Hybrid Transformer Model For Portfolio Allocation

Sai Anand Chandrasekaran[1], *Mehul Sharma*[2], *Sai Phyo Hein*[3]
[1,2,3] *WorldQuant University*
[1]c.saianand@gmail.com, [2]mehul.sharma2798@gmail.com,
[3]phyohein.1196@gmail.com

Abstract

Portfolio optimization is an art of selecting an optimal and a balanced set of assets from a variety of available assets in order to achieve the best risk adjusted returns. Traditional portfolio optimization methods such as mean-variance optimization have long been the cornerstone of portfolio optimization strategy. However, limitations such as not being able to capture complex relationships and handling large datasets hinders its ability to maximize risk-adjusted returns. Hence there exists a growing interest in leveraging deep learning based algorithms to enhance investment decision-making processes. Transformer based approach is one such technique which has proved promising when there is a need to explore complex patterns and relationships. Nevertheless the efficacy of these advanced techniques in optimizing portfolios, particularly when evaluated against the established Sharpe Ratio metric, remains an area of exploration. This research aims to address this gap by proposing a transformer-based hybrid model for portfolio optimization and comparing its performance with traditional mean-variance optimization, as well as with vanilla transformer based model. We hope to demonstrate that the hybrid transformer model will outperform the latter methods in terms of performance.

Keywords: *Hybrid Transformer, Transformer Models, Deep Learning, Machine Learning, Portfolio Management, Asset Allocation, Sharpe Ratio, Diversification, Risk Management*

1. Introduction

Portfolio management is the allocation of budgets in different financial assets to meet the investors' financial expectations based on the risk aversion. One may purchase a financial asset with their life savings or wealth that is not spending currently to generate return from it. However, there is a trade-off between the expected return and the risk. Higher rates of return entail greater risk; and, the greater the risk, the greater the uncertainty regarding future profits [1]. The Markowitz Model also known as Modern Portfolio Theory (MPT) [2] introduced by Harry Markowitz in 1952 had been a backbone in the field of portfolio management using quantitative finance. The objective function of this model is optimized by maximizing the return while minimizing the risk of the investment.

But the mean-variance optimization method includes an unrealistic statistical assumption [4]; while most of the returns of financial assets tend to have fat tail distributions, the mean-variance method requires the returns to follow a Gaussian distribution. [5] Other drawbacks of the Markowitz model are static and linear; the linear correlation matrix of the assets is applied to calculate the budget allocations.

When the mean, volatility, and the correlation patterns in the asset returns change drastically time to time[6], machine learning and deep learning strategies are applied for dynamic portfolio optimization which involves sequential decision making of continuously reallocating funds into assets with complex interrelationships in consecutive balancing periods based on real-time financial information to achieve desired performance [5].

Section 2 explains the innovations and related works in the field of Portfolio Management where machine learning and deep learning are highly focused. Section 3 presents the brief theoretical introductions to each component applied for the transformer architecture including the hybriding methods. Section 4 discusses the detailed methodology and architecture of the hybrid transformer model for constructing portfolios of equities in the Indian market.

2. Literature Review and Related Works

Machine learning seeks to automatically learn meaningful relationships and patterns from examples and observations [7]. There are a large number of machine learning techniques to identify the patterns in data which are used again to predict the future observations. Example techniques are linear and non linear regressions, decision trees for classification and regression, support vector machines (SVM) and nearest neighbors methods for classification and clustering, principal component analysis (PCA), hidden markov model (HMM) and partial least square methods to extract latent factors from the datasets [8]. Yijian Chuan, Chaoyi Zhao, Zhenrui He and Lan Wu (2021) successfully applied an adaptive boosting version of decision tree for portfolio management and the model outperforms the buy and hold strategy by 45% annualized return. Thomas Conlon, John Cotter, and Iason Kynigakis [9] used machine learning models like principal component analysis (PCA), partial least square regression (PLA) and autoencoder models to extract latent factors from assets' time series data for construction portfolios and compared the performance to a simple estimator of equal weight strategy. Their work shows that a portfolio based on autoencoders would result in a standard deviation and Sharpe ratio which improve by up to 8% and 17%, respectively, compared to the portfolio formed using the sample estimator.

Artificial neural network is one of the advancement in machine learning methods in search of sophisticated algorithms and efficient pre-processing techniques. Deep neural networks typically consist of more than one hidden layer, organized in deeply nested network architectures: they usually contain advanced neurons in contrast to simple ANNs and use advanced operations (e.g., convolutions) or multiple activations in one neuron rather than which allow deep neural networks to be fed with raw input data and automatically discover representative patterns. This is commonly known as deep learning [7]. Examples of deep learning neural networks are Feed Forward Net, Recurrent Neural Network (RNN), Long-short term memory (LSTM) and the most recent advancement - Transformer models. Zihao Zhang, Stefan Zohren, Stephen Roberts [4] optimized a portfolio's sharpe ratio using LSTM models and compared to Mean-Variance (MV), Maximum Diversification (MD) and Diversity Weighted Portfolio (DWP). The sharpe ratios using the LSTM model range from 1.4 to 1.9 while other methods' are ranging from negative 0.1 to 1.4, which indicates that the deep learning model can outperform the MV method. Yilin Ma, Ruizhu Han, and Weizhong Wang [10] applied Deep Multilayer Perceptron (DMLP), LSTM and CNN to create prediction-based portfolio models and the authors found that DMLP is a better model than the others in stock return prediction. Time series data implies a sequential dependency and patterns over time that need to be detected to form forecasts [9]. Thus, models which include sequential information as a feature like LSTM and RNN show better performance for financial time series forecasting.

RNN, LSTM and GRU had been the state of art approaches in sequence modeling and transduction problems. Throughout the computation steps, they generate a sequence of hidden states h(t), as a function of the previous hidden state h(t-1) and the input for position t. This process creates the problem of exceeding the computational limit that becomes critical as the lengths of the sequence become large. Thus, attention mechanisms which can handle sequential data parallely have become an integral part of compelling sequence modeling [11]. As LSTM and transformer models have been popular in large language modeling, those models are also used to predict stock prices and construct portfolios by analyzing the market sentiments from twitter and news. Jintao Liu and et al. (2022) [12] had proposed this approach of language processing using a transformer-based model in comparison with other deep learning models like CNN to predict the stock prices of 47 stocks listed in S&P 500. They showed that the performance of the transformer-based model was the highest with 64% accuracy in predicting stock price trends. In the field of directly applying sequential data for portfolio management, Tae Wan Kim, and Matloob Khushi (2020) used a modified transformer model which can successfully develop a strategy to create a 46% of cumulative return from a portfolio with assets of nine Dow Jones companies while its competing models are returning in negatives up to -50% for modern portfolio management. The authors had engineered a complex architecture where the actor, critic and target units are all replaced by special transformer structures.

Thus, transformer models are performing better than other sequential modeling methods in terms of both prediction performance and computation complexity. As portfolio management requires precise and timely information, a huge effort is being invested by researchers in developing transformer models to close the gap of rapid sequential decision making and prolong model training of RNN and LSTM.

Portfolio Transformer by Damian Kisiel and Denise Gorse (2022) [13] is one of the innovative usage of transformer models in portfolio strategy. Their Portfolio Transformer shows the best performance when compared to other methods like LSTM, MLP, XGBoost and Mean-Variance method for all portfolios of equities and ETFs.

3. Theoretical Background

The financial data usually exhibits the following stylized facts:

- Serial dependent are present in the data - autoregressive
- The volatility changes over time - heteroskedasticity
- Distribution is non-Gaussian.

The generalized autoregressive conditional heteroskedasticity model (GARCH) is usually used to study the serial data with heteroskedasticity [15]. Details of the GARCH model is mentioned in section 3.1. As long short-term memory (LSTM) models are the current state of art in predicting sequential data, we are hybriding the transformer model especially with LSTM and theoretical explanations are presented in section 3.2 followed by the discussion of the basic transformer architecture which will be our main focus. Mathematical formulations of Mean-Variance method of portfolio strategy which serves as the baseline for our proposing model and the benchmark one is also described in section 3.4.

3.1. GARCH

We plan to include the GARCH in our architecture to learn the volatility movement of the price time series data.

Generalized Autoregressive Conditional Heteroscedasticity is a type of statistical model where conditional expectation of process is modeled as an ARMA (autoregressive moving average) process with errors that are not white noise (i.e. they do not have constant variance). The architecture consists of a mean process and a volatility (variance) process.

An ARMA (mean) process can be expressed using below equation:

$$y_t = c + \sum_{i=1}^{p} \phi_i y_{t-i} + \sum_{j=1}^{q} \Theta_j \epsilon_{t-q} + \epsilon_t \tag{3.1}$$

where:

- yt is the observed time series data at time t.
- c is a constant term.
- ϕ_i and Θ_i are parameters representing the autoregressive (AR) and moving average (MA) components, respectively.
- ε_t is the error term at time t, assumed to be normally distributed with mean zero and constant variance.

The variance process is also assumed to follow an ARMA(p,q) process as suggested by Bollerslev [16] .

$$u_t = \epsilon_t \sqrt{h_t} \tag{3.2}$$

$$h_t = \omega + \sum_{i=1}^{p} \alpha_i u_{t-i}^2 + \sum_{j=1}^{q} \beta_j h_{t-j} \tag{3.3}$$

Where:

- h_t is the conditional variance (volatility) at time t.
- ω is the intercept term representing the long-term average variance.
- α is the coefficient of the lagged squared error term, representing the short-term impact of past shocks on volatility.
- β is the coefficient of the lagged conditional variance term, representing the persistence of volatility.
- is the squared residual (error) term at time t–i.

The GARCH (p,q) model essentially states that the current volatility depends on three components:

1. The long-term average variance (ω).
2. The short-term impact of past shocks on volatility ($\alpha_i \varepsilon^2_{t-i}$).
3. The persistence of volatility ($\beta_j h_{t-j}$).

By estimating the parameters (ω, α_i, β_j) of the volatility equation using historical data, future volatility can be forecasted. The model assumes that the errors (ε_t) are independently and identically distributed with zero mean and constant variance. The parameters are typically estimated using maximum likelihood estimation or other optimization techniques.

3.2. Long Short-Term Memory (LSTM)

Despite mentioning that LSTM are slower than the transformers, we aim to incorporate the ability of LSTM in learning the sequential dependency so that the model can learn from the auto-regression to predict the next data points.

LSTM introduced in [18] is a kind of recurrent neural network (RNN) architecture which has the ability to remember and adapt to learning information over a longer period of time. LSTM is mainly useful in modeling scenarios such as speech, time series forecasting etcIt addresses the main drawback of recurrent neural networks which performs poorly due to fewer parameters and prone to vaninish and exploding gradient problems.

LSTM Architecture consists of various cell states and each cell capable of a particular action. The sequential data is fed through these cells with a combination of linear and non-linear interactions involved. Further each cell consists of a gate system namely: the input gate, the forget gate and the final output gate.

Forget Gate: This gate is primarily used to moderate the amount of information to be preserved or discarded at each level. The output from the previous hidden units and the current state is considered and provides an output in the range of 0 to 1.This indicates how much of the respective information has to be remembered.

Input Gate: It determines what new information to be stored in the cell state.

Output Gate: It controls which parts of the cell state are used to compute the output.

Hidden State: This state is present at every time step and is influenced by all other states. The important purpose of this state is to track what the model has seen so far, further used to make predictions on future time steps.

The ability of LSTM to capture long-term dependencies and handle sequential data made it a preferred choice in modeling financial time series data.

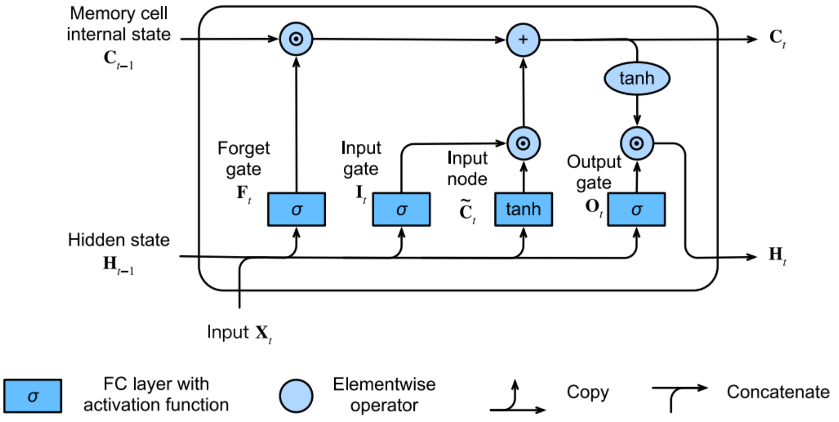

Fig 1: LSTM architecture [19]

3.3. Transformer Model

Transformer model is the main architecture of our proposed model. The detailed structure of the implemented model is presented in Section 4.2. The transformer model is one of the powerful deep learning techniques that is popularly used for the natural language processing domain. The capability of extracting the sequential relations of sequence data parallelly is the main focus we would like to apply in predicting the weight allocations on asset returns. It was first proposed by Ashish Vaswani [11] in the year 2017.. The basic building blocks of the Transformer model is elucidated below:

Encoder block: The encoder is responsible for processing the input sequence of data and extracting features out of it. It basically encodes the given input in some representative format. A encoder block usually consists of a stack of similar layers namely self attention mechanism followed by feedforward neural network (FFN).

Self-attention: This section helps in weighing importance to different sections of the sequence of data. Otherwise called as tokens. The main advantage of self-attention is helping the encoder in capturing long ranging dependencies.

Multi-head attention: In order to capture the diverse relationships across the sequence of data, multi-head attention is of prime importance. It helps in processing the data parallelly. Thereby attending to the different parts of the input sequence simultaneously.

Decoder: This block is responsible for getting the output sequence based on the representations that are encoded by the encoder. The design of the decoder block is so similar to the encoder block with a stacked set of self attention mechanisms and FFNs.

4. Methodology

Section 4.1 presents the dataset chosen, asset selection and exploratory data analysis on those selected assets. Section 4.2 explains the architecture of the proposing model - Hybrid Transformer model and benchmarking one - Vanilla Transformer model.

4.1. Data and Asset Selection

Daily prices of the stocks are downloaded from Yahoo Finance. Assets for the portfolio are mainly chosen from the ones listed on NSE (National Stock Exchange of India). NSE provides 3 asset classes for trading: equities, fixed income securities and derivatives. Our scope of work is specific to equities.

Several equities are selected from NIFTY Index - NIFTY 50 so there is no liquidity risk incorporated in the dataset. Nifty 50 is the benchmark stock market index of NSE, comprising the top largest and most liquid 50 stocks in India. The list of the NIFTY 50 Index constituents is rescheduled every 6 months. The assets in this paper are the listing of NIFTY 50 on 16th April 2024.

Price data of all NIFTY 50 assets from 2014 January to 2024 March are sampled. Stocks are then selected with the following criteria:

- All equities must have price data greater than zero for the sampled time window so that portfolios derived from them are balanced throughout the training, validating and testing process.
- Enhancing the diversification, assets with mutual linear correlation of their daily returns greater than 0.5 are deselected for portfolio construction.
- To save the computational demand of the work, only the top 3 of the highest market capitalization in each industry are filtered out.

Filtering with the above criteria results in a total of 17 assets as follows:

Table 1: Selected Stocks for Portfolio and their Information

No.	Ticker (Yahoo Finance)	Company	Industry	Market Capitalization (₹ - Billions)
1.	MARUTI.NS	Maruti Suzuki India Ltd.	Automobile and Auto Components	3965
2.	TATAMOTORS.NS	Tata Motors Ltd.	Automobile and Auto Components	3497
3.	M&M.NS	Mahindra & Mahindra Ltd.	Automobile and Auto Components	3000
4.	ASIANPAINT.NS	Asian Paints Ltd.	Consumer Durables	2702
5.	TITAN.NS	Titan Company Ltd.	Consumer Durables	2983

6.	HINDUNILVR.NS	Hindustan Unilever Ltd.	Fast Moving Consumer Goods	5476
7.	ITC.NS	ITC Ltd.	Fast Moving Consumer Goods	5452
8.	NESTLEIND.NS	Nestle India Ltd.	Fast Moving Consumer Goods	2410
9.	DIVISLAB.NS	Divi’s Laboratories Ltd.	Healthcare	1043
10.	CIPLA.NS	Cipla Ltd.	Healthcare	1133
11.	SUNPHARMA.NS	Sun Pharmaceutical Industries Ltd.	Healthcare	3681
12.	ONGC.NS	Oil & Natural Gas Corporation Ltd.	Oil Gas & Consumable Fuels	3509
13.	RELIANCE.NS	Reliance Industries Ltd.	Oil Gas & Consumable Fuels	19415
14.	COLINDIA.NS	Coal India Ltd.	Oil Gas & Consumable Fuels	2893
15.	NTPC.NS	NTPC Ltd.	Power	3549
16.	POWERGRID.NS	Power Grid Corporation of India Ltd.	Power	2941
17.	BHARTIARTL.NS	Bharti Airtel Ltd.	Telecommunication	8030

The resulting set of assets covers a range of industries - automobiles, consumer products, energies, healthcare and telecommunication. The market capitalization ranges from a minimum of 1043 Billion Rupees - DIVISLAB.NS to a maximum of 19415 Billion Rupees - RELIANCE.NS. Except for the RELIANCE.NS as the outlier for market capitalization and price trend, other assets are within similar bounds.

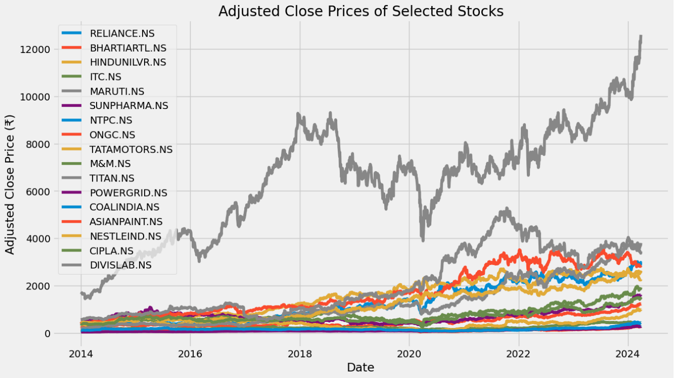

Fig 2: Time Series of Historical Prices of Selected Stocks (own source code)

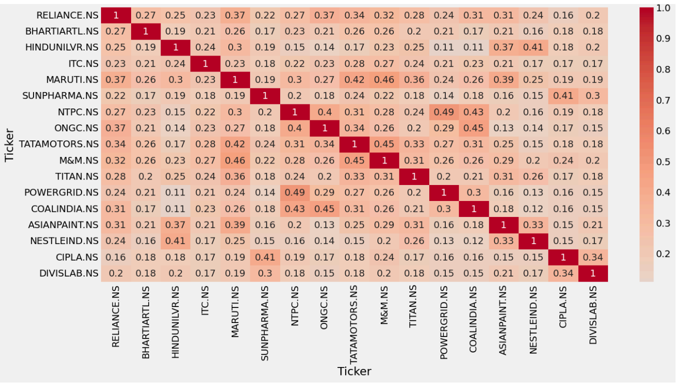

Fig 3: Correlation of selected stocks (own source code)

According to the span of the expected returns from the assets, the portfolios are expected to have expected returns between 0.0511% to 0.1314%. The risk (standard deviation) of are assumed to be ranging from 1.4% to 2.6%.

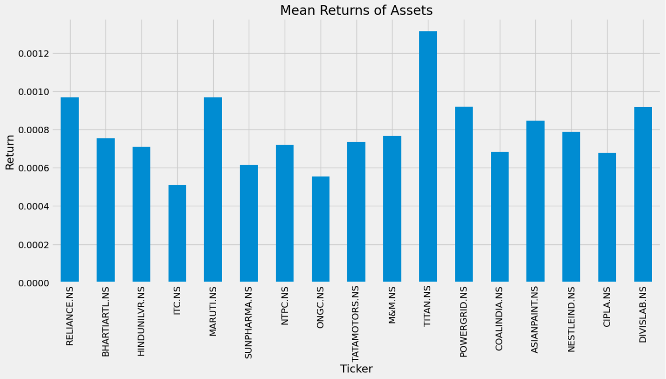

Fig 4: Expected return of selected stocks (own source code)

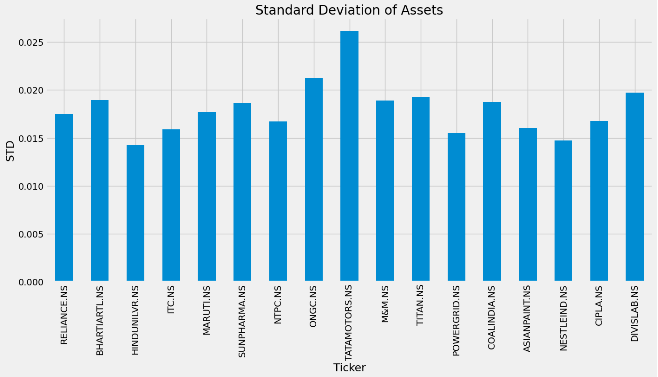

Fig 5: Standard deviation of return of selected stocks (own source code)

4.2. Model Architecture and Benchmarks

The hybrid architecture of the transformer model by hybriding with GARCH and LSTM models is largely derived from the work of Eduardo Ramos-Pérez,Pablo J. Alonso-González, and José Javier Núñez-Velázquez (2021) [14].

The main difference of this architecture from conventional/vanilla transformer models is the replacement of positional encoding with autoregressive models like GARCH or LSTM. One of the benchmarking models to justify the performance of our model is a vanilla transformer model without hybriding. Mean-variance method is applied as the baseline for both hybrid transformer model and vanilla transformer model.

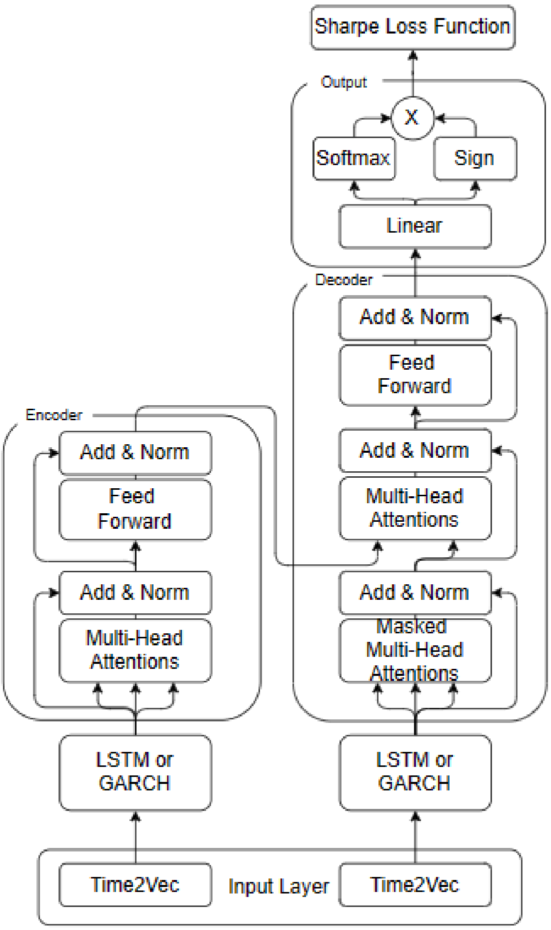

Fig 6: Architecture of Hybrid Transformer Model for Portfolio Allocation [14][17]

The two transformer architectures applied in this paper are similar except for the positional encoding step: LSTM or GARCH replaces the positional encoding layer in the hybrid architecture. The architectures also include encoder, decoder, output layer and loss function.

4.2.1. Positional Encoding for Time Series Financial Data

Positional encoder in vanilla transformer models embeds the information of the relative position of the observations in the sequential data. The following encoders are suggested by Vaswani et al. [11] as:

$$PE_{(pos,\,2_{i})} = sin\,(\,pos\,/\,1000^{2i/dim}) \qquad (4.1)$$

$$PE_{(pos,\,2_{i+1})} = cos\,(\,pos\,/\,1000^{2i/dim}) \qquad (4.2)$$

Positional encoder by Vaswani et al. [11] is intended for the usage in Natural Language Processing tasks. But the financial data does not use words as input, positional encoders should be modified and Eduardo Ramos-Pérez et al. modified the positional encoder by avoiding variation of inputs depending on the number of occurrences. Their suggested positional encoder changes depending on the lag, but it remains the same across the different explanatory variables. Equation for modified positional encoder is given as the following and the benchmarking vanilla model adopts this positional encoding method:

$$PE_{pos} = cos\,(\pi\frac{pos}{N_{pos}-2}) = sin\,(\frac{\pi}{2} + \pi\frac{pos}{N_{pos}-1}) \quad (4.3)$$

4.2.2. Encoder

For both hybrid and vanilla transformer models, there are encoder layers, consisting of a multi-head attention mechanism where the outputs from the positional encoding or LSTM or GARCH are input. The number of heads for the multi-head attention mechanism will be updated via the hyperparameter optimization process. The multi-head attention layers are followed by layer normalization and then the feed forward layers learn the correlation between the inputs which are then fed to another layer normalization. The residual connection provides a direct path for the gradient (and ensures that vectors are updated by the attention layers instead of replaced), while the normalization maintains a reasonable scale for the outputs [17].

4.2.3. Decoder

The decoders in the hybrid and vanilla transformer models are also composed of two blocks of attention mechanism. First attention mechanism is used to mask for ensuring that the predictions are only dependent on preceding observations and the second mechanism incorporates the output information from the encoder stack. The decoders also include the feed forward layers and there are layer normalizations at each attention mechanism and the feed forward.

4.2.4. Output Layer

Damian Kisiel and Denise Gorse (2022) [13] have proposed a modified output layer which allows short-selling for their Portfolio Transformer. The outputs from the decoder is firstly processed by a fully connected layer and the resulting vector is used to calculate the weights for the portfolio allocation with the following equation:

$$\omega_{i,t} = sign(s_{i,t}) * softmax(s_{i,t}) \quad (4.4)$$

$$\omega_{i,t} = sign(s_{i,t}) * \frac{e^{s_{i,t}}}{\sum_{j=1}^{N} e^{s_{j,t}}} \quad (4.5)$$

where, $\omega_{i,t}$ = weight of asset i at time t, $s_{i,t}$ = return of asset i at time t and N = number of assets.

4.2.5. Loss Function

Damian Kisiel and Denise Gorse (2022) [13] have also proposed the objective function of the transformer model for portfolio optimization. The objective function optimizes the asset allocation for maximizing the sharpe ratio. The sharpe ratio for portfolio is defined as the expected portfolio return divided by the portfolio's volatility:

$$SR = \frac{E(R_p)}{\sqrt{E(R^2_p)-(E(R_p))^2}} \quad (4.6)$$

4.3. Model Training, Validation and Testing

Following the usual practice of machine learning and deep learning model, the dataset is splitted into three sets - train, valid, and test.

Train dataset is used for the model training step in which the model learns the complex relations among or the self sequential relation of the input data. Validation set is applied to justify whether the model can generalize and predict unseen datasets. Different combinations of hyperparameters are used to test the model's capability to predict the validation set and the hyperparameters giving the best prediction performance is chosen by a method called - Grid Search Validation.

Time series data consists of a risk of leakage in model training if we split the train, valid and test sets sequentially due to autocorrelation [20]. The leakage is alleviated by the purging method - the start 30 data points of the testing set and 10 data points of the validation set are eliminated before performing validation or testing.

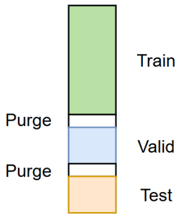

Fig 7: Data Purging

4.4. Modern Portfolio Theory

Nurfadhlina Abdul Halima and Ari Yuliatib (2020) [3] presented a comprehensive review on the Markowitz Theory. Suppose there are *N* risk assets with returns $r_1, \ldots, r_N$. The return expectation value vector is given by:

$$\Pi^T = (\mu_1, ..., \mu_N), \text{with } \mu_i = E[r_i],\ i = 1,..., N \tag{4.7}$$

and the covariance matrix is given by:

$$\Sigma = (\sigma_{ij})_{i,j=1,...,N}, \text{with } \sigma_{ij} = Cov(r_i, r_j),\ i, j = 1, ..., N \tag{4.8}$$

and budget allocation is given by:

$$w^T = (w_1, ..., w_N) \tag{4.9}$$

Thus, the expected return and variance of the portfolio constructed from those N assets is

$$\Pi_p = E[r_p] = \Pi^T w \tag{4.10}$$

$$\Omega^2_p = Var(r_p) = w^T \Sigma w \tag{4.11}$$

The Mean-Variance optimization defines an efficient portfolio as: A p^* portfolio is called (Mean-Variance) efficient if there is no p portfolio with $\Pi_p > \Pi_{p^*}$ and

$\Omega^2_p < \Omega^2_{p^*}$.

Thus, to get an efficient portfolio, the following objective function is maximize:

$$\textbf{Maximize } 2\tau\Pi_p - \Omega_p^2 \tag{4.12}$$

where, τ is the risk tolerance parameter of the investor and subjected to:

$$\sum_{i=1}^{N} w_i = 1 \tag{4.13}$$

The objective function can be written in a vector form as:

$$\textbf{Maximize } 2\tau\Pi^T w - w^T\Sigma w, \text{ subjected to } e^T w = 1 \tag{4.14}$$

An efficient portfolio that matches τ=0 is called the minimum variance portfolio.

4.5. Portfolio Construction

Using the weight allocations resulting from the models, portfolio holdings are adjusted with daily frequency.

Since having a high turnover rate will significantly drawback the portfolio performance, the cost adjusted return mechanism of Damian Kisiel and Denise Gorse (2022) [13] is also implemented for the hybrid and the vanilla transformer models in this paper.

$$R_{P,t} = \sum_{i}^{N} \omega_{i,\,t-1} * r_{i,t} - C * \sum_{i}^{N} |\omega_{i,\,t-1} - \omega_{i,\,t-2}| \tag{4.15}$$

C represents the constant cost rate, $\omega_{i,t-1}$, $\omega_{i,t-2}$ represents the weight of asset i on day t-1 and day t-2. As the portfolio is allowed for short-selling but no leverage, the weights are constrained by [-1, 1], and $\sum_{i}^{N} \omega_{i,t} = 1$.

4.6. Metrics

Performance of the portfolios resulting from each model are compared using the metrics like sharpe ratio, sortino ratio, maximum drawdown, cumulative returns, rolling returns and rolling risks.

4.6.1. Sharpe Ratio

It is a ratio of the excess return on an investment over the risk-free rate of return to the associated risk of investment. The excess return is calculated by taking a difference of realized or expected portfolio return and the risk-free rate of return and the risk of investment is calculated by the volatility of returns in a given period. The Sharpe ratio can be calculated using the below equation:

$$Sharpe\ Ratio = \frac{Rp-Rf}{\sigma_p} \tag{4.16}$$

The measure provides the risk adjusted relative returns of the portfolio with respect to the benchmark rate of return and can be used to determine the risk adjusted performance of the portfolio. The measure is based on the assumption that past risk adjusted excess returns can provide a view of the future risk adjusted returns of the portfolio. We neglect the risk free rate for evaluating the performance.

4.6.2. Sortino Ratio

It is a variation of the sharpe ratio where instead of the standard deviation of the portfolio returns, we take the standard deviation of downside returns. It is the ratio of the excess returns of a portfolio over risk-free rate of return to the standard deviation of the downside returns. It is calculated as follows:

$$Sortino\ Ratio = \frac{Rp-Rf}{\sigma_d} \tag{4.17}$$

It helps investors to evaluate portfolio returns for a given level of bad risk.

4.6.3. Maximum Drawdown

It is calculated as the maximum loss between the peak to a trough of a portfolio, before the new peak. It indicates the downside risk over a specified time period. It can be calculated as follows:

$$Maximum\ Drawdown = \frac{Trough\ Value-Peak\ Value}{Peak\ Value} \tag{4.18}$$

5. Results and Discussion

The comparison of the metrics for the portfolios constructed are as follows:

Table 2: Comparison metrics of the Portfolios from Different Allocation Methods

Allocation Method	**Mean Return ↑**	**Mean Standard Deviation ↓**	**Overall Sharpe Ratio ↑**	**Overall Sortino Ratio ↑**	**Overall Maximum Drawdown ↓**
Hybrid Transformer	**0.001805**	0.017366	0.103963	0.176388	0.198771
Vanilla Transformer	0.000531	0.008071	0.065788	0.095822	0.122163
Mean Variance Process	0.000903	0.007421	0.121687	0.180753	0.101954
Equal Weight Allocation	0.001129	**0.007410**	**0.152311**	**0.229462**	**0.074730**

The hybrid model achieves the highest return by taking the overall risk which in turn reduces the overall sharpe ratio, overall sortino ratio. The higher risks result in a higher maximum drawdown.

The vanilla model is incapable of learning current market data, showing the least performance in returns, sharpe ratio, and the second worst of maximum drawdown without having a higher movement.

The MVP and equal-weight methods exhibit the maximum sharpe ratios by lowering the risks which are incorporated with lower return.

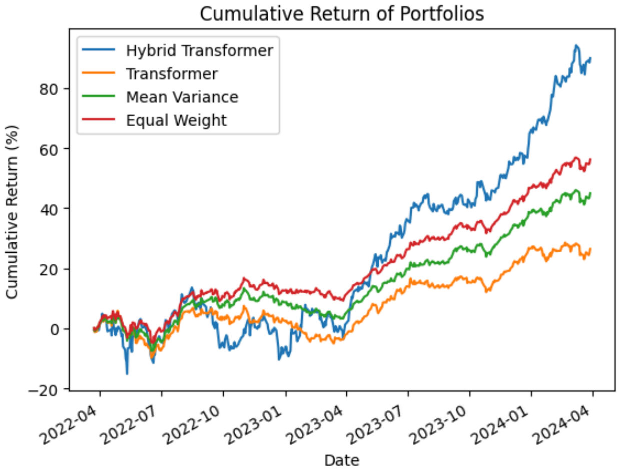

Fig 8: Cumulative Returns of Allocated Portfolios for Predicted Period

The allocations of the hybrid model starts making profit after 1 year of investment from 2022-Apr to 2023-Apr. Starting from there, the rate of cumulative return outperforms all other allocation methods. Up to 90% of return is expected for 2 years of total investment while the maximum of different methods - equal weight allocation - is only around 50%.

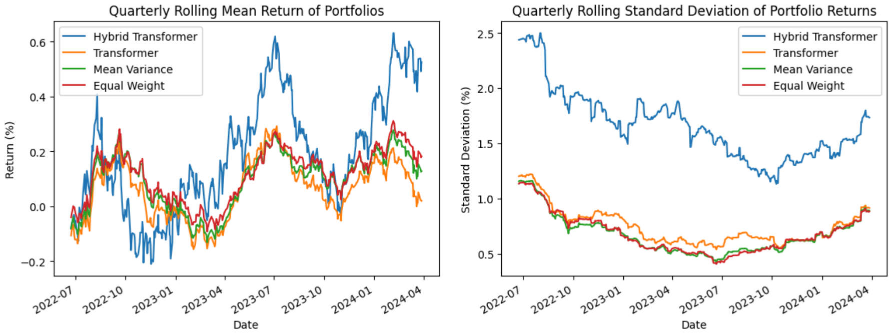

Fig 9: Quarterly Rolling Returns and Risk of Allocated Portfolios for Predicted Period

For quarterly rolling windows, the hybrid model still achieves the highest returns with the expense at a higher risk. Quarterly risk adjusted return - sharpe ratio is presented below.

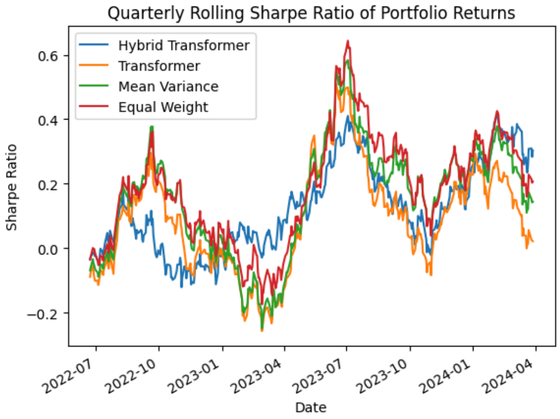

Fig 10: Quarterly Rolling Risk Adjusted Returns of Allocated Portfolios for Predicted Period

The rolling risk adjusted returns of all models are comparably similar. The hybrid model shows a greater sharpe ratio during the period of 2023-Jan to 2023-Jun.

6. Conclusion

This work could contribute to the usage of hybriding the transformer model with other sequential models for the portfolio allocation applications. Even for selecting the test period on the time when most of the assets' prices are in negative returns or slower growth rate, the hybrid transformer model is capable of optimizing the expected return by exhibiting higher risk without reducing the sharpe ratio for all quarterly windows. We had trained the model without constraints for allocation distributions and risk aversion factors. The future works will include the incorporation of those constraints to the hybrid transformer model in the usage of portfolio allocation.

Appendix

The code packages for model architecture, data loading and metric calculations are hosted in this github repository: Hybrid Transformer For Portfolio Allocation

Acknowledgement

We would like to express our gratitude to WorldQuant University (WQU) for providing an exceptional learning experience in the field of data sciences and financial engineering. We extend heartfelt thanks to the dedicated dean, professors, instructors at WQU. Their expertise, guidance, and passion for teaching have enriched our understanding of complex topics such as mathematical modeling, machine learning, and financial engineering. Lastly, we are grateful to my fellow students and the entire WQU community for their support, collaboration, and shared enthusiasm for knowledge.

References

[1] Schultz Collins Lawson Chambers, Inc. Portfolio Management: Theory & Practice. 2008,

[2] Harry Markowitz. Portfolio selection. The journal of finance, 7(1):77–91, 1952

[3] Hali, Nurfadhlina & Yuliati, Ari. (2020). Markowitz Model Investment Portfolio Optimization: a Review Theory. International Journal of Research in Community Services. 1. 14-18. 10.46336/ijrcs.v1i3.104.

[4] Zihao Zhang, Stefan Zohren, Stephen Roberts. Deep Learning for Portfolio Optimization. 2005. DOI: 10.48550/arXiv.2005.13665

[5] Kumar Yashaswi. Deep Reinforcement Learning for Portfolio Optimization using Latent Feature State Space (LFSS) Module. 2021. DOI: 10.48550/arXiv.2102.06233

[6] Ang, Andrew and Timmermann, Allan, Regime Changes and Financial Markets (June 20, 2011). SSRN: https://ssrn.com/abstract=1919497

[7] Christian Janiesch, Patrick Zschech, Kai Heinrich. Machine learning and deep learning
. 2021. DOI: 10.48550/arXiv.2104.05314

[8] Sen, Jaydip & Mehtab, Sidra & Sen, Rajdeep & Dutta, Abhishek & Kherwa, Pooja & Ahmed, Saheel & Berry, Pranay & Khurana, Sahil & Singh, Sonali & Cadotte, David & Anderson, David & Ost, Kalum & Akinbo, Racheal & Daramola, Oladunni & Lainjo, Bongs. (2022). Machine Learning: Algorithms, Models, and Applications.

[9] Thomas Conlon, John Cotter, Iason Kynigakis. Machine Learning and Factor-Based Portfolio Optimization. 2021. DOI: 10.48550/arXiv.2107.13866

[10] Y. Ma, R. Han and W. Wang, "Prediction-Based Portfolio Optimization Models Using Deep Neural Networks," in IEEE Access, vol. 8, pp. 115393-115405, 2020, doi: 10.1109/ACCESS.2020.3003819.

[11] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin. Attention Is All You Need. 2023. DOI: 10.48550/arXiv.1706.03762

[12] Qiuyue Zhang, Chao Qin, Yunfeng Zhang, Fangxun Bao, Caiming Zhang, Peide Liu,
Transformer-based attention network for stock movement prediction, Expert Systems with Applications. 2022. DOI: 10.1016/j.eswa.2022.117239.

[13] Damian Kisiel, Denise Gorse. Portfolio Transformer for Attention-Based Asset Allocation. 2022. DOI: 10.48550/arXiv.2206.03246

[14] Eduardo Ramos-Pérez,Pablo J. Alonso-González, and José Javier Núñez-Velázquez. "Multi-Transformer: A New Neural Network-Based Architecture for Forecasting S&P Volatility." *Mathematics*, vol. 9, no. 15, 2021, 10.3390/math9151794.

[15] Posedel, Petra. (2005). Properties and estimation of GARCH(1,1) model. Advances in Methodology and Statistics. 2. 10.51936/jjkd5433.

[16] Tim Bollerslev, Generalized autoregressive conditional heteroskedasticity, Journal of Econometrics, Volume 31, Issue 3, 1986, Pages 307-327, ISSN 0304-4076, DOI: 10.1016/0304-4076(86)90063-1.

[17] Neural machine translation with a Transformer and Keras. https://www.tensorflow.org/text/tutorials/transformer

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

[19] Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola. Deep Dive into Deep Learning. 370-374 (2021)

[20] Gort, Berend, and Bruce Yang. The Combinatorial Purged Cross-Validation Method. Towards AI, 31 March 2022.