Neural Network in Asset Allocation

I INTRODUCTION

Portfolio management is a well-known multi-factor optimization problem facing investment advisors. The resource allocation problem is a process of allocating a set of resources among a set of entities or activities. It is a complex problem encountered in a variety of areas in operations economics and operation researches, such as portfolio selection, production planning, and computer scheduling. The intelligent computational techniques such as artificial neural networks (ANNs) would be more suitable to improve the resource allocation problem. Artificial Neural Networks (ANN) are excellent approximators of non-linear functions, and so the use of these ANN models in computational finance research has persisted. It has been known for decades that multilayer feedforward neural networks are capable of approximating any measurable function to any desired degree of accuracy. The difference between ANN and other approximation methods is that it uses one or more hidden layers to transform the input variables, using a transfer function to deal with nonlinear statistical functions. ANNs can analyze huge quantities of data to recognize patterns and make sense of incomplete or noisy data, and therefore provide an excellent alternative to linear models for forecasting and estimating financial time-series.

One of the commonly used classes of ANNs is the Recurrent Neural Network (RNN). While feed-forward networks are designed to have no feedback loops, RNNs contain at least one directed cycle to create an internal memory. Long Short- Term Memory (LSTM) networks are a form of RNN designed to deal with modeling temporal sequences. LSTMs are explicitly designed to avoid the long-term dependency problem. Remembering information for long periods of time is practically their default behavior. All recurrent neural networks have the form of a chain of repeating modules of neural network. However, in standard RNNs, this repeating module will have a very simple structure, such as a single tanh layer.

This work applies Neural Networks (NN) into asset allocation problem. Our asset involves a diversified portfolio of 11 stocks from S&P500. Vanilla ANN and LSTM Recurrent Neural Network (RNN) are both conducted and compared for better performance. Annualized Return, volatility and sharpe ratio are examined to evaluate our models.

The following sections present related work (Section II), our asset allocation model (Section III), followed by a description of the dataset (Section IV), a description of the empirical experimentation and analysis of results, closing with conclusions and future improvement.

II RELATED WORK

Daniel Shapiro and Samer Obeidat[1] presented a Long Short-Term Memory approach to adaptive asset allocation, building upon prior work on training neural networks to model causality. The neural network model discussed in their work ingests historical price data and ingests macroeconomic data and market indicators using Principal Components Analysis. The model then estimates the expected return, volatility, and correlation for the selected assets. These neural network outputs were then turned into action recommendations using a Mean- Variance Optimization framework augmented to use a forward- looking rolling window technique. Testing was performed on a dataset with a 7.66 year duration. The observed mean annualized return for classical passive portfolio management approaches were 4.67%, 3.49%, and 4.57%, with mean Sharpe ratios of 0.46, 0.20, and 0.54. 10 simulations using the new Long Short- Term Memory model from this work provided a mean

annualized return of 10.07%, with a Sharpe ratio of 0.98. This work provides the conclusion that a Long Short-Term Memory model can generate better risk-adjusted returns than conventional strategic passive portfolio management.

Po-Chang Ko and Ping-Chen Lin[2] illustrate that portfolio selection is a resource allocation problem in a finance market. The investor's asset optimization requires the distribution of a set of capital (resources) among a set of entities (assets) with the trade-off between risk and return. The ANN with nonlinear capability is proven to solve a large-scale complex problem effectively. It is suitable to solve NP-hard resource allocation problem. However, the traditional ANN model cannot guarantee the summation of produced investment weight always preserves 100% in output layer. Their model introduces a resource allocation neural network model to optimize investment weight of portfolio. This model will dynamically adjust the investment weight as a basis of 100% of summing all of asset weights in the portfolio. The experimental results demonstrate the feasibility of optimal investment weights and superiority of ROI of buy-and-hold trading strategy compared with benchmark Taiwan Stock Exchange (TSE).

III OVERALL MODEL

Figure 1 presents the building blocks of our asset allocation model. The output from our model is a set of asset weights and the raw data of our model includes the close price of 11 stocks from S&P500. The input is prepared by calculation the variance and covariance of close prices within a certain time interval while the target output is acquired by Hierarchical Risk Parity (HRP) [3] model with future returns of a certain time interval.

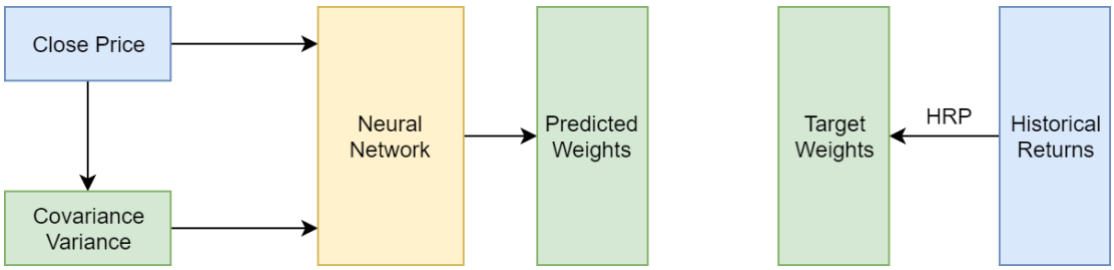

Figure 1. Block Digram for the overall asset allocation model

Once the input and target output data has been prepared, they are passed to Vanilla ANN or LSTM RNN to generate a set of optimal weighting of the assets.

IV DATA DESCRIPTION AND ANALYSIS

A. Stock Selection

11 stocks are selected to construct our portfolio from S&P500. To ensure diversity, they are from different sectors of S&P500 industry. The information of our target stocks is shown in

Table 1. All the stock selected must be traded in NASDAQ since December 31th, 2008. We take the available stock weighted the most in each sector. Data are collected from Bloomberg and there is no missing value from December 31th, 2008 and November 26th, 2019.

Table 1. Information of our target stocks

Company name	Ticker	Weight	Sector
Apple Inc.	AAPL	4.315146	Information Technology
Amazon.com Inc.	AMZN	2.794432	Consumer Discretionary
JPMorgan Chase & Co.	JPM	1.610674	Financials
Johnson & Johnson	JNJ	1.396042	Health Care
Procter & Gamble Company	PG	1.167607	Consumer Staples
Exxon Mobil Corporation	XOM	1.142851	Energy
AT&T Inc.	T	1.065165	Communication Services
Boeing Company	BA	0.759452	Industrials
NextEra Energy Inc.	NEE	0.446289	Utilities
Linde plc	LIN	0.432876	Materials
American Tower Corporation	AMT	0.367077	Real Estate

B. Input Data

Input data are comprised of three parts: stock price, the variance of prices of each stock within a certain time interval and the covariance between each pair of stocks within a certain time. Therefore, this time interval would lead to different input data so is taken as a hyperparameter in our model. It is decided by grid search in rolling backtest and is set to 60 days according to our result. Therefore, the input data is finally in shape of 2686 by 77, which means 2686 samples and 77 features.

C. Target Output Data

Target output data are a set of asset weights. This is calculated by HRP model in the hope of best performance. However, the effect of HRP seems suspicious after we examine the performance after experiment. The returns within another certain time in the future interval are utilized for the calculation of HRP. This time interval means our rebalancing frequency, namely our allocation will stay unchanged in this time interval until next rebalancing date so all the returns during this period are required to be taken into consideration. This is also a hyperparameter of our model and is set to 10 days according to the result of grid search and rolling backtest.

V EXPERIMENT

All the computations were based on a fixed 756-day (3-year) rolling window. Our whole dataset was divided into seven sliding windows as shown in Figure 2.

A. Data Split and Normalization

Our dataset is split to training data, validation data and test data in each rolling window as shown in Figure 2. The ratio is set to 0.7, 0.2 and 0.1.

Our input data is stock price as well as its variance and covariance, which calls for normalization to avoid the influence of each stock's absolute value. As the length of rolling time windows is set to 756 days, we consider there is no need to normalize it by cutting to several time period. Note that validation data and test data are normalized according to the statistic values of the training dataset because these are unseen data.

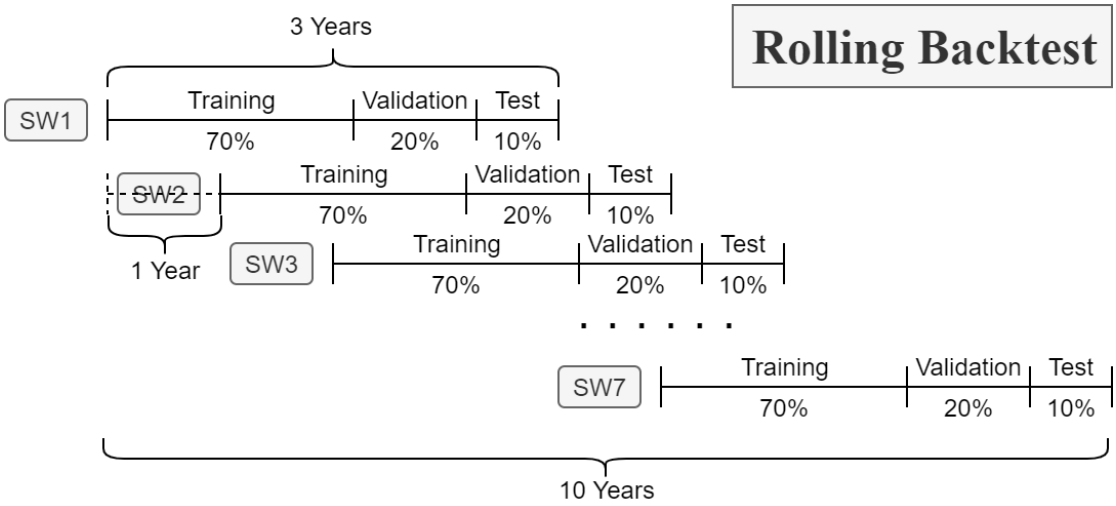

Figure 2. Sliding windows backtest process

B. Tune Hyperparameters

In each rolling window, hyperparameters are tuned by grid search on the validation data. As illustrated previously, the time interval for input calculation (TW1) and the time interval for output calculation (TW2) are two hyperparameters in our models. These two are common while some of other hyperparameters are different in Vanilla ANN and LSTM RNN. The time step in LSTM is an important hyperparameter and it is also carefully examined in our experiment.

Figure 3. Time flow schematic diagram

C. Final Model

After hyperparameters are tuned for each rolling window, we set our final model according to the comprehensive performance of each rolling window. The final model is retested with adding validation part to training data to make the best use of available information for each time window. This aims at evaluating the robustness of our model.

VI RESULT ANALYSIS

A. Benchmark

Classical passive portfolio management approach Equally Weighted Portfolio Rebalancing (EQWT) and a brand-new approach Hierarchical Risk Parity (HRP) based on the Markowitz's

Modern Portfolio Theory are evaluated for this work in comparison with LSTM-based approache.

For EQWT, the allocation in each of the 11 assets is set to 9.09%. For HRP, the only input is the historical data of the various stocks. The development of HRP motivated by some issues regarding widely used strategies, such as Markowitz's dependency on quadratic optimization of forecasted returns, frequently providing unstable and highly concentrated solutions; Traditional risk parity's ignorance of useful covariance information. In order to address these issues, HRP methodology was proposed to drop forecasted returns and rely completely on covariance data, and to cluster assets based on correlation in order to allocate less weight to similar assets.

The first step of HRP is tree clustering, which groups similar investments into clusters based on their correlation matrix. Having a hierarchical structure helps to improve stability issues of quadratic optimizers when inverting the covariance matrix. As the second part for quasi-diagonalization, it reorganizes the covariance matrix so similar investments will be placed together. This matrix diagonalization allows to distribute weights optimally following an inverse-variance allocation. Finally, recursive bisection distributes the allocation through recursive bisection based on cluster covariance.

B. Metrics

This paper measures model performance by three metrics including annualized return, volatility and Sharpe ratio. The annualized return is the geometric average amount of money earned by an investment each year (252 days) over a given time period. Volatility is the degree of variation of a trading price series over time as measured by the standard deviation of returns. The Sharpe ratio was developed by William F. Sharpe and is used to help investors understand the return of an investment compared to its risk. The ratio is the average return earned in excess of the risk-free rate per unit of volatility or total risk. Subtracting the risk-free rate from the mean return allows an investor to better isolate the profits associated with risk-taking activities. Generally, the greater the value of the Sharpe ratio, the more attractive the risk-adjusted return.

C. Result

a) Grid Search

A model hyperparameter is a characteristic of a model that is external to the model and whose value cannot be estimated from data. The value of the hyperparameter has to be set before the learning process begins. The hyperparameter in ANN model mainly includes the time interval for input calculation (TW1) and the time interval for output calculation (TW2). To get optimal hyperparameters, we use Grid Search method. Taking the loss and robustness of model into consideration, we list the (TW1, TW2) hyperparameter pairs of top three results in each sliding window, which is shown in Table 2. We can find that ANN model performs well when TW1=60 and TW2=10. Using the same method in LSTM model, we get the optimal value of time step in LSTM model which is equal to 60.

Table 2. Grid Search Results of ANN model

Rank	SW1	SW2	SW3	SW4	SW5	SW6	SW7
1	(10, 5)	(60, 60)	(5, 20)	(10, 10)	(60, 10)	(60, 60)	(5, 20)

2	(5, 5)	(20, 60)	(60, 10)	(10, 20)	(60, 60)	(5, 5)	(60, 10)
3	(20, 5)	(10, 60)	(20, 10)	(5, 10)	(10, 60)	(20, 20)	(60, 20)

b) Comparisons of ANN and LSTM

Each sliding window is done five times and their mean values are taken to obtain more general results. Let R_ANN and R_LSTM refer to the obtained annualized return by using ANN model or LSTM model. From Table 3, it is shown that the average of R_LSTM (31.51%) is greater than average of R_ANN(19.31%). Moreover, R_LSTM is superior to R_ANN in each sliding window. Even if the R_ANN in SW2 is negative, LSTM model also performs better than ANN. In conclusion, this result demonstrates that LSTM model obtain better asset allocations than ANN effectively.

Table 3. Comparisons of return of ANN and LSTM model

Sliding Windows	R_ANN(%)	R_LSTM(%)
SW1	39.60	41.74
SW2	-6.24	26.64
SW3	43.04	45.02
SW4	5.37	20.64
SW5	20.02	51.51
SW6	6.12	7.12
SW7	27.25	27.91
Average	19.31	31.51

c) Comparisons between model and benchmark

Figure 4 illustrates the annualized return (%) trend of LSTM and benchmarks for seven years. LSTM model ranks first in three sliding windows (SW4, SW5, SW7) and ranks second in three sliding windows as well. In general, it obtains efficient asset allocations from the return perspective. Meanwhile, the performance of HRP model is not robust enough considering that it ranks last in several sliding windows. The equally weight approach (Mean) has the same problem as HRP and it is much more volatile.

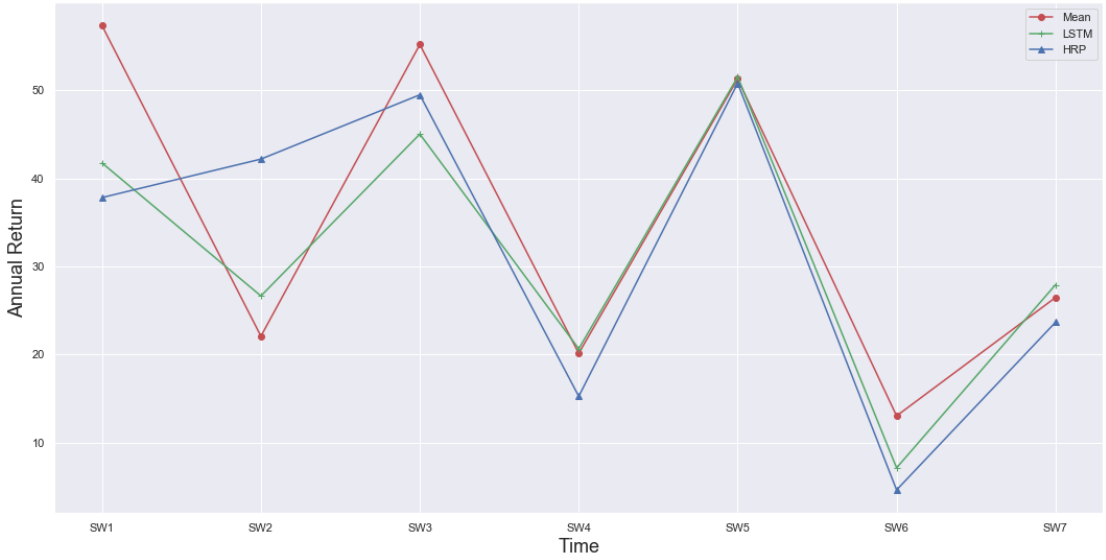

Figure 4. Percentage return comparison between LSTM and benchmarks

Figure 5 demonstrates the volatility change of LSTM and benchmarks during seven sliding windows. In general, the volatility of HRP is the least except last two sliding windows. This result is consistent with its principle and algorithm. To be specific, the goal of HRP is to minimize the risk for a given stock price series which is similar with the mean-variance optimizer. The volatility of LSTM model is the second largest in five sliding windows except SW3 and SW4. The average volatility of HRP (0.1014) and LSTM is close. As for the equally weight approach (Mean), its average volatility in all sliding windows is the largest (0.1088).

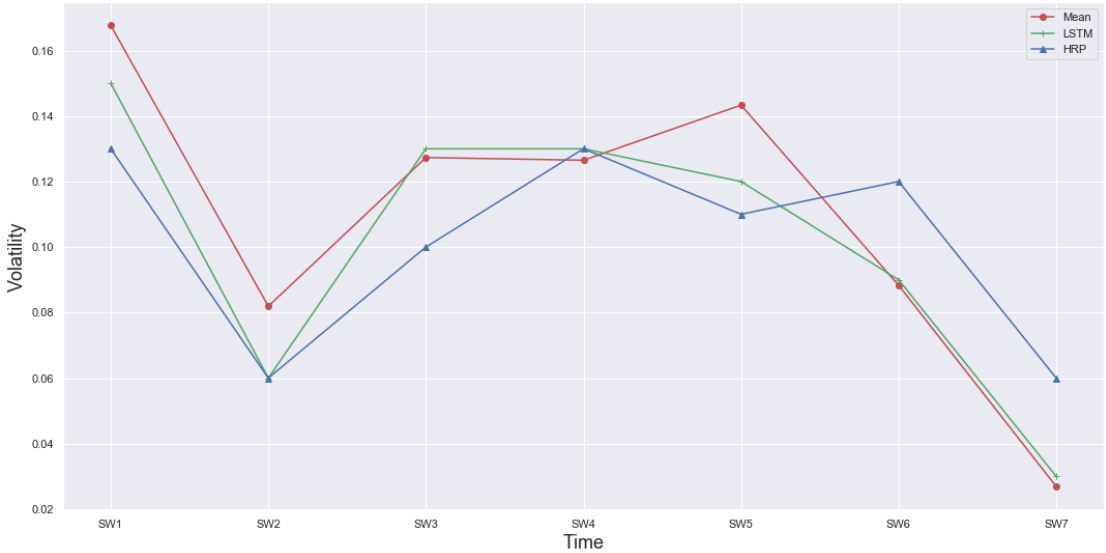

Figure 5. Volatility comparison between LSTM and benchmarks

Figure 6 compares Sharpe ratio of LSTM and benchmarks in each sliding window. The Sharpe ratio of LSTM model ranks first in two sliding windows (SW4, SW7) and ranks second in two sliding windows. The average Sharpe ratio in all sliding windows of Mean,

LSTM and HRP is 3.56, 3.67 and 3.33 respectively, which demonstrates that LSTM is well balanced in return and volatility.

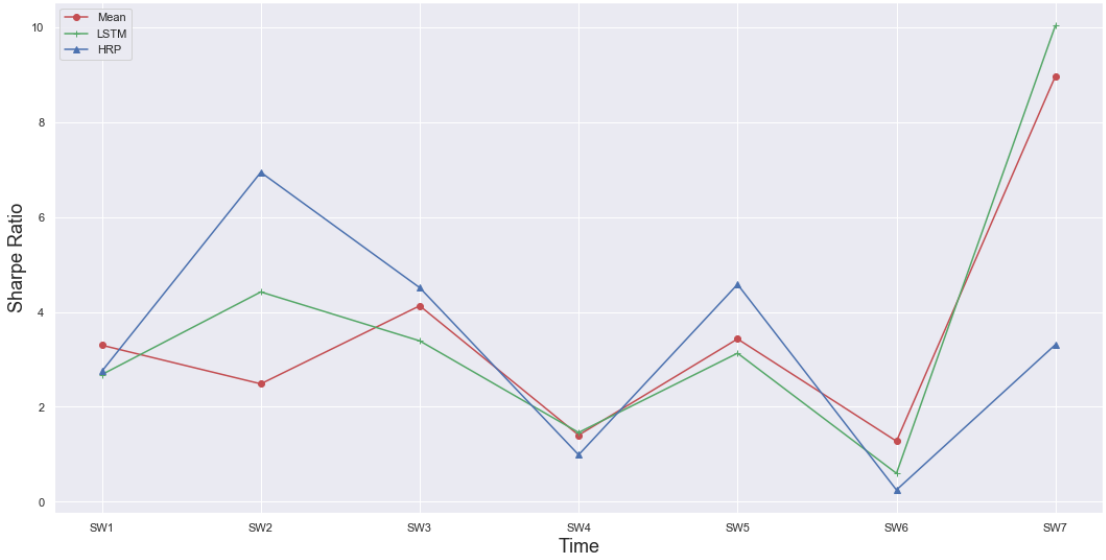

Figure 6. Sharpe ratio comparison between LSTM and benchmarks

D. Analysis

a) In the first place, LSTM RNN model outperforms vanilla ANN model apparently. Because LSTM RNN model remembers values over arbitrary intervals while vanilla ANN not. LSTM is well-suited to predict time series given time lags of a duration.

b) Though LSTM does not preform best in return and volatility comparing to the benchmarks, there is a balance of return and volatility in LSTM because its Sharpe ratio is better than benchmarks.

c) From the return perspective, the equally weight approach (Mean) obtains an excellent result but its volatility is also quite large. This may reflect stocks we choose are leading stocks in the market. As a result, just using equally weight approach also can get a good result. Meanwhile, it is affected by the market risk evidently. So, if we adjust the weights of them, we can decrease the market risk in our model in some extent.

d) The indictors' changes between sliding windows are sharp especially for the return. This may root in the fact of market changes and the time interval setting of rolling windows. For example, as time interval between two rolling windows increase, the time interval between two test sets also increases. Therefore, market may change a lot during that time.

VII CONCLUSION

A. Summary

In a word, we focus on applying Neural Network into asset allocation on stocks from S&P 500. We set Stock Price, Covariance and Variance as input and HRP weights as target output. To test the robustness of our model, we use conduct rolling back test. We compare the adaptability of vanilla ANN and LSTM RNN model in asset allocation problem and find that LSTM RNN model is an effective method to solve it. To tune our model, we use grid search to get optimal hyperparameters. Eventually, we compare our final LSTM with two benchmarks. Annualized return and Sharpe ratio

of our model is 31.51% and 3.67. Therefore, our model is well balanced in return and volatility.

B. Further improvement

Firstly, we can apply other algorithms or models as target weight output considering the drawback of HRP which cannot always achieve satisfied return rate. Secondly, the indicator's change between sliding windows is volatile. We should shorten the length of backtesting time step to further test the robustness. Last but not least, we can acquire more stock or data to conduct more complicated model relating to this problem in the future.

REFERENCE

[1] Obeidat, S., Shapiro, D., Lemay, M., MacPherson, M. K., & Bolic, M. (2018). Adaptive Portfolio Asset Allocation Optimization with Deep Learning. International Journal on Advances in Intelligent Systems, 11(1), 25-34.

[2] Ko, P. C., & Lin, P. C. (2008). Resource allocation neural network in portfolio selection. Expert Systems with Applications, 35(1-2), 330-337.

[3] de Prado, M. L. (2016). Building diversified portfolios that outperform out of sample. The Journal of Portfolio Management, 42(4), 59-69.