seed, re-trained, and retested. This was repeated until ten MSE values were obtained for each model. These values were then tested across different models in a pair-wise fashion to determine if they were from the same distribution[10]. The *Shapiro-Wilk* (SW) test was first applied to assess the normality of the distribution with significance level $\alpha = 0.05$[11]. If both distributions were normal then Student's t-test was used, otherwise the *Kruskal-Wallis* (KW) test was employed.

[9] The training set was used to train the models using different hyperparameters, which were then evaluated against the validation set to determine the performance with that given set of hyperparameters. Different combinations of hyperparameters were searched, and the best performing set of hyperparameters proceeded to the test phase. Here, the model was re-initialized and trained again with the union of both the training and validation set, then evaluated once using the test set.

[10] If so, the models were assumed to be equivalent, if not, then the model with the smaller mean MSE was assumed to be superior.

[11] The SW test was justified by the applicability of the test to the data with unspecified mean and variance, as well as its appropriateness for small sample sizes.

Table 2. Smallest error models (in bold), and models for which no statistically significant difference could be found from the smallest error models.

Task	Model	S&P500	NDX	Gold	Silver	Oil
HV	Smallest error	**TFT**	**TFT**	**ANN-GARCH**	**TCN var.**	**TFT**
	Equivalent	–	–	TFT	TCN	–
IV	Smallest error	**TCN var.**	**TCN var.**	**TCN var.**	**TCN var.**	**TCN var.**
	Equivalent	–	–	–	–	–

Table 3. Performance (MSE) of forecasting models on the test set. Smaller is better, bold is the best. HS refers to Hyperparameter Search.

Model Family and Task	Method	S&P500	NDX	Gold	Silver	Oil
Benchmark Models						
HV	Naïve	0.1281	0.1887	0.0216	0.0745	0.2799
	GARCH	0.2662	0.9430	0.0191	0.0611	7.9448
	MLP	0.2935	0.3542	0.0146	0.0666	**0.2080**
	ANN-GARCH	**0.1136**	**0.1735**	**0.0105**	0.0563	0.2191
	CNN-LSTM	0.1550	0.1869	0.0554	**0.0488**	0.3870
IV	Naïve	**0.0098**	**0.0084**	**0.0013**	**0.0023**	**0.0139**
	GARCH	0.2296	0.5859	0.0263	0.0279	7.6927
	MLP	0.0172	0.0151	0.0021	0.0032	0.0278
	ANN-GARCH	0.0248	0.0159	0.0022	0.0036	0.0312
	CNN-LSTM	0.0141	0.0104	0.0064	0.0104	0.0246
Experimental Models						
HV	Naïve	0.1281	0.1887	0.0216	0.0745	0.2799
	Best benchmark	0.1136	0.1735	**0.0105**	0.0488	0.2080
	TCN	0.1467	0.1912	0.0331	0.0364	0.2614
	TCN variants	0.1300	0.1728	0.0153	**0.0315**	0.2050
	TFT	**0.0294**	**0.0513**	0.0116	0.0341	**0.0864**
IV	Naïve	0.0098	0.0084	0.0013	0.0023	0.0139
	Best benchmark	0.0141	0.0104	0.0021	0.0032	0.0246
	TCN	0.0098	0.0083	0.0029	0.0032	0.0136
	TCN variants	**0.0093**	**0.0081**	**0.0013**	**0.0022**	**0.0127**
	TFT	0.0094	0.0083	0.0013	0.0025	0.0134
HS for TCN						
HV	Grid search	**0.1467**	**0.1912**	**0.0331**	**0.0364**	0.2614
	BOHB	0.1591	0.2010	0.0616	0.0800	**0.2107**
IV	Grid search	**0.0098**	**0.0083**	0.0029	0.0032	**0.0136**
	BOHB	0.0126	0.0093	**0.0017**	**0.0025**	0.0139

3.3 Result Evaluation and Analysis

A comparison of the benchmark models and experimental models gave evidence of a clear trend. Across almost all volatility forecasting tasks and assets investigated, the experimental models outperformed the benchmark models, with statistical significance. Based on 10 repetitions, for almost all assets and tasks, the performance values from the experimental models were superior and found to be statistically significant (Table 2).

Of the benchmark models, the ANN-GARCH model from literature performed best overall in forecasting HV, but for IV forecasting, the naïve model performed the best overall, achieving the smallest errors for all five assets (Table 3). However, a comparison of the traditional grid search hyperparameter optimization method against the more recent *Bayesian Optimisation HyperBand* (BOHB) search [9] indicated no clear trend; BOHB only produced a better forecasting model for the HV of oil, and the IV of gold and silver . Both methods were given roughly the same wall time and were both tested using the un-modified TCN. Though it is difficult to say if one method is superior to the other, the continued use of grid search is justified and was the primary hyperparameter optimization method for the remaining experimental TCN models.

Of the experimental models, the TFT performed best overall for HV forecasting, achieving the smallest errors for S&P500, NDX, and oil (Table 3). An encoder length of 21 days was optimal for all assets, with no set of input variables that were consistently best. S&P500 and NDX performed best with the addition of variables that describe time and the underlying asset, gold and silver performed best with the addition of variables that describe time, and oil performed best with the addition of market and time descriptors. The inclusion of exogenous variables only increased the performance for forecasting gold HV. The smallest error for gold was achieved by a benchmark model, specifically the ANN-GARCH.

The TCN variants were the best performing experimental model for IV forecasting, achieving the smallest errors for all assets (Table 3). The optimal modification was to use a secondary task of predicting the direction, as well as forecasting the residuals, consistent amongst all assets. S&P500, NDX, and gold also benefited from the inclusion of the volatility index value and previous direction of movements. For the TFT model, an encoder length of 10 days was preferred for all assets, except for S&P500 which preferred a length of 126.

4 Discussion

This experimental study exemplified the value of DL in forecasting financial volatility and expedited further progress in such DL applications by releasing open-source software and proposing a shared task. It created a benchmark of experimental evaluation results that consisted of the SOTA in NN-based financial volatility forecasting, several traditional models, and a naïve baseline model. This was then compared to several DL methods, representing the competing experimental models.

These results, however, come with some limitations. The main limitation is that the implementation of several models (GARCH and TFT) was open source and thus not necessarily under the same strict control as the other models used.

This study differs from prior publications by presenting a multidisciplinary approach to DL experimentation in forecasting financial volatility. While some other studies on financial volatility forecasting exist, they tend to be limited to literature reviews [10,27,29] or expert systems in economics [5,14]. Our results imply that DL may offer better volatility forecasting performance than traditional methods, and hence, our code release and proposed shared task should expedite this future work. The most obvious is an investigation into other DL models that have not yet been used for volatility forecasting. Combined with the larger capacity of deeper models, another avenue to enhance the models is to make use of multi-modal data (e.g., extend from numeric data to text [18,22]).

Moving forward, the most vital work is not further exploration of DL models and methods, but rather, the establishment of the proposed shared task that could include, for example, sharing of relevant resources (e.g., code to train models and/or the resulting trained models) and tracks for studying models on a given data modality or expanding them across modalities. This would allow easy and direct comparisons without the need to implement competing models, enabling the synthesis of publications, and propelling the field of financial volatility forecasting further and faster. This task should help gain a deeper understanding of the factors and mechanisms that may affect the economic feasibility of a statistical result. In conclusion, harvesting the diversity of thought and other community effects is likely to accelerate knowledge discovery and methodological innovations required to proceed from statistical significance to economic impact.

References

1. Andersen, T.G., Bollerslev, T.: Answering the skeptics: yes, standard volatility models do provide accurate forecasts. Int. Econ. Rev. **39**(4), 885 (1998). https://doi.org/10.2307/2527343
2. Bollerslev, T.: Generalized autoregressive conditional heteroskedasticity. J. Econometrics **31**(3), 307–327 (1986). https://doi.org/10.1016/0304-4076(86)90063-1
3. Bucci, A.: Cholesky-ANN models for predicting multivariate realized volatility. J. Forecast. **39**(6), 865–876 (2020). https://doi.org/10.1002/for.2664
4. Cavalcante, R.C., Brasileiro, R.C., Souza, V.L.F., Nobrega, J.P., Oliveira, A.L.I.: Computational intelligence and financial markets: a survey and future directions. Expert Syst. Appl. **55**, 194–211 (2016). https://doi.org/10.1016/j.eswa.2016.02.006
5. Chen, W.J., Yao, J.J., Shao, Y.H.: Volatility forecasting using deep neural network with time-series feature embedding. Econ. Res.-Ekonomska Istraživanja 1–25 (2022). https://doi.org/10.1080/1331677X.2022.2089192
6. Edwards, S., Biscarri, J.G., Pérez de Gracia, F.: Stock market cycles, financial liberalization and volatility. J. Int. Money Finance **22**(7), 925–955 (2003). https://doi.org/10.1016/j.jimonfin.2003.09.011
7. Engle, R.F.: Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation. Econometrica **50**(4), 987–1008 (1982). https://doi.org/10.2307/1912773

8. Engle, R.F., Patton, A.J.: 2 - What good is a volatility model?*. In: Knight, J., Satchell, S. (eds.) Forecasting Volatility in the Financial Markets, 3rd edn, pp. 47–63. Quantitative Finance, Butterworth-Heinemann, Oxford (2007). https://doi.org/10.1016/B978-075066942-9.50004-2
9. Falkner, S., Klein, A., Hutter, F.: BOHB: robust and efficient hyperparameter optimization at scale. In: Dy, J., Krause, A. (eds.) Proceedings of the 35th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 80, pp. 1437–1446. PMLR (2018). https://proceedings.mlr.press/v80/falkner18a.html
10. Ge, W., Lalbakhsh, P., Isai, L., Lenskiy, A., Suominen, H.: Neural network based financial volatility forecasting: a systematic review. ACM Comput. Surv. **55**(1), 14:1–14:30 (2022). https://doi.org/10.1145/3483596
11. Hansen, P.R., Lunde, A.: A forecast comparison of volatility models: does anything beat a GARCH(1,1)? J. Appl. Economet. **20**(7), 873–889 (2005). https://doi.org/10.1002/jae.800
12. Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., Muller, P.A.: Deep learning for time series classification: a review. Data Min. Knowl. Disc. **33**(4), 917–963 (2019). https://doi.org/10.1007/s10618-019-00619-1
13. Kim, H.Y., Won, C.H.: Forecasting the volatility of stock price index: a hybrid model integrating LSTM with multiple GARCH-type models. Expert Syst. Appl. **103**, 25–37 (2018). https://doi.org/10.1016/j.eswa.2018.03.002
14. Kristjanpoller, W., Hernández, E.: Volatility of main metals forecasted by a hybrid ANN-GARCH model with regressors. Expert Syst. Appl. **84**, 290–300 (2017). https://doi.org/10.1016/j.eswa.2017.05.024
15. Kumar, P.H., Patil, S.B.: Estimation forecasting of volatility using ARIMA, ARFIMA and neural network based techniques. In: 2015 IEEE International Advance Computing Conference (IACC), pp. 992–997. IEEE (2015). https://doi.org/10.1109/IADCC.2015.7154853
16. Lea, C., Flynn, M.D., Vidal, R., Reiter, A., Hager, G.D.: Temporal convolutional networks for action segmentation and detection. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, pp. 1003–1012. IEEE (2017). https://doi.org/10.1109/CVPR.2017.113
17. Lim, B., Arik, S.O., Loeff, N., Pfister, T.: Temporal fusion transformers for interpretable multi-horizon time series forecasting. arXiv:1912.09363 [cs, stat] (2020)
18. Malik, F.: Estimating the impact of good news on stock market volatility. Appl. Financ. Econ. **21**(8), 545–554 (2011). https://doi.org/10.1080/09603107.2010.534063
19. Masset, P.: Volatility Stylized Facts. SSRN Scholarly Paper ID 1804070, Social Science Research Network, Rochester (2011). https://doi.org/10.2139/ssrn.1804070
20. Mayhew, S.: Implied volatility. Financ. Anal. J. **51**(4), 8–20 (1995). https://doi.org/10.2469/faj.v51.n4.1916
21. Neupane, B., Thapa, C., Marshall, A., Neupane, S.: Mimicking insider trades. J. Corp. Finan. **68**, 101940 (2021). https://doi.org/10.1016/j.jcorpfin.2021.101940
22. Oliveira, N., Cortez, P., Areal, N.: The impact of microblogging data for stock market prediction: using Twitter to predict returns, volatility, trading volume and survey sentiment indices. Expert Syst. Appl. **73**, 125–144 (2017). https://doi.org/10.1016/j.eswa.2016.12.036
23. van den Oord, A., et al.: WaveNet: a generative model for raw audio. arXiv:1609.03499 [cs] (2016). https://arxiv.org/abs/1609.03499

24. Orhan, M., Köksal, B.: A comparison of GARCH models for VaR estimation. Expert Syst. Appl. **39**(3), 3582–3592 (2012). https://doi.org/10.1016/j.eswa.2011.09.048
25. Pandey, A., Wang, D.: TCNN: temporal convolutional neural network for real-time speech enhancement in the time domain. In: 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), ICASSP 2019, pp. 6875–6879 (2019). https://doi.org/10.1109/ICASSP.2019.8683634
26. Patton, A.J.: Volatility forecast comparison using imperfect volatility proxies. J. Econometrics **160**(1), 246–256 (2011). https://doi.org/10.1016/j.jeconom.2010.03.034
27. Poon, S.H., Granger, C.W.J.: Forecasting volatility in financial markets: a review. J. Econ. Lit. **41**(2), 478–539 (2003). https://doi.org/10.1257/002205103765762743
28. Ruder, S.: An overview of gradient descent optimization algorithms. arXiv:1609.04747 [cs] (2016). https://arxiv.org/abs/1609.04747
29. Sezer, O.B., Gudelek, M.U., Ozbayoglu, A.M.: Financial time series forecasting with deep learning?: a systematic literature review: 2005–2019. Appl. Soft Comput. **90**, 106181 (2020). https://doi.org/10.1016/j.asoc.2020.106181
30. Timmermann, A., Granger, C.W.J.: Efficient market hypothesis and forecasting. Int. J. Forecast. **20**(1), 15–27 (2004). https://doi.org/10.1016/S0169-2070(03)00012-8
31. Vaswani, A., et al.: Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS 2017, pp. 6000–6010. Curran Associates Inc., Red Hook (2017). https://papers.nips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf
32. Vidal, A., Kristjanpoller, W.: Gold volatility prediction using a CNN-LSTM approach. Expert Syst. Appl. **157**, 113481 (2020). https://doi.org/10.1016/j.eswa.2020.113481
33. Wan, R., Mei, S., Wang, J., Liu, M., Yang, F.: Multivariate temporal convolutional network: a deep neural networks approach for multivariate time series forecasting. Electronics **8**(8), 876 (2019). https://doi.org/10.3390/electronics8080876
34. Zhang, J., He, T., Sra, S., Jadbabaie, A.: Why gradient clipping accelerates training: a theoretical justification for adaptivity. In: International Conference on Learning Representations, p. 21 (2020). https://openreview.net/forum?id=BJgnXpVYwS
35. Zhang, X., Shi, J., Wang, D., Fang, B.: Exploiting investors social network for stock prediction in China's market. J. Comput. Sci. **28**, 294–303 (2018). https://doi.org/10.1016/j.jocs.2017.10.013