Portfolio Volatility Forecasting with Recurrent Neural Networks

Conly Sharp

Abstract

Forecasting the volatility of an asset remains an essential area of research in finance, and with the advent of deep learning, new methods are being introduced with strong forecasting performance. In this study, Recurrent Neural Networks are utilized to forecast the volatility of portfolios across different sectors of the world market. Portfolios are constructed across the healthcare, technology, and metals sectors which are then optimized with respect to maximizing the Sharpe Ratio. Multiple recurrent neural network architectures are trained over thirty year's worth of historical price data for the Microsoft Corporation stock. Each asset in the portfolio has a years worth of stock returns forecasted by multiple recurrent neural network models which are then used to compute portfolio volatility. We find best results with an ensemble model utilizing models comprising of Gated Recurrent Units, Long Short-Term Memory, and standard Recurrent Networks in predicting portfolio volatility, with this ensembled model performing the best across the technology and healthcare sectors. Additionally, a model utilizing solely Gated Recurrent Units found the best results for predicting portfolio volatility in the metals sector. Among the best performing models, we find 1.01%, 2.73%, and 2.81% percent error compared to the true historical portfolio volatility in the technology, metals, and healthcare sectors respectively.

I. Introduction

Volatility of an asset is an important metric when it comes to information regarding financial markets. An asset's volatility suggests the inherent riskiness of the security, and thus having an accurate forecast of future volatility values allows for an investor to diversify their investments to better protect and maximize their returns. Mathematically, we can consider an asset's volatility as the standard deviation of an asset's returns multiplied by the square root of a particular trading period. Likewise, the volatility of a portfolio is a similarly regarded metric which tells one the inherent risk associated with the assets that make up the portfolio. As diversifying investments and therefore mitigating risk is a top priority for any investor managing their assets, having a handle on future volatility is paramount for successful investing. It has been shown that portfolios consisting of low volatility stocks yield higher returns than their riskier counterparts, leading portfolios managers seeking low portfolio volatility [1]. Therefore, an accurate prediction for portfolio volatility is a valuable measure which can be used to maximize gains over markets. Furthermore, volatility is found to be the largest factor when it comes to low volatility stocks outperforming high volatility stocks [2], giving an advantage to any investor who is able to utilize future information about the volatility of an asset or portfolio.

In the field of deep learning, recurrent neural networks (RNNs) have been utilized to take advantage of time series data for forecasting purposes with strong results [3]. The recurrent connections inherent in the structure of RNNs allows for better predictions over sequences of data as each sequence can be stored in the memory of the network [4].

In this report, we utilize multiple different RNN architectures to predict the opening prices of multiple stocks listed in world markets across different sectors. With each opening price forecasted for a years worth of trading data on each asset, we compute asset volatilities. Each asset volatility is then used to compute the final portfolio volatility subject to the optimized weights dictated by a mean variance analysis optimization program. We train and tune multiple separate RNN architectures over the historical price data of the Microsoft Corporation stock over the time period of March 2nd, 1985 to October 15th, 2016, using an average of the stock's historical price data over the open, close, high, and low values. Once every model was tuned efficiently, we predicted the future prices of each asset over the entire 2016 trading period, and used these values to compute the predicted portfolio volatility for each respective portfolio.

We chose three separate and distinct sectors to choose our stocks from: technology (software infrastructure), metals (gold, aluminum, steel producers), and healthcare (drug manufacturers). The technology portfolio contained five stocks while the latter two contained four each. Optimal asset allocation weights were constructed for each portfolio using the mean variance method subject to maximizing the Sharpe ratio. The portfolios were optimized over the the same period we tuned our network models over (see Methods). The technology portfolio consisted of Microsoft Corporation (MSFT), Adobe Incorporated (ADBE), Palo Alto Networks Incorporate (PANW), Synopsys Incorporated (SNPS), and Fortinet Incorporated (FTNT). The metals portfolio consisted of the Franco-Nevada Corporation (FNV), Kaiser Aluminum Corporation (KALU), Royal Gold Incorporated (RGLD), and Steel Dynamics Incorporated (STLD). The healthcare portfolio consisted of Johnson and Johnson (JNJ), Eli Lilly and Company (LLY), Merck and Company Incorporated (MRK), and AbbVie Incorporated (ABBV). The assets from each sector were chosen arbitrarily from among the stocks with the biggest market capitalization in their respective sector.

II. Literature Review

There is ample research in forecasting prices, volatility, and portfolio metrics in both deep learning and quantitative finance fields. In regards to deep learning research, Multilayer Perceptron networks were found to be the most widely used architectures for forecasting volatility, with RNN architectures coming in second followed by Convolutional Neural Networks (CNNs) [5]. The same study finds a large discrepancy between the advanced models found in deep learning research and those used in volatility forecasting, with a concluding remark musing that financial forecasting could become a distinct sub-field of artificial intelligence on its own, analogous to the evolution of natural language processing and computer vision.

A quick scan among quantitative finance research finds a plethora of novel developments being used to forecast financial volatility, such as using CNNs over Gramian Angular Field Images [6], LSTM models combined with a novel likelihood based loss function [7], or combining LSTMs and CNNs to forecast volatility using previous volatility values and investor sentiment data [8].

The applications of deep learning to financial forecasting has amplified tremendously in recent years, and with quantitative based trading and investing taking more of the market share each year, this field is only going to become more important across all financial domains whether it be banks, hedge funds, or private equity firms.

III. Methods

For RNN architecture, we primarily utilized three separate RNNs into our models' architectures, that is, SimpleRNN, LSTM (long short-term memory), and GRU (Gated Recurrent Units). Three architectures were trained. The first architecture is comprised of solely LSTM layers, the second utilizes GRU layers, and the third utilizes a mix of SimpleRNN and LSTM layers. Upon tuning each architecture, an ensemble model was used to attempt to better predict future prices. GRU and LSTM were created to combat the vanishing gradient problem inherent in RNN structures [10].

Concerning the portfolios themselves, with each portfolio created from their respective sector, we then computed the optimal allocation weights for every asset in each portfolio. These weights are needed in the portfolio volatility formula. To find these weights, we used Mean Variance analysis adjusted for the highest returns on the Sharpe Ratio. Mean variance was first introduced by the economist Harry Markowitz in 1952 with his seminal paper on Modern Portfolio Theory, while the Sharpe Ratio, the difference between the returns of an investment and its risk free return, was introduced by the economist William Sharpe in 1966. Essentially, this optimization process allocates weights to each stock in the portfolio based on the highest rate of return relative to each asset's volatility over the given period. In this case, we used the same period that we trained our models over, from 1985 to 2016. For more on Mean Variance optimization and the Sharpe Ratio, see [9] and [10]. Figures 1, 2, and 3 display each sector's optimized portfolios. All portfolios were optimized through the online "Portvolio Visualizer" tool created by Silicon Cloud Technologies.

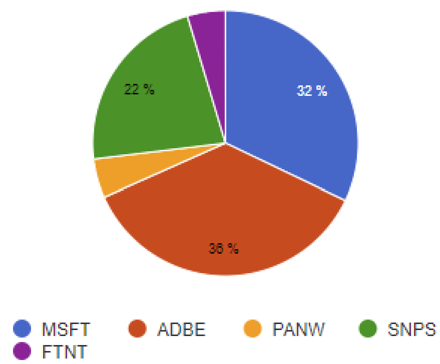

FIGURE 1. Optimized technology sector portfolio. Not listed: FTNT 4.62%, PANW 4.74%

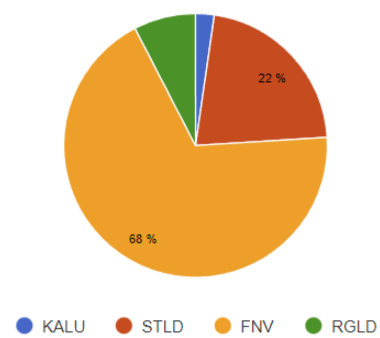

FIGURE 2. Optimized metals sector portfolio. Not listed: RGLD 7.59%, KALU 2.29%

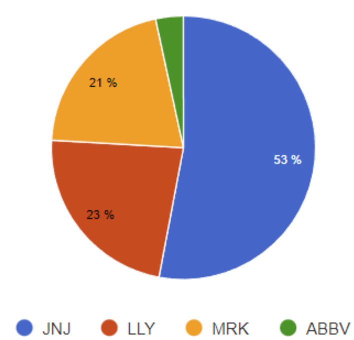

FIGURE 3. Optimized healthcare portfolio. Not listed: ABBV 3.38%

The dataset used for training comprises of the past thirty-one years of price data for the Microsoft Corporation. This stock was chosen for its high amount of volume and liquidity during the entirety of its history listed on the NASDAQ exchange. The dataset was split into training and testing by an 80 to 20 percent ratio. We constructed a custom column of data which averaged the Open, High, Low, and Close price of the stock for each trading day. Using this "Average" column, sequences containing thirty days of the average prices were constructed, with each successive sequence iterating one day forward. The model then learns to predict the thirty-first day for each sequence. For the testing dataset, we removed the first thirty days after the end of the training set to ensure there is no overlap between training and testing. Once every model was tuned, we began to predict the open price values for each day in assets we have selected. For the predictions on each asset, we predict over the 2016 trading year excluding January to account for the testing dataset losing the first thirty days. With each asset's predicted future open prices, we compute the asset's volatility using the predicted returns over the year. With each asset's volatility computed, we then construct the overall portfolio volatility for each sector using the assumption that stock correlations are null (for simplicity) and the following equations:

$$\textbf{Portfolio Volatility}= \sqrt{\sum_{i=1}^{n} \omega_i^2 \tau_i^2 + \sum_{i=1}^{n} \sum_{i \neq j}^{n} \tau_i \sigma_{ij} \omega_j} \tag{0.1}$$

Where τ is the variance of the returns of each respective asset, ω is the optimized weight of the respective asset in the portfolio, and σ_{ij} is the covariance between i and j.

$$\textbf{Portfolio Volatility with Independent Assets} = \sqrt{\sum_{i=1}^{n} \omega_i^2 \tau_i^2} \tag{0.2}$$

With each portfolio volatility constructed, we can then compare each predicted volatility to the true historical portfolio volatility for each sector with the same independent asset assumptions.

IV. Results

With every model tuned, we predicted the portfolio volatility for each sector's portfolio using three distinct architectures. These included the Ensemble model which combined an LSTM, GRU, and LSTM-Simple RNN model, the LSTM model, and the GRU model. See Figure 4 for further information on each model's architecture. We found the predicted portfolio volatility from each model to be quite close for each sector, with the worst prediction being 7.47% off from the actual historical volatility by percent error. Among the models, the ensemble Model and GRU model performed the best, with the LSTM predicting slightly less accurately across all sectors. Concerning each sector, the Ensemble model predicted the Tech sector portfolio and the Healthcare sector portfolio with the highest accuracy, while the GRU Model predicted the Metals sector portfolio volatility most accurately. While the LSTM Model did not predict as close to the historical portfolio volatility compared to the other architectures, it was not egregiously erroneous by any means. See Figure 5 for a complete listing of each model's predicted portfolio volatility across each sector compared to the true historical volatility. Figure 6 displays each model's percent error across each sector.

Model	Description	Optimizer	Loss
LSTM	LSTM-1000, Bidirectional:LSTM - 624, SimpleRNN - 624, LSTM-624, Dense:relu - 32, Dense:sigmoid -1	Adam (learning_rate=0.001)	Mean Squared Error
GRU	GRU - 1000, GRU:relu - 624, GRU-624, Dense:relu - 32, Dense: sigmoid-1	Adam (learning_rate=0.0001)	Mean Squared Error
Ensemble: LSTM	LSTM - 1000, LSTM:relu - 624, LSTM-624, Dense:relu - 32, Dense: sigmoid-1	Adam (learning_rate=0.001)	Mean Squared Error
Ensemble	The ensemble model is a combination of the results of the three models described above.		

Figure 4. Final model architectures used for predicting portfolio volatility.

Sectors	Historical Portfolio Volatility	Predicted Portfolio Volatility		
		Ensemble	LSTM	GRU
Tech	6.30%	6.23%	6.67%	5.97%
Metals	11.61%	12.10%	12.45%	11.93%
Healthcare	5.90%	6.06%	6.34%	5.66%

FIGURE 5. Results for each model's predicted portfolio volatility.

Sectors	Ensemble	LSTM	GRU
Tech	1.01%	5.85%	5.21%
Metals	4.17%	7.21%	2.73%
Healthcare	2.81%	7.47%	4.02%

FIGURE 6. Percentage error between predicted portfolio volatility and historical volatility values.

V. Discussion and Conclusion

As we have seen, the predictive power of RNN, GRU, and LSTM networks on time-series data is strong. With each architecture showing close and accurate results with respect to each portfolio volatility, we can safely conclude the reliable efficacy of these models when it comes to forecasting time-series data. Of note is that the GRU model outperformed the LSTM model in each sector. We hypothesize the less complex nature of the GRU structure might lead to better training on simple and short numerical sequences that we have in our data, in addition to being less computationally expensive as a bonus. We see the ensemble model performs as intended, giving us the most robust predictions, only outperformed by the GRU model in one sector.

In a financial context, forecasting volatility for both assets and larger portfolios is tantamount when it comes to work in trading, asset management, and risk management. While these architectures are relatively simplistic and have been trained and predicted over non-complex data, the general accuracy of each predicted portfolio volatility across three vastly different sectors sets a positive tone towards recurrent neural networks utilization in quantitative finance or any domain which uses time-series data.

There are many avenues for which one can improve upon this work. For one, combining recent advances in Natural Language Processing in Deep Learning could be utilized to improve upon predictions of the change in asset prices. For example, gathering information on investor sentiment through news articles, social media, and consumer reports combined with time-series data could potentially lead to higher prediction power. Another improvement is that of higher complexity data, such as including technical indicators, smaller price frames (minutes or seconds), and utilizing novel advancements in deep learning. Additionally, it is possible semi-supervised transformers could be used on our sequential price data. Innovations in computer vision might also lead to stronger predictive performance.

Neural networks are here to stay when it comes to all realms of finance. As the entire industry is dominated by sequential time series data, recurrent neural networks are an obvious fit for moving forward in cutting edge research and forecasting in any quantitative finance topic.

References

[1] Baker, Nardin L. and Haugen, Robert A. (2012): *"Low Risk Stocks Outperform within All Observable Markets of the World"*, Available at SSRN: https://ssrn.com/abstract=2055431 or http://dx.doi.org/10.2139/ssrn.2055431

[2] Dudek, G., Smyl, S., and Pełka, P. (2022): *"Recurrent Neural Networks for Forecasting Time Series with Multiple Seasonality: A Comparative Study"*, Electrical Engineering Faculty, Czestochowa University of Technology, Poland. https://arxiv.org/abs/2203.09170

[3] Blitz, D., van Vliet, P., and Baltussen, G. (2019): *"The Volatility Effect Revisited"*, Available at SSRN: https://ssrn.com/abstract=3442749 or http://dx.doi.org/10.2139/ssrn.3442749

[4] Gjylapa, D. and Proko, E., (2017): *"Recurrent Neural Networks in Time Series Prediction"*, Conference: New Research and Advances on Computer Science and Information Technology NRACSIT-2017 At: Vlora, Albania.

[5] Ge, W., Lalbakhsh, P., Isai, L., Lenskiy, A., and Suominen, H., (2022): *"Neural Network–Based Financial Volatility Forecasting: A Systematic Review"*, ACM Computing Surveys, Volume 55, Issue 1, pp 1–30.

[6] Chen, W., Yao, K., and Shao, Y. (2022): *"Volatility forecasting using deep neural network with time-series feature embedding"*, Economic Research-Ekonomska Istraživanja, DOI: 10.1080/1331677X.2022.2089192

[7] Jia, F., and Yang, B. (2021): *"Forecasting Volatility of Stock Index: Deep Learning Model with Likelihood-Based Loss Function"*, *Complexity*, vol. 2021, Article ID 5511802, 13 pages, 2021. https://doi.org/10.1155/2021/5511802

[8] Ncume, V., L van Zyl, T., and Paskaramoorthy, A. (2022): *"Volatility forecasting using Deep Learning and sentiment analysis"*, SACAIR 2022 conference, https://doi.org/10.48550/arXiv.2210.12464

[9] Markowitz, H.M. (1952): *"Portfolio Selection"*, The Journal of Finance. 7 (1): 77–91.

[10] Sharpe, W.F. (1966): *"Mutual Fund Performance"*, Journal of Business. 39 (S1): 119–138.

[11] Hu, Y., Huber, A., Anumula, J., and Liu, S., (2018): *"Overcoming the vanishing gradient problem in plain recurrent networks"*, https://arxiv.org/abs/1801.06105

Email address: `conlysh@proton.me`