IJRRETAS

INTERNATIONAL JOURNAL FOR RAPID RESEARCH

IN ENGINEERING TECHNOLOGY & APPLIED SCIENCE

Volume:	Issue:	Month of publication:
12	4	April 2026

www.ijrretas.com

+91 77710 84928

info@iijrretas.com

An Intelligent Machine Learning System for Identifying Financial Volatility in Corporate Entities

Dr. Amit Verma

Prof, Sage University, Bhopal

ABSTRACT

Forecasting the volatility of financial assets can be useful because volatility is frequently used in several financial sectors. In this study, we employ long short-term memory (LSTM) and deep neural network (DNN) models to estimate stock index volatility. The majority of related research projects train machine learning models using the distance loss function, but this has two drawbacks. The first is that their models cannot be fairly compared to econometric models since they create mistakes when using estimated volatility as the forecasting aim. We also implement a probability-based loss function to train the deep learning models and test all the models against the likelihood of the test sample in order to address these two issues. The findings demonstrate that our deep learning models with likelihood-based loss functions are more accurate at forecasting volatility than the econometric model and the deep learning models with distance loss functions. Of the two deep learning models with likelihood-based loss functions, the LSTM model performs the best.

Keywords: Machine Learning, Financial Volatility, Corporate Companies, Detection, Deep Learning

INTRODUCTION

Monitoring the volatility of market factors, such as commodity prices, interest rates, and the variables that determine the value of a portfolio, is one of the most crucial duties in finance. As the underlying asset for many derivatives, it is also a significant determinant of the pricing of many financial products. The stylized facts of volatility, the efficient market hypothesis, and the transient nature of financial relationships are only a few of the numerous reasons why it is not an easy process, as is the case with all financial forecasting and prediction (terms used interchangeably in this book). In spite of this, volatility can still be predicted to some extent.

Many various definitions have been put forth because volatility is frequently described as hidden and unobservable, even ex-post. Additionally, it can be difficult to quantify or predict volatility. Despite being simply approximate measures of latent volatility, these still have application since they offer a quantitative means of comparison and frequently coincide with market definitions.

There are numerous techniques that make an effort to model, comprehend, and forecast volatility in addition to the numerous volatility proxies. The generalized autoregressive conditional heteroscedasticity (GARCH) model and its family of variations are among the most frequently used models. In contrast to these conventional models, intelligent methods—which include techniques like machine learning (ML) and deep learning (DL), evolutionary algorithms (EAs), and fuzzy logic—have recently acquired significant attention. These methods are frequently nonlinear. Due to a flurry of successful neural network (NN) applications, machine learning (ML) and deep learning (DL) in particular have soared in popularity in recent years. This trend can also be seen in financial volatility forecasts.

We have the following three objectives in particular: The goals of this project are to: (1) produce a paper that may be used as an introduction to the subject of financial volatility forecasting; (2) offer a snapshot of the state-of-the-art in NN volatility forecasting; and (3) highlight some common problems, potential solutions, and future directions.

LITERATURE REVIEW

Using a deep belief restricted Boltzmann machine and a special nonmating genetic algorithm, Rafiei, M. H., and Adeli, H. (2016) presented a novel and comprehensive model for estimating the cost of new housing in any given city at the design stage or beginning of the construction.

Using large-scale, high-frequency time-series derived from the order book of financial exchanges as input, Tsantekidis, Passalis, Tefas, Kanniainen, J., Gabbouj, & Iosifidis (2017, July) proposed a deep learning methodology based on Convolutional Neural Networks (CNNs) that predicts the price movements of stocks.

Following a brief explanation of the many forms of machine learning, Lee, I., and Shin, Y. J. (2020) provide three different ways that machine learning is used in organizations. The trade-off between machine-learning algorithms' accuracy and interpretability is then covered. This is an important factor to take into account while choosing the best algorithm for the task at hand.

A state-of-the-art overview of the created DL models for financial applications was attempted by Ozbayoglu, A. M., Gudelek, M. U., and Sezer (2020). We not only divided the works into categories based on the desired subfield of finance, but we also examined them using DL models. Additionally, we sought to identify potential future applications and highlighted the direction for the field's current research.

Boluk, S. A., Ozbayoglu, A. M., & Gudelek, M. U. (2017, November). By engaging in paper trading and determining the total capital, we can assess our approach. We contrast the effectiveness of our strategy with popular traditional trading tactics. Our findings show that, when realistic transaction costs are taken into account, we can forecast prices for the following day with an accuracy of 72% and end up with 5:1 of our starting capital.

DL Minh, A Sadeghi-Niaraki, HD Huy, K Min, & H Moon (2018). featured the suggestion of a brand-new Stock2Vec-a sentiment word embedding trained on financial news dataset and Harvard IV-4 and two-stream gated recurrent unit network. There are two main experiments carried out: the first experiment forecasts the direction of the S&P 500 index stock price using historical S&P 500 prices and articles crawled from Reuters and Bloomberg, and the second experiment forecasts the direction of the VN-index price trend using Viet Stock news and stock prices from cophieu68.

The analytical models developed by Almalis, Kouloumpris, and Vlahavas (2022) are improved using a semi-supervised learning technique that depends on the identification and correction of data that may have been incorrectly labelled.

(2017) Hajek, P., and Henriques. In terms of true positive rate (the percentage of fraudulent firms that were correctly identified as such), ensemble approaches performed better than the other methods.

RESEARCH AND METHODLOGY

2.1.DATA

This study examines the S&P 500 Index, Dow Jones Industrial Average Index, and NASDAQ Composite Index, three significant US stock market indices. From the beginning dates up until June 30, 2020, we gather the closing prices for various stock indexes. The S&P 500 sample data span the years January 2, 1928 to June 30, 2020, the Dow Jones sample data span the years May 26, 1896 to June 30, 2020, and the NASDAQ sample data span the years February 5, 1971 to June 30, 2020. Separate trade days from the three sample periods total 23240, 31096, and 12457.

Then, using the following equation, we determine daily returns as the logarithms of relative daily closing prices:

$$r_t = \log\left(P_t/P_{t-1}\right)$$

where pt is the closing price at time t and rt is the daily return at time t.

The three-return series' summary data are displayed in Table 1. Within the survey period, all three indices have positive average returns. Compared to NASDAQ, the S&P 500 and Dow Jones saw greater single-day losses. The three-return series' standard deviations are similar. Positive skewness and strong kurtosis are present in all of the return series. The Jarque-Bera test demonstrates that these return series do not follow a normal distribution.

Table 1: Descriptive statistics of daily returns of stock indices

	S&P 500	Dow Jones	NASDAQ
Mean	0.022%	0.021%	0.037%
Maximum	15.36%	14.27%	13.25%
Minimum	-22.90%	-25.63%	-13.15%
Std.dev.	1.19%	1.16%	1.25%
Skewness	-0.47	-0.86	-0.38
Kurtosis	22.12	27.72	13.57
Jarque-Bera test	0.00	0.00	0.00
Observations	23239	31095	12456

2.2. DEEP NEURAL NETWORK

Undoubtedly one of the most well-known machine learning techniques is the artificial neural network (ANN). It is made to mimic the way that neurons in the human brain are structured. Artificial neural networks that are interconnected can learn to modify the weights of their connections to develop problem-solving abilities.

An ANN with a specific level of complexity is a deep neural network. DNN is a common term for a neural network that has two or more hidden layers. the DNN architecture as a whole. The input layer is used to import inputs into the model. The activation function is used each time to determine the hidden layers and output layer by multiplying the preceding layer by the connection weights.

2.3. LSTM

The deep neural network known as the LSTM was created by Hochreiter and Schmid Huber [25]. The advantages of the recurrent neural network (RNN) have been passed down to the LSTM, which has demonstrated significant power to model sequential data, such as time series. LSTM, in contrast to a standard RNN, employs gates to regulate how information moves through the sequence, enabling it to learn long-term dependencies and resolve the vanishing gradient problem.

the standard LSTM architecture. Like a standard RNN, the model accepts inputs at each time step, and it also allows for the delivery of outputs at each time step. Particularly, LSTM substitutes memory blocks for neurons in RNN. A memory cell, an input gate, a forget gate, and an output gate make up a memory block.

The following vector formulas can be used to describe LSTM:

$$X = \begin{bmatrix} h_{t-1} \\ x_t \end{bmatrix},$$

$$f_t = \sigma(W_f X + B_f),$$

$$i_t = \sigma(W_i X + B_i),$$

$$o_t = \sigma(W_o X + B_o),$$

$$c_t = f_t \nabla c_{t-1} + i_t \nabla \tanh(W_c X + B_c),$$

$$h_t = o_t \nabla \tanh(c_t),$$

where xt stands for the inputs at time t and ht stands for the hidden state at time t. The often used logistic sigmoid is in the activation function. The output gate is described by ot, the input gate by it, and the forget gate by ft. The symbol for multiplication is points.

2.4. Simple Historical Statistics as Benchmark

In addition to ARMA-GARCH and deep learning models, we also provide a straightforward technique to predict volatility. We compute the standard deviation of the return series in a n trading day window and use it as the prediction of volatility for the following trading day, much like Markowitz who utilized historical statistics to be the expected risk. We can use this straightforward but logical approach as a standard by which to compare other approaches.

Just the window length n must be estimated as the parameter. In order to obtain the estimation, we also employ an optimization technique similar to that used in econometric models, so the value of the ideal window length is that which may maximize the likelihood of the sample data.

2.5. Experimental Setup of Our Deep Learning Models

We train our DNN and LSTM models using a likelihood-based loss function, as was mentioned in the introduction. rt If is the return series that we track from time 1 to time T, and t is the volatility that our models predict at the same time step, we may compute the sample likelihood under the premise that the data is normally distributed as follows:

$$L = \prod_{i=1}^{T} \frac{1}{\sqrt{2\pi}\,\sigma_t} \exp\left(-\frac{r_t^2}{2\sigma_t^2}\right)$$

The log-likelihood function is then available:

$$\text{Log L} = \sum_{i=1}^{T} \left(-\frac{1}{2}\log(2\pi) - log\ \sigma_t - \frac{r_t^2}{2\sigma_t^2}\right)$$

While deep learning models are always taught by minimizing their loss function, the log-likelihood function should be optimized through optimization. Therefore, in deep learning models, the loss function must be the negative log likelihood. We employ a simplified function as the loss function of our DNN and LSTM models in order to further lower the computational cost of deep learning models:

$$\text{Loss} = \sum_{i=1}^{T} \left(2\ log\ \sigma_t + \frac{r_t^2}{\sigma_t^2}\right)$$

We will use an equation to generate the test sample's log-likelihood function and compare the results when we test all the models we discuss in this paper.

The forecasting procedure is put into practice every day, whether it is during training or testing for all the approaches. This implies that we predict volatility in a rolling window, which advances one trading day at a time. We can utilize the volatility series from time 1 to time T, which we obtain after forecasting the volatility day by day, to solve equations and carry out various activities.

The following are the other settings for our DNN model. Because the input layer's unit number is set to 10, we always utilize a 10-day return series as our input. There are two secret layers in total. There are 40 units in the first secret layer and 80 units in the second hidden layer. ReLU is selected as the hidden layers' activation function, and the dropout for these two layers is set to 0.3. The output layer's activation function is configured to be sigmoid. The optimizer for training the model is RMSprop. 2048 are designated as the batch size. If the loss function of the validation set does not decrease any longer, there will be an early termination.

2.6. Experimental Setup of Models for Comparison

As we have explained, the distance between estimated realized volatility and predicted volatility is employed as the loss function to train the models in practically all reference articles using machine learning to forecast volatility. We additionally evaluate DNN and LSTM models with similar distance loss functions in order to compare them to our deep learning models with likelihood-based loss functions. By training deep learning models by minimizing the distance loss function and testing them by computing the likelihood value of the test sample, we may investigate whether they can still make accurate forecasts. Deep learning models with distance loss functions are at a disadvantage when we compare them to other models in this research since their testing and training methodologies are inconsistent.

MSE is selected to be the distance loss function for DNN and LSTM. Among the linked research efforts, this loss function has been the most popularly chosen. The realized volatility is estimated in a manner similar to these studies by taking the standard deviation of the return series over a window of 21 trading days, which is roughly equivalent to the average number of trading days in a month. When forecasting volatility, the activation function of the output layer is specified to be the linear function, which is also a popular option for deep learning models with distance loss function. The likelihood-based loss function used in our deep learning models is used in all other situations as well.

We set m, n, p, and q to be 1 to 3 while training the ARMA (m, n)-GARCH (p, q) model, and we estimate all possible combinations. The model with the lowest BIC is chosen, and it will be applied to the test set to make predictions. Simple way only requires selecting the ideal train set window length, as we covered in the previous part.

CONCLUSION

For the purpose of predicting the volatility of three US market indexes, we combine deep learning models, econometric models, and a straightforward statistical approach. We further offer a likelihood-based loss function to train the deep learning models and test all the approaches by the likelihood of the test sample, which is different from comparable research papers. When employing deep learning models, we can reduce errors in the volatility forecasting process by doing this. We can also compare the models we look at in a more equitable way.

The empirical study's findings demonstrate that our deep learning models with likelihood-based loss functions predict volatility more accurately than the econometric model, and LSTM (likelihood-based loss) is the superior deep learning model of the two. The volatility series predicted by the six models exhibit remarkably comparable long-term tendencies. Under the harsh conditions of the US stock market, around March 2020, LSTM (likelihood-based loss) seems to capture the property better.

Forecasting financial volatility will always be a crucial task. It is developing and getting more popular and accessible than ever because to the growth of AI research. Financial markets will, however, also continue to develop, revealing new relationships to exploit. These relationships will eventually burn out, making room for new ones that are hidden from view. This ongoing change will result in new possibilities and study directions.

FUTURE WORK

Model interpretation will be a future study direction. We did not examine the methodology used by our ML models to make their price forecasts or the most crucial impacting variables; we simply forecasted prices. Permutation importance and other feature importance techniques are frequently used to find pertinent predictor variables. However, the majority of feature importance techniques give highly inaccurate findings, particularly when there are significant dependencies between the predictor variables. Because the relationships between our real estate variables cannot be completely eliminated (for instance, size and number of rooms have a strong correlation), specific techniques like conditional permutations are required in real estate pricing.

REFERENCES

1. Alam, Md. Iftekhar Hossian Md Tasnim, and Jyotirmoy Ghose. “Image Forgery Detection Using Copy-Move Technique.” International Journal of Research Publication and Reviews, vol. 4, no. 3, Genesis Global Publication, Mar. 2023, pp. 1103–07. Crossref, https://doi.org/10.55248/gengpi.2023.32077.
2. Almalis, I., Kouloumpris, E., & Vlahavas, I. (2022). Sector-level sentiment analysis with deep learning. *Knowledge-Based Systems*, *258*, 109954.
3. Babu, S. B. G. Tilak, and Ch Srinivasa Rao. “Efficient Detection of Copy-move Forgery Using Polar Complex Exponential Transform and Gradient Direction Pattern.” Multimedia Tools and Applications, vol. 82, no. 7, Springer Science and Business Media LLC, Feb. 2022, pp. 10061–75. Crossref, https://doi.org/10.1007/s11042-022-12311-6.
4. F. Black and M. Scholes, “The pricing of options and corporate liabilities,” *Journal of Political Economy*, vol. 81, no. 3, pp. 637–654, 1973.
5. F. Corsi, “A simple approximate long-memory model of realized volatility,” *Journal of Financial Econometrics*, vol. 7, no. 2, pp. 174–196, 2009.
6. Gudelek, M. U., Boluk, S. A., & Ozbayoglu, A. M. (2017, November). A deep learning-based stock trading model with 2-D CNN trend detection. In *2017 IEEE symposium series on computational intelligence (SSCI)* (pp. 1-8). IEEE.
7. H. Markowitz, “Portfolio selection,” *The Journal of Finance*, vol. 7, no. 1, pp. 77–91, 1952.
8. Hajek, P., & Henriques, R. (2017). Mining corporate annual reports for intelligent detection of financial statement fraud–A comparative study of machine learning methods. *Knowledge-Based Systems*, *128*, 139-152.
9. Lee, I., & Shin, Y. J. (2020). Machine learning for enterprises: Applications, algorithm selection, and challenges. *Business Horizons*, *63*(2), 157-170.
10. Minh, D. L., Sadeghi-Niaraki, A., Huy, H. D., Min, K., & Moon, H. (2018). Deep learning approach for short-term stock trends prediction based on two-stream gated recurrent unit network. *Ieee Access*, *6*, 55392-55404.
11. Ozbayoglu, A. M., Gudelek, M. U., & Sezer, O. B. (2020). Deep learning for financial applications: A survey. *Applied Soft Computing*, *93*, 106384.
12. R. F. Engle, “Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation,” *Econometrica*, vol. 50, no. 4, pp. 987–1007, 1982.
13. Rafiei, M. H., & Adeli, H. (2016). A novel machine learning model for estimation of sale prices of real estate units. *Journal of Construction Engineering and Management*, *142*(2), 04015066.
14. Suresh, Gulivindala, and Chanamallu Srinivasa Rao. “Copy Move Forgery Detection Using GLCM Based Statistical Features.” International Journal on Cybernetics & Informatics, vol. 5, no. 4, Academy

and Industry Research Collaboration Center (AIRCC), Aug. 2016, pp. 165–71. Crossref, https://doi.org/10.5121/ijci.2016.5419.

15. Tsantekidis, A., Passalis, N., Tefas, A., Kanniainen, J., Gabbouj, M., & Iosifidis, A. (2017, July). Forecasting stock prices from the limit order book using convolutional neural networks. In *2017 IEEE 19th conference on business informatics (CBI)* (Vol. 1, pp. 7-12). IEEE.