Reinforcement Trading for Multi-Market Portfolio with Crisis Avoidance

CAI Lingzhi

I. ABSTRACT

As innovations of programmed trading emerges rapidly, researchers have developed many portfolio management applications based on Artificial Intelligence. Recently the global financial market comes to a new crisis in 2020 triggered by the COVID-19 pandemic. During such period, it is crucial for portfolio manager to adopt policies that can preserve the value of the portfolio, however, the existing works on programmed trading rarely considered this issue. In this paper, we proposed a trading framework based on Reinforcement Learning with crisis avoidance algorithm. We implemented a Reinforcement Learning Environment that describes the market behavior with technical analysis and finite rule-based action sets. The framework further implements a crisis detection and avoidance algorithm. The experiment result shows that the models trained by the framework performed as well as buy-and-hold strategy benchmark in bullish period of 2015-2019. Furthermore, very much accredited to the crisis avoidance algorithm, the models acted 17% better than buy-and-hold during all testing windows no less than 5 years in 2000-2019.

II. INTRODUCTION

Computational finance and program trading for portfolio management in recent year emerges rapidly as the computational power enhances as more computer technology innovations come into being. While the early program trading relies on rule-based strategies and expert system that has been used for decades, researchers and traders start to be interested in introducing Artificial Intelligence (AI) into portfolio management.

There are many approaches to for the program to describe the market condition and take action. The most intuitive way is to consider a single market index (such as S&P 500), use the raw price as the input, and output a binary 0/1 decision indicating whether to buy-in or sell-out. However, if we wish to invest in multiple assets and manage a portfolio, the market condition and the transaction procedure will be much more complicated.

In common practice of portfolio management, allocating assets in multiple geographical locations is widely adopted for diversifying market risks. The actual gain or loss may therefore be affected by the exchange rates among different currencies. The impact of currency change and leakage should be considered by the model.

Furthermore, most of the previous models are trained and tested in selected markets during a specific short period of time, the majority of which are bullish market. Such models are therefore time and data dependent. However, black-swan and grey-rhino events always exist, and statistically there has been a financial crisis for almost every decade, just as what we are experiencing in 2020. Hence, it is very plausible that a portfolio for five years or longer encounters some sorts of crisis. Therefore, a crisis detection and avoidance algorithm is crucial for a sustainable trading framework.

To address the above problems, in this paper, we choose a broader perspective than historical price with more quantifiable features to describe the market, and develop a portfolio management framework covering multiple markets/countries while considering the currency fluctuations. We implement a Reinforcement Learning framework in OpenAI Gym [1], and use Stable Baseline for implementing and customizing the models [2]. The models are built using with an Actor-Critic Model, optimized by Proximal Policy Optimization algorithm. We define a novel Environment for training the model using technical indicators that cover trend, momentum and currency leakage. A Crisis Avoidance algorithm is further implemented in the model.

The remaining part of this paper is organized as follows: Section III introduces the technical analysis indicators and the theory foundation of reinforcement learning. Section IV describes the structure and setup of the framework and the training methods. The performance of the proposed framework under different circumstances will be verified in section V. Section VI concludes this paper and discusses future research directions.

III. RELATED WORKS

a. PORTFOLIO MANAGEMENT

A portfolio in investment generally selects and invests in a group of assets for long-time benefits within reasonable risk [3]. The classic Markowitz portfolio theory [4] uses mean-variance with covariance matrix. Throughout the years till now, researchers have developed a wide range of models considering many aspects and variables, including VIX index and Volatility [5] [6] [7], Momentum [8] [9], Reversals [10], Liquidity [11], long- and short-term risks [12] [13], macro-economic factors [14], and even longevity risk [15]. These asset allocation models may improve the performance of portfolio management and asset allocation, however, as Laborda and Olmo pointed out, they can only be solvable with strong assumptions of the objection function or the distribution of the asset return [16]. Such assumptions are idealistic when considering the real market behaviours, and the conclusion drew from such works may therefore be hard to implement or alter in the actual trading.

b. MACHINE LEARNING, ARTIFICIAL NEURAL NETWORK AND REINFORCEMENT LEARNING

In very short words, Machine Learning techniques are programs that are designed to extract patterns from historical data during its training phase, in order to make judious prediction on the new data [17]. Some of the basic techniques include Support Vector Machines (SVMs), Random Forests (RFs), Genetic Algorithms (GAs) and Artificial Neural Networks (ANNs) [18].

From the aspect of targets and goals , Machine Learning can be categorized into Supervised Learning, Unsupervised Learning and Reinforcement Learning. In Supervised Learning, the developers provide the correct labels and the model is evaluated base on the accuracy. In Unsupervised Learning, such as clustering, or recommender systems, the goal is to find the similarities or differences from the given data. The Reinforcement Learning, however, is meant to optimize the reward from a series of actions. Inspired by neural networks in human brain, the artificial neural networks (ANN) consists of layers of “neurons” that takes in output signals. Most ANN has three types of layers, the input layer, the hidden layer, and the output layer. The neurons between layers are interconnected, and the receivers assigns certain weight to the input indicating its importance. The receiver neuron then sum up the input signals and produces output signal based on its propagation functions.

Reinforcement Learning (RL), introduced by Sutton and Barto [19], is one of the machine learning subfields which is concerned with decision-making. Unlike Supervised Learning, it studies how an agent can learn and achieve targets in a dynamic, unpredictable environment [20].

c. PROGRAMMED TRADING

Researchers in the field of Computer Science and Artificial Intelligence started to develop trading algorithms about two decades ago [21] [22]. Recent AI-based trading applications develops in multiple approaches, including machine learning [23], generic algorithm [24] [25], reinforcement learning [26] [27] [28] [29] and so on. However, most of these approaches use supervised learning [29]. Many works are generally conducted with rather idealistic assumptions and simple action spaces. Many of the works only consider one index, such as S&P 500 Index, and a binary action, i.e. 100%-Buy to 100%-Sell. Such assumption can rarely be introduced into actual trading as well.

There is currently not much consensus in the community of computer science regarding what aspects should a realistic and intelligent trading framework consists of. For example, the work of Kang et. al. experimented on stock selection based on history prices [30], and Kim’s work investigates on the impact of volatility for asset allocation. While also dealing with risk, Gupta’s research focused more on the risk-sensitivity of the investors level [27]. On a more macro perspective, Miharja proposed that the cycle-based analysis should be considered for stocks and ETFs traded in a single market -- most likely in US Market [29].

IV. IMPLEMENTATION OF REINFORCEMENT TRADING FOR MULTI-MARKETS PORTFOLIO WITH CRISIS AVOIDANCE

a. REINFORCEMENT LEARNING FRAMEWORK

In a general Reinforcement Learning setting, the *Environment* and the *Agent* are the two main components. The data flow can be described as a set of *Observation*, *Action*, and *Reward*. The Agent interacts with the Environment with discrete time steps. At each time step t, the Agent fetches an Observation o_t from the Environment. o_t will go through its Policy π so the Agent can decide an action a_t. After the action is decided, the Environment move the time step forward to $t+1$, and fed the Agent with a new state o_{t+1} along with a reward r_{t+1} based on its previous action a_t.

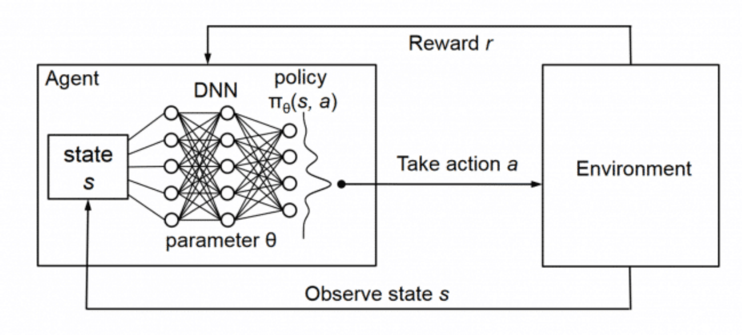


Figure 1: Reinforcement Learning Framework, Adapted from [32]

Therefore, we can define the state s_t that an agent ends up with as a function of all its previous experiences [31], that is the previous observation, action and reward at each time step. Hence,

$$s_t = f[(o_1, a_1), (r_2, o_2, a_2), \dots, (r_{t-1}, o_{t-1}, a_{t-1}), (r_t, o_t)]$$

The Policy π mentioned above, is defined as a probability distribution of choosing the action a given the observation state s. The policy can be written as $\pi(s, a)$ or $\pi(a|s)$, while both are carrying the same meaning. The policy has a range of [0, 1], that is, $\pi : \pi(s, a) \rightarrow [0, 1]$.

For a given state s_t, the n-step return, $R_{t:t+n}$, is defined as the discounted sum of rewards during the n steps where γ is the discount factor, which is computed by in Equation (1).

$$\begin{aligned} R_{t:t+n} &= r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots + \gamma^n r_{t+n} \\ &= \sum_{i=1}^{n} (\gamma^i \cdot r_{t+i}), \gamma \in (0,1] \dots \dots \dots \dots (1) \end{aligned}$$

We use $V^{\pi}(s)$ to represent the *value function* of a state s to an Agent with policy π. The value function shows how "good" the state is, and is defined as the Agent's expected return from state s, which is computed by in Equation (2).

$$V^{\pi}(s) = \mathbb{E}[R_{t:+\infty}|(s_t = s, \pi)] \dots\dots\dots\dots (2)$$

The *action value function*, or the *Q value function* written as $Q^{\pi}(s, a)$, measures how "good" an action would be, and is defined as the expected return when an action a is taken in state s, which is computed by in Equation (3).

$$Q^{\pi}(s, a) = \mathbb{E}[R_{t:+\infty}|(s_t = s, a_t = a, \pi)] \dots\dots\dots\dots (3)$$

The ultimate maximization goal for an Agent is the *expected cumulative discounted reward*, which is computed by in Equation (4).

$$goal\colon \max[\mathbb{E}(R_{0:+\infty})] = \max\left[\mathbb{E}\left(\sum_{t=0}^{+\infty}(\gamma^{t} \cdot r_t)\right)\right], \gamma \in (0,1] \dots\dots\dots\dots (4)$$

In order to define the loss function for optimizing the model, we define the *Advantage Estimate* $\hat{A}_t$, which the *discounted sum of the rewards* minus the *baseline estimate*. The *discounted sum of the rewards* is obtained at the end of the trajectory, which is a known value. The *baseline estimate*, on the other hand, is the Value Function of state s_t estimated by the Critic, which is the output of a Neural Network in this framework. The Advantage Estimate $\hat{A}_t$ is computed by in Equation (5).

$$\hat{A}_t = R_{t:+\infty} - V^{\pi_\theta}(s_t) = R_{t:+\infty} - \mathbb{E}[R_{t:+\infty}|(s_t, \pi_\theta)] \dots\dots\dots\dots (5)$$

Base on Equation (5), we can use the *Advantage Estimate* to represent the estimate of the *relative value* of the selected action in the current state. The *relative value* generally means that it shows how much better off because of the action a_t that the Agent took in state s_t, comparing to what generally happens in state s_t as the value function estimated. Intuitively, if $\hat{A}_t$ is positive, the action a_t performs better than average expectation, and the model increases the possibility.

The framework utilizes Policy Gradient to update the model, which directly learns the pattern when interacting with the Environment. A typical Loss Function or the Target Function [32] for a Policy Gradient algorithm at time step t is computed by in Equation (6).

$$L^{PG}(\theta) = \hat{\mathbb{E}}_t\left[\log \pi_\theta(a_t|s_t) \cdot \hat{A}_t\right] \dots\dots\dots\dots (6)$$

In Equation (6), $\pi_\theta(a_t|s_t)$, as explained above, is the probability that a policy π chooses action a_t when facing the state s_t at time step t, and $\log \pi_\theta(a_t|s_t)$ takes the logarithm of the probability. $\hat{A}_t$ is the Advantage Estimate defined in Equation (5).

b. TECHNICAL ANALYSIS INDICATORS

In order to quantify the performance of the market performance, researchers and investors have invented numerous market indicators. In an open-source indicator package, the author Padial splits them into 5 categories, including Volume, Volatility, Trend, Momentum and Returns [33]. Among those indicators, we adopted the Exponential Moving Average (EMA), the Moving Average Convergence/Divergence (MACD) as the indicator of market trend and use the Relative Strength Index (RSI) as the momentum indicator for the study.

1. Exponential Moving Average (EMA)

The Exponential Moving Average (EMA) is a weighted moving average that smoothen the price by taking average of the previous value and giving more importance to recent data. Given a series $x_1, \dots, x_N$, the EMA at time step t is computed in Equation (7)

$$EMA(t) = k \cdot x_t + (1-k) \cdot EMA(t-1) \ \dots\dots\dots\dots (7)$$

where

$$EMA(1) = x_1,$$

$$k = \frac{2}{N+1}$$

2. Moving Average Convergence/Divergence (MACD)

Moving Average Convergence/Divergence (MACD) measures the relationship of momentum and trend of a series. MACD consists of two lines, the Signal Line and the MACD Line. The Signal Line is the 26-period EMA ($EMA(t, 26)$), and the MACD Line is the 12-period EMA($EMA(t, 12)$). In short, $EMA(t, n)$ is the n-day EMA from time step t backwards. The MACD difference at step t, represented by $MACD(t)$, is the difference between the MACD Line and the Signal Line. The definitions are computed in Equation (8):

$$MACD(t) = EMA(t, 12) - EMA(t, 26) \dots\dots\dots\dots (8)$$

where

$$EMA(t, n) = \ k \cdot x_t + (1-k) \cdot EMA(t-1, n-1),$$

$$EMA(t-n+1,1) = x_{t-n+1},$$

$$k = \frac{2}{n+1}$$

$MACD(t) > 0$ when $EMA(t, 12) > EMA(t, 26)$, which means the short-term value is higher than the long-term value. $MACD(t) < 0$ when $EMA(t, 12) < EMA(t, 26)$, which means the short-term value is lower than the long-term value.

Therefore, when a crossover of the MACD Line and the Signal Lines appears, it indicates that the trend has changed. When the MACD difference moves from positive to negative, the trend is going downwards; otherwise, if the MACD difference moves from negative to positive, it indicates that the trend is going upwards.

3. Relative Strength Index (RSI)

The Relative Strength Index (RSI) indicates the momentum of the price and indicates a possible reversal in trend. It is defined in Equation (9):

$$RSI(n) = 100 - \left[\frac{100}{1 + RS(n)}\right] \dots\dots\dots\dots (9)$$

where

$$RS(n) = \frac{Average\ Gain}{Average\ Lost} = \frac{\frac{\sum Gain\%}{n_{Gain}}}{-1 \times \frac{\sum Loss\%}{n_{Loss}}}$$

$$n = n_{Gain} + n_{Loss}$$

The RSI has a range of [0, 100], and generally it is considered “oversold” when $RSI \leq 30$, and considered “overbought” when $RSI \geq 70$.

c. OBSERVATION SPACE

As the framework targets to allocate assets in multiple markets, we categorize the markets into high- medium- and low- risk classes. We use the technical analysis indicators for high and medium risk market in the past as the observation space. The time stamps are the past 1-5, 10, 15, 20, 40, and 100 days. In specific, the technical analysis indicators are defined in Equation (10).

$$\boldsymbol{Observation}_t = [o_{t-1}, o_{t-2}, o_{t-3}, o_{t-4}, o_{t-5}, o_{t-10}, o_{t-15}, o_{t-20}, o_{t-40}, o_{t-100}] \dots\dots\dots\dots (10)$$

where

$$o_{t_0} = \left[EMA_h, MACD_{diff_h}, \Delta Time_h, EMA_m, MACD_{diff_m}, \Delta Time_m\right]_{t=t_0}$$

EMA and MACD is defined in the previous section. The delta_time is the count of steps since the previous sign changes (positive to negative or negative to positive) of the MACD value.

Instead of the approach taken by Lim [28] using z-score normalization of all the history that results in “peeking in future”, we normalize the EMA with the first data point (i.e. the EMA 6 days ago) to show the pattern in the window period.

d. ACTION SPACE

Generally, the expected return from the high-, medium- and low-risk market is positively related to the risk level. Therefore, the portfolio is designed to leverage on the high-risk market when the market is bullish, and to shift to medium- or low-risk market when the market condition of high-risk market goes bear. In short, the model shifts its risk appetite among “Aggressive”, “Stable” and “Conservative” policies based on the market condition. We therefore define four actions for the Actor to choose from based on the rationale of the policies as in Table-1.

Action/Policy #	**High-Risk**	**Medium-Risk**	**Low-Risk**	**Remark**
1	80%	10%	10%	Aggressive
2	10%	80%	10%	Stable
3	45%	45%	10%	Stable
4	10%	10%	80%	Conservative

Table 1: Discrete Action Space Settings Base on Risk Appetite

For each day, the Actor will provide a four-dimensional one-hot output, in a format of $[x_1, x_2, x_3, x_4]$, indicating its willingness for taking each action. The action with the greatest value will be the action taken. If two or more values are the same, the action with the smallest position number will be taken. Furthermore, if all of the four number are 0, the model will skip this round without any action. This is to avoid the model giving a bias to Action 1 as a 0.0 output has a much greater possibility than any other exact value.

e. REWARD FUNCTION

The Reward Function is used for training both the Actor and the Critic in the Actor-Critic Reinforcement Learning Algorithm. It determines how good the Actor’s decision is. As such, we would naturally wish to maximize the Reward. However, as the Reward function is customized and not bounded during on-policy training, it is hard to directly set an optimization approach for maximizing the Reward. Therefore, the Critic is placed here to “predict” the Reward, and to transform the maximization problem into minimizing the difference between the predicted result by the Critic and the Actual Reward from the Environment.

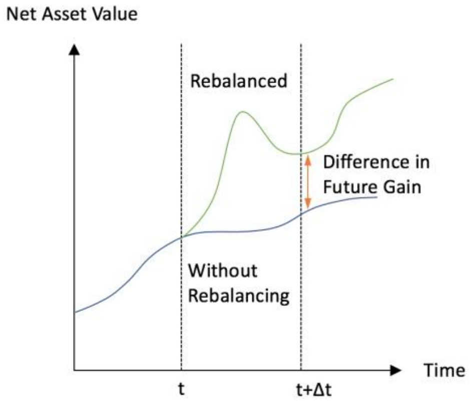


Figure 2: Difference in Future Gain

We adopt the *Difference in Future Gain* as the Reward, as proposed in Lim's algorithm [28]. The Reward of an Action is defined the difference between the NAV after m days with and without taking that Action. Such reward scheme is able to provide direct feedback of the action. The mathematical definition is computed in Equation (11):

$$r_t = \frac{\sum_k i_k^{a_t} \cdot p_{t+m}^k - \sum_i i_k^{\sim a_t} \cdot p_{t+m}^k}{\sum_k i_k^{\sim a_t} \cdot p_{t+m}^k} \dots\dots\dots\dots (11)$$

where a_t is the action at time step t, $i_k^{a_t}$ is the inventory number of Asset k after executing action a_t, $i_k^{\sim a_t}$ is the number of inventory where a_t is hypothetically not executed, and p_{t+m}^k is the market price of Asset k after m time steps. In the experiment we take $m = 10$.

As the future price is only computed in the reward, it also reduces the impact of "Peeking into the Future" issue. The Reward is generally bounded in [-0.050, +0.125], and the difference between a positive reward and a negative reward still remains clear.

f. TRANSACTION PROCESS AND COMMISSION FEE

We assume the commission fee rate is 0.125% for buying and selling the assets. Many papers, such as Lim's model [28], simply raises all the prices up with the commission fee rate. Such approach is not accurate because it overestimated the selling price and therefore claims more gain than reality. The error in calculation may affects the decision of the model and lead to significant difference of the result.

Since we design the model with no cash on hand, for each step the capital we use for buying in the asset must come from the cash we received for selling the other assets.

Suppose there are N number of different assets in the portfolio, the time step is t, commission fee is r.

- The inventory vector (i.e., the number of stocks on hand) is $\boldsymbol{I}_t = [i_1, i_2, \ldots, i_N]$.
- The current price vector is $\boldsymbol{P}_t = [p_1, p_2, \ldots, p_N]$.
- The previous weight of each asset is $\boldsymbol{W}_t = [w_1, w_2, \ldots, w_N]$ and the target weight vector after rebalancing is $\boldsymbol{W}_{t+1} = [w_1', w_2', \ldots, w_N']$.
- We want to compute the inventory vector after rebalancing $\boldsymbol{I}_{t+1}$

The change in weights due to selling or buying is computed in Equation (12) and (13)

$$\boldsymbol{W}_{sell} = -\boldsymbol{1} \times \min\big(\boldsymbol{0}, (\boldsymbol{W}_{t+1} - \boldsymbol{W}_t)\big) \ldots\ldots\ldots\ldots (12)$$

$$\boldsymbol{W}_{buy} = \max\big(\boldsymbol{0}, (\boldsymbol{W}_{t+1} - \boldsymbol{W}_t)\big) \ldots\ldots\ldots\ldots (13)$$

The min(0, w) functions changes all positive values in the vector **w** to 0, while the max(0, w) function changes all negative values in the vector **w** to 0.

The Cash Flow from selling is computed in Equation (14)

$$C_{in} = sum(\boldsymbol{W}_{sell} \times \boldsymbol{I}_t \times \boldsymbol{P}_t) \times (1 - r) \ldots\ldots\ldots\ldots (14)$$

Change in inventory due to selling is computed in Equation (15)

$$\Delta\boldsymbol{I}_{sell} = -\boldsymbol{1} \times \boldsymbol{W}_{sell} \times \boldsymbol{I}_t \ldots\ldots\ldots\ldots (15)$$

Change in inventory due to buying is computed in Equation (16)

$$\Delta\boldsymbol{I}_{buy} = \frac{C_{in} \times \boldsymbol{W}_{buy}}{\boldsymbol{P}_t \times (1 + r)} \ldots\ldots\ldots\ldots (17)$$

Combining Equation (12)-(17), we can therefore compute the new inventory number as Equation (18) states,

$$\boldsymbol{I}_{t+1} = \boldsymbol{I}_t + \Delta\boldsymbol{I}_{sell} + \Delta\boldsymbol{I}_{buy} \ldots\ldots\ldots\ldots (18)$$

Specifically, if the rebalancing frequency is set to i days ($i > 1$), the model is prevented to execute any other transaction during the period of $t + 1, t + 2, \ldots, (t + i - 1)$. Therefore, the weight matrix and the inventory number will remain the same during this period, as shown in Equation (19) and (20).

$$\boldsymbol{W}_{t+i-1} = \boldsymbol{W}_{t+i-2} = \cdots = \boldsymbol{W}_{t+1} \ldots\ldots\ldots\ldots (19)$$

$$\boldsymbol{I}_{t+i-1} = \boldsymbol{I}_{t+i-2} = \cdots = \boldsymbol{I}_{t+1} \ldots\ldots\ldots\ldots (20)$$

g. CURRENCY LEAKAGE

One more important concern is the fluctuation of the currency. Suppose we are investing as a US Dollar portfolio, therefore, all the assets in the foreign market are affected by the change of Foreign Exchange (FOREX) Rate. Some currencies has a strong and stable bond with US Dollar, such as New Taiwan Dollar or Hong Kong Doller. Some currencies, on the other hand, are greatly affected by other factors. For example, the US Dollar/Japan Yen Pair is traditionally tided to the US Treasuries yield.

Suppose there are N assets in the portfolio, and the time steps range from 1 to T.

The history price of the assets in local currency can be stored in a matrix $P[N, T]$. The history exchange rate to certain global currency (e.g. US Dollar) can be represented as a matrix $R[N, T]$. N represents the index of the asset, and T represents the time step. The exchange is 1 if the local currency is identical to the selected global currency.

The change in currency rate can therefore be computed in Equation (21),

$$\Delta R[n,t] = \begin{cases} 0, (t = 1) \\ \dfrac{R[n,t] - R[n,t-1]}{R[n,t-1]}, (t > 1) \end{cases} \dots\dots\dots\dots (21)$$

where $n \in \{1,2,\dots,N\}$, and $t \in \{1,2,\dots,T\}$.

The cumulative change in currency rate is computed in Equation (22):

$$\mathrm{Cum_R}[n,t] = \begin{cases} 1, (t = 1) \\ (\Delta R[n,t] + 1) \times \mathrm{Cum}_R[n,t-1], (t > 1) \end{cases} \dots\dots\dots\dots (22)$$

The Leakage rate can be therefore computed in Equation (23):

$$L[n,t] = 1 - \mathrm{Cum_R}[n,t] \dots\dots\dots\dots (23)$$

Combining Equation (21)-(23), Actual Price is therefore computed as in Equation (24).

$$P_{Actual}[n,t] = P[n,t] \times \mathrm{Cum_R}[n,t] \dots\dots\dots\dots (24)$$

h. CRISIS AVOIDANCE

As we are writing this paper in Mar 2020, the epidemic of COVID-19 spread all over the world and causes a great fall in the global market. Hence specific “circuit breaker” rules for crisis avoidance is essential, and we therefore introduce the monthly maximum drawdown into the models as an indicator of cash-out action.

The maximum drawdown (MDD) is defined as the difference between the maximum value and the minimum value afterwards of the previous m working days divided by the rolling maximum, as computed in Equation (25). Specifically, a monthly MDD is defined as the MDD parameter $m = 20$.

$$MDD = \frac{peak - trough}{peak} \dots\dots\dots\dots (25)$$

where

$$peak = \max(p_N, p_{N-1}, \dots, p_{N-m+1}) = p_i,$$

$$trough = \min(p_N, p_{N-1}, \dots, p_i),$$

The Crisis-Avoidance Policies forces the model to cash out when the monthly MDD is greater than the threshold, and to return to the markets evenly only after the monthly MDD moves below the threshold.

V. EXPERIMENT RESULTS

a. DATASET AND BENCHMARK

The dataset is derived from Yahoo Finance from Jan/2000 to Dec/2019. There are three indexed used in the experiment as the follows:

Risk Level	Ticker	Name	Market	Location	Currency Volatility
High	^BVSP	IBOVESPA	Brazil	Latin America	High
Medium	^TWII	Taiwan Weighted Index	Taiwan	East Asia	Low
Low	^IXIC	Nasdaq Composite	US	North America	-

The indices were selected based on the geographical location and their volatility. The index number is treated as the "Price" of an hypothetical ETF traded in the market in its local currency.

We define an hypothetical Buy-and-Hold (B&H) Portfolio as the benchmark of the whole section. The B&H Portfolio initially holds equal value of the three assets, that is, invests 1/3 of the total capital value in each market. It holds the assets for the whole period and sells at the end.

In Figure-3 and Figure-4, the logged market value of high-, medium- and low-risk market is plotted in blue, yellow and green line, respectively. The Net Asset Value of the benchmark is plotted in red line. Figure-3 shows the log price change during the training period (2005-2014), and Figure-4 shows the log price change during the testing period (2015-2019).

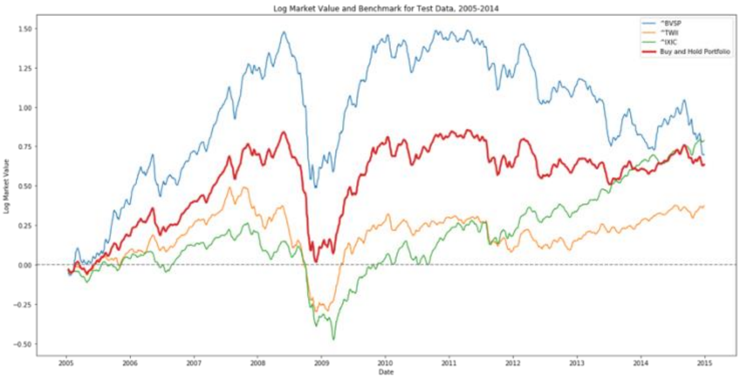


Figure 3: Log Market Value of Training Dataset and Benchmark (2005-2014)

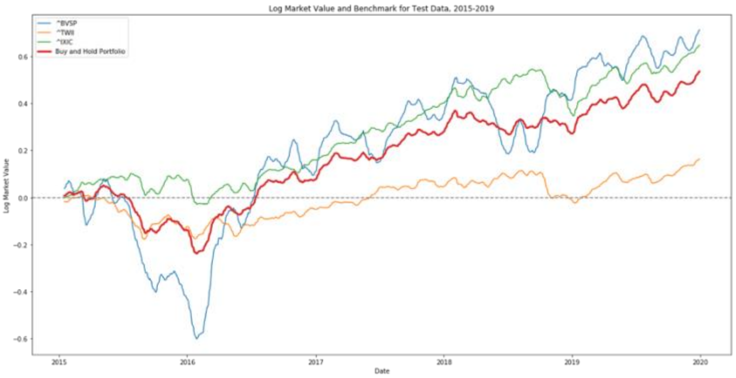


Figure 4: Log Market Value of Testing Dataset and Benchmark (2015-2019)

b. NETWORK STRUCTURE AND MODEL SETTING

The two Agents, the Actor and the Critic, are both Multi-Layer Perceptron (MLP) Neural Networks implemented by Stable Baseline. Each MLP model contains 2 layers of 64 neurons. For each experiment, we trained a batch of 10 independent models and calculate the average performance of these models.

The Observation Space is defined in Equation (10) using the actual price defined in Equation (24). The Action Space is defined in Table-1. The Reward Function is computed in Equation (11). The transaction process is computed in Equation (18) and (20).

c. COMPARISON AMONG ACTION FREQUENCIES

The first task is to determine the action frequency of the model. The action frequency indicates how many times that the model is allowed to rebalance the portfolio and execute transaction within a certain period of time. In practice, the frequency ranges from once every millisecond to once per year. Although the model may not necessarily act at the maximum allowed frequency, there are still possibilities that the model abuses the transactions and results in excessive transaction cost.

We first trained the models with daily frequency with 10k, 50k, 100k and 500k, and discovered that the 10k training epochs are obviously insufficient, while the improvement of the model from 100k to 500k is very little. Therefore, in the later experiment (for weekly and monthly frequency), we reset the training epochs sets to 50k, 100k and 200k. The testing period is from 2015-2019. The grey dotted line is the Actual Return Rate or the Sharpe Ratio of benchmark, respectively.

We measure the performance using the Nominal Return, the Actual Return and the Sharpe Ratio. The Nominal Return Rate equals the gain/loss divided by initial value, as defined in Equation (26). It simply reflects how much the portfolio return at the end of the period, yet it does not reveal whether the return comes from the decision it takes, or from the market trend, or just simply by luck.

$$NRR = \frac{NAV_{end} - NAV_{start}}{NAV_{start}} \ldots\ldots\ldots\ldots (26)$$

where NRR is the Nominal Return Rate of a model. NAV is the Net Asset Value, that is, the market value of the portfolio. NAV_{end} and NAV_{start} are respectively the start and end nominal value of the portfolio without adjustment of the currency.

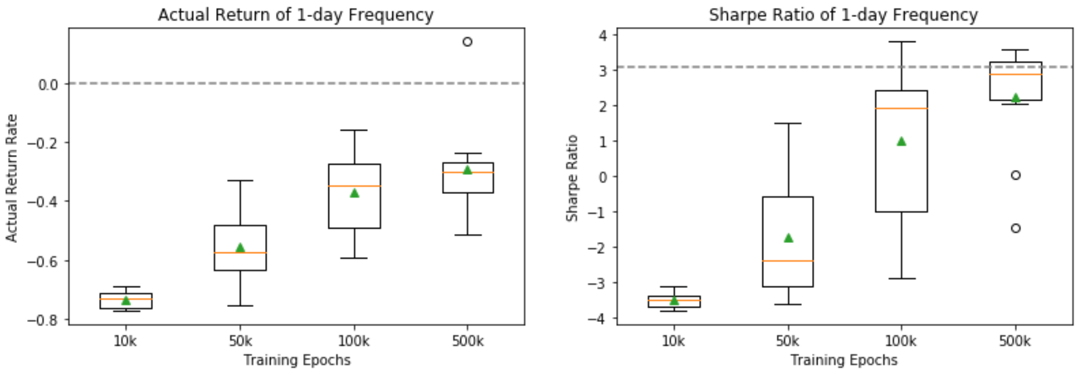
Actual Return Rate therefore eliminate the impact of the market trend, and it equals the relative gain/loss to the benchmark divided by the final benchmark net value, adjusted by the currency rate as defined in Equation (24). Basically the Actual Return shows how much better the model performs than the average. As the impact of currency is included in the Actual Return Rate, there might be negative ARR even if the NRR of the model is higher than the benchmark and vice versa. The Actual Return is defined in Equation (27).

$$ARR = \frac{NAV_{end}^{model}|_{P_{Actual}} - NAV_{end}^{benchmark}|_{P_{Actual}}}{NAV_{end}^{benchmark}|_{P_{Actual}}} \ldots\ldots\ldots\ldots (27)$$

The Sharpe Ratio, on the other hand, shows how much of the profit is by luck. The Sharpe Ratio is computed by the average return divided by the standard deviation of the price, and shows the average return in "one unit of risk", as shown in Equation (28). For the same Nominal Return or Actual Return, the higher the Sharpe Ratio, the more stable the profit will be.

$$Sharpe\ Ratio = \frac{R_p - R_f}{\sigma_p} \ldots\ldots\ldots\ldots (28)$$

where $R_p = NAV_{end} - NAV_{start}$ indicates the return of the portfolio. R_f is the risk-free return, and we substitute it using the US 3-month T-bill return. σ_p is the standard deviation of R_p as the end time stamp moves along the calendar.



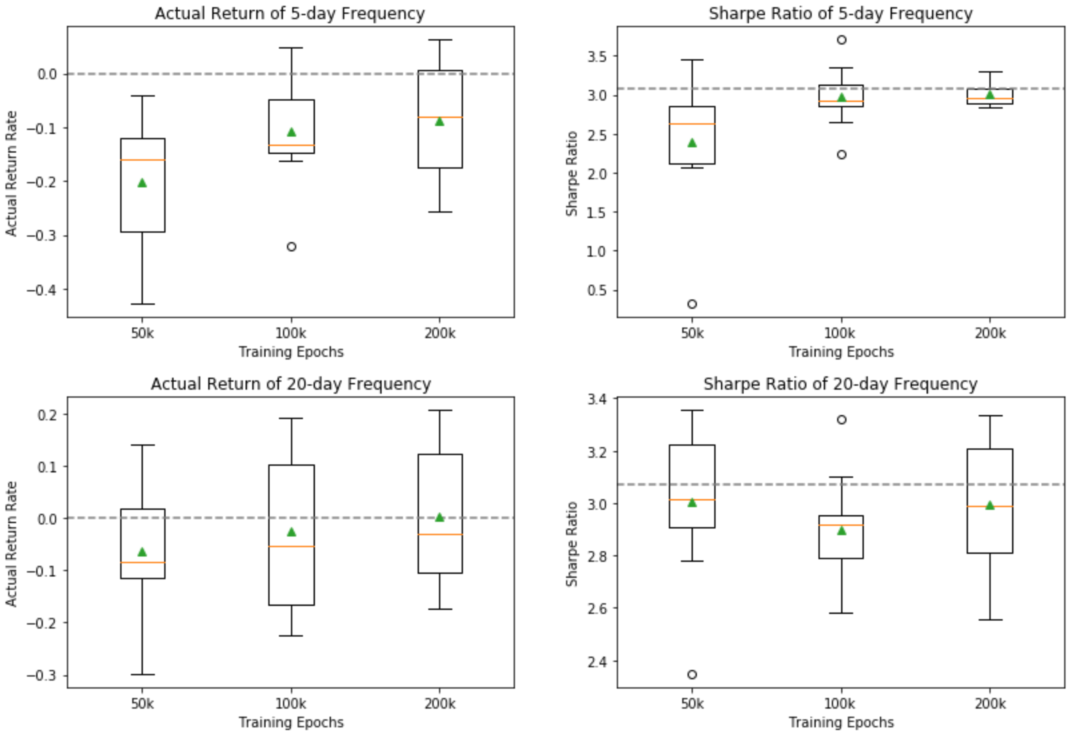


Figure 5: Actual Return and Sharpe Ratio for Different Frequencies

Frequency	Training Epochs	Nominal Return Rate	Benchmark Nominal Return	Actual Return Rate	Sharpe Ratio	Benchmark Sharpe Ratio
1-day	10k	-46.3%	+72.7%	-73.3%	-3.513	3.069
	50k	-15.5%		-55.5%	-1.724	
	100k	+16.6%		-36.9%	0.984	
	500k	**+29.6%**		**-29.3%**	**2.230**	
7-day	50k	+42.1%		-20.2%	2.400	
	100k	+58.6%		-10.7%	2.973	
	200k	**+61.8%**		**-8.8%**	**3.011**	
20-day	50k	+66.8%		-6.3%	3.002	
	100k	+73.4%		-2.4%	2.896	
	200k	**+77.9%**		**+0.2%**	**2.995**	

Table 2: Actual Return and Sharpe Ratio for Different Frequencies

The experiment result shows that as the training epochs increases and the frequency decreases, the performance of the model actually gets better in 2015-2019. The Actual Return Rate of daily trading frequency is way below the benchmark, and the Sharpe Ratio is negative. This shows that it is not suitable as the high-frequency trading leads to high commission fee leakages (defined in Equation (18)), while some random action hinders the performance of its strategies.

The testing result of weekly frequency significantly improved comparing to the daily frequency. The Actual Return Rate increases but still the majority is below the benchmark. The Sharpe Ratio is close to the benchmark but the majority is still lower.

The monthly frequency models with 200k training epochs receive 100% gain for all 10 models in the batch, and has 50% of the models performing better than the benchmark. As highlighted in the table, the performance of the models is significantly better than the weekly and daily rebalancing models. The performance improves as the training epochs increases, which means the model converges and the setting of the model is feasible. Statistically, the monthly models can achieve a return as good as the benchmark, with more information provided and less trading frequency forced on the Agent.

d. TEST OF MONTHLY REBALANCE MODEL IN ALL PERIODS

Instead of simply testing in 2015-2019, we further evaluate the performance of the best monthly rebalance model in the whole history in 2000-2019, for a minimum period of 5 years. That is, we tried all combination of holding period for no less than 5 years, such as (2000, 2004), (2000, 2005) and so on. The testing period starts on 1/Jan of the starting year, and ends on 31/Dec of the ending year. We plot the Actual Return is as the Figure-6, where the y-axis is the starting year and the x-axis is the ending year.

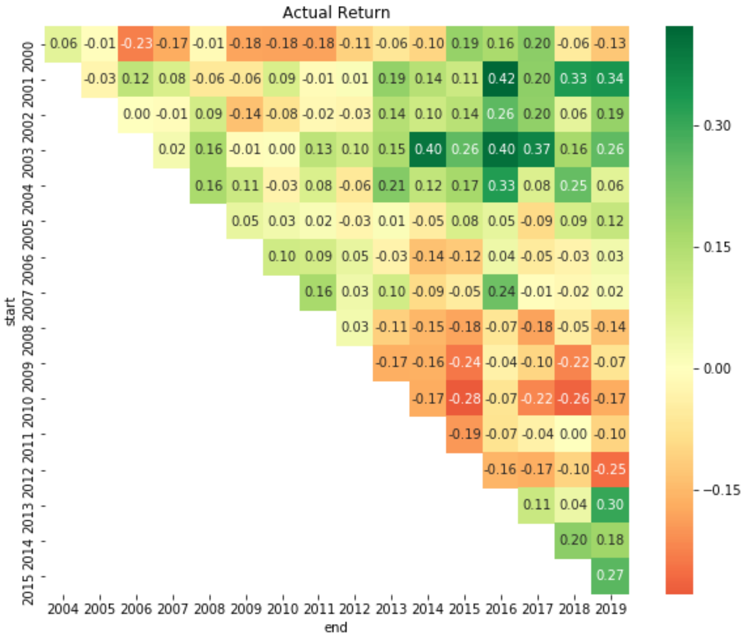


Figure 6: Actual Return Rate of Trained Model from 2000-2019

The Green colour shows a positive Actual Return Rate, which means the model performs better than the benchmark, and the Red colour indicates a negative Actual Return Rate. Putting all the data together, we can compute the average *actual return rate* (defined in Equation (27)) of all cells in the matrix as in Equation (28)

$$\overline{ARR} = \frac{\sum_{s=2000}^{2015} \sum_{e=s+5}^{2019} [ARR(s,e)]}{N} \dots \dots \dots \dots (28)$$

where

$$N = \frac{(1+m) * m}{2}, m = (2019 - 2000 - 4 + 1) = 16$$

The calculated result of the average actual return rate is **2.57%,** indicating that the overall performance of the Monthly Rebalance Policy is slightly better than the benchmark in 2000-2019. The best period is buying in during 2001-2003 and selling out in 2015-2017, which has a maximum profit 42% higher than the benchmark. The worst period is buying in during 2009-2011, and also selling out in 2015-2017, which suffers a -28% lower than the benchmark.

We further compared the Actual Return Rate of the models in Figure-6 with the Nominal Return Rate of the benchmark in Figure-7. The Nominal Return Rate of the benchmark indicates the average market performance. We discovered that the distribution of the Actual Return Rate of the models is similar to the Nominal Return Rate of the benchmark, which shows that the model gains higher when the market is bullish, while loss more during the bearish market.

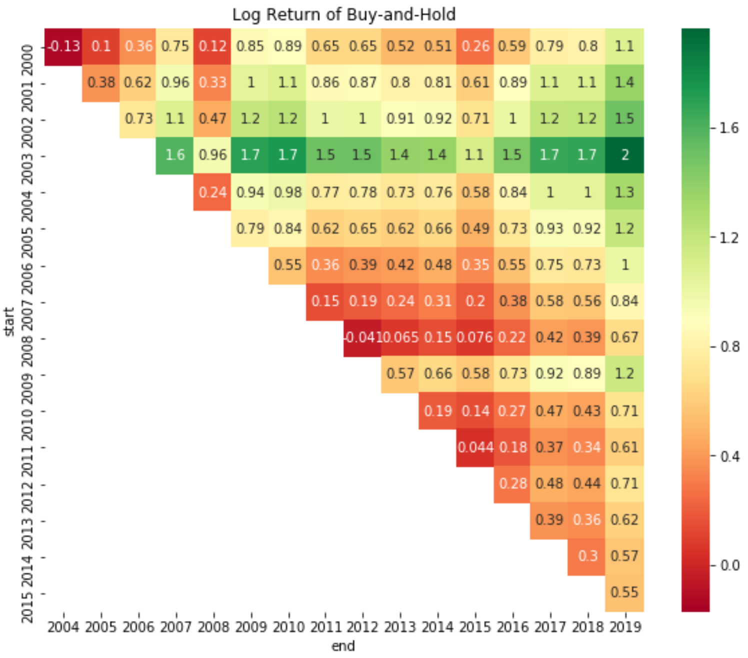


Figure 7: Nominal Return Rate of B&H Benchmark from 2000-2019

e. COMPARISON OF CRISIS AVOIDANCE ALGORITHM IN SELECTED PERIOD

As the previous result shows, the Monthly Rebalanced Model losses more during the bad market condition. Therefore, we implement a risk-avoidance policy that cash-out the assets if all the assets surge in a short period.

The risk-avoidance policy is based on the Maximum Drawdown (MDD) defined in Equation (25). We set a monthly window size with maximum drawdown threshold among 20%, 15% and 10%. That is, if the maximum drawdown of all markets in the past month exceeds the threshold, the model should cash out from all the market. Only after the MDD of one or more markets decreases below the threshold will the model return to the market. In the following section, we call the model without crisis-avoidance algorithm as the “Vanilla Policy”, and call the model with crisis-avoidance algorithm as the “Crisis-Avoidance Policy”

We tested the pre-trained model with and without the crisis avoidance policy. We select two periods to examine the behaviour for disaster avoidance of our models. The first period is 2001-2004 (in Fiture-8), which is not in the previous training or testing set. During the period, the market encounters the burst of dot-com bubble and 911 in 2001 and the SARS in 2003. The second period is 2007-2010, which is in the training set and covers the whole cycle of 2008 crisis (in Fiture-9).

The performance in 2001-2004 exhibits both similarities and differences in global markets. The markets went into different direction during the second quarter of 2001, the first quarter of 2002 and the first half of 2003 (Green Box in Figure-8). Certain rebalancing and relocation can be taken to ride the wave. On the other hand, the market behaviour during second half of 2001 and the second half of 2002 (Blue Box in Figure-8) shows the time frame when all the markets moving down simultaneously, in which the loss is unavoidable if no cash-out option is allowed.

The pattern from 2007-2010 (Figure-9) shows another story. All markets perform in nearly the same pattern, but the volatility differs. Therefore, the model may not be able to avoid losses during the falling period, but still has the option to control the degree of loss and rebalance to the fastest growing asset when the market recoveries.

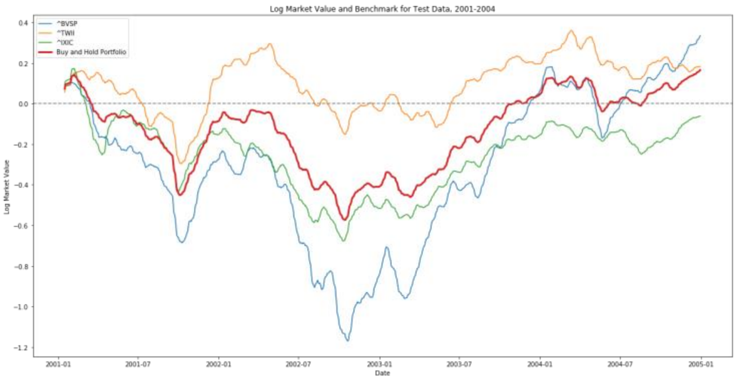


Figure 8: Log Market Value of Testing Dataset and Benchmark (2001-2004)

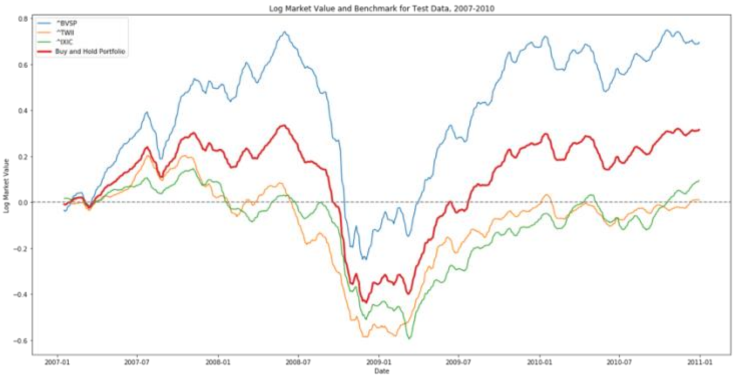
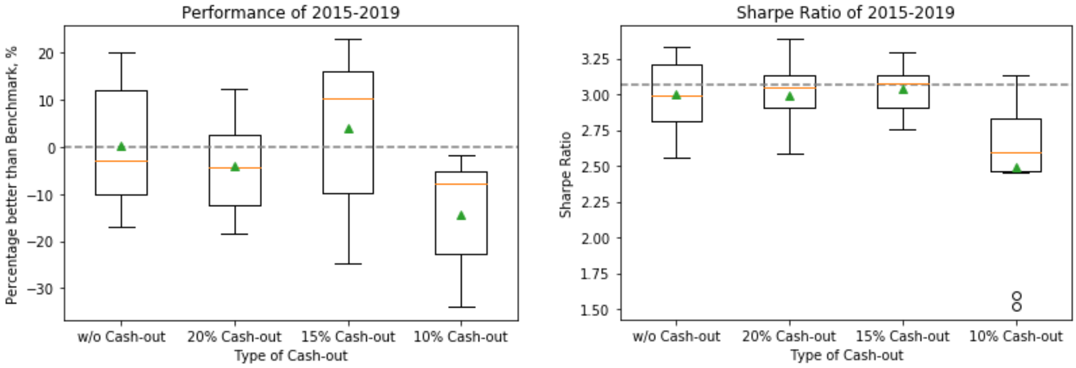


Figure 9: Log Market Value of Testing Dataset and Benchmark (2007-2010)

The detailed result is listed in Table-3:

Period	Cash-Out Threshold	Nominal Return Rate	Benchmark Nominal Return	Actual Return Rate	Sharpe Ratio	Benchmark Sharpe Ratio
2015-2019	w/o Cash-out	+77.9%	+72.7%	+0.2%	2.995	3.069
	20% MDD	+74.4%		-2.0%	2.995	
	15% MDD	**+78.1%**		**+0.0%**	**3.037**	
	10% MDD	+71.7%		-3.7%	2.489	
2001-2004	w/o Cash-out	+1.8%	+19.6%	-10.7%	-0.317	1.073
	20% MDD	-3.6%		-15.3%	-0.803	
	15% MDD	-0.2%		-12.4%	-0.006	
	10% MDD	**+3.2%**		**-9.5%**	**0.396**	
2007-2010	w/o Cash-out	+25.5%	+39.2%	-1.0%	1.365	1.894
	20% MDD	+32.8%		+4.2%	2.349	
	15% MDD	+40.6%		+9.8%	2.600	
	10% MDD	**+46.9%**		**+14.3%**	**2.711**	

Table 3: Returns and Sharpe Ratio in Different Periods with Crisis-Avoidance Policies



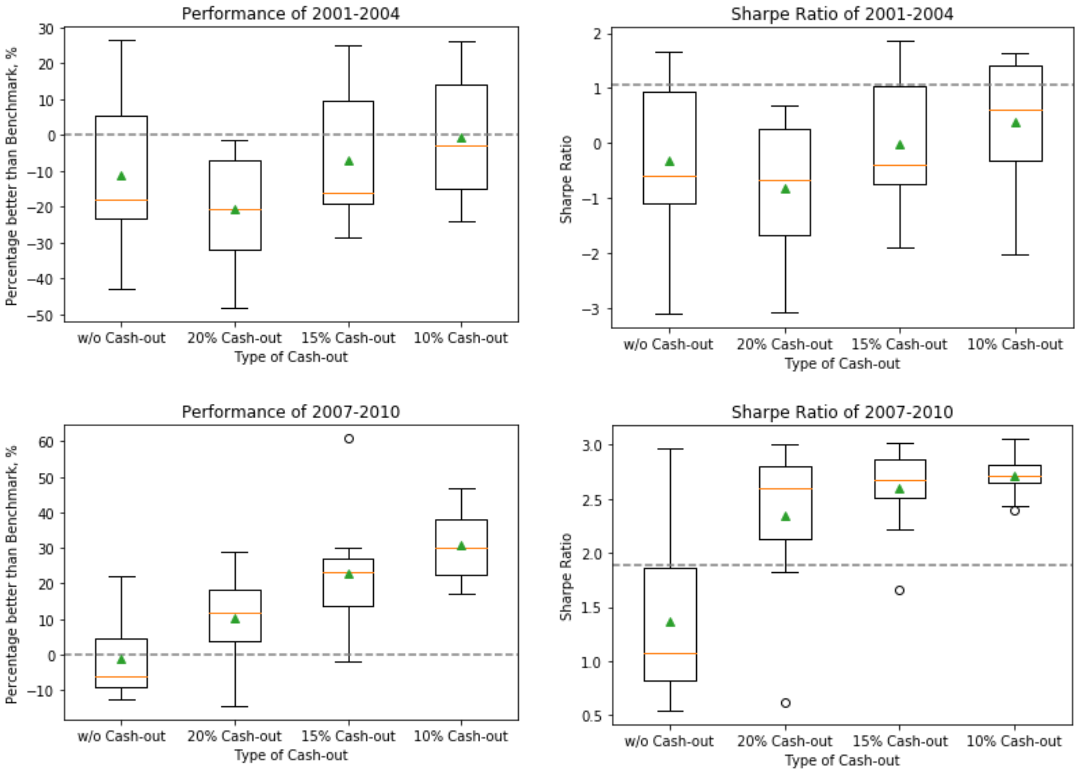


Figure 10: Returns and Sharpe Ratio in Different Periods with Crisis-Avoidance Policies

The grey dash line in the boxplot in Figure-10 represents the performance of the benchmark. As shown in Table-3 and Figure-10, during 2015-2019 the models perform as good as the benchmark in both absolute return and the sharpe ratio. The average return of the 15% Crisis-Avoidance Policies performs even higher than the benchmark. During 2001-2004, the performance of the models varies greatly. Some of the best performing model are still able to perform neary 30% better than the benchmark. During 2007-2010, the models significantly outperforms the benchmark with the cash-out options. The sharpe ratio of the best batch reaches 1.43 times against the benchmark.

The above experiment shows that the crisis avoidance algorithms is able to avoid 2008 crisis without hampering the performance in the 2015-2019 bullish market. The parameter should be adjusted from time to time, as the models with 20% or 15% MDD threshold gets the best result in 2015-2019, while the 10% MDD threshold receives the highest return in 2007-2010. The market performance in 2001-2004, on the other hand, is much more turbulence than the other two period. The trend changes within one month or two and therefore becomes a challenge to our monthly rebalanced framework. Hence, an algorithm to decide trading frequency is therefore important in the future work.

f. COMPARISON OF CRISIS AVOIDANCE ALGORITHM IN ALL PERIODS

In order to obtain the bigger picture about how the model works eventually, there is a need to conduct a check that run through the model in all period from 2000-2019 with all combination of window size and drawdown threshold. The definition of “all periods” is the same as in Section V/d. The cash-out window size is tested among monthly (20 working days), bi-weekly (10 working days) and weekly (5 working days). The cash-out threshold is selected in 0.20, 0.15 and 0.10. Therefore, there are in total 9 combinations of the parameters. We can plot the cash-out period given different parameter, as shown in Figure-11.

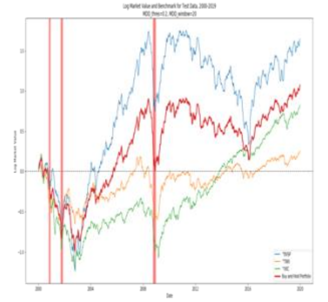
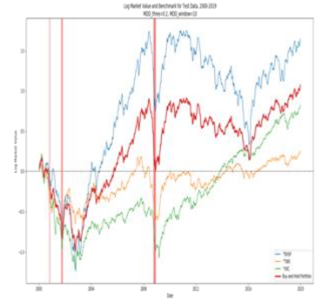
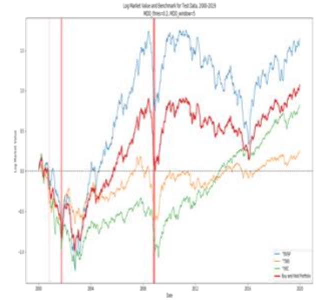
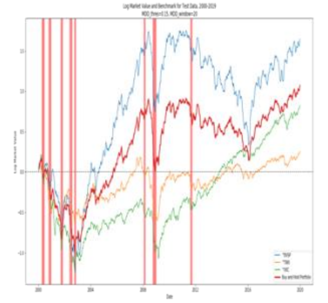
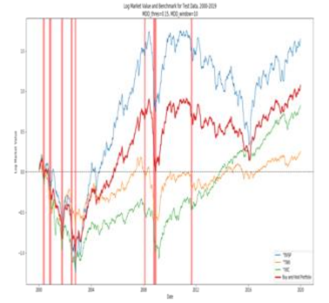
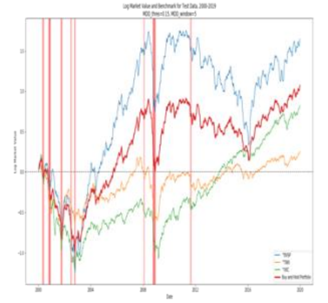
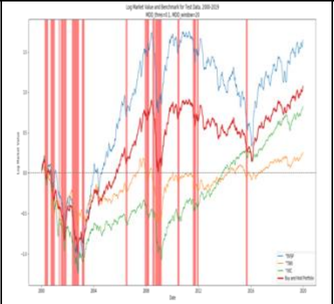
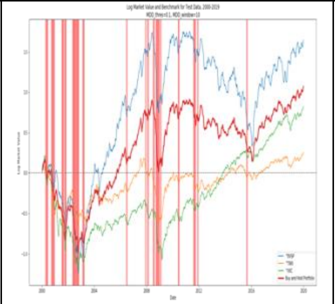
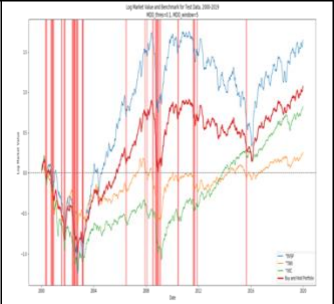
Thres	Monthly	Biweekly	Weekly
0.20			
0.15			
0.10			

Figure 11: Cash-out Windows in All Periods

As shown in Figure-11, the monthly window size has the longest minimum cash-out period, as it has the longest window size that the a MDD need to pass through before it has no effect on the decision. The bi-weekly and weekly window size is apparently more agile and the status changes faster than the monthly window size. However, it also causes excessive commission fee as there will be a fixed 0.25% loss for each cash-in/cash-out action.

We use the Log Nominal Return Rate and the Average Nominal Return Rate to measure the performance of the combination of parameters of the Crisis-Avoidance Policies. Instead of the Nominal Return Rate, we take the logarithm to make the colour on the heatmap linearly distributed, as show in Figure-12. Table-4 shows the average Nominal Return Rate of all the models and the benchmark in all periods. Figure-13 and Table-5 shows the Actual Return Rate the Crisis-Avoidance Policies in all periods.

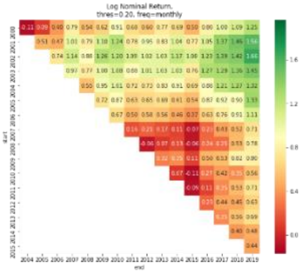
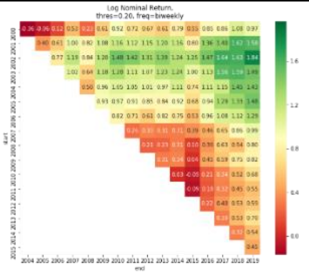
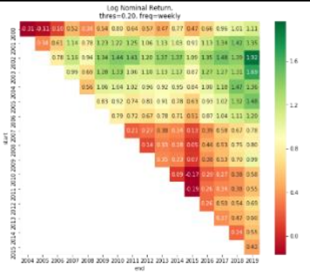
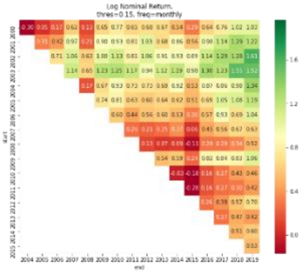
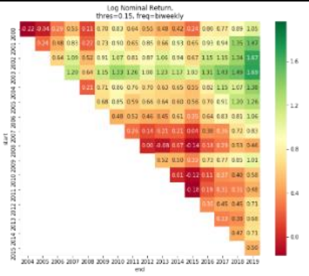
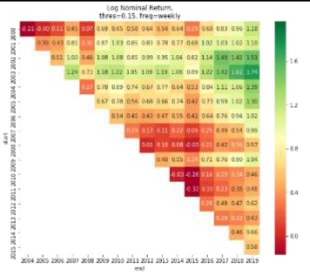
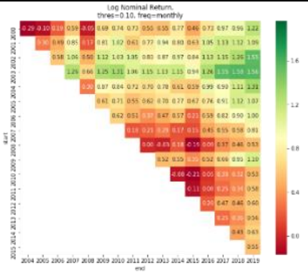
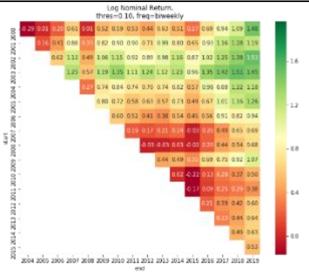
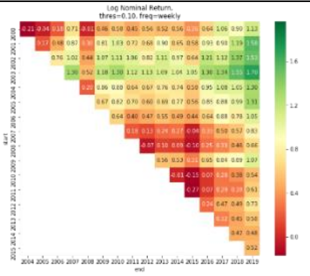
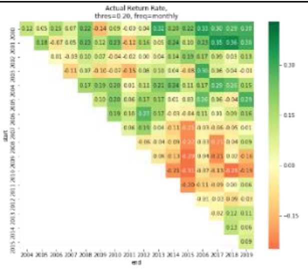
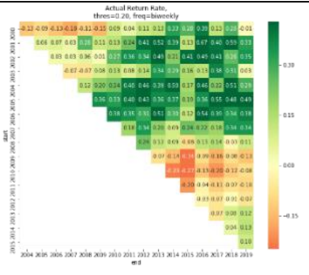
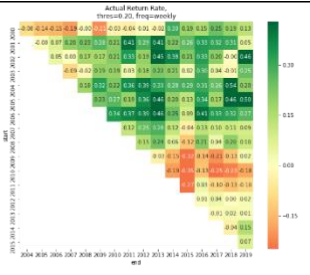
Thres	Monthly	Biweekly	Weekly
0.20			
0.15			
0.10			

Figure 12: Log Nominal Return Rate for Crisis-Avoidance Policies in All Periods

Threshold	Monthly	Biweekly	Weekly
0.20	+121.70%	+148.46%	+139.99%
0.15	+110.66%	+111.12%	+110.89%
0.10	+110.29%	+111.69%	+111.40%
Benchmark	+126.53%		

Table 4: Average Nominal Return Rate for Crisis-Avoidance Policies in All Periods

Thres	Monthly	Biweekly	Weekly
0.20			

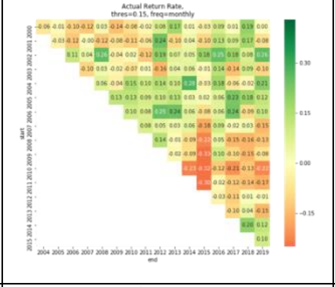
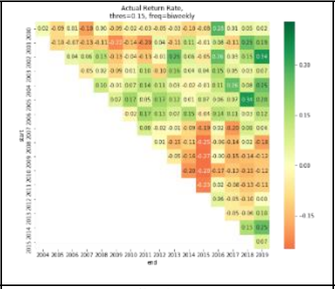
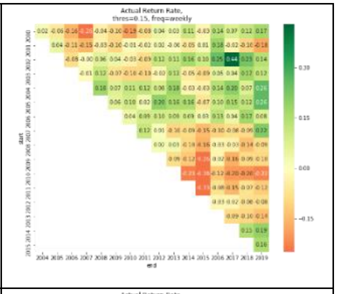
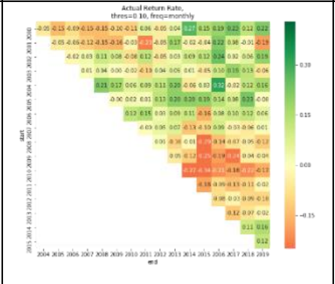
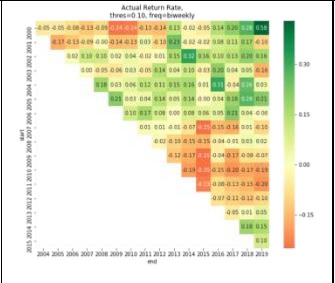
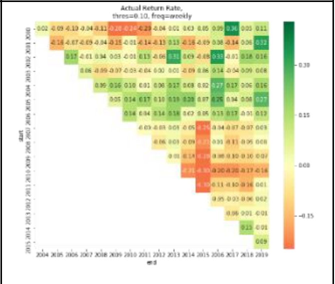
0.15			
0.10			

Figure 13: Actual Return Rate for Crisis-Avoidance Policies in All Periods

Threshold	Monthly	Biweekly	Weekly
0.20	+5.60%	+17.20%	+13.43%
0.15	+0.96%	+0.45%	+0.44%
0.10	+0.37%	+0.72%	+0.60%

Table 5: Average Actual Return Rate for Crisis-Avoidance Policies in All Periods

Although the Nominal Return Rate of the Benchmark is higher than some Crisis-Avoidance Policies, considering the impact of currency, all the actual return of the Crisis-Avoidance Policies are positive. The result shows a significant improvement of actual return rate with 20% bi-weekly maximum drawdown as the crisis avoidance policy that is 17.2% higher than the benchmark. There is also improvement of 20% weekly drawdown compared to the monthly drawdown, but the extent is less than the bi-weekly window size. As shown in Figure-13, the improvement are mostly the period starting in 2002-2003. Comparing the monthly, bi-weekly and weekly window size in Figure-11, although the cash-out period is similar, the difference execution date in the crisis period has a large impact in the following period. The improvement over 2007-2010 period is significant indicating that the crisis avoidance policy is able to tackle the 2008 crisis. However, the 2009-2015 period is still challenging for both actual return rate and the log nominal return rate, indicating that there are challenges for the current policies to handle the horizontal markets.

VI. CONCLUSION

Comparing to models in [27] [28] [29], which generally develop one model with one group of assets and measures its performance in one period of time, this paper proposes a more general training scheme for a Reinforcement Agents that is extensible to indefinite number of financial assets.

In this paper, we proposed a novel open-source Reinforcement Learning trading framework with OpenAI Gym Environment plus Actor-Critic Model, and Stable Baseline Proximal Policy Optimization method. The framework can theoretically support unlimited assets in the portfolio, and we experiment with three assets throughout the paper. The experiments demonstrated that the Environment with an Observation Space of EMA, MACD, Delta Reverse Days and Leakage information provide sufficient information for an Agent to act upon. The rule-based finite Action Space proves to be significantly better than the infinite Action Space for reducing the excessive dimensionality. A monthly-rebalancing scheme has been proven to be better than weekly and daily rebalancing as the Agent tends to abuse the window period and cause unnecessary leakage in the current setting. We further tested the crisis detection and avoidance algorithm, and proved that a bi-weekly 20% maximum drawdown achieves significantly better performance with 17.2% higher profit than the benchmark.

VII. References

1. Brockman, G., et al., *Openai gym.* arXiv preprint arXiv:1606.01540, 2016.
2. Hill, A., et al., *Stable Baseline*. 2018, GitHub: GitHub repository.
3. Stewart, S., C. Piros, and J. Heisler, *Portfolio Management*. 1st ed. 2019: Weiley.
4. Markowitz, H., *Portfolio Selection.* The Journal of Finance, 1952. **7**(1): p. 77-91.
5. Kim, Y. and D. Enke, *Using Neural Networks to Forecast Volatility for an Asset Allocation Strategy Based on the Target Volatility*, in *Complex Adaptive Systems*, C.H. Dagli, Editor. 2016. p. 281-286.
6. Vorlow, C.E., *Simple Tactical Asset Allocation Strategies on the S&P 500 and the Impact of VIX Fluctuations*. Handbook of Investors' Behavior during Financial Crises, ed. F. Economou, et al. 2017. 383-400.
7. Kim, Y. and D. Enke, *A dynamic target volatility strategy for asset allocation using artificial neural networks.* Engineering Economist, 2018. **63**(4): p. 273-290.
8. Mattei, M.D. and N. Mattei, *Analysis of fixed and biased asset allocation rebalancing strategies.* Managerial Finance, 2016. **42**(1): p. 42-50.
9. Wu, H., C.Q. Ma, and S.J. Yue, *Momentum in strategic asset allocation.* International Review of Economics & Finance, 2017. **47**: p. 115-127.
10. Faber, M., *A Quantitative Approach to Tactical Asset Allocation Revisited 10 Years Later.* Journal of Portfolio Management, 2018. **44**(2): p. 156-167.
11. Hayes, M., J.A. Primbs, and B. Chiquoine, *A Penalty Cost Approach to Strategic Asset Allocation with Illiquid Asset Classes.* Journal of Portfolio Management, 2015. **41**(2): p. 33-41.
12. Wang, P. and J. Spinney, *Strategic Asset Allocation: Combining Science and Judgment to Balance Short-Term and Long-Term Goals.* Journal of Portfolio Management, 2017. **44**(1): p. 69-82.
13. Diris, B., F. Palm, and P. Schotman, *Long-Term Strategic Asset Allocation: An Out-of-Sample Evaluation.* Management Science, 2015. **61**(9): p. 2185-2202.
14. Franz, R., *Macro-Based Parametric Asset Allocation.* 2013.
15. Bisetti, E., et al., *A Multivariate Model of Strategic Asset Allocation with Longevity Risk.* Journal of Financial and Quantitative Analysis, 2017. **52**(5): p. 2251-2275.
16. Laborda, R. and J. Olmo, *Optimal asset allocation for strategic investors.* International Journal of Forecasting, 2017. **33**(4): p. 970-987.
17. Liu, J.S., et al., *Data envelopment analysis 1978–2010: A citation-based literature survey.* Omega, 2013. **41**(1): p. 3-15.
18. Henrique, B.M., V.A. Sobreiro, and H. Kimura, *Literature review: Machine learning techniques applied to financial market prediction.* Expert Systems with Applications, 2019. **124**: p. 226-251.
19. Sutton, R.S. and A.G. Barto, *Introduction to reinforcement learning*. Vol. 135. 1998: MIT press Cambridge.
20. OpenAI. *Background: Why Gym? (2016)*. 2016; Available from: https://gym.openai.com/docs/.
21. Moody, J., et al., *Performance functions and reinforcement learning for trading systems and portfolios.* Journal of Forecasting, 1998. **17**(5-6): p. 441-470.
22. Neuneier, R. *Enhancing Q-learning for optimal asset allocation*. in *Advances in neural information processing systems*. 1998.
23. Dash, R. and P.K. Dash, *A hybrid stock trading framework integrating technical analysis with machine learning techniques.* The Journal of Finance and Data Science, 2016. **2**(1): p. 42-57.
24. Liu, F., C. Quek, and G.S. Ng, *A novel generic hebbian ordering-based fuzzy rule base reduction approach to Mamdani neuro-fuzzy system.* Neural Computation, 2007. **19**(6): p. 1656-1680.
25. Quek, C., et al., *Investment portfolio balancing: application of a generic self-organizing fuzzy neural network (GenSoFNN).* Intelligent Systems in Accounting, Finance & Management: International Journal, 2009. **16**(1-2): p. 147-164.
26. Garcia-Galicia, M., A.A. Carsteanu, and J.B. Clempner, *Continuous-time reinforcement learning approach for portfolio management with time penalization.* Expert Systems with Applications, 2019. **129**: p. 27-36.
27. Gupta, S., *A risk-sensitive stock trading system with the application of reinforcement learning (Q-learning).* 2017.

28. Lim, Q.Y.E., *Dynamic Portfolio Rebalancing using Genetic Algorithm and Reinforcement Learning.* 2019.
29. Miharja, T., *CERS-DR: cycle-based ETF rotation strategy with dynamic rebalancing with the application of reinforcement learning and neural network.* 2019.
30. Kang, Q.M., et al., *An Asynchronous Advantage Actor-Critic Reinforcement Learning Method for Stock Selection and Portfolio Management*. Proceedings of the 2018 2nd International Conference on Big Data Research. 2018. 141-145.
31. Jaderberg, M., et al., *Reinforcement learning with unsupervised auxiliary tasks.* arXiv preprint arXiv:1611.05397, 2016.
32. Schulman, J., et al., *Proximal policy optimization algorithms.* arXiv preprint arXiv:1707.06347, 2017.
33. Padial, D.L., *Technical Analysis Library in Python*. 2018, GitHub.