R04

Optimizing the Transformer Model Architecture for trading in Equity and FX

By

CHRISTANTO, Nicholas
DYCHENGBENG, Matthew
GOEL, Kalpa
SAXENA, Srijan

Advised by

Prof. David Paul Rossiter

Submitted in fulfilment
of the requirements for COMP 4981

in the

Department of Computer Science and Engineering

The Hong Kong University of Science and Technology

2022-2023

Date of submission: April 20, 2023

ABSTRACT

This project aimed to design and optimize a transformer architecture-based multifactor model to predict financial returns in the Equity and Foreign Exchange (FX) markets. Additionally, we developed a trading strategy that utilizes the proposed model to make daily trades and generate returns. For securities within the Equity market, we focused on four different pillars of factors to act as input data for our model: Technical, Fundamental, Value, and Macroeconomic. For each equity that we analyzed, we trained and compared five different model instances based on the different combinations of factors used as input data: Technical Factor Model, Technical + Fundamental Factors Model, Technical + Value Factors Model, Technical + Macroeconomic Factors Model, and Technical + Fundamental + Value + Macroeconomic Factors Model.

Our results showed that for all securities in the FX market, our model and trading strategy consistently outperform Buy-and-Hold returns (e.g., for AUD/USD, our system achieves a CAGR of 45.98% versus a CAGR of -5.46% for the Buy-and-Hold strategy). For securities in the Equity Market, our Technical + Fundamental Factors Model consistently performs better than the other four models and even outperforms buy-and-hold returns (e.g., for AAPL Technical + Fundamental Factors Model, this model achieves a CAGR of 75.69% versus a CAGR of 50.11% for the Buy-and-Hold strategy). Furthermore, while trading SPY, both our Technical and Technical + Macroeconomic Factors Model outperform buy-and-hold returns (Buy-and-Hold CAGR: 0.22%, Technical CAGR: 6.39%, Technical + Macroeconomic CAGR: 0.39%).

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF EQUATIONS

TABLE OF TABLES

1 INTRODUCTION

1.1 OVERVIEW

In the financial industry, machine learning is growing at a rapid pace with over 80% of financial firms making a significant investment in this area in recent years [1]. Although breakthroughs in machine learning have led to multiple applications in the financial industry, such as fraud detection and algorithmic trading, one of the most important applications in this industry is forecasting financial returns [2]. With the benefits of machine learning, firms now train models to analyze large historical data sets for exploring potential historical trends that can be used to forecast future data. In recent years, convolutional neural networks (CNNs) and recurrent neural networks (RNNs) are machine learning models that have become industry standards in the financial industry for forecasting time series data such as financial returns. The rise of these models' widespread application in this industry can be attributed to its advantage of revealing opportunities not necessarily discoverable through traditional technical analysis methods [3]. Although the performance of these models may suit the needs of the current financial industry, these models still have drawbacks. For example, a disadvantage of using CNN models is that they lack a structural understanding of the time-series nature of financial data. Meanwhile, a disadvantage of RNN models is the utilization of sequential processing, as this causes an inability to deal with long-range dependencies and an inability to perform parallelization. As these drawbacks bring caveats to the quality and accuracy of forecasted data, our team looked at newer methods to forecast time-series data.

A model architecture that has not yet been explored in this this context is the Transformer architecture. First introduced in the paper, “Attention is All You Need” [4], Transformers have consistently outperformed CNN and RNN based architectures and have shown great successes in Natural Language Processing (NLP), Video Processing [5], and other time series domains [6]. Therefore, they represent a potentially valuable approach to return prediction in finance. This success can be mostly attributed to its positional embeddings and “attention” philosophy [4]. In contrast to CNN and RNN sequential computation, the transformer model avoids recursion in order to allow parallel computation and make use of the positional embeddings to signify sequential arrangement. Self-attention, which is the newly introduced ‘unit’ used to compute similarity scores between different time points in a sequence, is another important aspect of the transformer model, allowing long distance dependencies. This allows the transformer model to not only rely on past hidden states to capture dependencies, but also processes the time-series sequence as a whole. Therefore, the model is able to effectively perform pattern recognition within a sequential context. Considering the benefits of the transformer model in NLP over RNN and CNN, we sought to explore the transformer model on price movement forecasting based on historical data.

The goal of this project was to build a transformer-based deep learning model to predict financial equity and foreign exchange (FX) returns. To assess the practical effectiveness of the model, we also developed a trading strategy that considers market friction and an optimized stoploss system. We developed different models using different combinations of input factor pillars (technical, value, fundamental, and macroeconomic) to analyze and compare the impact of those pillars on the trading performance of our model.

1.2 Objectives

In our project, we aimed to develop a stock price movement forecaster that leverages the transformer model. To achieve this, we had three objectives:

1. Conduct preliminary statistical data analysis to identify the most important factors (e.g., price data, macroeconomic indicators etc.) to act as input for the model.
2. Design and train a transformer-based model to develop a profitable trading strategy using our transformer model as a backbone, that accounts for market friction, such as trading commission cost, in its decision-making process. Profitability is defined by cumulative annual growth rate (CAGR).
3. Systematically evaluate the performance of each factor pillar model to understand the benefits of each factor pillar and to determine the best performing model.

To achieve the first objective, we sourced technical, fundamental, value, and macroeconomic data from various sources and employed decision trees to ascertain to most important factors. For the second objective, we studied several transformer model architectures in the literature and performed extensive iterative experiments to select the best hyperparameters. We also studied the different costs associated with market friction and derived appropriate risk management techniques to address the same. To achieve the final objective, we studied the benefits of each factor pillar and evaluated both how it contributed to the model's performance and how it outperformed or underperformed against the other models.

We faced several obstacles while working on this project:

1. Optimizing hyperparameters and model architecture.
2. Sourcing time series data for non-price related factors at the desired frequency.
3. Reducing the noise of the input data and designing an appropriate input projector

Our solutions to these obstacles are discussed in section 2.2.

1.3 LITERATURE SURVEY

In this section, we will first introduce and summarize existing literature on predicting financial returns using deep learning architectures such as CNNs, Long Short-Term Memory (LSTMs), and CNN-LSTMs, along with their limitations. Next, we will introduce literature on the Transformer architecture, how it addresses the aforementioned limitations, and its successes in various time series domains.

1.3.1 Stock Market Prediction using CNN and LSTM

In the paper by Hamdy Hamoudi and Mohamed A Elseif [7], the authors introduce a CNN and LSTM-based supervised learning model approach to identify trading signals on time series data of securities that outputs a binary prediction of gain or loss on each trade. The model was trained on Jane Street's proprietary securities market data set with over 130 independent factors during a 500-day duration [8].

However, the study shows certain challenges with CNN and LSTM-based models in predicting either gains or losses with a time series data set for trades and returns. This was highlighted by low precision numbers ranging from 0.3 to 0.4 achieved by the authors, when using several iterations of their CNN and LSTM-based models.

To understand the limitations, one must evaluate how CNN and LSTM models are structured independently.

CNN

A 1D-CNN architecture is optimized to eliminate noise and recognize patterns within data. However, it lacks any structural understanding of the time series nature of data. This means that not only is it unable to make inferences using an essential sequential component of the data, but also that the number of different kernels required to capture dependencies grows with the size of the inputted data. This tends to make the model very large in terms of learned parameters, thus making it infeasible for a thorough analysis with a long sequential data set.

LSTM & RNN

LSTM on the other hand is a type of RNN model. RNN is a step forward from CNNs on time series data. It addresses the sequential factor neglected by CNNs. Each data point "t", while training, also considers the exact previous "t-1" data point for training in addition to all the input factors at time "t". However, this very feature, proves to be a limitation for RNN during the training process. RNNs are subject to a vanishing/exploding gradient problem, whereby the training weights for long time-series data closer to the inception time, "t = 0", tend to not get updated, given the multiple layers for time periods the training process iterates through. The LSTM structure tries to address this issue with multiple gates (input, forget and output) at each node.

However, by its very structural design, the model architecture still provides greater emphasis on data points at the end of the sequence, potentially ignoring long term and cyclical patterns.

Furthermore, since the LSTM model architecture is sequential in nature, the training of the model cannot be effectively parallelized since each node “t” has to wait for the hidden embeddings from the previous nodes before performing calculations. This leads to long training times and limits the size of the training data.

CNN-LSTM

The CNN-LSTM is an LSTM architecture specifically designed for sequence prediction problems. Successful use cases can be found in the domain of visual recognition and description. It aims to combine a CNN’s noise reduction and pattern recognition capabilities with an LSTM’s sequential architecture. While often performing better than each of the architectures individually, it still retains certain structural issues inherent in both models, such as the number of kernels, difficulties in understanding long-term sequential dependencies, and the lack of parallelization.

1.3.2 Transformers in Natural Language Processing

Transformers were first introduced in the paper “Attention is all you need” [4], with its primary applications in NLP and machine translation. Language, which constitutes sentences and phrases, can be equated to be as a form of sequential data set, where word ordering and positioning directly impacts the output. This is where parallels can be drawn with time series data of security prices, for example, that can be a major application area for transformers.

While most prominently still used in NLP, transformer models highlight three key benefits namely, parallelization, self-attention, and positional encoding. To understand the benefits, one must underscore the major structural differentiators of a transformer model:

1. Transformer models use "self-attention" to identify sequential data snippets within the larger dataset that most significantly impact the prediction of the next data point. This eliminates the vanishing gradient problem and the inability to identify long-distance dependencies between points far apart in a data sequence.
2. Instead of having a single stream of sequential input, as we see in RNN models, the entire sequence can be split and input simultaneously for parallel computing. This significantly reduces training time and allows one to train over a larger dataset.
3. Information about the sequence of the data is instead entered through positional encoding, allowing identification of any potential sequential dependencies.
4. Finally, transformer models are able to perform a so-called "multi-headed attention" approach, whereby they run several self-attention learning modules simultaneously. Thus, being able to parallelly infer different types of associations between different parts of the sequence.

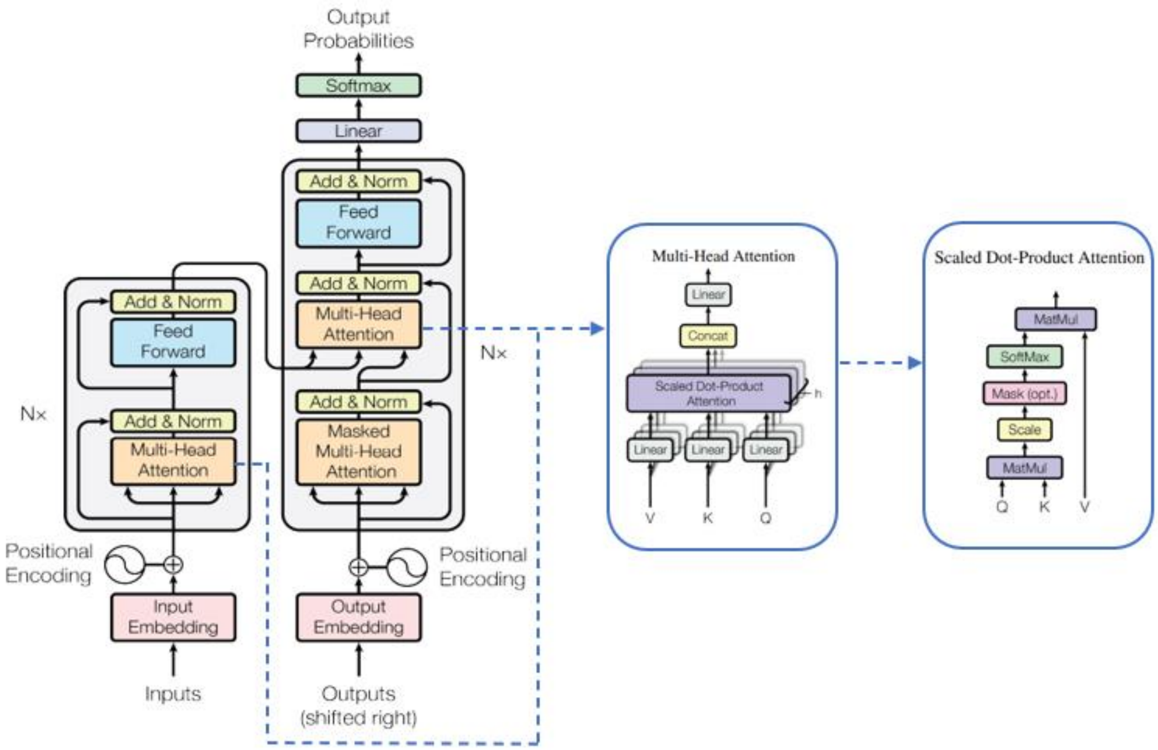

Figure 1: The transformer model detailed breakdown overview from [4]

1.3.3 Transformers in Multifactor Time Series Predictions

The paper, "Transformer-based time series prediction of the maximum power point for solar photovoltaic cells" [6], highlights the pivot into the application of transformer models in analyzing non-textual time-series data.

The paper uses transformers to predict voltage output data of photovoltaic cells with respect to time/duration. Figure 2, below, shows the encoder architecture of the transformer taking in a stream of sequential continuous and one-hot encoded categorical variables, as multiple input factors.

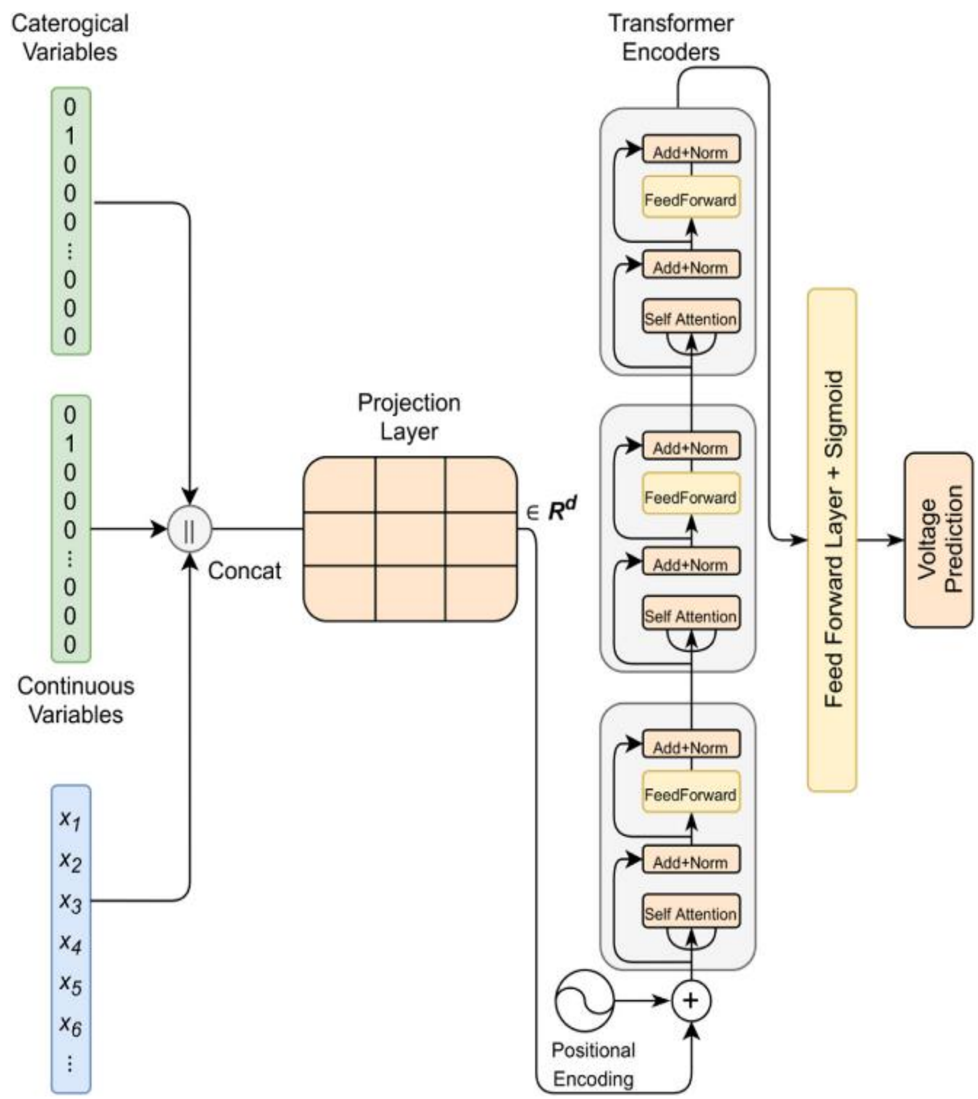

Figure 2: Transformer model architecture for photovoltaic cell voltage prediction [6]

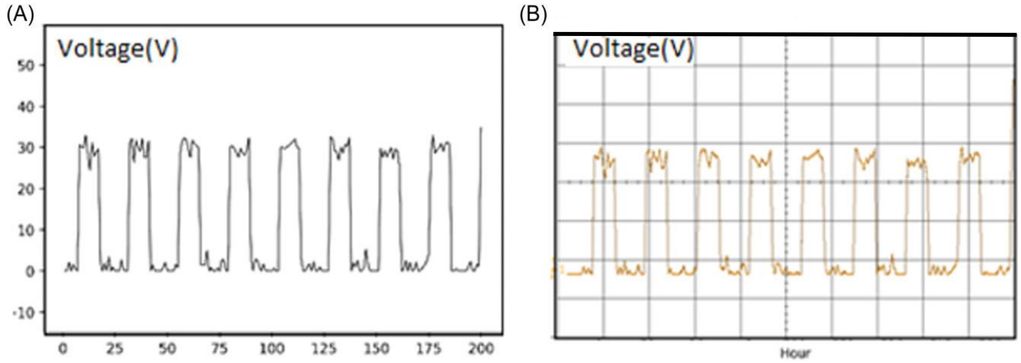

Figure 3: Photovoltaic cell transformer model predicted (A) vs actual outputs (B) [6]

Whereas Figure 3 highlights the model predicted (A) and actual measured (B) voltages respectively. What stands out is the model's ability to not only underscore the major periodical trend, but also intra-period fluctuations of voltages in tandem.

Altogether, through our literature survey and extensive background research, it laid the foundation for us to explore and research the application of transformer in predicting returns in various financial markets.

2 METHODOLOGY

2.1 DESIGN

We limited the scope of this research to four securities in the equity market: SPY, AAPL, AMZN, MSFT, and seven currency pairs in the FX Market: USD/CAD, EUR/USD, AUD/USD, USD/CHF, USD/JPY, GBP/USD, CNY/USD. Within the equities market, we chose SPY since the S&P500 index is indicative of the overall US equity market performance, while the other three equities were chosen since they are the largest companies by market capitalization within the S&P500. For the FX market, we chose all the currency pairs that form the U.S. Dollar Index. In an effort to limit the complexity of our decision space, we decided to train a different instance of the model for each security.

One of the first steps was to source the factors that form our time series input data. For this research, we explored factors from four key pillars: Technical (price data), Fundamental (company fundamentals), Value (bargain metrics), and Macroeconomic (key economic indicators) data. The key motivation behind using factors from a variety of pillars, instead of just price data, was to allow the model to gain a more comprehensive understanding of the respective economic and financial climate, and thus, infer the context behind price movements. The exact factors within each pillar were chosen after a thorough analysis of their relationship with returns. To achieve the fourth objective (section 1.2), we trained five different model instances for each security in the equity market, based on the factor pillars used to train the model:

1. Technical (T)

2. Technical + Fundamental (T + F)

3. Technical + Macroeconomic (T + M)

4. Technical + Value (T + V)

5. Technical + Fundamental + Value + Macroeconomic (T + F + V + M)

Models 2, 4 and 5 are not trained for SPY since the S&P500 index does not have fundamental and value factor data. For FX securities, only a technical factor model is trained.

The next step was to design the high-level architecture of our model. The system (Figure 4) is designed such that 10 days of continuous daily data is used to predict the following day's closing return. The input data goes through an input projection layer to perform automatic feature extraction, before being concatenated with the positional encoding values. This data is then input into a series of 3 transformer encoders, which implement the Multi-headed Self-Attention mechanism, as described in Section 1.3.2. The output from the encoders is utilized by a final layer to predict the return. Finally, this predicted price is used by our trading algorithm to take one of three decision – Buy, Sell, Hold.

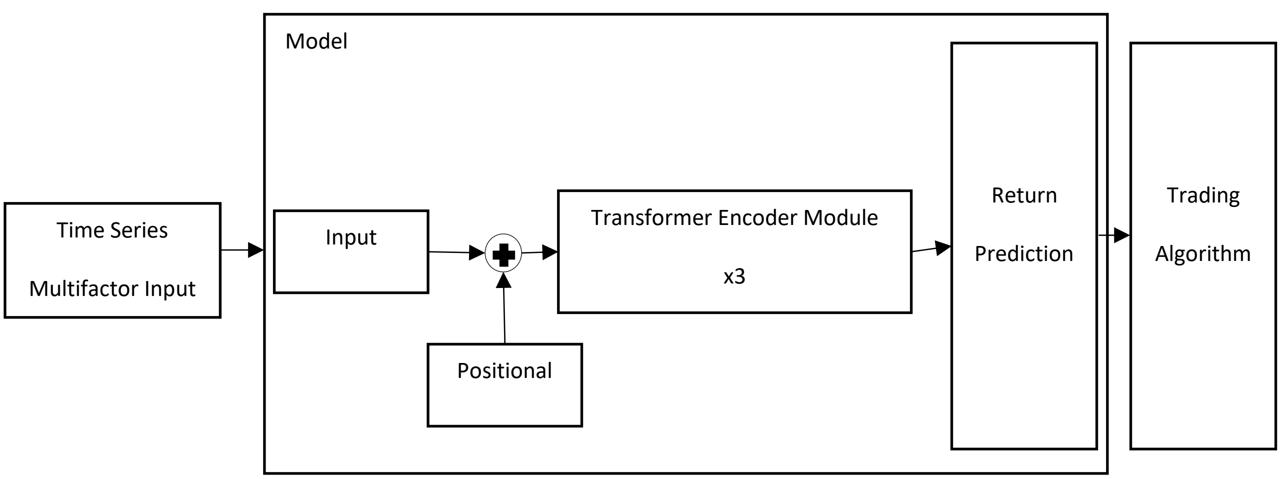

Figure 4: High-level model architecture (self-made)

Finally, we evaluated each model instance using three metrics: cumulative annual growth rate (CAGR), Sharpe Ratio and Maximum Drawdown, and used that to compare models using different factor pillars.

2.2 IMPLEMENTATION

2.2.1 Data Sourcing

To collect data across all factor pillars, we utilized three main sources: Yahoo Finance [9], Bloomberg [10], and EODHD [11]. Yahoo finance and EODHD data were scraped using the website's Python API to collect daily Technical and Macroeconomic data, as well as quarterly Fundamental data. Bloomberg terminals were used to collect Value data and Macroeconomic data. We wanted to ensure that the time period of the dataset for all factors for each security was consistent, so that we could make accurate comparisons. Due to limited fundamentals data, our dataset for securities within the Equity market was limited. The following time period were used for each security.

Table 1: Equity and FX data time period

Security	Start Date	End Date	Number of Trading Day Data Points
AAPL	02-Jan-04	21-Sep-21	4410
AMZN	02-Jan-04	29-Sep-20	4163
MSFT	02-Jan-04	29-Sep-20	4163
SPY	02-Jan-04	17-Feb-23	4766
All FX	17-Sep-03	17-Feb-23	4998

2.2.2 Factor Selection

2.2.2.1 Technical Factor Pillar

For the Technical Factor Pillar, we used daily price data (Opening Price, Closing Price, Highest Price, Lowest Price), as well as the daily trading volume.

2.2.2.2 Fundamental Factor Pillar

To ascertain which factors to include from the Fundamental Pillar, we conducted exploratory and statistical analysis in two stages:

First, we evaluated feature importance according to explainable and descriptive statistical models. We trained an Extratree Regressor [12] and a linear XGBoost Regressor [13] on the fundamental factors to predict daily opening price. Figure 5 is a sample result by the XGBoost model which shows the features importance ranked. The model highlights total liabilities, total assets, total stockholder equity, common stock, and other current assets as the most important fundamental factors. Meanwhile, the Extratree Regressor similarly highlighted total liabilities, total assets, total stockholder equity and 13 other features which are also found in the XGBoost results in Figure 5. As both the XGBoost Regressor and Extratree Regressor return overlapping key features, we took the 16 most important features that were present in both results.

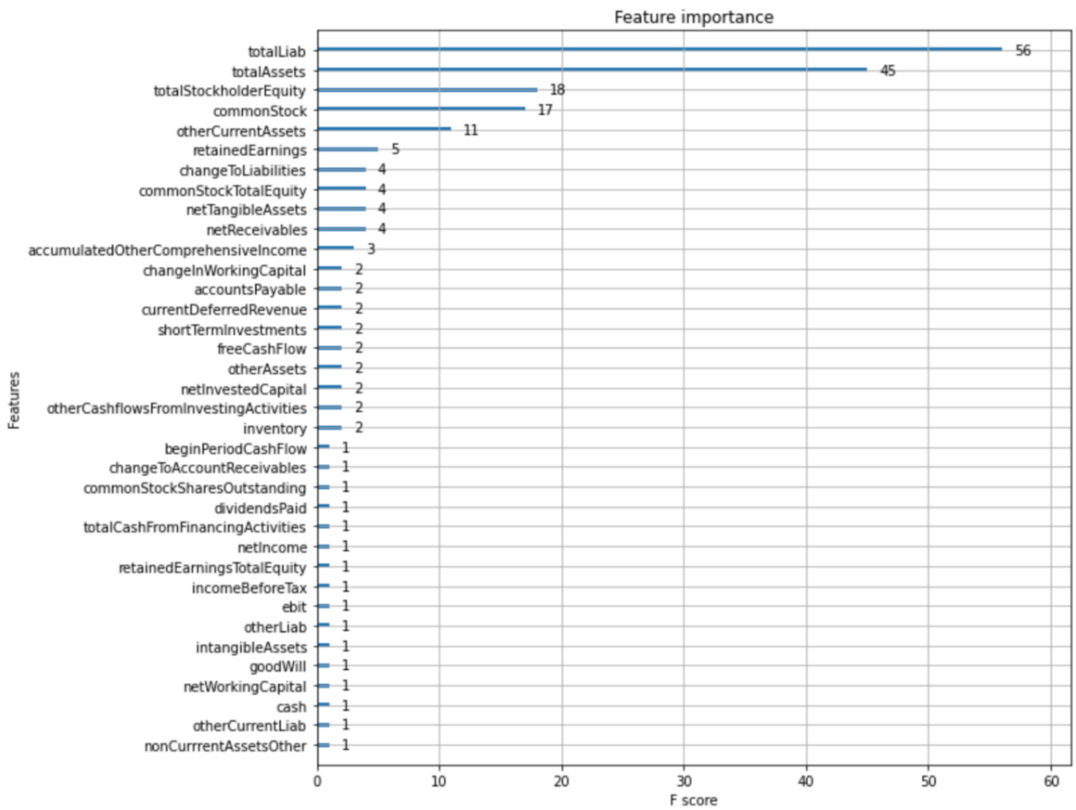

Figure 5: Feature Importance of Fundamental Factors according to a linear XGBoost Regressor

Once the factors had been narrowed down to 16 key factors, we constructed mock factor portfolios and evaluated the performance of these factors. Mock factor portfolios allowed us to rank each factor based on the portfolio's profitability and hence determine which features were most correlated with returns. The mock factor portfolio for each factor was constructed by ranking securities based on the value of that factor and then creating a portfolio that buys the top five stocks and shorts the bottom five stocks. We calculated the returns of these portfolios and ranked the features based on their corresponding mock factor portfolio's return. Finally, we selected the factors which had the top performing portfolios to include in our fundamental model. The selected factors were: Total Assets, Total Liabilities, Net Debt, Intangible Assets, and Total non-current Assets.

2.2.2.3 Value Factor Pillar

For Value Factor Pillar, we decided to use the following factors: price to earnings ratio, price to sales ratio, enterprise value to EBITDA ratio, and put call ratio. We chose the first three factors because they indicate underlying valuations of the company relative to the firms' earnings and revenue streams. These can be useful indicators for investors to understand whether the stock price is trading at a premium or discount compared to its own historical valuations and peers. We chose the Put Call Ratio because it summarizes the stock options trading flow amongst investors and highlights an alternative investor sentiment on value.

2.2.2.4 Macroeconomic Factor Pillar

For the Macroeconomic Factor Pillar, we decided to use the following factors: 13-week treasury bill, 10-year treasury bond, 30-year treasury bond, crude oil, gold, Canadian dollar to US dollar exchange rate, Japanese Yen to US dollar exchange rate, Euro to US dollar exchange rate, and the Chinese yuan to US dollar exchange rate. The 13-week, 10-year, and 30-year treasury bills underscore the overall interest rate conditions by estimating the level, slope, and curvature of the yield curve. The currency pairs represent the largest trading partners of the US. Crude Oil prices are included since they impact all major industries and consumption across the industries in the US. Finally, gold prices are included because gold is considered to be a safe haven asset for investors during weaker economic outlooks and thus, it is an indicator of broader macroeconomic and business conditions influencing stock price movements.

2.2.3 Data Pre-processing and Transformation

Data pre-processing and transformation was conducted on Python Notebooks using libraries such as NumPy [14], Pandas [15], Seaborn [16], Matplotlib [17], Scikit-learn [18] and PyTorch [19].

To ensure effective learning by the transformer model, it was necessary to introduce stationarity to the raw stock prices to a certain extent. Stationarity refers to the condition where the statistical properties of the data remain unchanged over time. One approach used to introduce stationarity to the time-series data was to use percentage change which removes trends and frame the data into rate of change. This is done to all the feature columns across all pillars. For price features such as daily closing price, doing percentage change is equivalent to finding the daily closing return. Following this transformation, a rolling geometric mean transformation was used to provide a consistent mean in different time spans. The rolling mean's window is set to 10, which corresponds to two weeks average in our daily data.

We also performed outlier selection to remove data points where the factor values were higher than 10 times the Inter Quartile Range of that factor. Finally, to normalize the data, we performed max absolute scaling for features that includes negative values. This scaling process divides the values in each factor with the maximum absolute value of that feature. Maximum absolute scaling does not shift the centre of data and preserves the sparsity of the data. Additionally, features that only include positive values, such as trading volume, were scaled using min-max scaling, which takes the minimum and maximum values of the feature and scale the data into the range from 0.0 to 1.0.

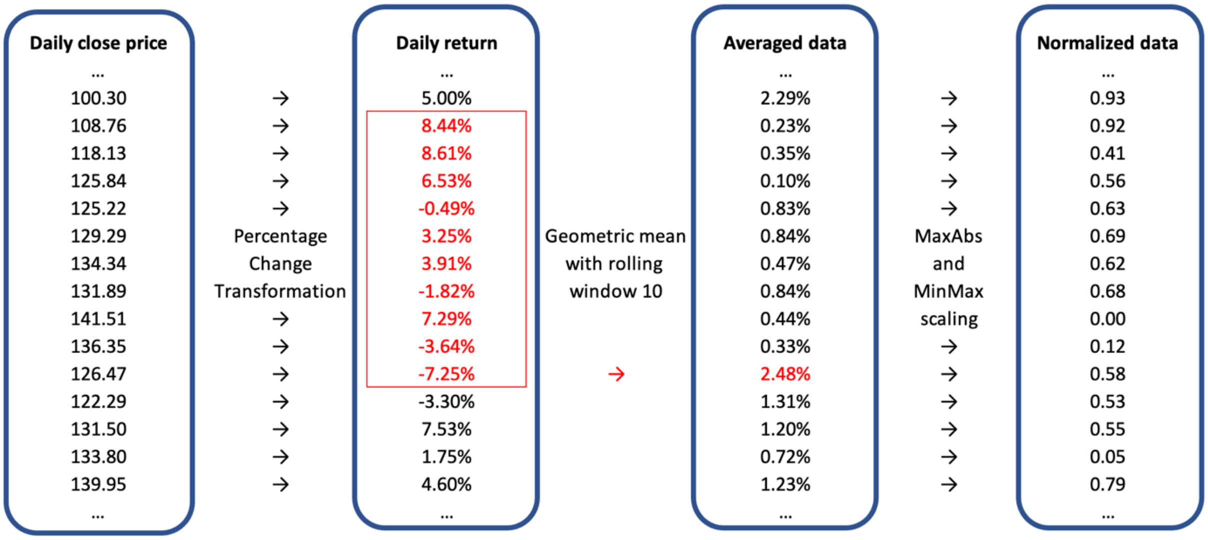

Figure 6: Pre-processing flow

Finally, we performed Principal Component Analysis (PCA) [18] on all factors except those from the Technical Factor Pillar and chose the components that cumulatively explained greater than 99% of covariance between the factors.

To make sure our model is evaluated properly, the dataset is split into three parts: train data, validation data, and test data with a proportion of 80:10:10 respectively.

2.2.4 Model Design and Training

The model was designed such that 10 days of continuous daily data is used to predict the following day's closing returns. We used Time2Vec [20] for positional encoding and used a simple feedforward layer as the input projection layer. After concatenating the positional encoding values, the data was input through three consecutive transformer encoder modules, followed by global average pooling. Finally, two dropout-enabled feedforward layers were used to output a single value: the model's predicted price. We used the Adam [21] algorithm for optimization and Huber Loss [22] with delta 0.01 as the loss function. We implemented our model using Keras [19] and trained the model on Google Collab using its hardware accelerator.

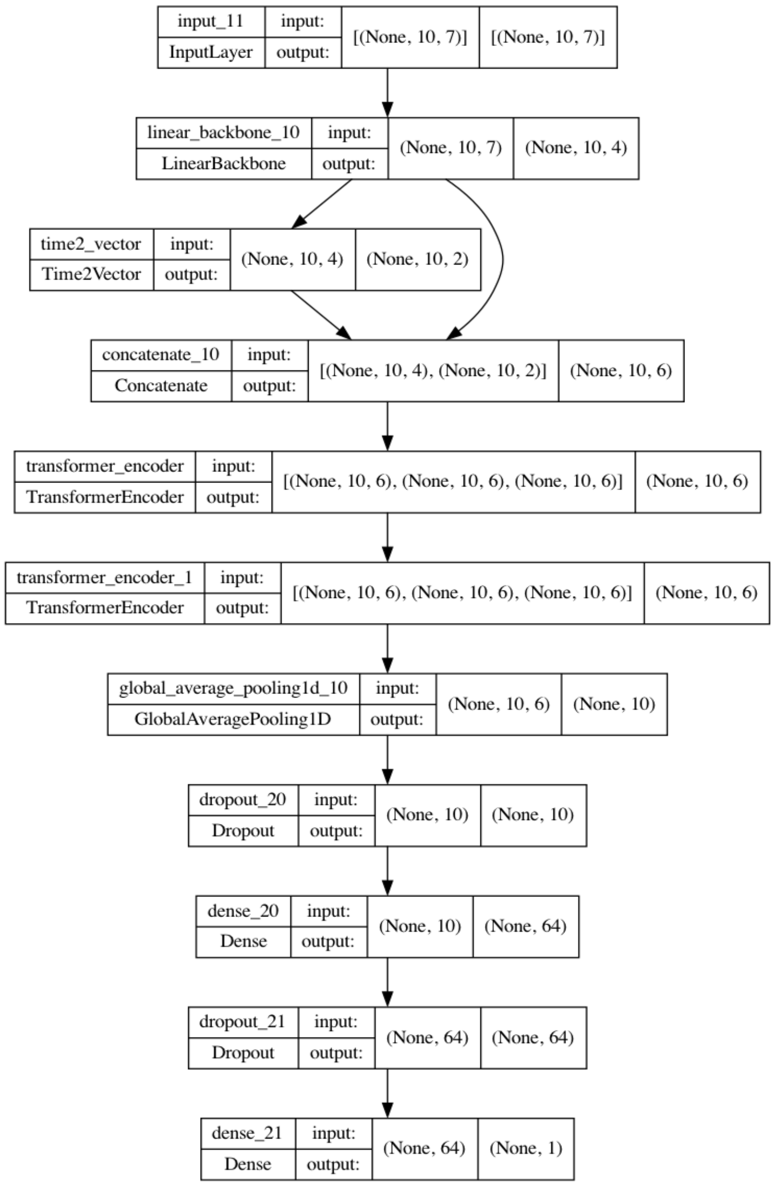

Figure 7: Architecture of implemented transformer model

2.2.5 Trading strategy

The current model has the capability to generate a single value that represents the daily return percentage after the input data sequences. To analyse the price movement as a result of the model's output, we need to examine the label data processing. Assume $Close_0$ is the result of pre-processing of the following sequence, where x_i is the close price percentage change and $Close_1$ is the result of pre-processing on the day after $Close_0$.

$$Close_0 = [(1 + x_0) * (1 + x_1) * (1 + x_2) * \ (1 + x_3) * \ (1 + x_4)]^{\frac{1}{5}} - 1$$

$$Close_1 = [(1 + x_1) * (1 + x_2) * \ (1 + x_3) * \ (1 + x_4) * \ (1 + x_5)]^{\frac{1}{5}} - 1$$

Equation 1: Pre-processed close price

By comparing the four daily return data from the two equations, we can determine the relationship between $Close_0$, $Close_1$, and the percentage change of the close price. If $Close_1$ is greater than $Close_0$, then x_5 is greater than x_0, indicating a positive trend. In a situation where x_0 is positive, then x_5 will also always be positive. If $Close_1$ is less than $Close_0$ then x_5 is less than x_0, indicating a negative trend. At the same time, when x_0 is negative, then x_5 will also always be negative.

Therefore, if the output value, $Close_1$, is higher than the label value from the last day of input data, $Close_0$, we should take a long position for the next 24 hours. Conversely, if the output value $Close_1$ is lower than the label value $Close_0$ from the last day of input data, we should take a short position for the next 24 hours since the market closed on the last day of input data.

2.2.6 Back-testing

To evaluate the performance of the trading strategy, back-testing was performed using the test data. In order to simulate the real-world conditions, several factors were considered. Firstly, an arbitrary initial trading balance of 10,000 USD was established to ensure that calculation of the commission fee was possible. Secondly, commission fees were considered using Interactive Broker's rates [23] since it is one of the largest trading brokers in Hong Kong. A commission fee of 0.05 USD per share with minimum 1 USD and maximum 1% of trade value was applied for equity trading, while a commission fee of 0.2 basis points of the trade value was used for foreign exchange trading. The commission fee was charged when entering and exiting trades for both equity and foreign exchange trading.

Risk management is also an important aspect of a proper trading behaviour. In our research, we optimized the stop-loss percentage by testing out different stoploss level based on the validation data and implemented it when back-testing with the testing data. Since we did not have minutely data, the stop loss works by comparing the percentage change between the daily closing prices, and if the change created a loss that was more than the stop-loss percentage, then the return on that trade was capped at the stop-loss percentage. However, the limitation of this approach is that the next day opening price might produce a loss bigger than the stoploss level. In this case, the return on that is set as the gap between the close price and next day open price, and the trade is closed.

2.3 TESTING

2.3.1 Data testing

To test our data collection, testing was done through cross-referencing data sources. We tested the validity of our data by cross-referencing our data with other sources such as Investing.com [24]. For example, technical and macroeconomic data collected through the yahoo finance API was cross referenced with data found on Investing.com. Similarly, value data collected from the Bloomberg terminals and fundamentals data collected from EODHD were also cross referenced with data from Investing.com and Yahoo finance.

2.3.2 Pre-processing testing

We also tested our pre-processing results in order to ensure our pre-processing methodology was done correctly. The way we tested the pre-processing process is by inverting the final output into its original value. Our pre-processing consists of three steps. The first one is transforming each value to be the percentage change from the previous value. Next, a geometric mean rolling window of 10 is applied to the percentage change values. Lastly, normalization such as min-max scaling and maximum absolute scaling is done on the geometric mean values.

To invert the values, we first collected the essential values from the normalization transformation. For features that used min-max scaling, we recorded the minimum and maximum values. On the other hand, only the maximum absolute value was needed to invert features that uses maximum absolute scaling. The inversion on normalized values is done by switching the variables in the equation to end up with the unnormalized values.

Min-max scaling inversion equation:

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}} \rightarrow x = x_{scaled}(x_{max} - x_{min}) + x_{min}$$

Equation 2: Min-max scaling inversion

Maximum absolute scaling inversion equation:

$$x_{scaled} = \frac{x}{|\max(x)|} \rightarrow x = x_{scaled} * |\max(x)|$$

Equation 3: Maximum absolute scaling inversion

Furthermore, to do inversion on the geometric mean, the product for the previous four data points was recorded. The following equation was applied to obtain the original percentage value from the result of the geometric mean.

$$gmean_0 = [(1 + x_0) * (1 + x_{-1}) * (1 + x_{-2}) * (1 + x_{-3}) * (1 + x_{-4})]^{\frac{1}{5}} - 1$$

Equation 4: Rolling geometric mean

$$x_0 = \frac{[g_{mean} + 1]^5}{(1 + x_{-1}) * (1 + x_{-2}) * (1 + x_{-3}) * (1 + x_{-4})} - 1$$

Equation 5: Rolling geometric mean inversion

Lastly, to obtain the original closing price, the following equation was used.

$$percentage\_change_0 = \frac{close_0 - close_{-1}}{close_{-1}}$$

Equation 6: Percentage change

$$close_0 = [percentage\_change_0 * close_{-1}] + close_{-1}$$

Equation 7: Percentage change inversion

By comparing the original values and the inversion results, we were able to confirm that the pre-processing was applied correctly.

2.3.3 Model testing

To evaluate our model's proper functioning, we also tested our model on self-generated data that does not represent any security's actual historical and current factor data. To do so, we created two sets of data to be tested upon.

The first one is a time series data set that is defined by the function defined below to represent close prices of a hypothetical security. Through our first set of testing, the objective is to evaluate whether our model can predict the subsequent day's closing returns by learning the trend of historical close prices.

The following is the function for the first self-generated close price data set, where C_t stands for close price on day t and C_1 to C_{10} are defined as consecutive integers from 1 to 10.

$$C_t = 1.001 \times C_{t-1} - 1.0009 \times C_{t-2} + 1.0008 \times C_{t-3} - 1.0007 \times C_{t-4} + 1.0006 \times C_{t-5}$$
$$- 1.0005 \times C_{t-6} + 1.0004 \times C_{t-7} - 1.0003 \times C_{t-8} + 1.0002 \times C_{t-9} - 1.0001 \times C_{t-10}$$

Equation 8: First self-generated close price: Weighted Sum

The purpose of using the above-mentioned function is to generate a C_t that takes in information of the 10 prior time periods, thus mimicking the type of function our transformer model is expected to learn. The choice of a linear function and the choice of coefficients was arbitrary.

Figure 8: Model testing using first set of self-generated close price.

The test results aptly highlight that the closing price returns predicted (green) from the model follows the actual returns of the close prices for the time series data generated (blue) closely. This underscored the transformer's ability to learn from sequential data following a systematic relational trend as defined by the equation, where each close price is dependent on the trend from the time period earlier.

On the other hand, the second dataset is discrete in nature, where each day's closing price (C_t) is calculated as listed below. Again, the objective is to evaluate whether our model can predict the subsequent day's closing returns by learning the trend of historical close prices.

$$C_t = \sin(x)\,, where\ x\ is\ a\ random\ generated\ positive\ integer$$

Equation 9: Second self-generated close price: sin(x) transformation

The key differentiation in this case from the first one explained above is that there is no relational trend between any C_t and $C_{t\pm i}$. This is because each C_t is generated from a random input.

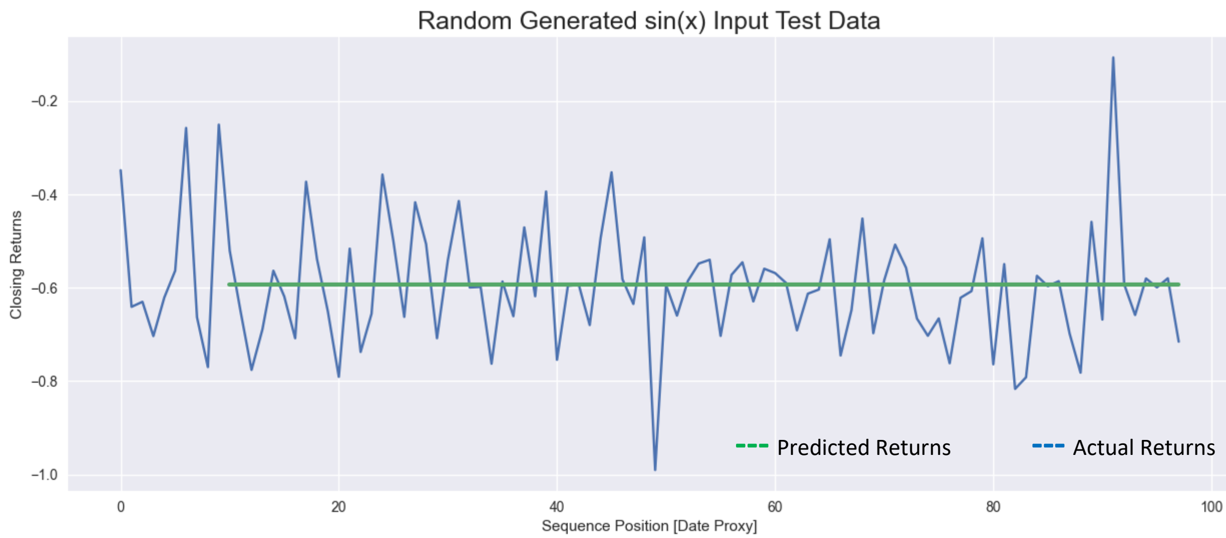

Figure 9: Model Testing using Second Set of Self-Generated Close Price

As expected from the graph above, the transformer is not able to learn the trend to predict the closing price returns for the second scenario. It underscores the reasoning that transformers learn by understanding sequential patterns in historical data to predict the next time instance. However, since each C_t is random and is unrelated to any other close price, there is no learning taking place.

2.4 EVALUATION

2.4.1 Evaluation metrics for model trading performance

Evaluating a trading strategy is an essential step to produce successful trades. There are many factors that traders need to consider when evaluating a trading strategy like **CAGR** (cumulative annual growth rate)**, Sharpe Ratio,** and **maximum drawdown**. These factors can aid traders in determining the long-term profitability and risk of a trading strategy.

- CAGR calculates the average return over a period while considering compounding. A higher CAGR shows that the trading strategy is more profitable over time. Instead of using overall return, using CAGR is more objective since the timeframe is standardized into annual growth. A trading strategy with a higher CAGR is preferable.
- The Sharpe Ratio measures risk-adjusted returns. It is calculated by dividing the average returns over the standard deviation of returns. A high Sharpe Ratio shows that the trading strategy can produce higher returns while taking on less risk. A high Sharpe Ratio can also show consistent returns as the standard deviation is low. Trading strategy with higher Sharpe Ratio is preferable.
- Maximum drawdown is the largest loss in percentage experienced by a trading strategy. It measures the maximum decline of the trading balance from its peak to its trough. Traders need to be aware of high maximum drawdown even in the face of high returns because it might indicate that the trading strategy is inconsistent.

CAGR, Sharpe Ratio, and maximum drawdown are used in this research to evaluate each model.

2.4.2 Evaluation of model trading performance on equity

2.4.2.1 Technical model

Starting with the simplest transformer, we only used the technical data such as closing price, opening price, high price, and low price. Based on figure 10, we can see how our transformer model was able to closely predict the actual closing return labels of the AAPL stock. This shows that we were successful in creating a transformer model that can be used in a financial context to predict stock return movement.

Now moving on to the performance of our model, the CAGR results in figure 11 showed that while this model has the capabilities to be profitable, the returns did not exceed the simple buy-and-hold strategy except only for SPY. The Sharpe ratio results also followed similar trend as the CAGR results. Our model, however, was able to reduce maximum drawdown across all securities by taking less trades, especially losing trades.

The addition of stoploss has proven to improve performance across all metrics. The stoploss has increased CAGR, Sharpe ratio, and decrease the maximum drawdown.

For more detail on the result of this model, refer to appendix 7.1 which includes trading balance for each equity and how it changes over time.

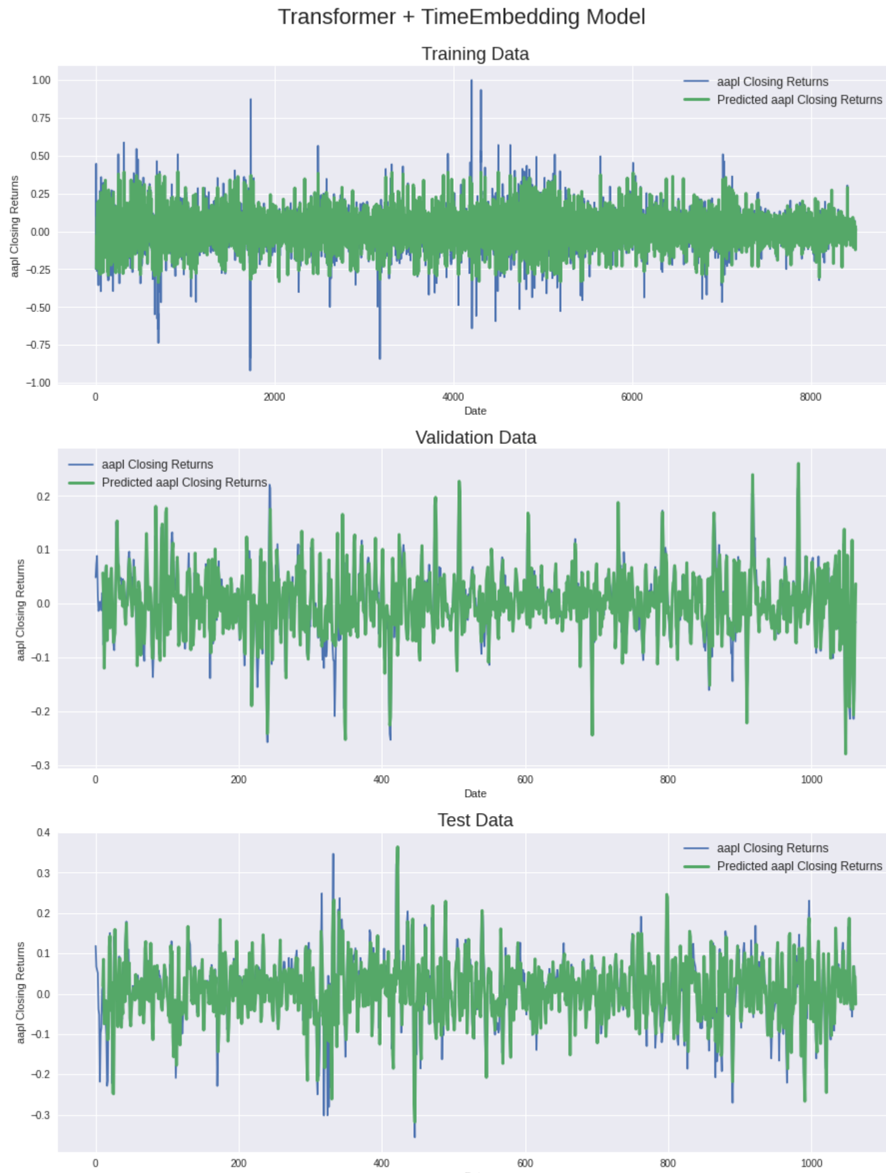

Figure 10: Equity technical model AAPL predicted label vs actual label

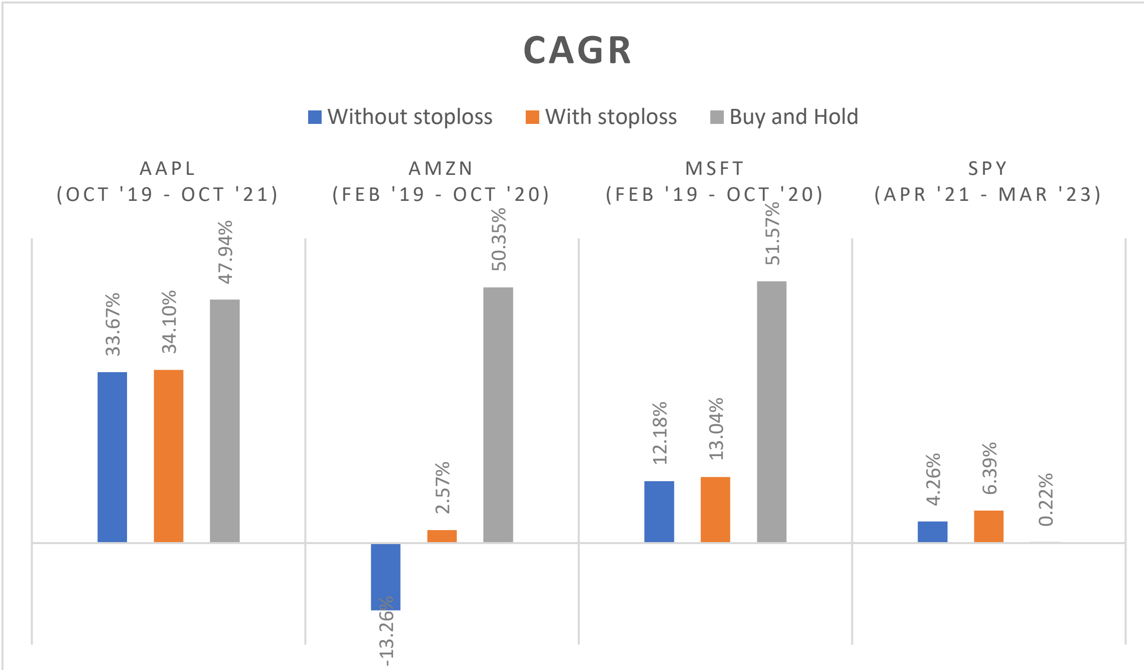

Figure 11: Equity Technical model CAGR bar chart

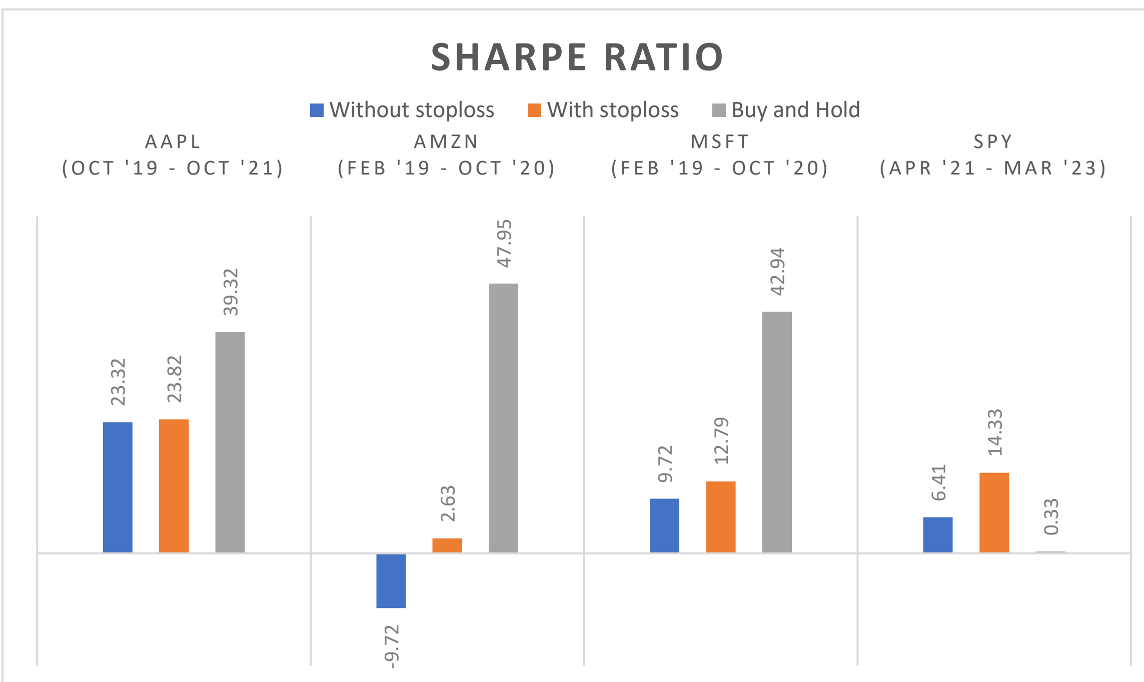

Figure 12: Equity Technical model Sharpe ratio bar chart

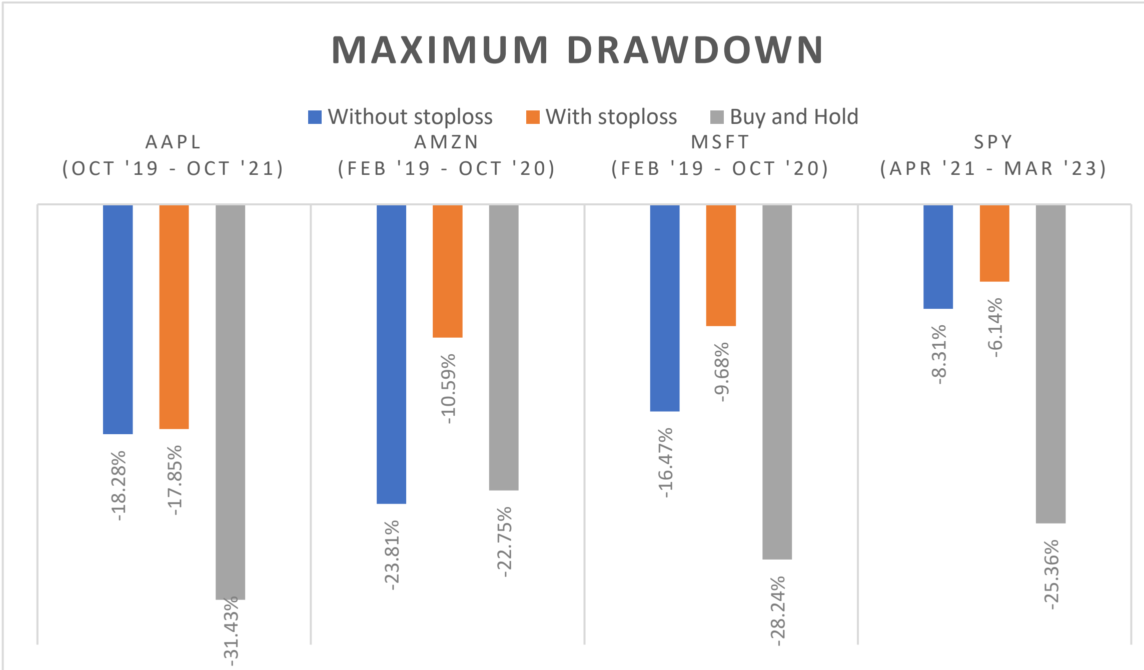

Figure 13: Equity Technical model maximum drawdown bar chart

2.4.2.2 Technical + Fundamental model

Compared to the technical model, the technical + fundamental model includes two new features which are the PCA results of the top 5 fundamental features. The results of adding in the fundamental features show that the model was able to generate a CAGR that outperforms the buy and hold CAGR for all three stocks. Furthermore, the Sharpe ratio and maximum drawdown also followed a similar trend as the CAGR results where both indicators in the stoploss model outperformed the buy and hold strategy. These results indicate that key fundamental data was able to contextualize the outlook of the company's performance per quarter, thus influencing the model's consideration to buy or sell.

For this model, the addition of stoploss has again proven to improve performance across all metrics. The stoploss has increased CAGR, Sharpe ratio, and decreased the maximum drawdown.

For more detail on the result of this model, refer to appendix 7.2 which includes trading balance for each equity and how it changes over time.

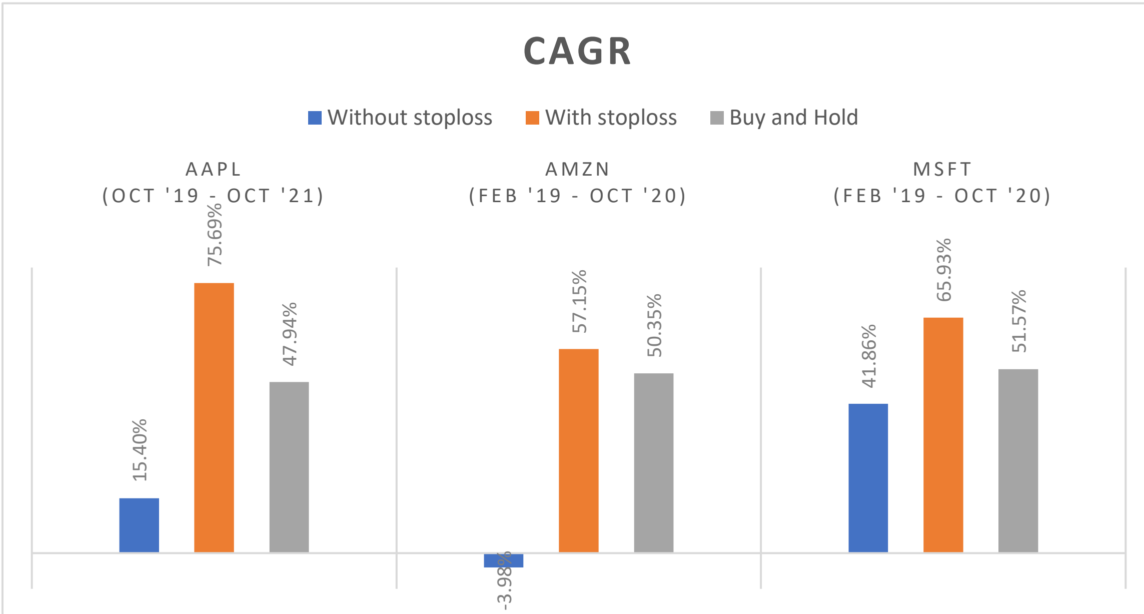

Figure 14: Equity Technical + Fundamental model CAGR bar chart

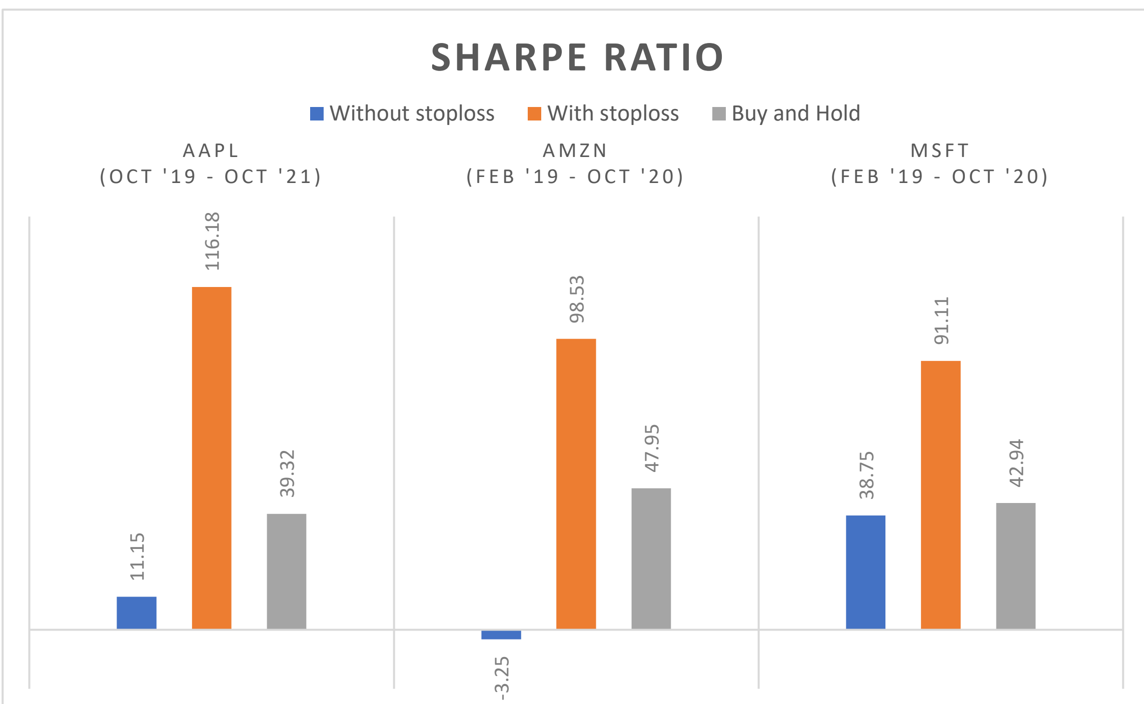

Figure 15: Equity Technical + Fundamental model Sharpe ratio bar chart

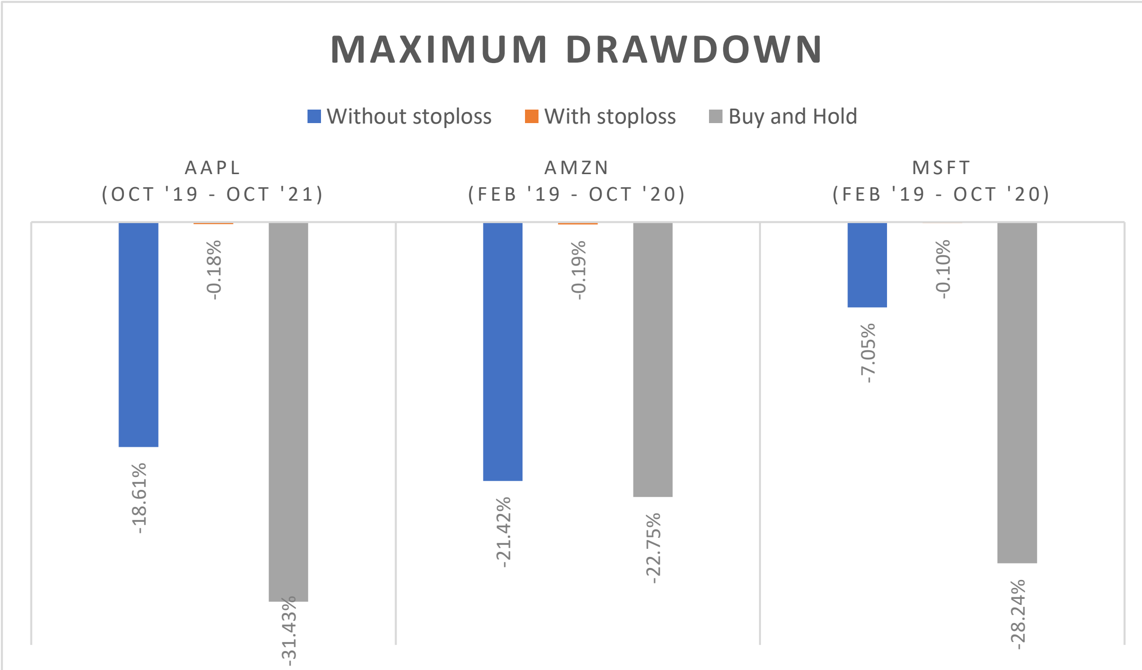

Figure 16: Equity Technical + Fundamental model maximum drawdown bar chart

**For the technical + fundamental model, the SPY stock was not included as the required fundamental data for indexes are unavailable.*

2.4.2.3 Technical + Macroeconomic model

Compared to the technical model, the technical + macroeconomic model includes 5 new features which are the PCA results of the nine macroeconomic features. The results of adding in the macroeconomic features showed that this model, although having the capabilities to be profitable, was not able to generate returns that exceed the simple buy-and-hold strategy. The Sharpe ratio results also followed similar trend as the CAGR results as seen in figure 18 below. This model, however, was able to reduce maximum drawdown across all securities by taking less trades, especially losing trades.

For this model, the addition of stoploss has again proven to improve performance across all metrics. The stoploss has increased CAGR, Sharpe ratio, and decrease the maximum drawdown.

For more detail on the result of this model, refer to appendix 7.3 which includes trading balance for each equity and how it changes over time.

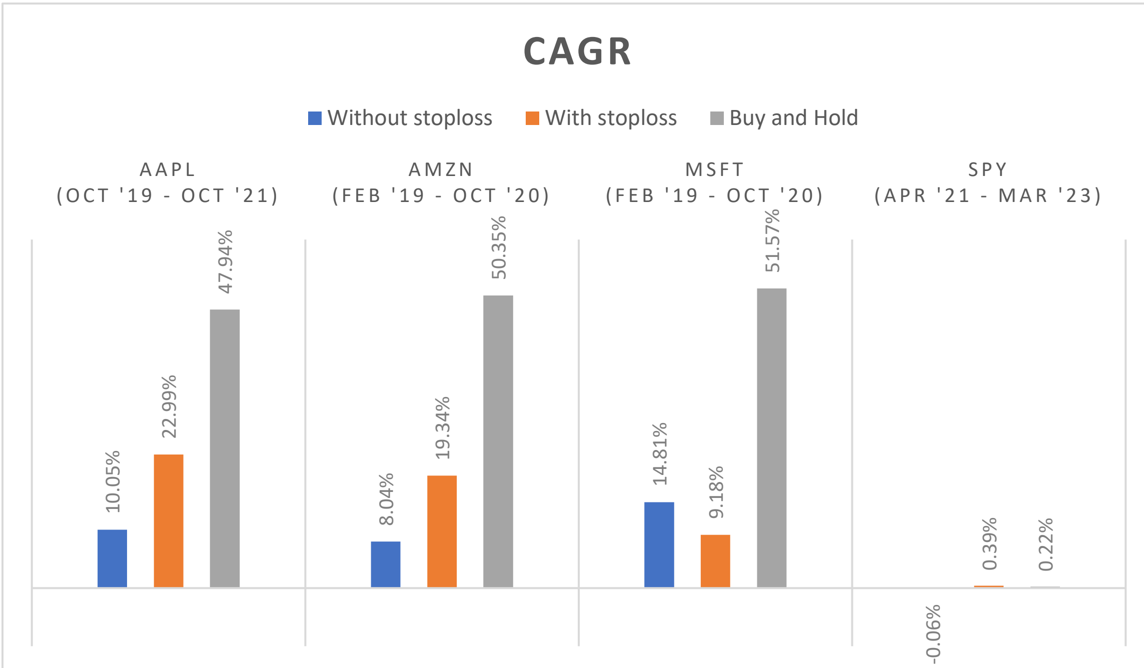

Figure 17: Equity Technical + Macroeconomic model CAGR bar chart

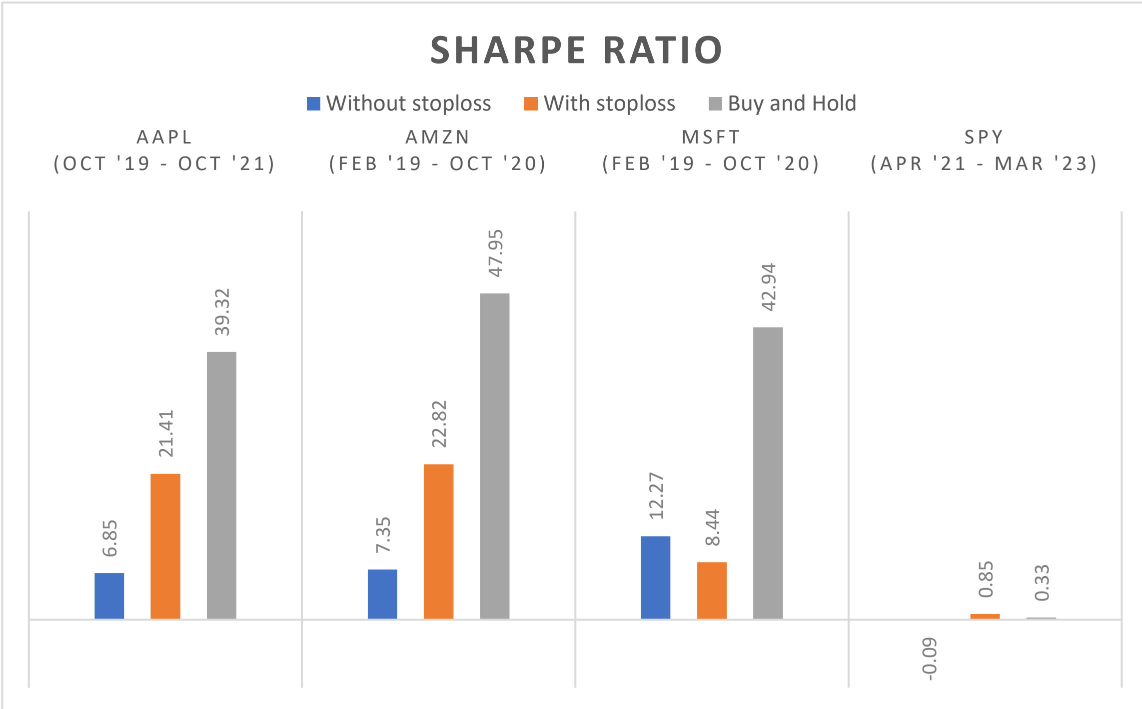

Figure 18: Equity Technical + Macroeconomic model Sharpe ratio bar chart

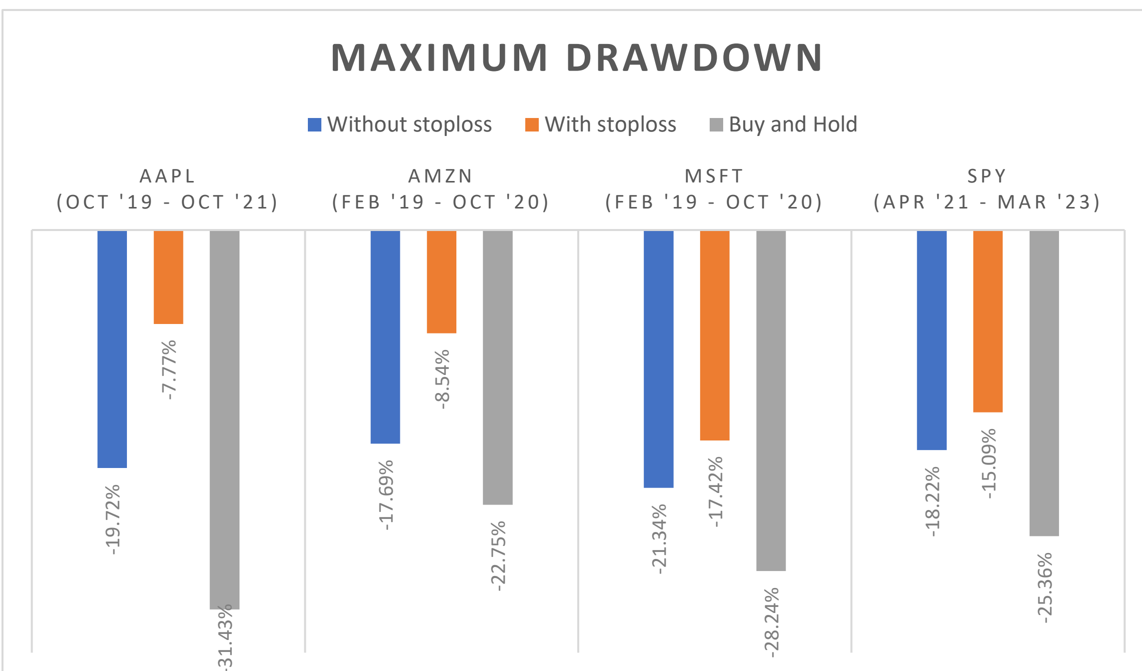

Figure 19: Technical + Macroeconomic model maximum drawdown bar chart

2.4.2.4 Technical + Value model

Compared to the technical model, the technical + value model includes 2 to 3 (stock dependent) added features which are the PCA results of the four value features. The results of adding in the value features show that the model was able to generate a CAGR that is profitable however underperforms when compared to the buy and hold CAGR for all three stocks. The same underperformance also holds true for the Sharpe Ratio as well. However, in the case of maximum drawdown both our stoploss and without stoploss outperformed with a lower drawdown. Nevertheless, there is no blanket generalizations in technical + value model because it depends on investor preference to prioritize impact on CAGR or drawdown, for example, of greater individual significance. Furthermore, when it comes to with or without stoploss, the model with stoploss outperforms with respect CAGR and Sharpe Ratio for AAPL, AMZN and MSFT. Finally, as expected having a stoploss would produce the least drawdown.

Overall, one could explain the mixed results from the addition of the value factor, which includes ratios such as price to earnings or price to sales, have a key repetitive component of price. The price component of the ratio is already captured by the technical factors. The ratio only implicitly adds new information pertaining to sales or earnings which is dampened by combining with price.

For more detail on the result of this model, refer to appendix 7.4 which includes trading balance for each equity and how it changes over time.

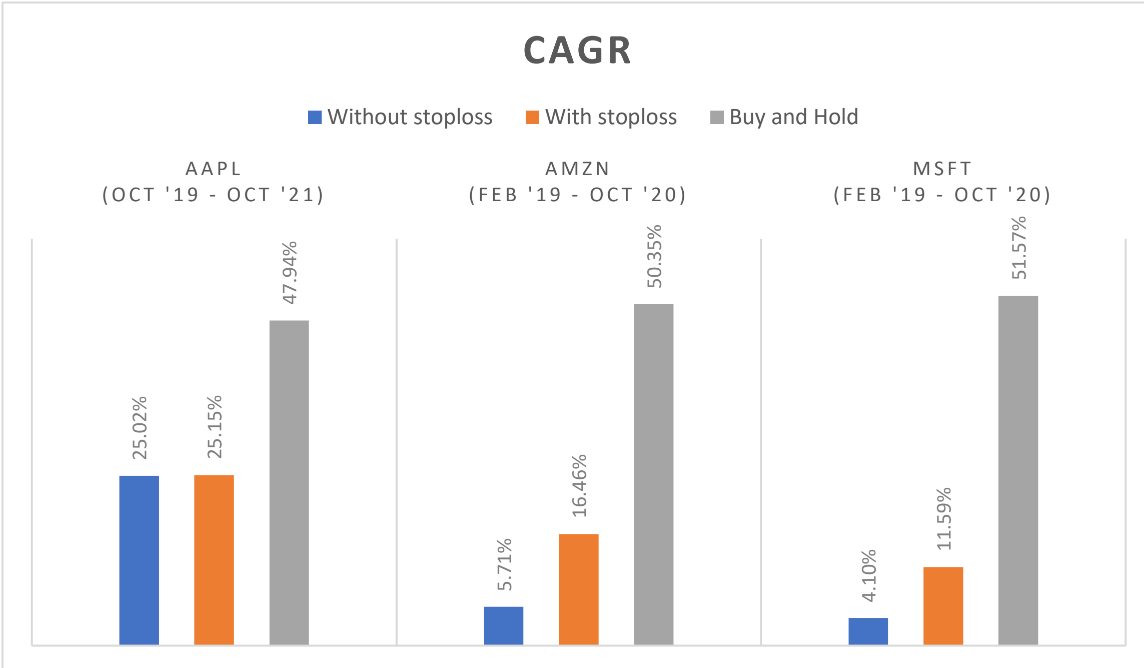

Figure 20: Equity Technical + Value model CAGR bar chart

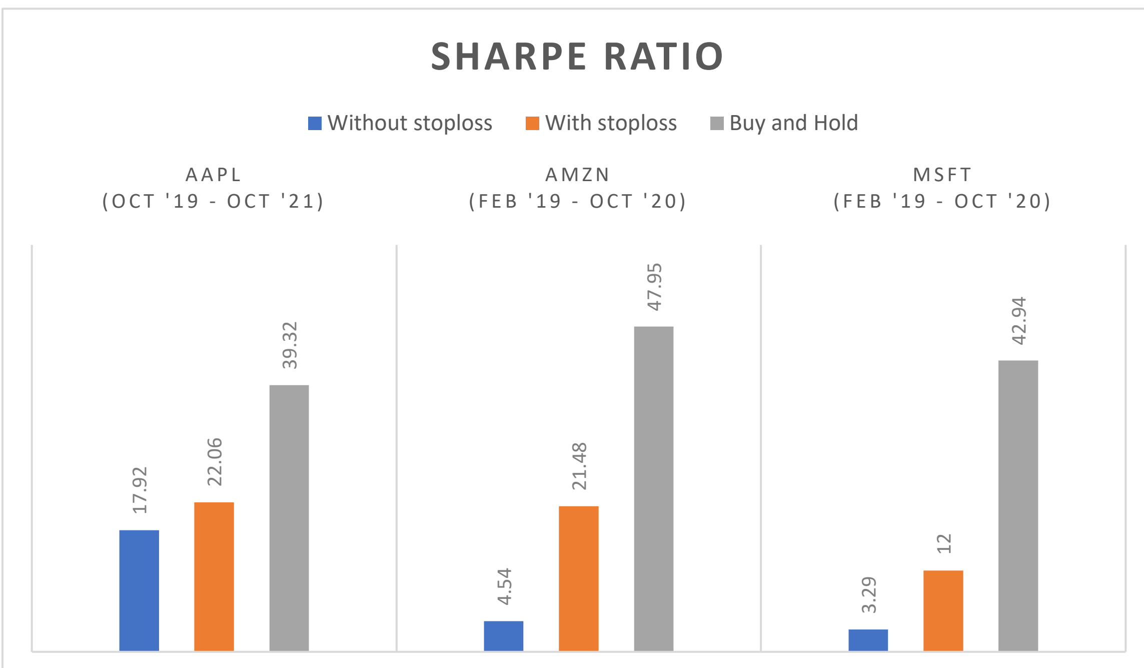

Figure 21: Equity Technical + Value model Sharpe ratio bar chart

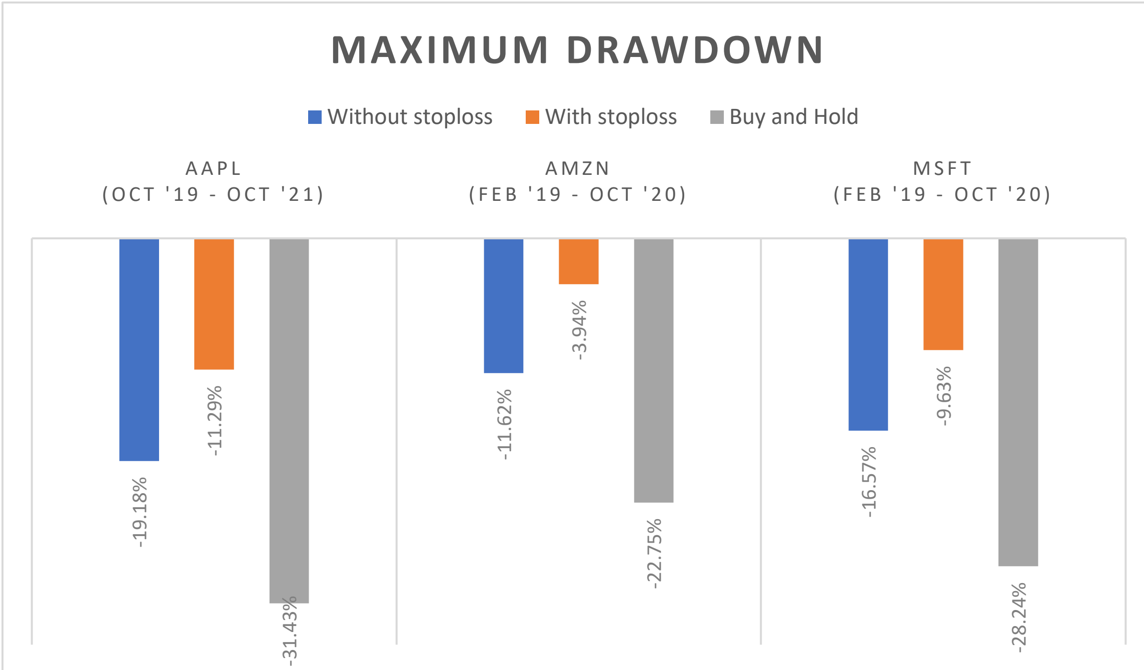

Figure 22: Equity Technical + Value model maximum drawdown bar chart

**For the technical + Value model, the SPY stock was not included as the required value data for the indexes are unavailable.*

2.4.2.5 Technical + Fundamental + Macroeconomic + Value model

Compared to the technical model, this combined model includes 16 new features which are the PCA results of the all the features. The results of adding in all factor pillars showed that this model, although having the capabilities to be profitable, was not able to generate returns that exceed the simple buy-and-hold strategy. While the Sharpe ratio results also followed similar trend as the CAGR results for AMZN and MSFT, the Sharpe ratio results for AAPL were better compared to the buy and hold strategy's Sharpe ratio. This model was also able to reduce maximum drawdown across all securities by taking less trades, especially losing trades.

For this model, the addition of stoploss has again proven to improve performance across all metrics. The stoploss has increased CAGR, Sharpe ratio, and decrease the maximum drawdown.

For more detail on the result of this model, refer to appendix 7.5 which includes trading balance for each equity and how it changes over time.

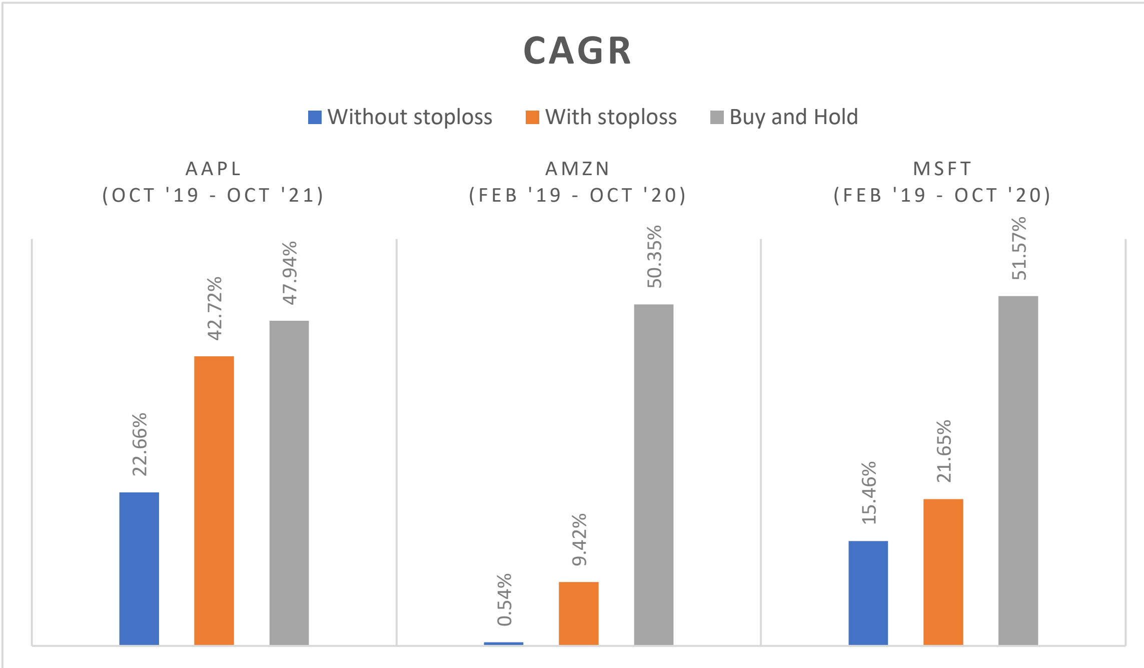

Figure 23: Equity Technical + Fundamental + Macroeconomic + Value model CAGR bar chart

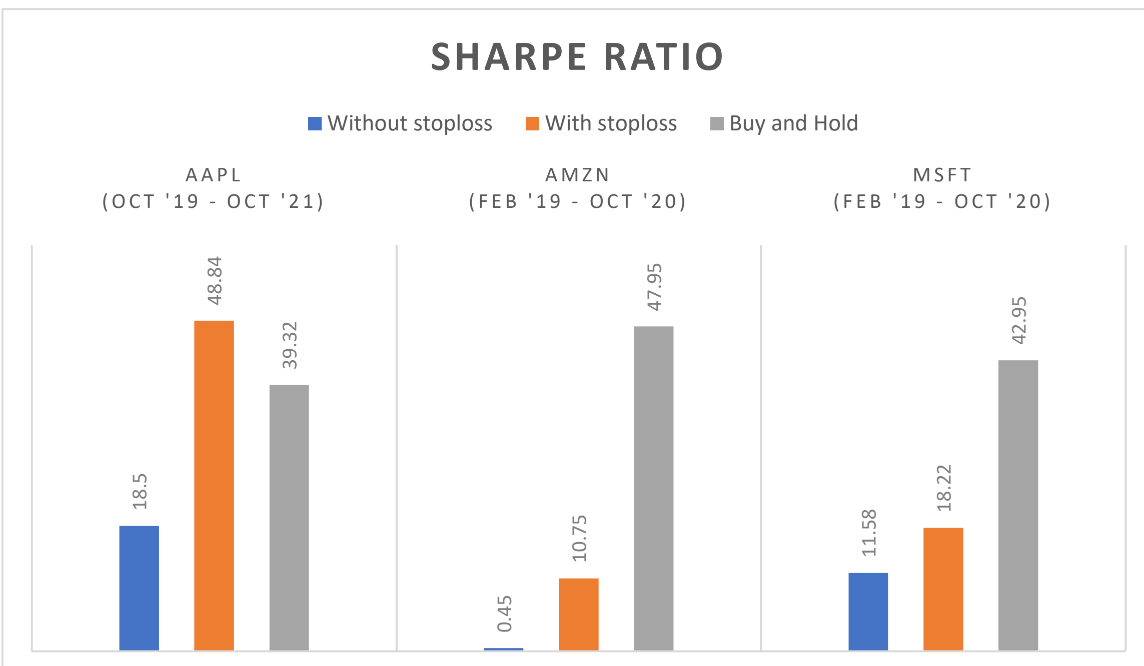

Figure 24: Equity Technical + Fundamental + Macroeconomic + Value model Sharpe ratio bar chart

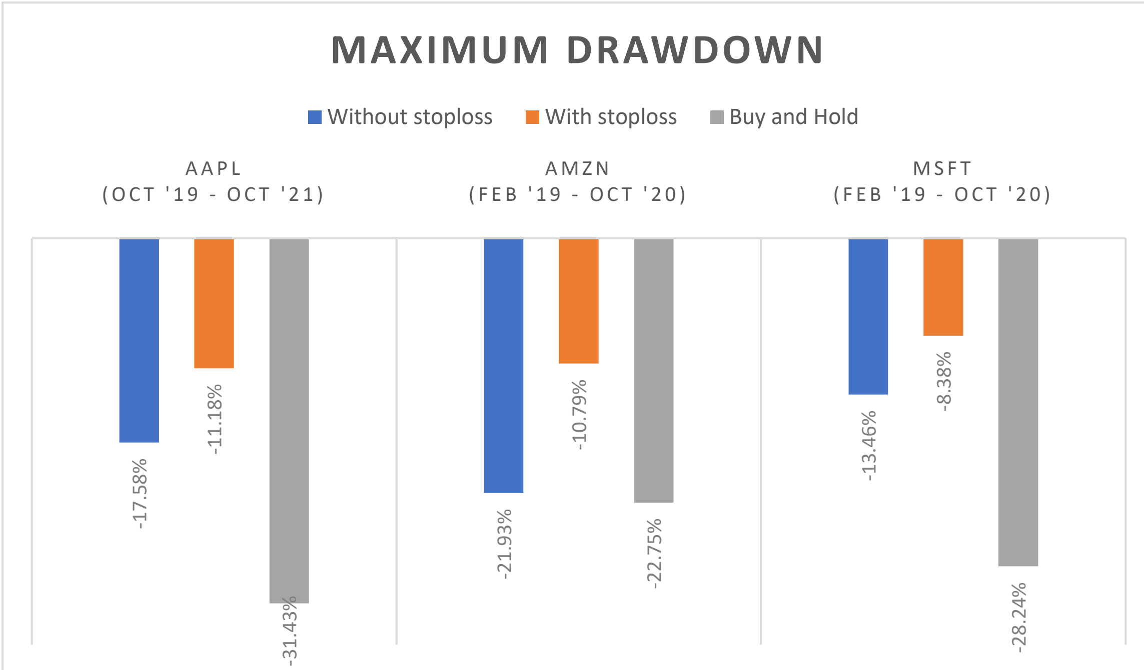

Figure 25: Equity Technical + Fundamental + Macroeconomic + Value model maximum drawdown bar chart

2.4.3 Evaluation of model trading performance on Foreign Exchange

The trading performance of our model when trained using FX technical data has shown significant trading performance. The CAGR across all seven currency pairs were positive and exceed the CAGR of buy-and-hold. For AUDUSD and USDJPY, the optimized stoploss level was so large that none of the trades were closed due to the stoploss, hence the CAGR for both with stoploss and without stoploss were the same. Overall, the addition of stoploss levels either improved or did not affect the CAGR except for USD/CHF. In this case, the optimized stoploss level that was determined from the validation data does not fit for the testing data. Still, USD/CHF's CAGR with and without stoploss were above the CAGR of buy-and-hold.

The Sharpe ratio also followed the same trend as the CAGR, as the Sharpe Ratios for all our models performed better than buy-and-hold. The addition of stoploss level should improve the Sharpe ratio since it prevents significant losses and thus, decreases the standard deviation of return. This can be seen for most of the currency pairs. For AUD/USD and USD/JPY, where the stoploss levels bring no impact towards the CAGR, it also did not affect the Sharpe ratio. On the other hand, USD/CHF's CAGR decreased with the addition of the stoploss level, and this can also be seen on its Sharpe ratio.

Since the trades only lasted for 24 hours and the model was able to predict the direction of the trades, the maximum drawdown for all currency pairs transformer model were significantly less than the buy-and-hold. The addition of stoploss level also prevented the trades to incur large losses. This is why the maximum drawdown for models with stoploss level was less than the ones without stoploss.

For more detail on the result of this model, refer to appendix 7.6 which includes trading balance for each equity and how it changes over time.

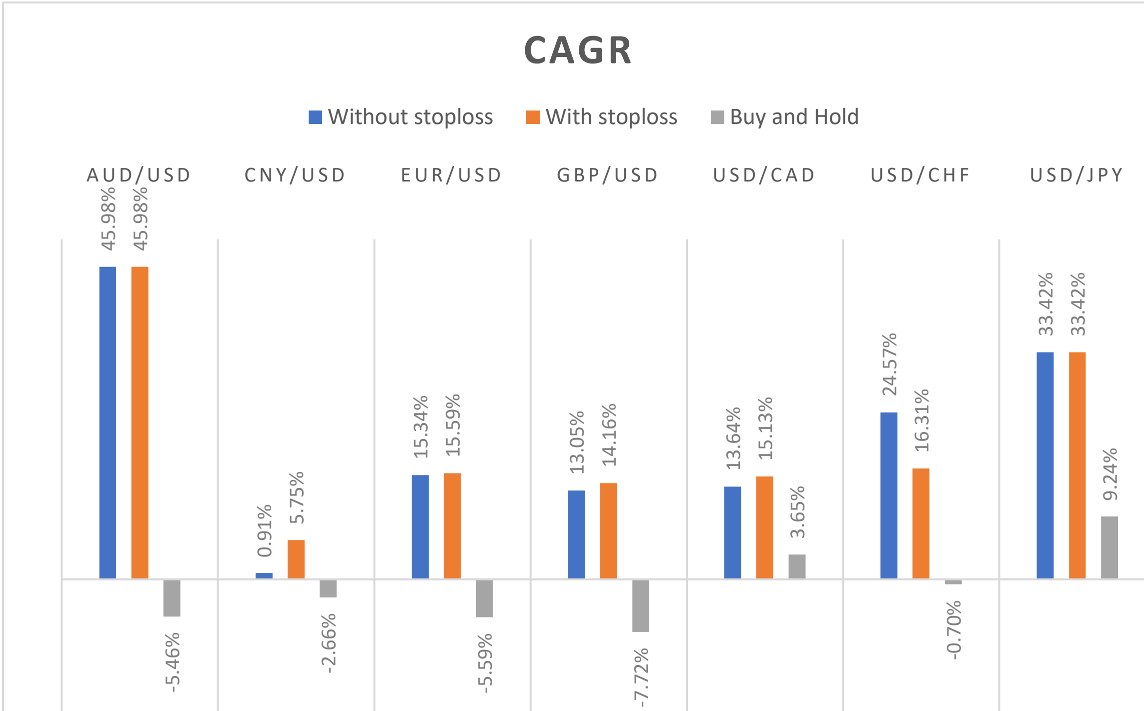

Figure 26: FX Technical Model CAGR Bar Chart (Test Period: Apr '21 - Mar '23)

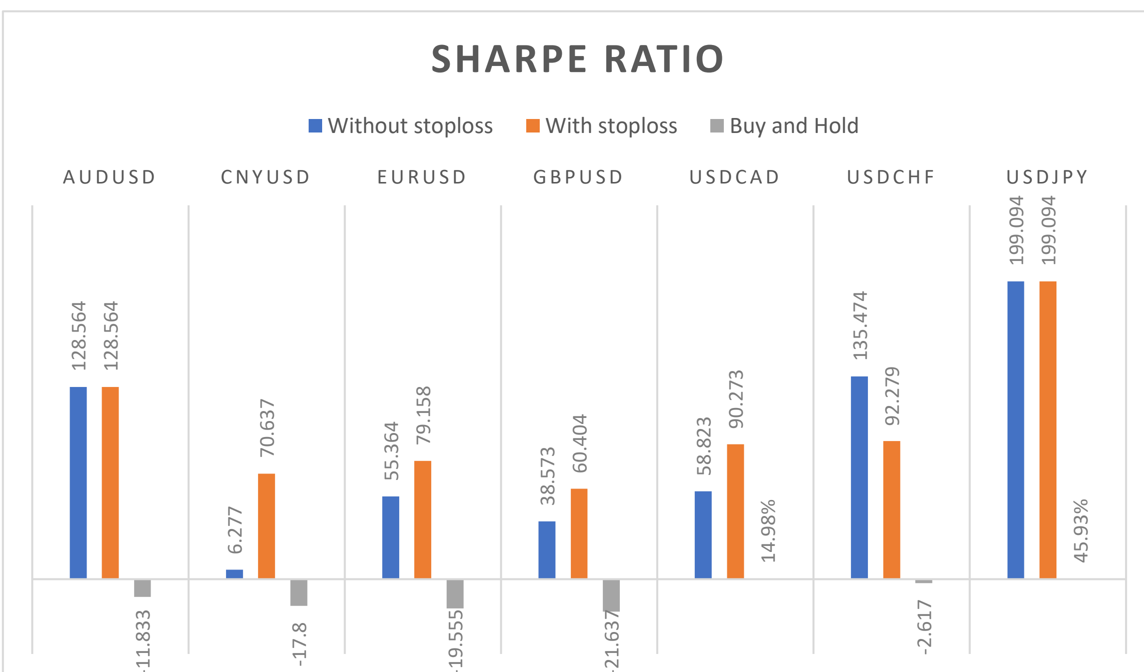

Figure 27: FX Technical Model Sharpe Ratio Bar Chart (Test Period: Apr '21 - Mar '23)

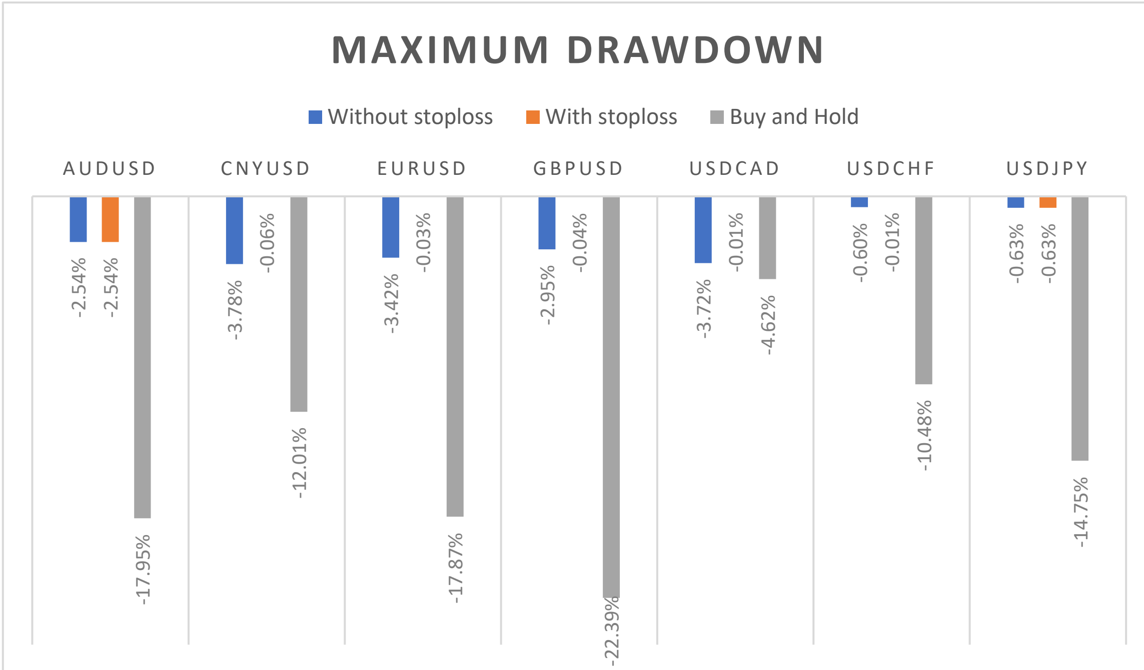

Figure 28: FX Technical Model Maximum Drawdown Bar Chart (Test Period: Apr '21 - Mar '23)

3 Discussion

3.1 Comparison between equity model performances

One of the key objectives of this project was to develop a profitable trading strategy using our transformer model. We have successfully achieved this objective as seen in the results of the technical plus fundamental model where the CAGR of this model even outperforms that of the buy and hold CAGR. While this model has succeeded in beating the benchmark of the buy and hold strategy, we are surprised to see the other pillars being unable to do so. Looking at the equity technical model, the results show that while technical factors such as the price are important in forecasting future returns, we may need to turn towards external factors for the model to beat the buy and hold. Although this model was able to produce a CAGR higher than

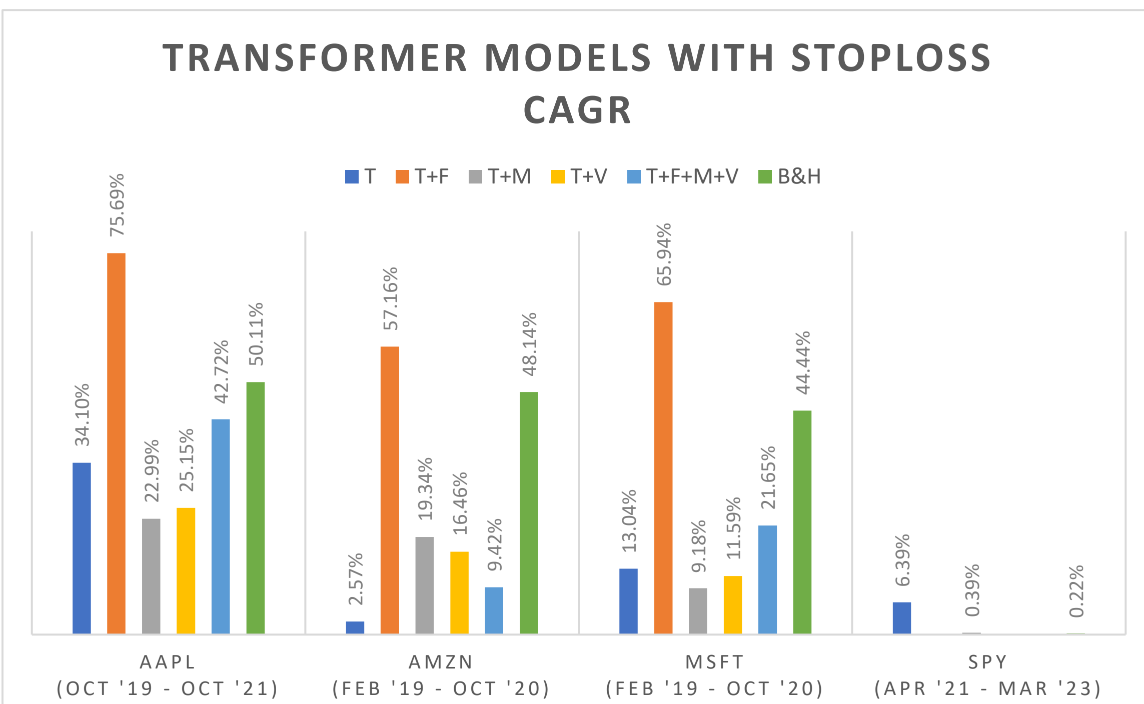

Figure 29: Transformer models with stoploss CAGR

buy-and-hold for SPY, this might be a special case due to the fact that SPY tracks S&P500 stock market. This means that the SPY data removes the idiosyncratic risk that is inherent for individual stocks, hence, the model might have found clearer patterns in price movement.

The technical plus fundamental model is by far the best performing equity model across the different stocks with respect to CAGR, Sharpe Ratio, and Drawdown. To recap, we used mock factor portfolio and feature importance to help identify fundamental features that correlate with highest return profitability. These are namely: total assets, total liabilities, net debt, intangible assets, and total non-current assets. The impact of adding these factors on top of the technical data is that they add new company specific intrinsic information that is uncorrelated to the stock pricing information captured from the technical but has a direct impact on the performance of the company and investor sentiment.

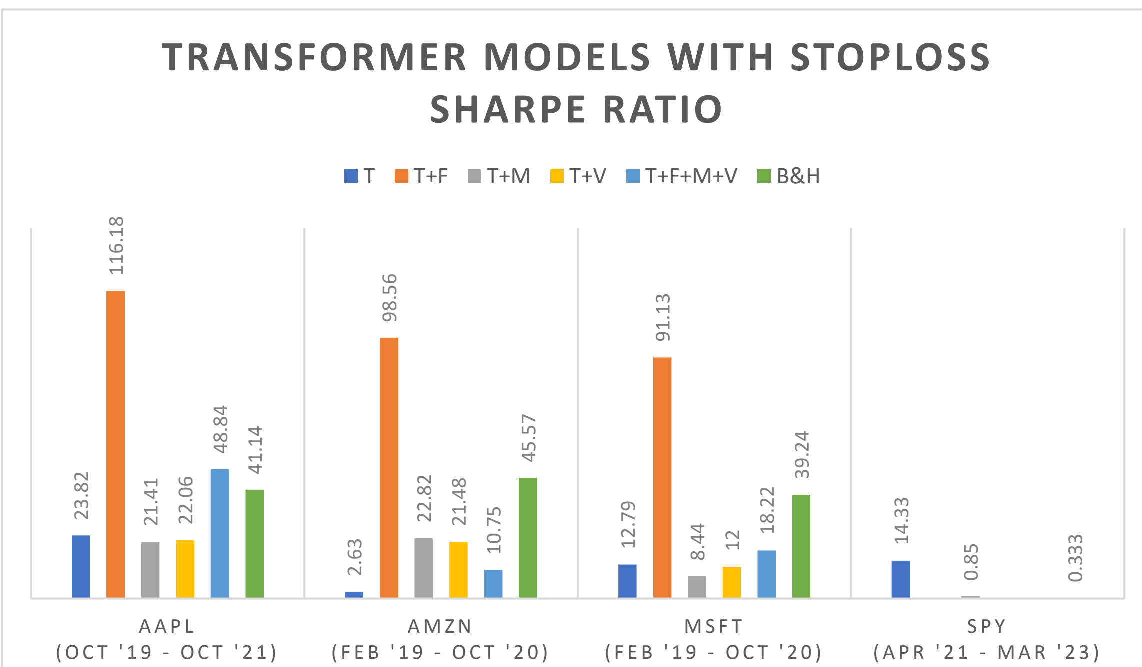

Figure 30: Transformer models with stoploss Sharpe ratio

Looking at the technical plus value factor model, we can see that it outperforms the technical only model when trading AMZN, however it still underperforms the buy-and-hold strategy for all securities. This indicates that the value factors are providing some new useful information compared to just the technical factors, however, it is still not enough for the model to gain an accurate understanding of price movements. It is likely that the value factors, which consists of ratios such as price to earnings or price to sales, dilute the new information regarding a firm's earnings and sales by being divided by redundant pricing information already captured in the technical pillars.

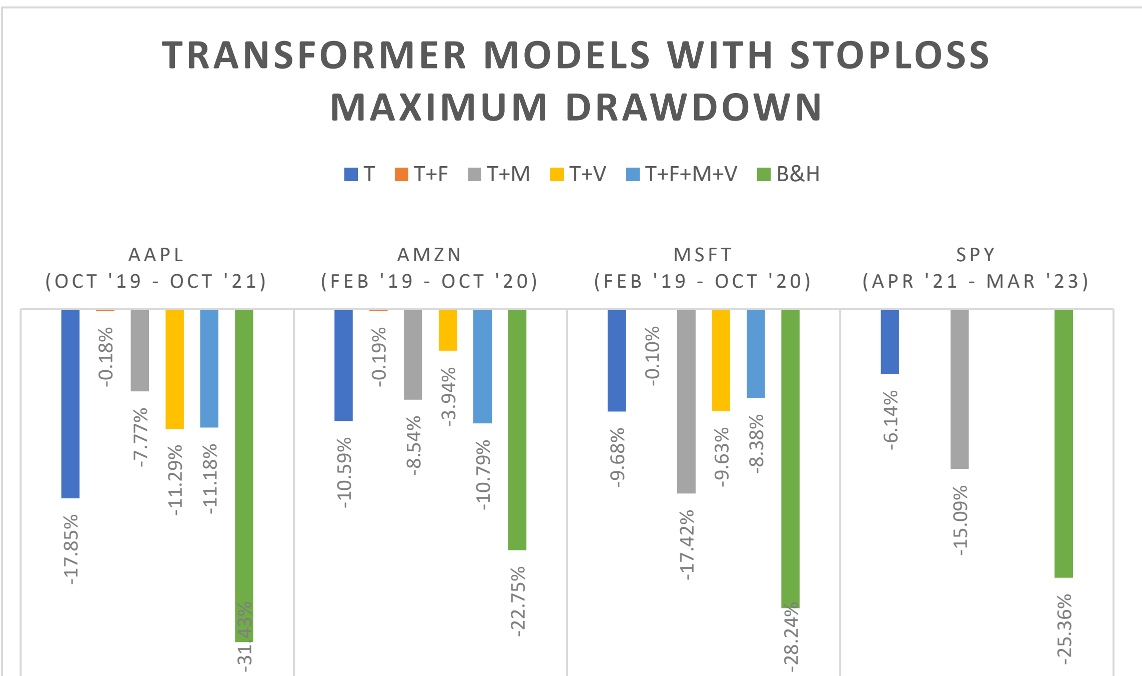

Figure 31: Transformer models with stoploss maximum drawdown

Moving on to the technical plus macroeconomic model, we believe that this model performed poorly against the buy and hold strategy mainly because the high number of macroeconomic features greatly increase the number of trainable parameters, however there isn't enough data for the model to sufficiently learn these parameters. There might also be low correlation between certain stocks and the macroeconomic factors. While macroeconomic factors have been proven to influence the direction of the stock's price, the influence of macroeconomic factors on each stock may vary as seen in our results. For example, macroeconomic factors were able to improve the CAGR for AMZN however not for AAPL and MSFT hence showing that macroeconomic factors greatly influencing AMZN but not necessarily for AAPL and MSFT. The macroeconomic factors were marginally able to beat the buy and hold strategy only for SPY, and this might be because the overall market performance takes macroeconomic factors into account, without the idiosyncratic movements associated with each single company.

Finally, for the model that uses all factors, we believe that the reason it wasn't able to perform well consistently was because the high number of input factors (20) added too many new parameters for the model to learn, and there wasn't sufficient data for the model to train. Despite that, the addition of new factors allowed it to beat the Technical only model while trading AAPL and MSFT.

3.2 Analysis of Equities vs FX

In comparing the performance of transformer models trained on technical data for trading FX and equities, it was observed that the FX transformer model exhibited significantly better results, as evidenced by its higher CAGR across currency pairs compared to the buy-and-hold strategy. We believe that there are two reasons for this. First is that, unlike Equity markets, FX markets are open 24/7. While trading equities, there is a risk of realizing a big loss (larger than the stoploss) because the opening price on a given day can be significantly lower than the previous closing price. This might be due to events that transpired the preceding night that our models cannot consider. 24/7 FX markets do not face this issue. Second, it is possible that technical data alone is able to encapsulate more of the trend of prices in FX, than it is able to do so for equities. This might be because each equity's prices are also subject to idiosyncratic risk from external company-specific factors.

It is important to note that even though our FX transformer model outperformed equity transformer models, it should be considered that FX trading differs from investing in equities. FX is characterized by stable prices, and price movements may create arbitrage opportunities which financial institutions may exploit. Conversely equities are meant to show the value of a company in which in an ideal case should always be on an uptrend. This is also supported by the observation that the maximum drawdown of currency pairs being lower in general than the maximum drawdown of equities which indicated that currency pairs are less volatile and more stable.

4 CONCLUSION

4.1 SUMMARY OF MAJOR TECHNICAL ACCOMPLISHMENTS

In our project, we were able to develop transformer-based models that could learn and follow the movement of the price changes. Using this capability, we were also able to formulate a trading strategy directly using the model output. We used technical data for our baseline model and systematically added data from multiple pillars such as fundamental, value, and macroeconomic to bring valuable insights for the model to utilize.

Based on the results shown in the evaluation section, we have two transformer-based models that consistently outperformed buy-and-hold in terms of CAGR, Sharpe ratio, and maximum drawdown, which are equity technical + fundamental model and FX technical model. The addition of fundamental data aided the model to contextualize the company performance for the equity technical + fundamental model. While for FX technical model, the model was able to outperform due to the stable nature of the currency pairs.

4.2 FUTURE WORK

While this project was able to achieve the objectives set in section 1.2, there is scope for improvement. The first major improvement would be to source more data. Due to data sourcing limitations, we were only able to operate using daily data, with only approximately 4000 datapoints for each model. The transformer model architecture has a significant number of parameters, almost 86,000 per encoder module and thus, more data would be required for the models like Technical + Macroeconomic and Technical + Fundamental + Value + Macroeconomic to achieve their full potential.

With more data, the model can also be further finetuned in future studies in two ways. First, a more complex input projection layer can be used. For example, multiple convolutional input projection layers can be used to take 60 days' data as input and perform learned durational feature extraction. We tried this in our project; however, the increased parameters could not be effectively trained with the limited amount of data. Finally, further studies could explore adding a transformer decoder module to the architecture, that can predict the future values for all factors for multiple days instead of just the closing price for the next day.

5 REFERENCES

[1] A. West, "New research reveals machine learning is maturing in finance," Refinitiv, 28 October 2020. [Online]. Available: https://www.refinitiv.com/perspectives/ai-digitalization/machine-learning-new-research-reveals-is-maturing-in-finance/. [Accessed 12 September 2022].

[2] "scalefocus," July 2022. [Online]. Available: https://www.scalefocus.com/blog/how-financial-institutions-can-leverage-the-power-of-machine-learning. [Accessed 12 September 2022].

[3] D. Conko, "Neural Networks: Forecasting Profits," Investopedia, 9 May 2022. [Online]. Available: https://www.investopedia.com/articles/trading/06/neuralnetworks.asp#:~:text=Neural%20networks%20do%20not%20make,using%20traditional%20technical%20analysis%20methods. [Accessed 12 Septmeber 2022].

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser and I. Polosukhin, "Attention Is All You Need," arXiv, 2017.

[5] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov and S. Zagoruyko, "End-to-End Object Detection with Transformers," *CoRR,* 2020.

[6] P. Agrawal, H. O. Bansal, A. R. Gautam, O. P. Mahela and B. Khan, "Transformer-based time series prediction of the maximum power point for solar photovoltaic cells," *Energy Science & Engineering,* vol. 10, no. 9, pp. 3397-2410, 2022.

[7] H. Hamoudi and M. A. Elseif, "Stock Market Prediction using CNN and LSTM," Stanford University, 2021.

[8] Jane Street, "Jane Street Market Prediction | Kaggle," Kaggle, 2021. [Online]. Available: https://www.kaggle.com/competitions/jane-street-market-prediction/overview. [Accessed 12 September 2022].

[9] Yahoo Finance, [Online]. Available: https://finance.yahoo.com/. [Accessed 12 September 2022].

[10] "Bloomberg Terminal," [Online]. Available: https://www.bloomberg.com/professional/solution/bloomberg-terminal/. [Accessed 12 September 2022].

[11] "EODHD APIs," [Online]. Available: https://eodhistoricaldata.com/. [Accessed 27 December 2022].

[12] Environmental Systems Research Institute, "How Extra trees classification and regression algorithm works," Environmental Systems Research Institute, [Online]. Available: https://pro.arcgis.com/en/pro-app/latest/tool-reference/geoai/how-extra-tree-classification-and-regression-works.htm. [Accessed 18 April 2023].

[13] Environmental Systems Research Institute, "How XGBoost algorithm works," Environmental Systems Research Institute, [Online]. Available: https://pro.arcgis.com/en/pro-app/latest/tool-reference/geoai/how-xgboost-works.htm. [Accessed 18 April 2023].

[14] NumPy, [Online]. Available: https://numpy.org/. [Accessed 12 September 2022].

[15] Pandas, [Online]. Available: https://pandas.pydata.org/. [Accessed 12 September 2022].

[16] Seaborn, [Online]. Available: https://seaborn.pydata.org/. [Accessed 12 September 2022].

[17] matplotlib, [Online]. Available: https://matplotlib.org/. [Accessed 12 September 2022].

[18] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller and O. Grisel, "{API} design for machine learning software: experiences from the scikit-learn," in *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, 2013.

[19] F. Chollet, *Keras,* 2015.

[20] S. M. Kazemi, R. Goel, S. Eghbali, J. Ramanan, J. Sahota, S. Thakur, S. Wu, C. Smyth, P. Poupart and M. Brubaker, "Time2Vec: Learning a Vector Representation of Time," Ahsanullah University of Science and Technology Dhaka, Bangladesh, 11 July 2019. [Online]. Available: https://arxiv.org/abs/1907.05321. [Accessed 20 January 2023].

[21] D. P. Kingma and J. L. Ba, "Adam: A Method for Stochastic Optmization," 2015. [Online]. Available: https://arxiv.org/pdf/1412.6980.pdf. [Accessed 25 January 2023].

[22] P. J. Huber, "Robust Estimation of a Location Parameter," *The Annals of Mathematical Statistics,* vol. 35, no. 1, pp. 73-101, 1964.

[23] Interactive Brokers, "Commissions | Interactive Brokers Hong Kong Limited," [Online]. Available: https://www.interactivebrokers.com.hk/en/index.php?f=1590. [Accessed 15 January 2023].

[24] Investing.com, [Online]. Available: https://www.investing.com/. [Accessed 2 January 2023].

[25] python, [Online]. Available: https://www.python.org/. [Accessed 12 September 2022].

[26] Visual Studio Code, [Online]. Available: https://code.visualstudio.com/. [Accessed 12 September 2022].

[27] Microsoft, [Online]. Available: https://www.office.com/. [Accessed 12 September 2022].

[28] Github, [Online]. Available: github.com. [Accessed 12 September 2022].

[30] Oracle, "What is Natural Language Processing (NLP)? | Oracle Hong Kong SAR, PRC," Oracle, [Online]. Available: https://www.oracle.com/hk/artificial-intelligence/what-is-natural-language-processing/#:~:text=Natural%20Language%20Processing%3F-,Natural%20Language%20Processing%20(NLP)%20Defined,natural%20language%20text%20or%20voice. [Accessed 12 September 2022].

[31] S. Goled, "Why Transformers Are Increasingly Becoming As Important As RNN And CNN?," Analytics India Mag, 17 March 2021. [Online]. Available: https://analyticsindiamag.com/why-transformers-are-increasingly-becoming-as-important-as-rnn-and-cnn/. [Accessed 12 September 2022].

[32] A. Lu, A. Parulekar and H. Xu, "Cluster-Based Statistical Arbitrage Strategy," 10 June 2018.

[33] J. Fernando and C. Potters, "Relative Strength Index (RSI) Indicator Explained With Formula," Investopedia, [Online]. Available: https://www.investopedia.com/terms/r/rsi.asp. [Accessed 25 January 2023].

[34] "Companies Market Cap," [Online]. Available: https://companiesmarketcap.com/usa/largest-companies-in-the-usa-by-market-cap/. [Accessed 10 February 2023].

[35] G. Siddhad, A. Gupta, D. P. Dogra and P. P. Roy, "Efficacy of Transformer Netwroks for Classification of Raw EEG Data," arXiv, 11 February 2022. [Online]. Available: https://arxiv.org/pdf/2202.05170.pdf. [Accessed 31 August 2022].

[36] B. Dolan, "MACD Indicator Explained, with Formula, Examples, and Limitations," Investopedia, [Online]. Available: https://www.investopedia.com/terms/m/macd.asp. [Accessed 31 August 2022].

[37] The Hong Kong University of Science and Technology Information Technology Services Centre, "X-GPU CLUSTER," [Online]. Available: https://itsc.hkust.edu.hk/services/academic-teaching-support/high-performance-computing/x-gpu-cluster. [Accessed 27 December 2022].

[38] Seeking Alpha, [Online]. Available: https://seekingalpha.com/. [Accessed 12 September 2022].

[39] Trading View, [Online]. Available: https://www.tradingview.com/. [Accessed 12 September 2022].

6 APPENDIX A

6.1 DISTRIBUTION OF WORK

Tasks	Srijan	Kalpa	Matthew	Nicholas
Formulate ideas	L	A	A	A
Perform the literature survey	A	L		
Prepare and source the data		A	L	A
Design the transformer model	L	A	A	A
Perform factor selection		A	L	
Develop the transformer model	L	A	A	A
Train the model	A	A	A	L
Testing	A	L	A	A
Do hyperparameter optimization	A	A	A	L
Develop the base technical model	A			L
Develop the model including the factor pillars	L	A	A	A
Develop the FX technical model			A	L
Evaluate the overall model performance	L	A	A	A
Write the reports	A	A	A	L
Make the video trailer	A	A	A	L

* L = Leader, A = Assistant

6.2 Gantt Chart

Project Schedule:

Tasks	**Jun**	**Jul**	**Aug**	**Sep**	**Oct**	**Nov**	**Dec**	**Jan**	**Feb**	**Mar**	**Apr**
Formulate ideas	■	■	■	■							
Perform the literature survey			■	■							
Prepare and source the data				■	■	■	■	■			
Design the transformer model					■	■	■	■			
Perform factor selection					■	■	■	■	■		
Develop the transformer model							■	■	■	■	
Train the model								■	■	■	
Testing									■	■	
Do hyperparameter optimization									■	■	
Develop the base technical model									■	■	
Develop the model including the factor pillars									■	■	
Develop the FX technical model									■	■	
Evaluate the overall model performance									■	■	■
Write the reports					■	■		■	■	■	■
Make the video trailer										■	■

6.3 HARDWARE

- Development PC: Windows 10/11 or MAC OS
- GPU: Google Collab, NVDIA GeForce 1660Ti

6.4 SOFTWARE

- Python [25]: Programming Language
- Visual Studio Code [26]: IDE
- Microsoft Office Tools (Excel, PowerPoint, Word) [27]: Documentation
- Github [28]: Code Repository
- Numpy [14], Pandas [15], Keras [19], Matplotlib [17], Seaborn [16], Scikit-learn [18]: Essential Libraries

6.5 MINUTES OF THE 1ST MEETING

Date: 14 June 2022
Time: 9 pm
Via: Zoom

Participants:

Prof. Rossiter
CHRISTANTO, Nicholas
DYCHENGBENG, Matthew
GOEL, Kalpa
SAXENA, Srijan

Meeting discussions:

- Creating a 5 steps framework on how we want to proceed with the project
  - Decide Asset Class and Corresponding Factor Pillar
  - Find Factors, Scrape Data and Analysing Correlations (Pre-processing, Analysis, Graphs)
  - Design and Implement Model
  - Backtesting
  - Evaluate and Repeat
- Deciding on two trading strategies to test
  - Pair trading with commodities as underlying assets
  - CNN-LSTM model where we include equity factor pillars as channels

Task distribution:

- Pair trading with commodities as underlying assets (Kalpa and Srijan)
- CNN-LSTM model (Matthew and Nicholas)

Next Meeting: 14 August 2022 (everyone)

6.6 Minutes of the 2nd Meeting

Date: 18 June 2022
Time: 9 pm
Via: Zoom

Participants:
GOEL, Kalpa
SAXENA, Srijan

Meeting discussions:

- Asset Class
    - Dependant on Data Availability
    - Data Sources
        - Monthly Data Commodities
        - Alternative Data
            - ECTs
            - Headlines
- Pair Trading
    - Analysing Spread between two Assets
    - Employing Mean Reversion or other techniques to estimate spread and seek alpha in variance
    - Commodity Futures & Equity?
        - A potential strategy
            - Find commodity futures and Equities that are highly correlated (>|95%|)
            - Train a multifactor ML model to estimate the appropriate equilibrium spread at a given time
                - Factor Pillars
                    - Macroeconomic Factors
                        - E.g. interest rate
                    - Equity Fundamentals
                    - Open Low Close High Prices
            - Employ mean reversion based on current spread and estimated equilibrium spread
            - Risk estimation:
                - Time to return to equilibrium (higher risk for HFT)
                - Risk of breaking equilibrium (higher risk for LFT)

Next meeting: 14 August 2022 (consolidate with everyone)

6.7 Minutes of the 3rd Meeting

Date: 14 August 2022
Time: 11 pm
Via: Zoom

Participants:
CHRISTANTO, Nicholas
DYCHENGBENG, Matthew
GOEL, Kalpa
SAXENA, Srijan

Meeting discussions:

- Instead of dividing the team into two groups, we decided to just focus on one strategy which is the exploration of transformer model in forecasting stock prices.
- Introducing the Transformer model
  - How is it better compared to other model when it comes to timeseries?
  - How it has been proven to be better in NLP and other domain which uses time series?

Research distribution

- Read and understand the transformer model
  - Be clear on how the model works
- In the case that we agree to go with this model:
  - Decide the universe of stocks that we are going to analyze

Next meeting: 19 August 2022

6.8 MINUTES OF THE 4TH MEETING

Date: 19 August 2022
Time: 9 pm
Via: Zoom

Participants:
CHRISTANTO, Nicholas
DYCHENGBENG, Matthew
GOEL, Kalpa
SAXENA, Srijan

Meeting discussions:

- Motivation on why the transformer was invented?
  - Before Transformer was RNN which has several problems
    - Vanishing gradient
    - Unable to do parallelization
- Why is it better than the other models (RNN, CNN, LSTM)?
  - Solve the vanishing gradient since all encoder h can
  - Parallelization
  - Understand positional context with positional context
  - Usage of attention simplifies which part to relate
  - Do sequential work like RNN and find patterns like CNN
- How it can be applied to trading.
  - Predicting stock returns
  - The first step is about predicting one stock
  - The next step is to create a portfolio which is readjusted
- What steps are needed to build/fine tune a transformer model.
  - PyTorch has encoder module
- Stock to trade
  - Start with one stock (blue chip): AAPL, IBM
  - Timeframe used: Daily
- how will the trading strategy be? (buy and hold until sell signal, can we short or should we have the shares first)

Research distribution

- Overview and objectives (Nicholas)
- Design and implementation (Srijan)
- Testing and evaluation (Matthew)
- Literature survey, project planning (Kalpa)

6.9 MINUTES OF THE 5TH MEETING

Date: 20 September 2022
Time: 9 pm
Via: Zoom

Participants:
CHRISTANTO, Nicholas
DYCHENGBENG, Matthew
GOEL, Kalpa
SAXENA, Srijan

Prof. Rossiter

Meeting discussions:

- Data Collection
    - Four pillars
        - Volatility
            - VIX
            - Derived from the price data
        - Technical (price data),
            - Open, High, Low, Close prices
            - Volume
        - Value
            - Multiples: P/E, B/M
        - Macroeconomic
            - Interest rate
    - Subjects
        - Blue chip stock
            - AAPL
            - BRK.A
            - KO
            - JNJ
    - Daily returns derived from price data
- Model development

    - Research on the backbone, input projector, and positional encoding

Research distribution

- Data collection: Matthew, Nicholas
- Model development: Srijan, Kalpa

Next meeting: 20 October 2022

6.10 Minutes of the 6th meeting

Date: 20 October 2022
Time: 9 pm
Via: Zoom

Participants:
CHRISTANTO, Nicholas
DYCHENGBENG, Matthew
GOEL, Kalpa
SAXENA, Srijan

Meeting discussions:

1. Begin Data collection
    a. Collected data from Bloomberg terminals
    b. Focused on four main stocks: Apple, Coca Cola, Berkshire Hathaway, and Johnson & Johnson
    c. Collected the following data indicators:
        i. volatility (VIX)
        ii. technical data (Open, High, Low, Close, Volume)
        iii. value (P/E and P/B indicators)
        iv. macro (interest rate)
2. Began data preparation and exploratory data analysis
    a. Clean and merge data
    b. Graph and view overall data statistics for each indicator
3. Continue to research the transformer model
    a. Learned that we need to collect more data (quantity) with higher frequency (hourly instead of daily)
4. Positional Encoding:
    a. Trigonometric model that enables unique identification of an input data point normalized within a 0 to 1 range.
    b. Incorporates both positional and object information into the encoded data.

$$P(k, 2i) = \sin\left(\frac{k}{n^{2i/d}}\right)$$

$$P(k, 2i+1) = \cos\left(\frac{k}{n^{2i/d}}\right)$$

i.

5. Baseline Model: (Diagram at the last page)
    a. Input: Technical Data (HLOC, V)
    b. Backbone: Feedforward Layer for Projection and a Low Dimension Feature Space + Trig-sine positional encoding
    c. 2x Encoder Modules
    d. 2x Decoder Modules (with inputs shifted right)
    e. Binary Classification: Linear Layer + Sigmoid. Single output: P(daily return > 0)

Research distribution

- Continue to perform exploratory data analysis on new data to be collected (Matthew and Nicholas)
    - data to be collected are on more factors and higher frequency. Utilize the requested APIs if approved for more efficient data collection.
- Continue to research on building the transformer model (Kalpa and Srijan)
- Concatenate the database into a structured dataset to be fed into the baseline model. (Matthew and Nicholas)
- Start model implementation of the baseline and the positional encoding. (Srijan and Kalpa

Next meeting: 3 November 2022

6.11 MINUTES OF THE 7TH MEETING

Date: 3 November 2022
Time: 9 pm
Via: Zoom

Participants:
CHRISTANTO, Nicholas
DYCHENGBENG, Matthew
GOEL, Kalpa
SAXENA, Srijan

Meeting discussions:

1. Data collection
    a. Purchased API subscription
    b. Begin Data collection of other S&P 500 stocks
    c. Changed data collection target to hourly data to allow more stocks
2. Data preparation and exploratory data analysis
    a. Research how to detail with outliers
        i. Test Random Forest Methods: MISS, MICE
        ii. Test KNN Imputation
    b. Research Outlier detection methods
3. Continue to research the transformer model
    a. Learned that we need to collect more data (quantity) with higher frequency (hourly instead of daily)

Research Distribution:

1. Implement outlier detection method (PCA or isolated forest) (Srijan)
2. Research and design manual and learned feature selection (Srijan)
3. Collect more fundamental data from https://eodhistoricaldata.com/pricing (50USD) for two months (Nicholas)
    a. general information (i.e. Market Cap)
    b. Numbers for valuation (Price/Sales)
    c. Share statistics
    d. Tehcnical indicators (Beta, 50 day moving average)
    e. Splits and dividends
    f. Financial reports (quarterly)
    g. Government bonds (Macroeconomic)
4. Finish Development of v0.0.0 model and test it with technical data only. (Matthew and Kalpa)

Next meeting: After exam

6.12 Minutes of the 8th Meeting

Date: 14 January 2023
Time: 9 pm
Via: Zoom

Participants:
CHRISTANTO, Nicholas
DYCHENGBENG, Matthew
GOEL, Kalpa
SAXENA, Srijan

Meeting Discussion:

- Data
  - Imputation
  - Outlier Detection
  - Factor Selection
    - Raw Data
      - Technical (hourly)
      - Macro
        - Government bonds (https://eodhistoricaldata.com/financial-apis/macroeconomic-data-api/)
      - Value
        - general information
        - Numbers for valuation
        - Share statistics
        - Tehcnical indicators
        - Splits and dividends
        - Financial reports (quarterly)
      - Momentum
        - Moving Averages
      - Quality
      - Value
        -
    - Manual Feature Selection
      - Fourier Transform, Wavelet
      - Ratios
      - ….
      - Low VAriance
      - Highly Corr Features
      - Low Corr Target Variable
      - Univariate
    - Learned Featured Selection
      - ExtraTree Regressor's top factors
      - L1 Regularization (LASSO Linear Regression)
      - Baruta

Next Meeting: 1 February 2023

6.13 MINUTES OF THE 9TH MEETING

Date: 1 February 2023
Time: 9pm

Via: Zoom

Participants:

DYCHENGBENG Matthew

SAXENA Srijan

CHRISTANTO Nicholas

GOEL Kalpa

Meeting Discussion:

- Preparatory python file
    - Execution issue
    - Adding lambda functionality
- Shows transformer model result on AAPL
    - Trained using different sequence length = 10, 20, 128
    - Translated the normalized predicted output into next day closed price
    - Did linear regression to see the t-stat and correlation based on the percentage change of price
    - Performed f1 metrics with confusion matrix:
        - F1 score: `0.9074355083459787`

Next Meeting: 2 March 2023.

6.14 MINUTES OF THE 10TH MEETING

Date: 2 March 2023
Time: 9 pm
Via: Zoom

Participants:
CHRISTANTO, Nicholas
DYCHENGBENG, Matthew
GOEL, Kalpa
SAXENA, Srijan

Meeting discussions:

Models

- Regression
    - Close Price Prediction Model
        - AAPL Loss: 0.0002
        - SPY Loss: 0.0009
    - Gmean prediction (rolling: 3)
        -
        - AAPL Loss: 0.0004
        - SPY Loss: 0.0008
- Classification (good f1-score around 60% according to literature)
    - AAPL f-1: 55%

Trading Strategy

- Our model provides us with a daily prediction of whether the expected return for the next day is positive or negative. Based on that have 3 strategies
    - A: Holding the stock as long as the prediction is positive and buying t-bills when it's negative. Low risk.
    - B: Holding the stock as long as the prediction is positive and shorting the stock when it's negative. Low risk.
    - C: Holding the stock as long as the prediction is positive and shorting the stock when it's negative. Monitoring price live and closing position if loss exceeds a predefined stop loss.
- Formula for GMEAN

Simulation Results on strategy C:

- Close Price Model
    - SPY - 0.1%
    - SPY - 10%.
        - 746%, 103%
- Gmean-3 Model
    - AAPL - 0.1% stop loss; 39% CAGR,
    - AAPL -0.5%
    - SPY - 0.1%

    - AMZN 0.1 percent

Factor Selection

- Added Pillars:
    - Treasury Bills
    - Corporate Bonds and Spreads
    - Exchange rate of U.S. dollar
    - Commodities
    - Data from other markets
- Fundamental Data factor selection using Mock Factor Portfolios

Planned Improvements

- CNN module for higher level features and duration extraction
- Training classification with the added features + feature extraction

Final Deliverable

- Paper comparing our performance vs holding and other benchmark models on same strategy
- Analysis based on returns, Sharpe and CEQ, Drawdown (MDD)?

Feedback

- Stop Loss, Slippage, Inability to create short position
- Benchmark might be arbitrary, focus
- Effect of adding each pillar

Next Meeting: 17 March 2023

6.15 MINUTES OF THE 11TH MEETING

Date: 17 March 2023
Time: 10:05am
Via: Zoom

Participants:
CHRISTANTO, Nicholas
DYCHENGBENG, Matthew
GOEL, Kalpa
SAXENA, Srijan
Prof. Rossiter

Meeting discussions:

- Shared preprocessing techniques
  - Simple mean
  - GMean
- Shared stock universe
  - Equity
  - FX
- Shared Trading Strategies
  - Close-close
  - Open-close
  - Shared testing results of the code
- Shared Model Results
  - Technical Equity Model
  - Technical FX Model
    - Concerns on validity of results
    - Professor recommends that we test the model.
      - Random discrete data
      - Known function
    - Evaluate why FX performs so well
  - Tech + Macro
    - Outperforms SPY
    - Stop loss helps
  - Tech + Fundamental
    - Sometimes outperforms B&H
- To do:
  - Test Model with dummy data
  - Implement stoploss for all models.
  - Improve Fundamental Model
  - Trading Hyperparameter Opt

Next Meeting: 1st April 2023

6.16 MINUTES OF THE 12TH MEETING

Date: 1st April 2023
Time: 4pm

Venue: HKUST Learning Commons

Participants:

DYCHENGBENG Matthew

SAXENA Srijan

CHRISTANTO Nicholas

GOEL Kalpa

Meeting Discussion:

- Make a new model that combines all pillars models (T+F+V+M) model
- Standardize output across all models
- Explore use of technical model on FX market
- Research possible FX indexes (i.e. DXY)
- Research possibility of investing cash being held
- Discuss FYP report structure
    - Methodology
        - Design
        - Implementation
            - Data sourcing
            - Factor selection
            - Preprocessing
                - Gmean
                - Scaling
                - 80:10:10 data split
            - Model design and training
                - Model architecture
                - Loss function
                - Classification
                - Regression
                - Optimizer
                - Learning rate scheduling
            - Trading strategy
                - Stop loss
                - Long and short
                - Accounting for market friction
                - Evaluation metrics
        - Testing
            - Data testing
            - Pre-processing testing
            - Model testing

            - Evaluation
                - Discuss evaluation metrics
                - Equity
                    - Technical model
                    - Technical + Fundamental
                    - Technical + Value
                    - Technical + Macroeconomic
                    - Technical + Fundamental + Value + Macroeconomic
                - FX
                    - Technical model
        - Discussion
            - Comparison between models
            - Objectives met
            - Limitations

Next Meeting: 8th April 2023.

6.17 MINUTES OF THE 13TH MEETING

Date: 8th April 2023
Time: 4pm

Venue: Zoom

Participants:

DYCHENGBENG Matthew

SAXENA Srijan

CHRISTANTO Nicholas

GOEL Kalpa

Meeting Discussion:

- Waive out the scope of using domestic interest rate in FX model
- Finalize FX model results
- Finalize combined pillar equity model (T + F + M + V) model results
- Build a model testing plan and evaluation metrics
- Devising an action plan for writing the draft final report

Action items:

- Implementing the testing regime
- Start writing the draft final report

Next Meeting: 16th April 2023

6.18 MINUTES OF THE 14TH MEETING

Date: 16th April 2023
Time: 4pm

Venue: HKUST Learning Commons

Participants:

DYCHENGBENG Matthew

SAXENA Srijan

CHRISTANTO Nicholas

GOEL Kalpa

Meeting Discussion:

- Evaluate multiple results on the testing being conducted
- Ensure any unexpected discrepancies between model results and testing is addressed
- Discuss and write Section 2.4 - Model Evaluation and Section 3 - Discussion together
- Finalize all the remaining report sections and ensure coherence between the section to give a consistent narrative
- Send the draft report to supervisor for feedback

Next meeting: 18th April 2023

6.19 MINUTES OF THE 15TH MEETING

Date: 18th April 2023
Time: 10pm

Venue: Zoom

Participants:

DYCHENGBENG Matthew

SAXENA Srijan

CHRISTANTO Nicholas

GOEL Kalpa

Meeting Discussion:

- Collate feedback from supervisor on the draft final report
- Make relevant formatting and content amendments to enhance readability and coherence of the report
- Complete the report for submission
- Discuss code submission process, file architecture, and relevant supporting documentation such as README files to be attached

**** Other meetings did not have digitized notes. Quite a few meetings were face-to-face where we worked together, and discussions were recorded on the white board. Meanwhile, some of the meetings with Professor only involved us sharing our updates without any notes.**

7 APPENDIX B: TRADING STRATEGY RESULT

7.1 EQUITY TECHNICAL MODEL RESULT

Table 2 Equity Technical model result

Stock Name	Evaluation Metrics	Without stoploss	With stoploss	Buy and hold
AAPL	Win rate	65.38%	65.38%	52.99%
	CAGR	33.67%	34.10%	47.94%
	Sharpe Ratio	23.32	23.82	39.32
	Max. Drawdown	-18.28%	-17.85%	-31.43%
	Trade turnover	5.69 days	5.69 days	1.46 days
AMZN	Win rate	50.49%	25.74%	54.88%
	CAGR	-13.26%	2.57%	50.35%
	Sharpe Ratio	-9.72	2.63	47.95
	Max. Drawdown	-23.81%	-10.59%	-22.75%
	Trade turnover	5.673 days	5.673 days	1.44 days
MSFT	Win rate	50.00%	27.27%	58.58%
	CAGR	12.18%	13.04%	51.57%
	Sharpe Ratio	9.72	12.79	42.94
	Max. Drawdown	-16.47%	-9.68%	-28.24%
	Trade turnover	5.22 days	5.22 days	1.44 days
SPY	Win rate	53.33%	25.18%	50.213
	CAGR	4.26%	6.39%	0.223%
	Sharpe Ratio	6.41	14.33	0.333
	Max. Drawdown	-8.31%	-6.14%	-25.361%
	Trade turnover	4.837 days	4.837 days	1.46 days

Figure 32: AAPL Trading Balance: Technical Model

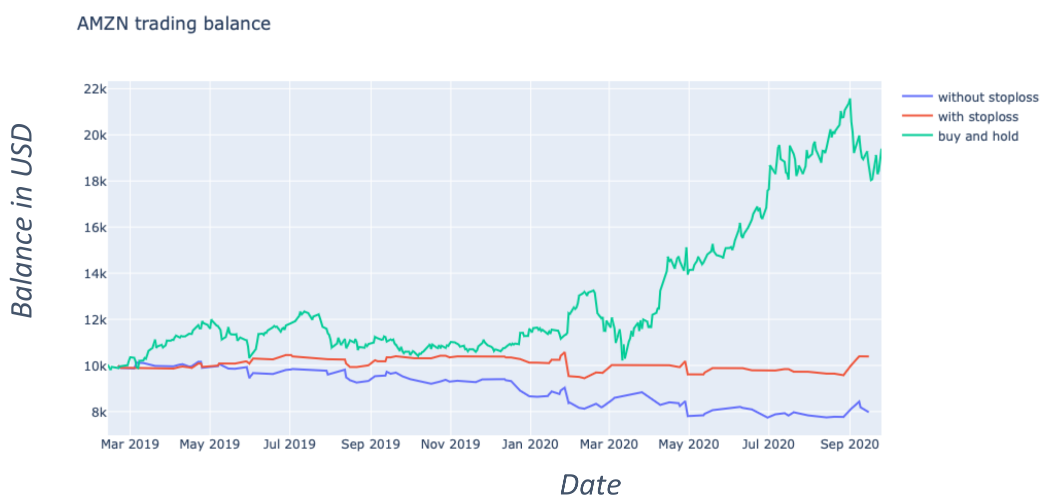

Figure 33: AMZN Trading Balance: Technical Model

Figure 34: MSFT Trading Balance: Technical Model

Figure 35: SPY Trading Balance: Technical Model

7.2 Equity Technical + Fundamental model trading result

Table 3 Equity Technical + Fundamental model trading result

Stock Name	Evaluation Metrics	Without stoploss	With stoploss	Buy and hold
AAPL	Win rate	55.32%	34.75%	52.99%
	CAGR	15.40%	75.69%	47.94%
	Sharpe Ratio	11.15	116.18	39.32
	Max. Drawdown	-18.61%	-0.18%	-31.43%
	Trade turnover	4.37 days	4.37 days	1.46 days
AMZN	Win rate	46.59%	23.30%	54.88%
	CAGR	-3.98%	57.16%	50.35%
	Sharpe Ratio	-3.25	98.56	47.95
	Max. Drawdown	-21.42%	-0.19%	-22.75%
	Trade turnover	3.31 days	3.31 days	1.44 days
MSFT	Win rate	61.60%	37.60%	58.58%
	CAGR	41.87%	65.94%	51.57%
	Sharpe Ratio	38.75	91.13	42.94
	Max. Drawdown	-7.05%	-0.099%	-28.24%
	Trade turnover	4.64 days	4.64 days	1.44 days

Figure 36: AAPL Trading Balance: Technical + Fundamental Model

Figure 37: AMZN Trading Balance: Technical + Fundamental Model

Figure 38: MSFT Trading Balance: Technical + Fundamental Model

7.3 EQUITY TECHNICAL + MACROECONOMIC MODEL TRADING RESULT

Table 4 Equity Technical + Macroeconomic model trading result

Stock Name	Evaluation Metrics	Without stoploss	With stoploss	Buy and hold
AAPL	Win rate	50.70%	29.58%	52.99%
	CAGR	10.05%	22.99%	47.94%
	Sharpe Ratio	6.85	21.41	39.32
	Max. Drawdown	-19.72%	-7.77%	-31.43%
	Trade turnover	5.69 days	5.69 days	1.46 days
AMZN	Win rate	52.89%	30.58%	54.88%
	CAGR	8.04%	19.34%	50.35%
	Sharpe Ratio	7.35	22.82	47.95
	Max. Drawdown	-17.69%	-8.54%	-22.75%
	Trade turnover	5.673 days	5.673 days	1.44 days
MSFT	Win rate	51.20%	21.08%	58.58%
	CAGR	14.81%	9.18%	51.57%
	Sharpe Ratio	12.27	8.44	42.94
	Max. Drawdown	-21.34%	-17.42%	-28.24%
	Trade turnover	3.86 days	3.86 days	1.44 days
SPY	Win rate	55.28%	23.60%	50.213
	CAGR	-0.06%	0.39%	0.22%
	Sharpe Ratio	-0.09	0.85	0.33
	Max. Drawdown	-18.22%	-15.09%	-25.36%
	Trade turnover	4.08 days	4.08 days	1.46 days

Figure 39: AAPL Trading Balance: Technical + Macroeconomic Model

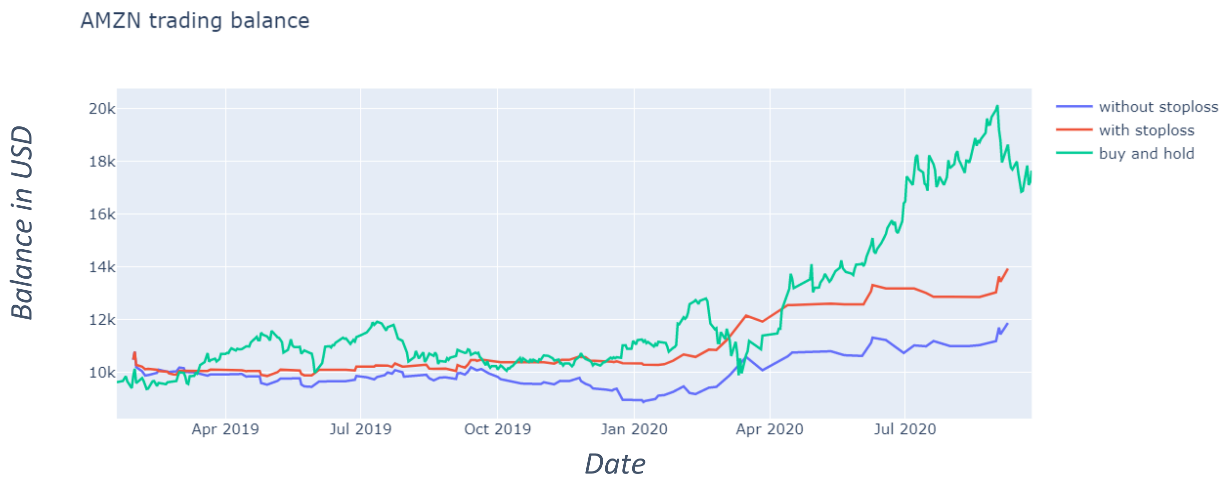

Figure 40: AMZN Trading Balance: Technical + Macroeconomic Model

Figure 41: MSFT Trading Balance: Technical + Macroeconomic Model

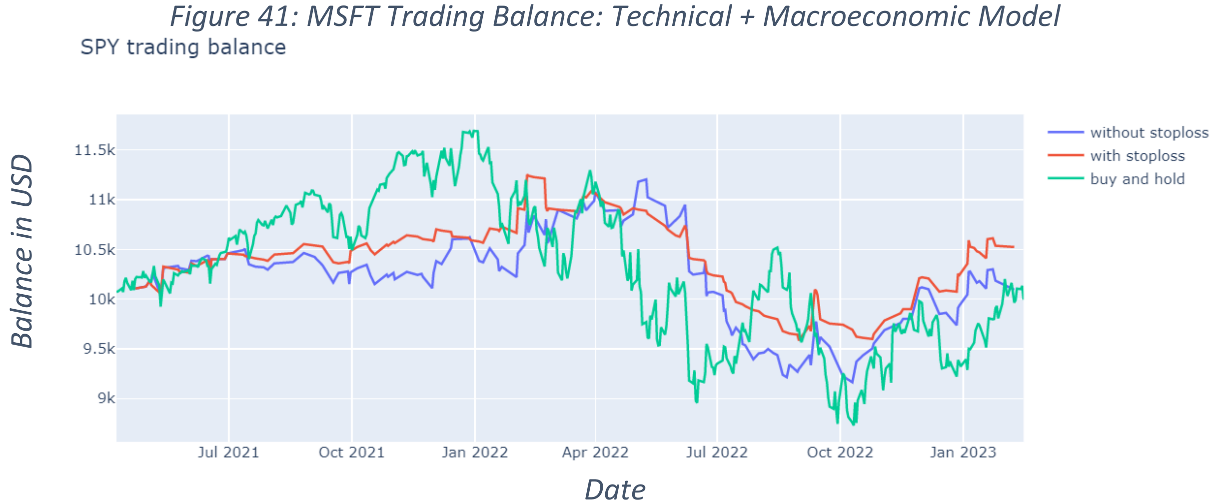

Figure 42: SPY Trading Balance: Technical + Macroeconomic Model

7.4 Equity Technical + Value model trading result

Table 5 Equity Technical + Value model trading result

	Evaluation Metrics	Without stoploss	With stoploss	Buy and hold
AAPL	Win rate	55.13%	30.77%	52.99%
	CAGR	25.02%	25.15%	47.94%
	Sharpe Ratio	17.92	22.06	39.32
	Max. Drawdown	-19.18%	-11.29%	-31.43%
	Trade turnover	3.95 days	3.95 days	1.46 days
AMZN	Win rate	50.00%	23.00%	54.88%
	CAGR	5.71%	16.46%	50.35%
	Sharpe Ratio	4.54	21.48	47.95
	Max. Drawdown	-11.62%	-3.94%	-22.75%
	Trade turnover	5.67 days	5.67 days	1.44 days
MSFT	Win rate	52.69%	33.33%	58.58%
	CAGR	4.11%	11.59%	51.57%
	Sharpe Ratio	3.29	12	42.94
	Max. Drawdown	-16.57%	-9.63%	-28.24%
	Trade turnover	6.14 days	6.14 days	1.44 days

Figure 43: AAPL Trading Balance: Technical + Value Model

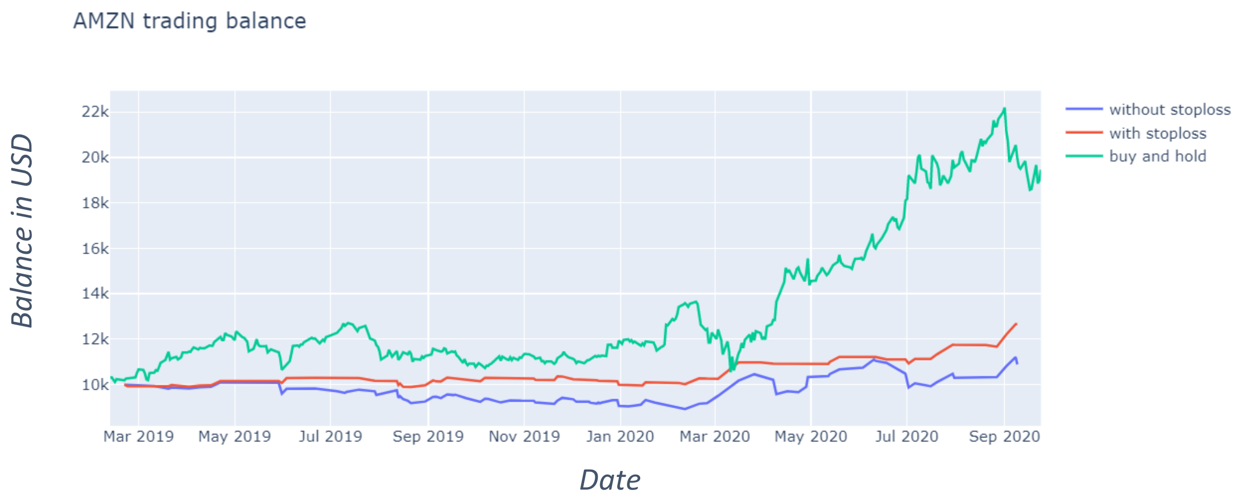

Figure 44: AMZN Trading Balance: Technical + Value Model

Figure 45: MSFT Trading Balance: Technical + Value Model

7.5 EQUITY TECHNICAL + FUNDAMENTAL + MACROECONOMIC + VALUE MODEL TRADING RESULT

Table 6 Equity Technical + Fundamental + Macroeconomic + Value model trading result

Stock Name	Evaluation Metrics	Without stoploss	With stoploss	Buy and hold
AAPL	Win rate	56.21%	34.91%	52.99%
	CAGR	22.66%	42.72%	47.94%
	Sharpe Ratio	18.50	48.84	39.32
	Max. Drawdown	-17.58%	-11.18%	-31.43%
	Trade turnover	3.81 days	3.81 days	1.46 days
AMZN	Win rate	50.00%	28.76%	54.88%
	CAGR	0.54%	9.42%	50.35%
	Sharpe Ratio	0.45	10.75	47.95
	Max. Drawdown	-21.93%	-10.79%	-22.75%
	Trade turnover	4.10 days	4.10 days	1.44 days
MSFT	Win rate	55.37%	32.23%	58.58%
	CAGR	15.46%	21.65%	51.57%
	Sharpe Ratio	11.58	18.22	42.94
	Max. Drawdown	-13.46%	-8.38%	-28.24%
	Trade turnover	4.88 days	4.88 days	1.44 days

Figure 46: AAPL Trading Balance: Technical + Fundamental + Macroeconomic + Value Model

Figure 47: AMZN Trading Balance: Technical + Fundamental + Macroeconomic + Value Model

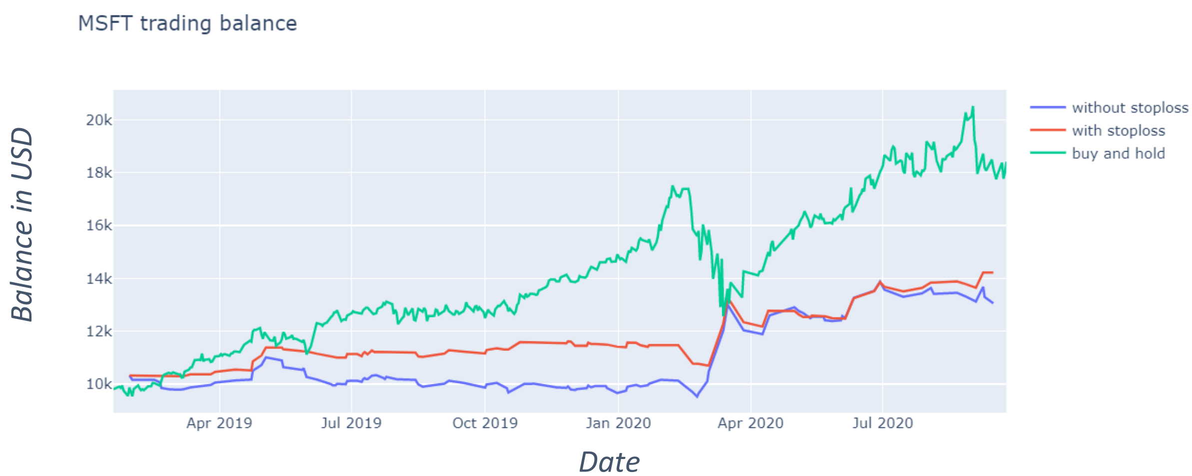

Figure 48: MSFT Trading Balance: Technical + Fundamental + Macroeconomic + Value Model

7.6 FOREIGN EXCHANGE TECHNICAL MODEL TRADING RESULT

Table 7 Foreign exchange technical model trading result

Currency pair name	Evaluation metrics	Without stoploss	With stoploss	Buy and hold
AUDUSD	Win rate	77.612%	77.612%	48.471%
	CAGR	45.979%	45.979%	-5.457%
	Sharpe Ratio	128.564	128.564	-11.833
	Max. Drawdown	-2.537%	-2.537%	-17.952%
	Trade turnover	4.291	4.291	1.398
CNYUSD	Win rate	53.896%	31.169%	50.487%
	CAGR	0.913%	5.753%	-2.658%
	Sharpe Ratio	6.277	70.637	-17.800
	Max. Drawdown	-3.781%	-0.055%	-12.011%
	Trade turnover	4.773	4.773	1.410
EURUSD	Win rate	70.161%	42.742%	47.942%
	CAGR	15.341%	15.592%	-5.590%
	Sharpe Ratio	55.364	79.158	-19.555
	Max. Drawdown	-3.424%	-0.030%	-17.866%
	Trade turnover	5.355	5.355	1.395
GBPUSD	Win rate	65.812%	32.479%	47.541%
	CAGR	13.049%	14.161%	-7.716%
	Sharpe Ratio	38.573	60.404	-21.637
	Max. Drawdown	-2.952%	-0.035%	-22.394%
	Trade turnover	5.701	5.701	1.395
USDCAD	Win rate	64.748%	35.971%	50.405%
	CAGR	13.640%	15.125%	3.654%
	Sharpe Ratio	58.823	90.273	14.982%
	Max. Drawdown	-3.718%	-0.011%	-4.618%
	Trade turnover	4.892	4.892	1.395
USDCHF	Win rate	85.714%	49.107%	51.822%
	CAGR	24.565%	16.306%	-0.704%
	Sharpe Ratio	135.474	92.279	-2.617
	Max. Drawdown	-0.595%	-0.008%	-10.475%
	Trade turnover	6.018	6.018	1.395
USDJPY	Win rate	85.106%	85.106%	54.627%
	CAGR	33.420%	33.420%	9.238%
	Sharpe Ratio	199.094	199.094	45.929%
	Max. Drawdown	-0.631 %	-0.631%	-14.749%
	Trade turnover	4.888	4.888	1.398

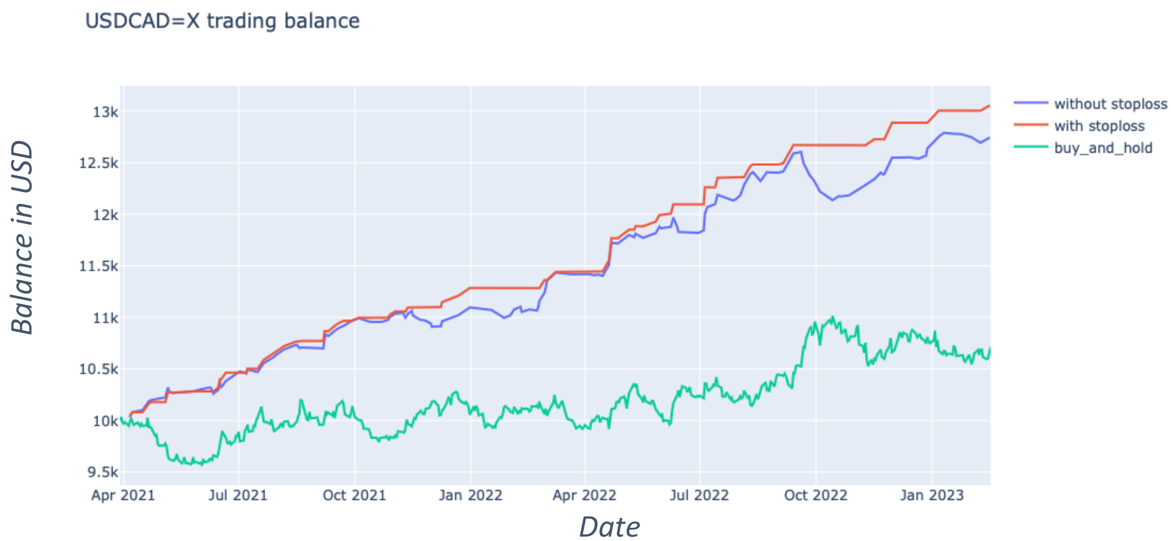

Figure 49: USD/CAD Trading Balance

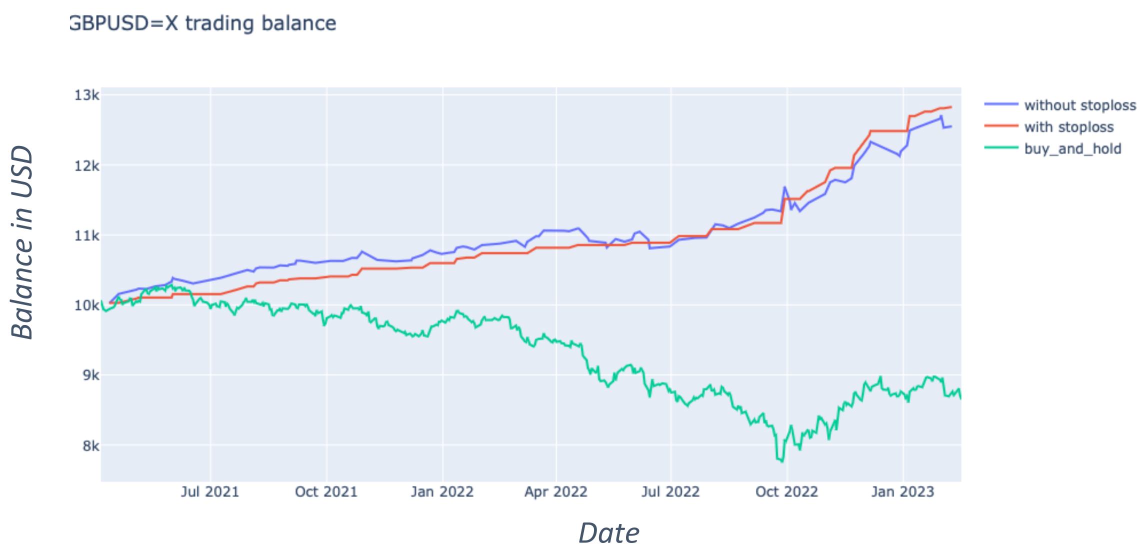

Figure 50: GBP/USD Trading Balance

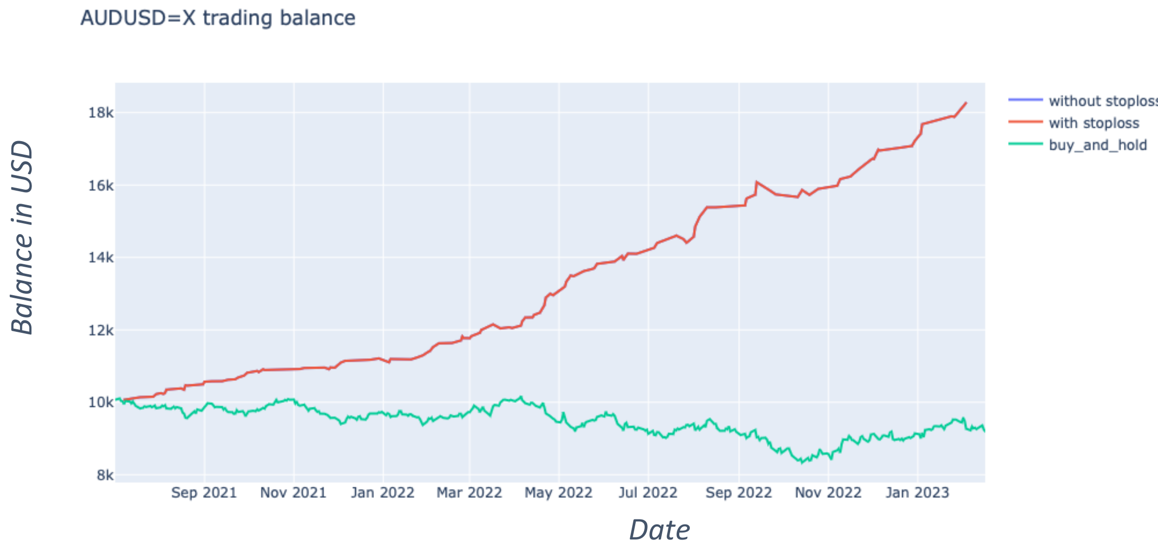

Figure 51: AUD/USD Trading Balance

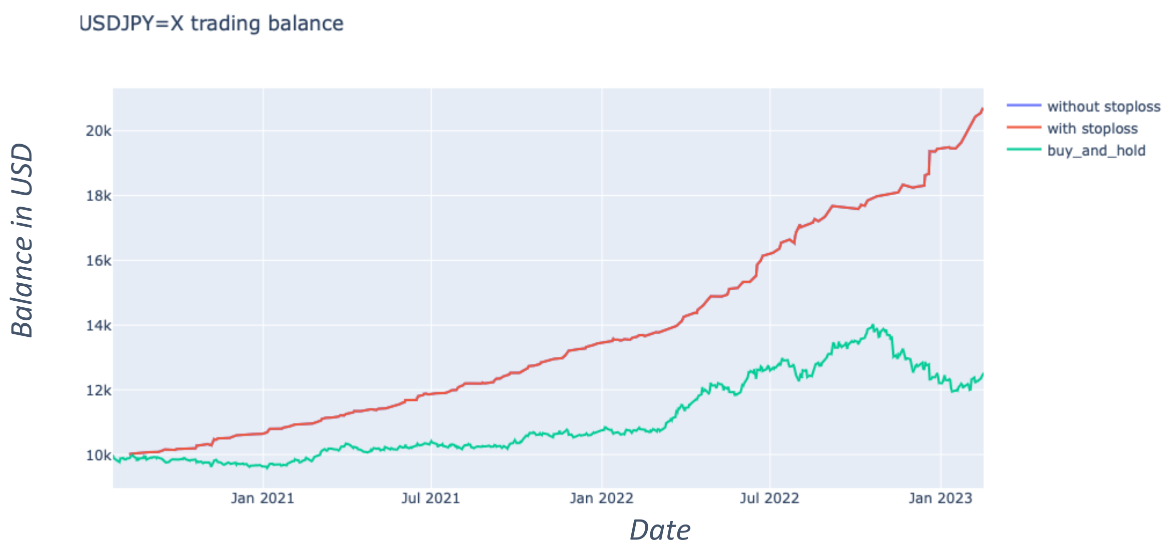

Figure 52: USD/JPY Trading Balance

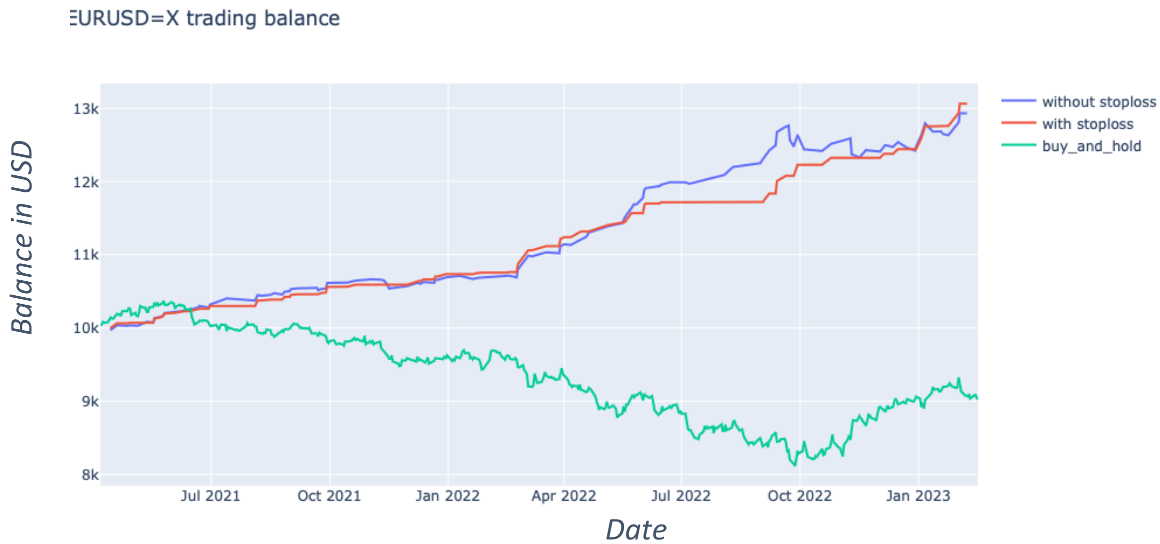

Figure 53 EUR/USD Trading Balance

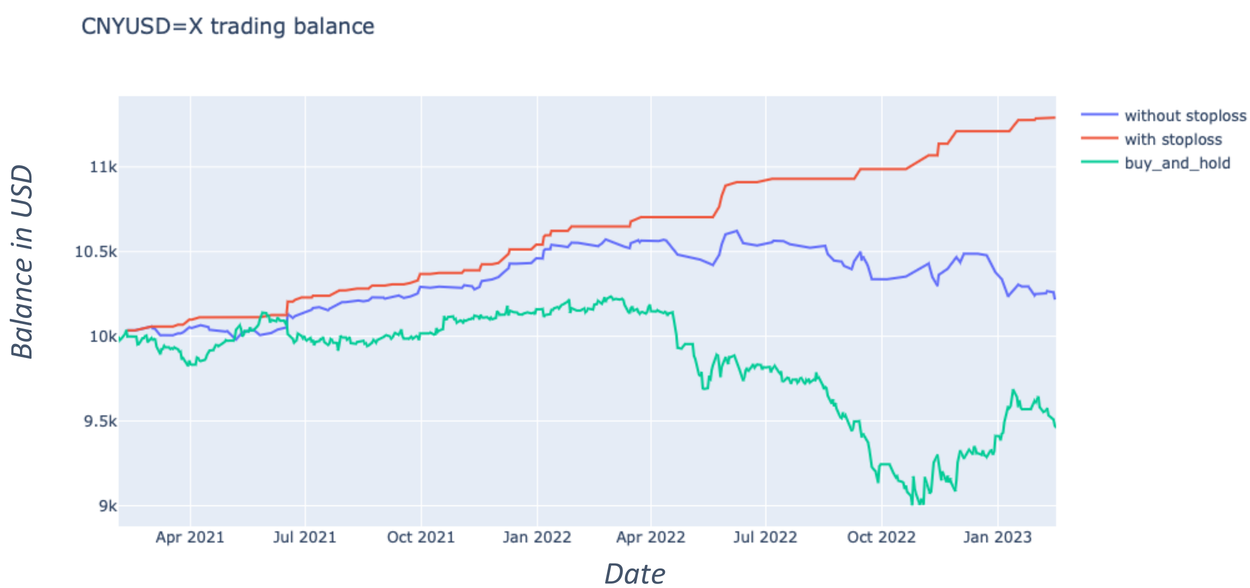

Figure 54: CNY/USD Trading Balance

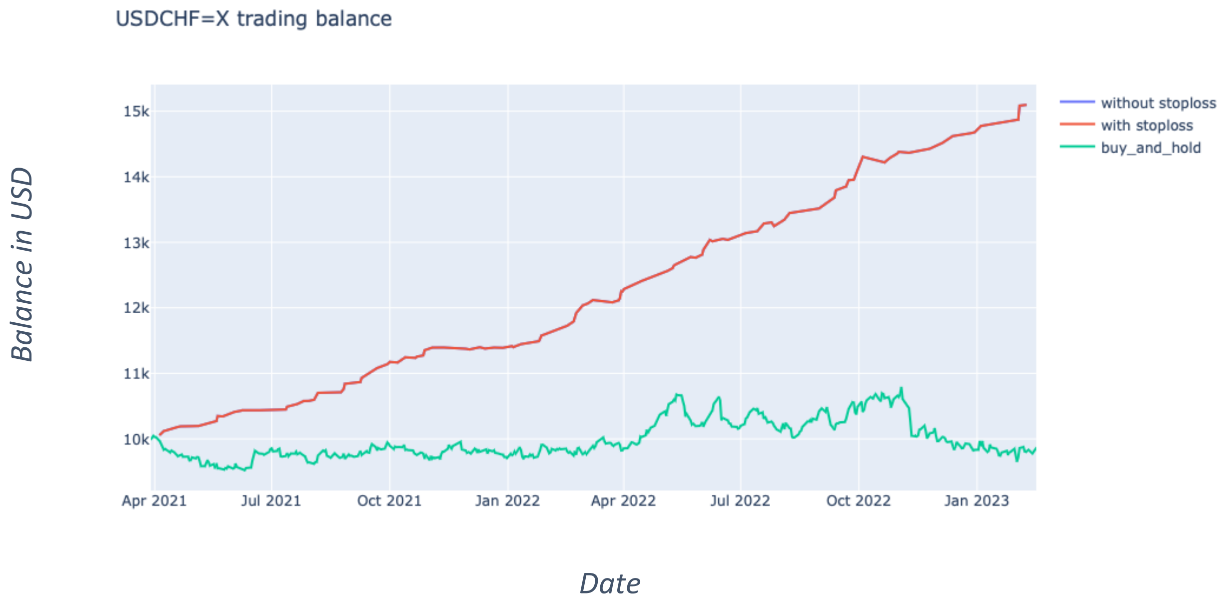

Figure 55: USD/CHF Trading Balance