Regime-switching recurrent reinforcement learning for investment decision making

Liting Chiang, Siddharth Somani, Chris Yi

Class Projection Presention for AMS 691

Dec 16, 2020

Outline

Outline

Motivations

- Recently years, machine learning models have been a hop topic in the quantitative finance.
- Many efforts have been spent by researchers of searching and developing trading strategies based on ML models.
- Recurrent reinforcement learning (RRL), an online learning technique which finds approximate solutions to stochastic dynamic programming problems, are explored by researchers to tune financial trading systems for the purpose of utility maximization.
- Regime switching (RS) models have been well developed but few applications of RS models concerned with ML being found in the financial trading field.

Major Results

- Researchers propose RSRRL model which is a regime-switching extension of the recurrent reinforcement learning (RRL) algorithm and describes its application to investment problems.
- Maringer and Ramtohul (2012) gives two variants of the RSRRL, namely a threshold version and a smooth transition version.
- RSRRL model shows better out of sample performance than RRL in the performance comparison of automated trading experiments.

Contribution

- This project has shown that a reinforcement learning approach can outperform a buy-and-hold strategy, however results between the regular reinforcement learning trader regime switching approaches are mixed. Further refinement of the hyper-parameters may be needed in order to perfect the strategy.
- We have also moved one-step forward and extend the one-time regime switching to a rolling window dynamic regime switching.
- With no external library to call, we build the entire algorithm of RSRRL from button up and packed the codes into nice class structure and could be published as a new simple Python library.

Outline

Outline

Recurrent Reinforcement Learning: Foundation

- The stereotype type or the basic type of reinforcement learning in quantitative finance was proposed in these 2 papers.
- Moody and Saffell (1998), Reinforcement Learning for Trading Systems and Portfolios
- Moody and Saffell (1999), Reinforcement learning for trading
- Let's first consider the simplest case that a trader can only trade a single security.

Single Asset Model

- Assume the action set at time t is $F_t = [-1, 1]$, this is also the position this trader takes at time t.
- Let the return time-series of the traded security be r_t
- The decision function that takes into account transactions costs and market impact would be:

$$\begin{aligned} F_t &= \tanh\left(\sum_{i=0}^{m-1} w_i r_{t-i} + w_m F_{t-1} + w_{m+1}\right) \\ &= \tanh(\theta^T x_t) \end{aligned} \tag{1}$$

- It means that the position at time t is decided by the rolling m-day returns and previous position, with an additional bias term.

Single Asset Model

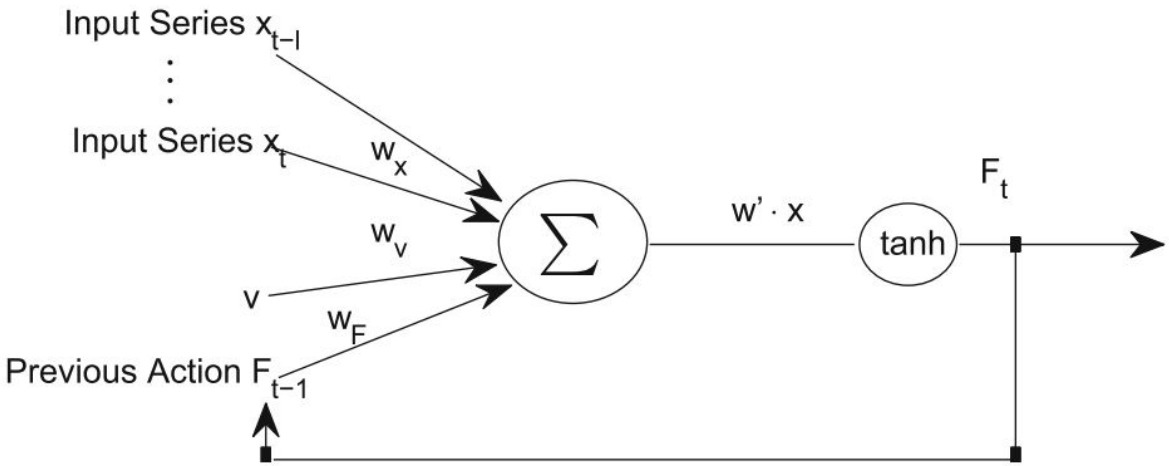


Fig. 1. Recurrent reinforcement learning

Figure: Recurrent reinforcement learning The, source: Zhang (2014)

Single Asset Model

- The realized return at t is given by:

$$R_t = F_{t-1} * r_t - \delta |F_t - F_{t-1}| \tag{2}$$

where δ is the transaction cost

- The return is modeled as asset return multiplied by the position the trader takes subtracted by the transaction cost rate times change in position.
- Rather than maximizing profits, most modern fund managers attempt to maximize risk-adjusted return as advocated by MPT.
- Choose Sharp Ratio as the objective function:

$$S_t = \frac{A_t}{\sqrt{B_t - A_t^2}} \tag{3}$$

where $A_t = \frac{1}{T}\sum_{t=1}^{T} R_t$ and $B_t = \frac{1}{T}\sum_{t=1}^{T} R_t^2$

Gradient Ascent

- The entire calculation is to find the derivative of S with respect to w.
- First is to split the derivative of S with respect to w into different pieces using the chain rule

$$\begin{aligned}\frac{dS_t}{dw} &= \frac{dS_t}{dA}\frac{dA}{dw} + \frac{dS_t}{dB}\frac{dB}{dw} \\ &= \sum_{t=1}^{T}\left(\frac{dS_t}{dA}\frac{dA}{dR_t} + \frac{dS_t}{dB}\frac{dB}{dR_t}\right)\frac{dR_t}{dw} \\ &= \sum_{t=1}^{T}\left(\frac{dS_t}{dA}\frac{dA}{dR_t} + \frac{dS_t}{dB}\frac{dB}{dR_t}\right)\left(\frac{dR_t}{dF_t}\frac{dF_t}{dw} + \frac{dR_t}{dF_{t-1}}\frac{dF_{t-1}}{dw}\right)\end{aligned} \tag{4}$$

Gradient Ascent

- Second is the calculate the derivative of each sub-pieces.

$$\frac{dR_t}{dF_t} = -\delta \cdot sgn(F_t - F_{t-1}) \tag{5}$$

$$\frac{dR_t}{dF_{t-1}} = r_t + \delta \cdot sgn(F_t - F_{t-1}) \tag{6}$$

$$\frac{dF_t}{dw} = (1 - \tanh(\theta^T x_t)^2)\left(x_t + w_m \frac{dF_{t-1}}{dw}\right) \tag{7}$$

- Besides security returns r_t, the RRL model can easily accommodate technical indicators or other economic variables that might have an impact on the security.

Regime-switching RRL

- Since the development of this single layer RRL model, many efforts have been spent on improving the performance.
- Maringer and Ramtohul (2012) proposed an interesting method that incorporate regime switch model with RRL and introduce so called Regime-switching RRL.
- The core idea is quite simple: RRL model learns the action or position at each period from the entire training data
- However, what if there are several significant market changes, thus multiple market pattern, in the historical data used for training.
- RSRRL defines more than one learners and let each learn from different market regime.

2 Regimes RSRRL Model

- There are many different methods for market regime change detection, like Markov-Switching model, smooth transition model, etc.
- Suppose a certain regime change detection is applied and results in a regime indicator time-series G_t and $G_t = 0$ or 1.
- Then the action function is given by:

$$F_t = y_{t,1}(G_t) + y_{t,2}(1 - G_t) \tag{8}$$

- And each $y_{t,i}$ i $= 1$ or 2 has a similar form with (1):

$$y_{t,j} = \tanh\left(\sum_{i=0}^{m-1} w_{i,j} r_{t-i} + w_{m,j} F_{t-1} + w_{m+1,j} v\right) \tag{9}$$

2 Regimes RSRRL Model

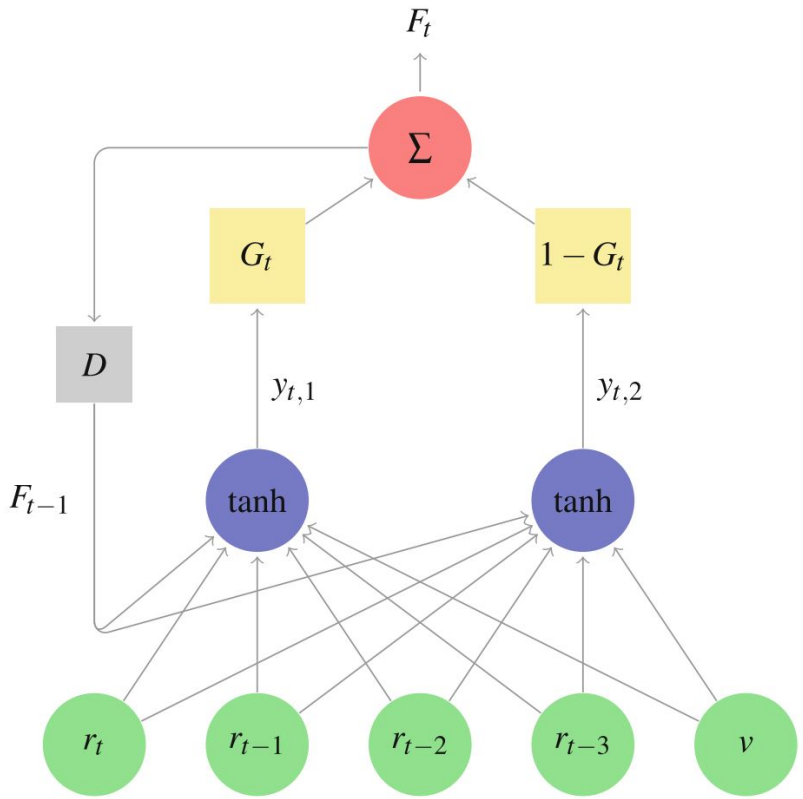


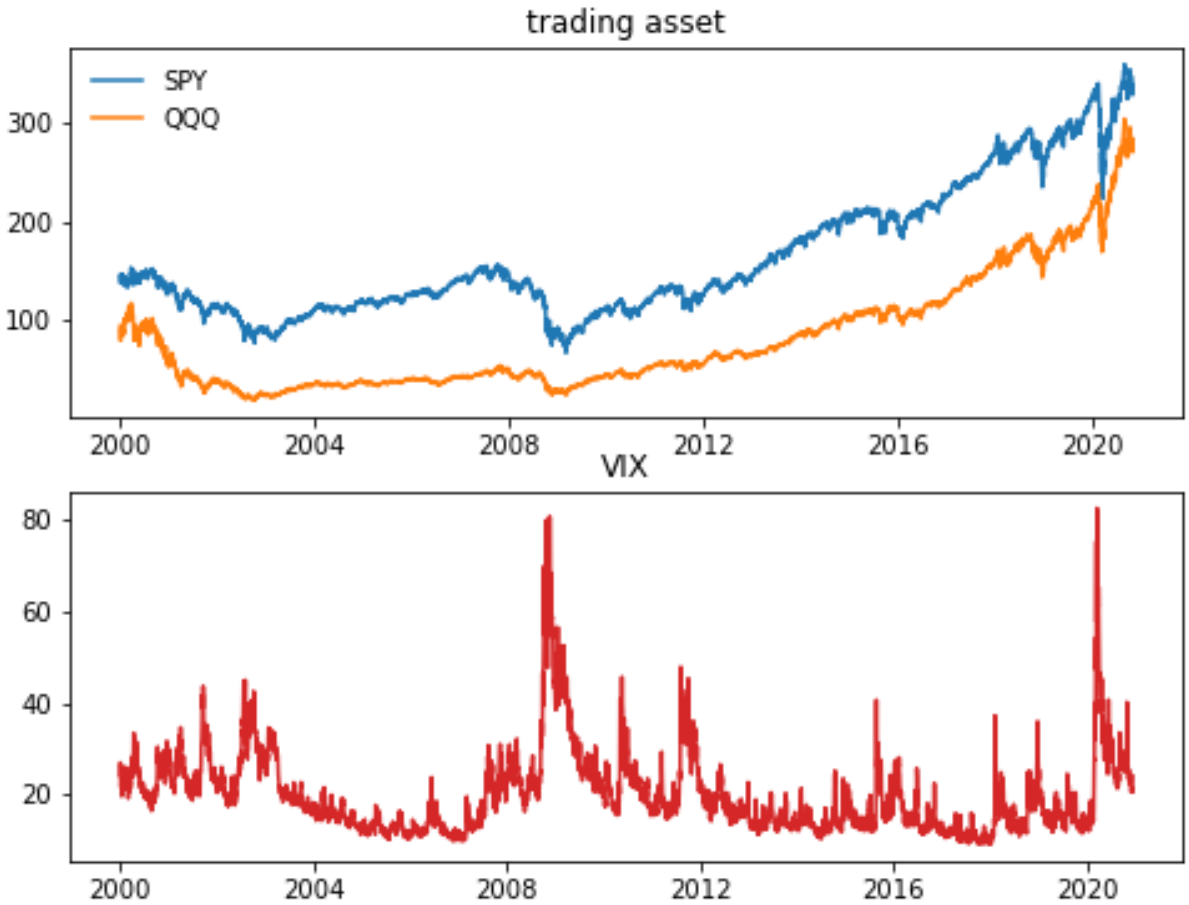
Figure: RSRRL network structure where D is the delay operator and m = 4 variable, source: Maringer and Ramtohul (2012)

Data and Methodology

- In this project, historical prices and returns of SPY ETF and QQQ ETF data are used as trading assets. And historical price of VIX is used as an alternative market regime indicator.
- For the regime switching model, we use a simple 2 states Hidden Markov Model. The original paper uses TAR or 'threshold auto-regressive model' and STAR or 'smooth transition auto-regressive'.
- Transaction cost or commission is set as 0.0025%.

Data and Methodology

- Below is the trading assets and VIX index.



Outline

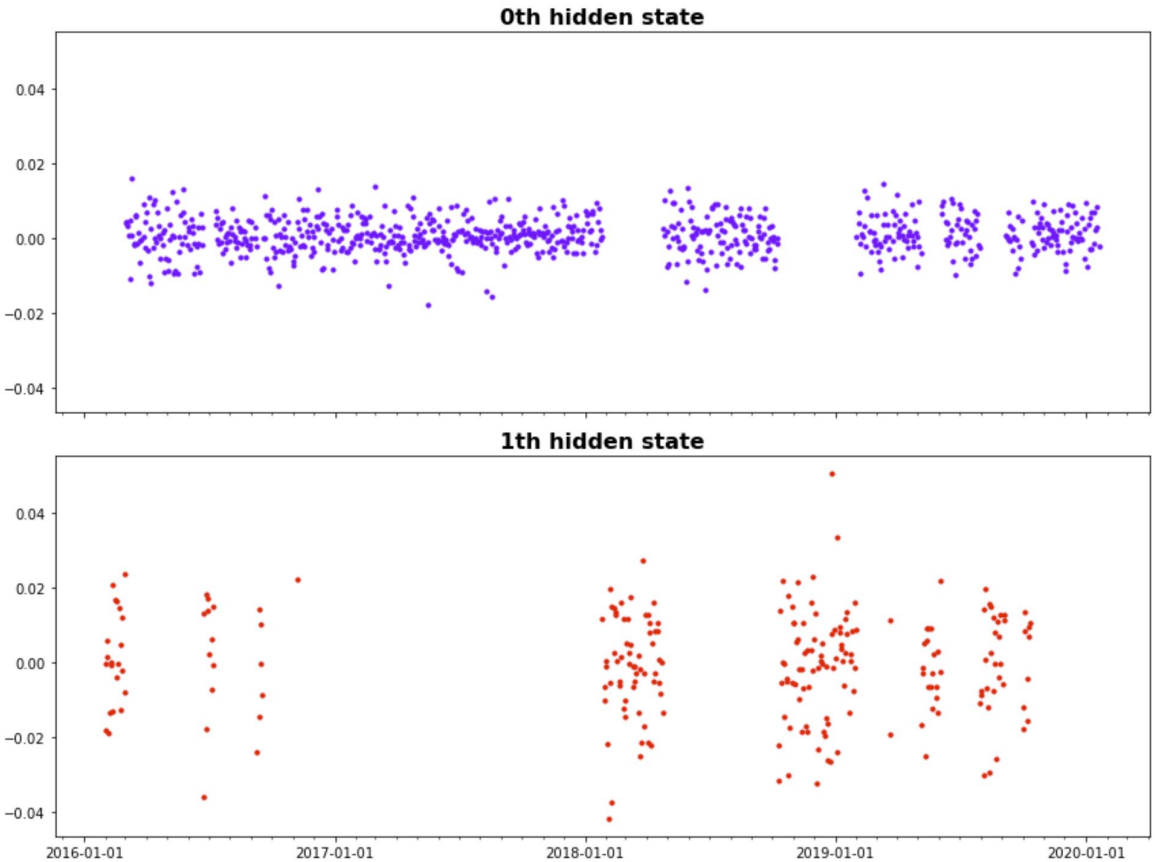
Outline

Experiment 1 & 2

- The time-series data consists of 1200 trading days ending at Oct.30 2020. While the first 1000 trading days are used as training data and the remaining 200 trading days are used as testing data.
- The training process is split into two parts: the first part fitted a Hidden Markov model to the training data, and the second part trained the trading agent to the predicted regimes
- For regime estimation, we used the hmm.learn package (link: `https://hmmlearn.readthedocs.io/en/latest/`
- For the hyper-parameters of our model, we used 5 trailing days and a learning rate of 0.3, with 2000 training epochs.
- First two major experiments compare the use of 2 different market regime indicators: one is the normalized log-returns of the trading asset, the other one is VIX index.

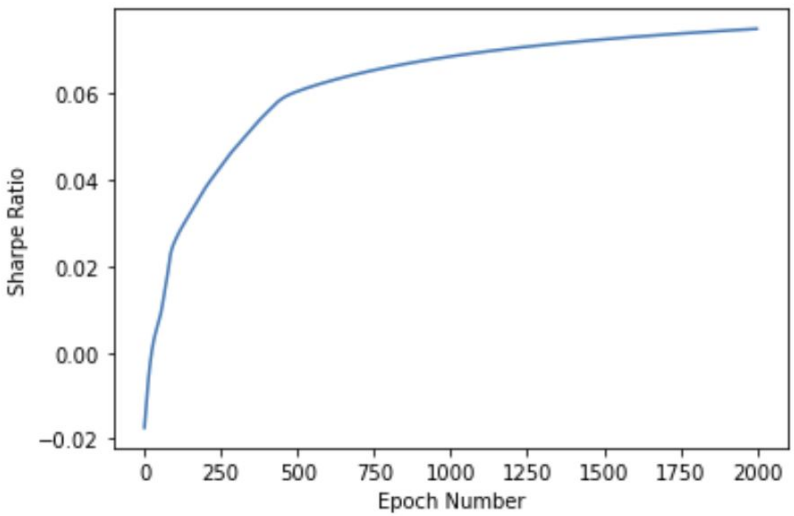
HMM for Regime-Switching: Asset Return

- Exp1 trains a 2 state HMM model for regime-switching based on normalized SPY returns.
- 2 distinct regimes for low/high volatility.

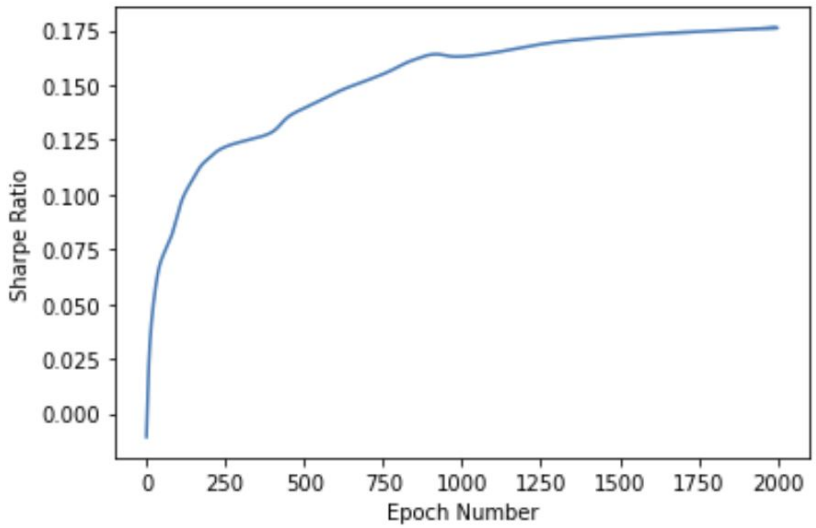


Model Training

- These graphs demonstrate the Sharpe ratio of the trading agents throughout each training epoch
- The increasing Sharpe ratio is a positive sign that the learning process is heading in the proper direction



(a) RRL



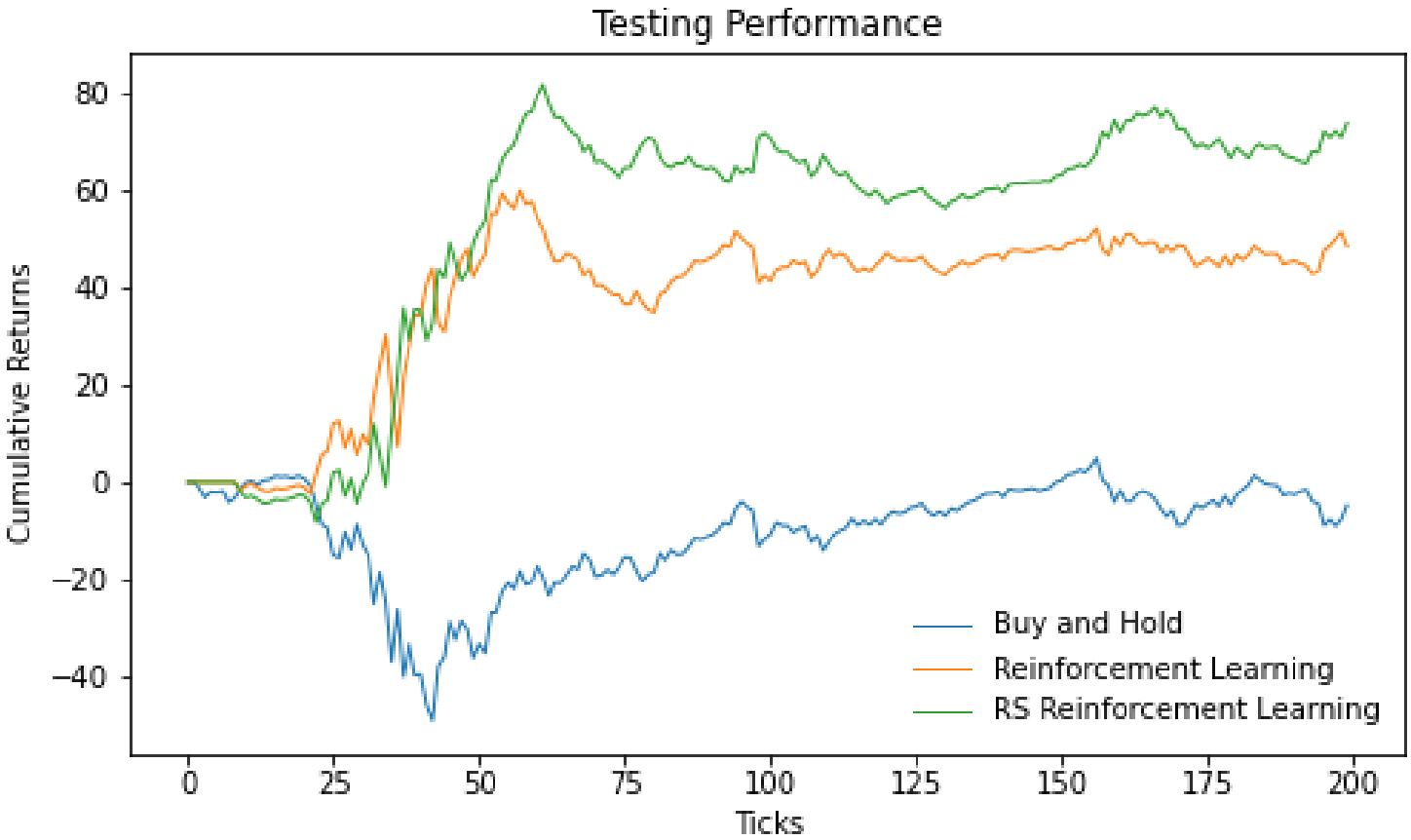
(b) RSRRL

Training Performance

- The training performance 1000 training days is shown below:

Testing Performance

- The training performance is as below:



HMM for Regime-Switching: VIX Index

- First training a 2 state HMM model for regime-switching based on VIX data
- Notice the distinct regimes

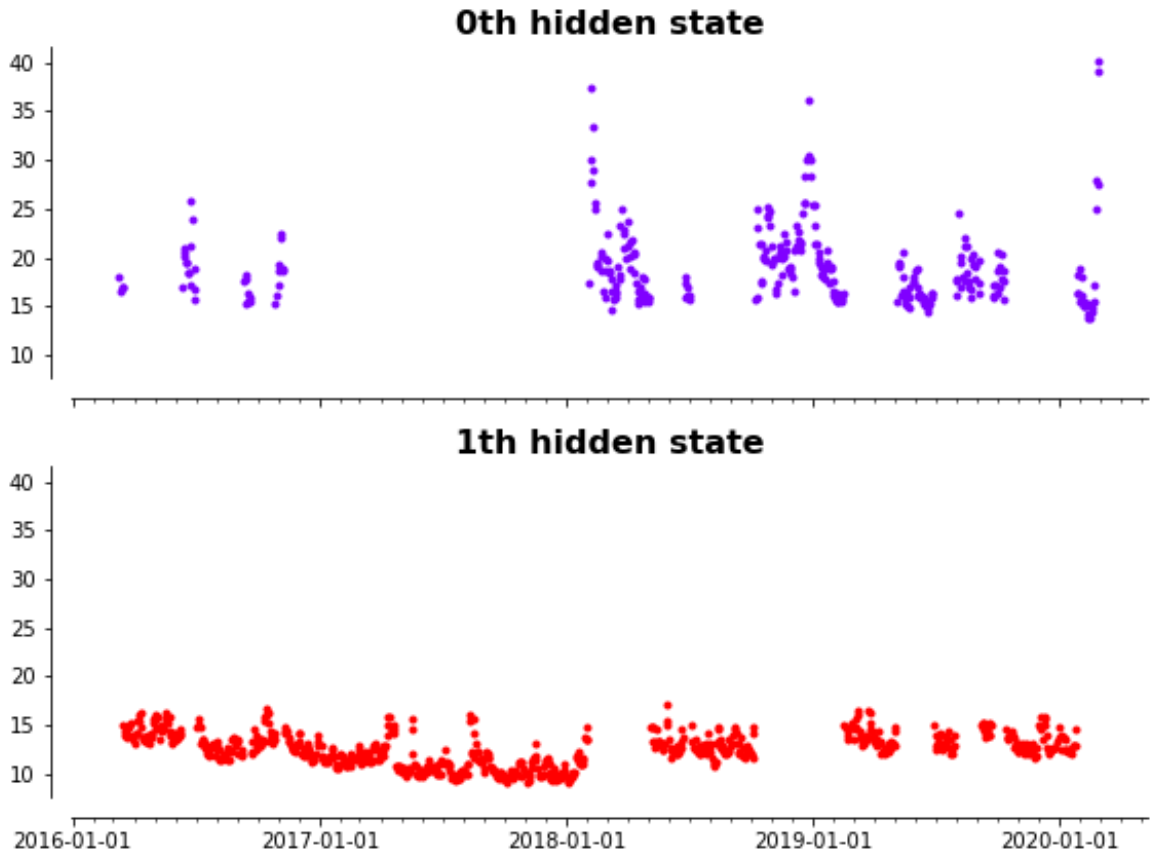
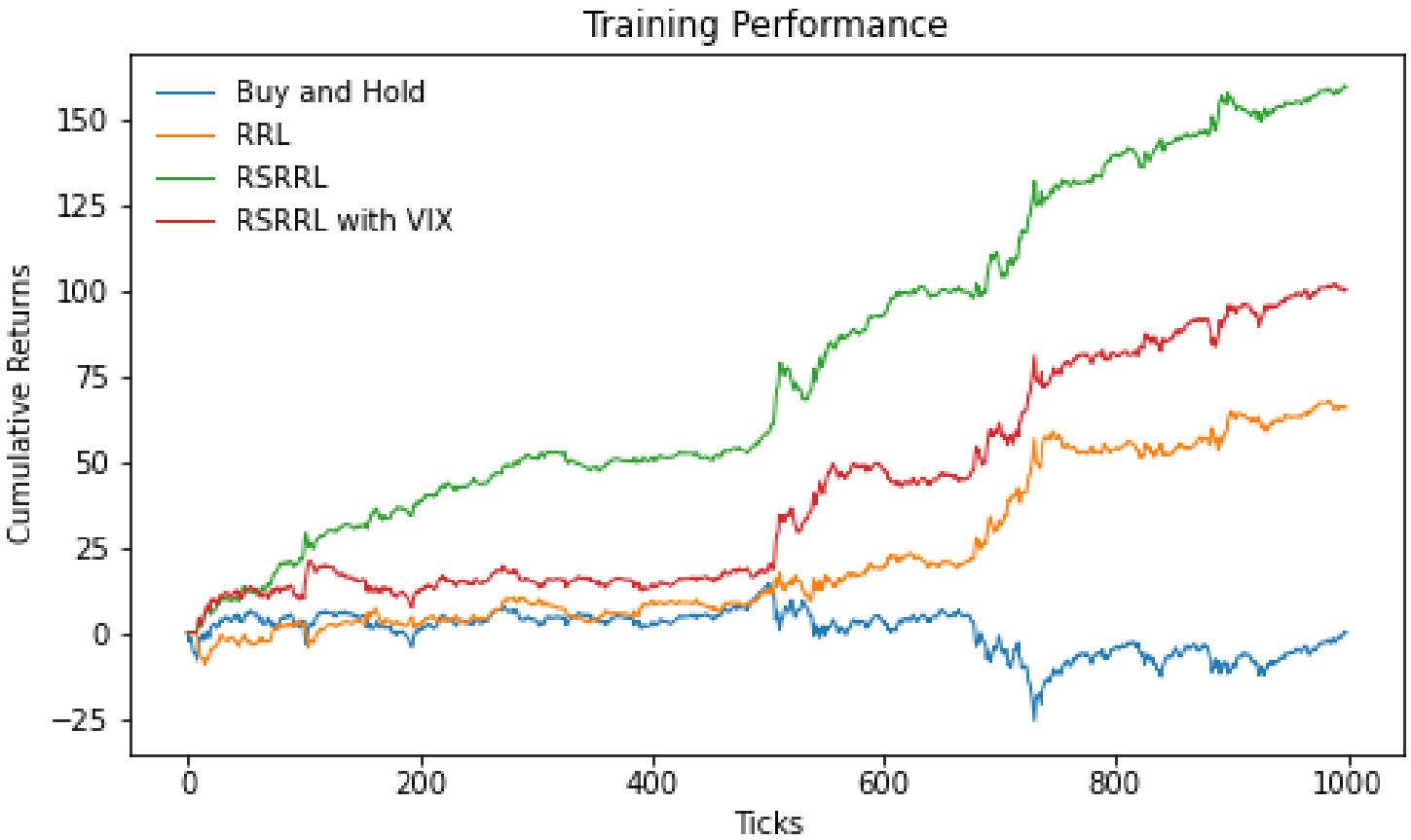


Figure: Regime-switching within training period

Training Performance

- The training performance of 1000 training days is shown below:



Testing Performance

- The testing performance of 200 testing days is as below:

Outline

Outline

Experiment 3

- For our third experiment, we enhanced the complexity of our model
- Rather then fitting regimes to the SPY log-return data, we used the VIX index as the indicator for overall market risk, and we estimtaed regimes based on VIX and traded based on the underlying VIX-predicted regime
- We also enhanced our calculations by implementing a rolling window exercise - this increases the number of hyperparameters we overall have to adjust for, since we now introduce a training window and a test window
- The other hyper parameters were kept the same as from the first experiment.

Experiment 3

- Our rolling window exercise is implemented as follows: We segment our data into rolling windows of $\{250, 500, \text{or } 750\}$ trading days, with 15 subsequent test days. For each window, a 2-regime model is fitted, and a RRL or RSRRL trader is trained on each window and then tested on the subsequent 15 trading days. The training window is then shifted by 15 days. The test results are compiled and calculated
- Based on experiments, changing the testing window did not seem to have significant impacts on the results - however, different training window lengths did. We show the results of setting the training window at 250, 500, and 750 days

250 Training Days, 15 Test Days

- Despite relatively strong early performance, the 250-RSRRL trader under-performs a benchmark buy-and-hold strategy, as well as the benchmark single-regime trader.

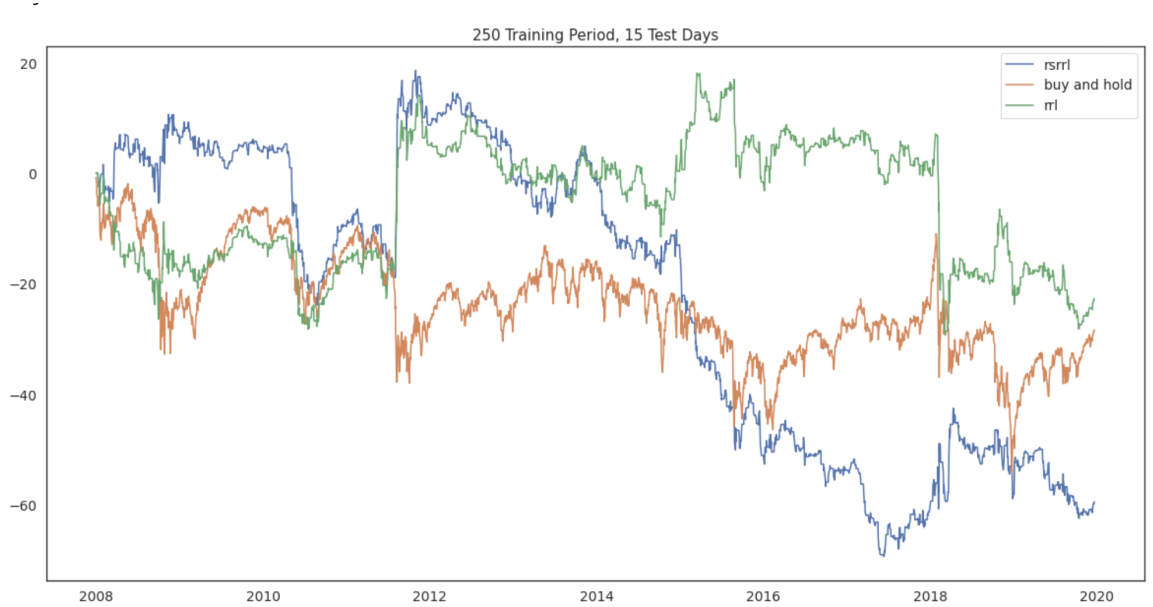


Figure: Rolling Window Exercise: 250 Days

500 Training Days, 15 Test Days

- As the training window was expanded, the reinforcement learning trading performance actually declined, contrary to expectations
- The single regime-model still outperformed the regime-switching model as well in terms of cumulative returns

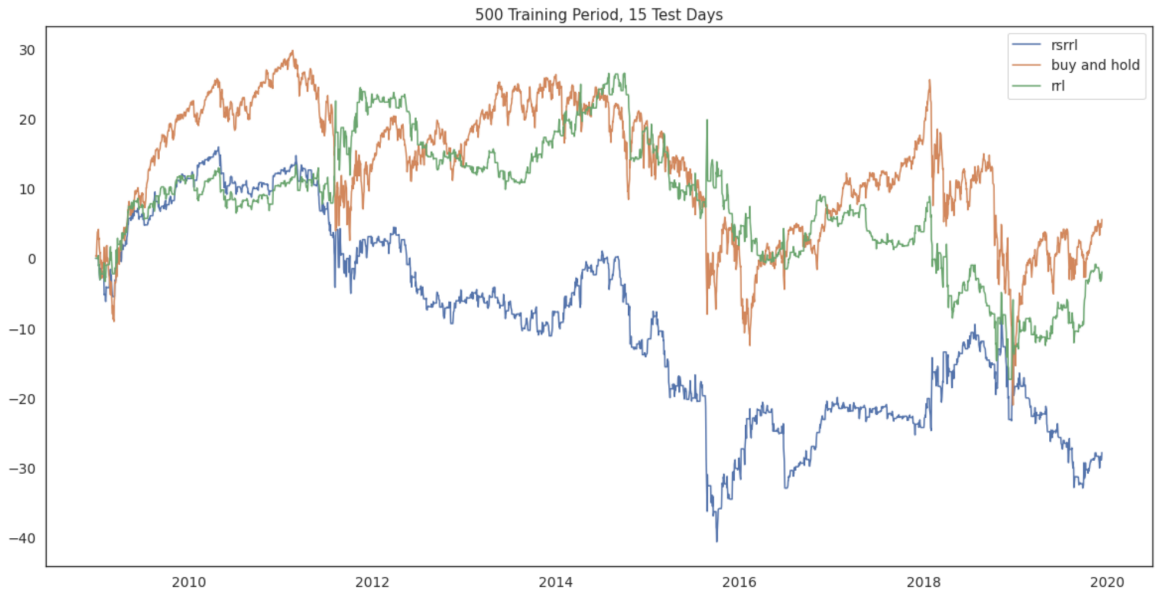


Figure: Rolling Window Exercise

750 training Days, 15 test days

- Once the training window was expanded to 750 days, however, we began to see significant improvements in the model performance
- It was at a 750 training day window (roughly 3 years worth of data) that both reinforcement learning models finally outperformed the benchmark buy and hold strategy, however the single regime model still outperformed the regime-switching model

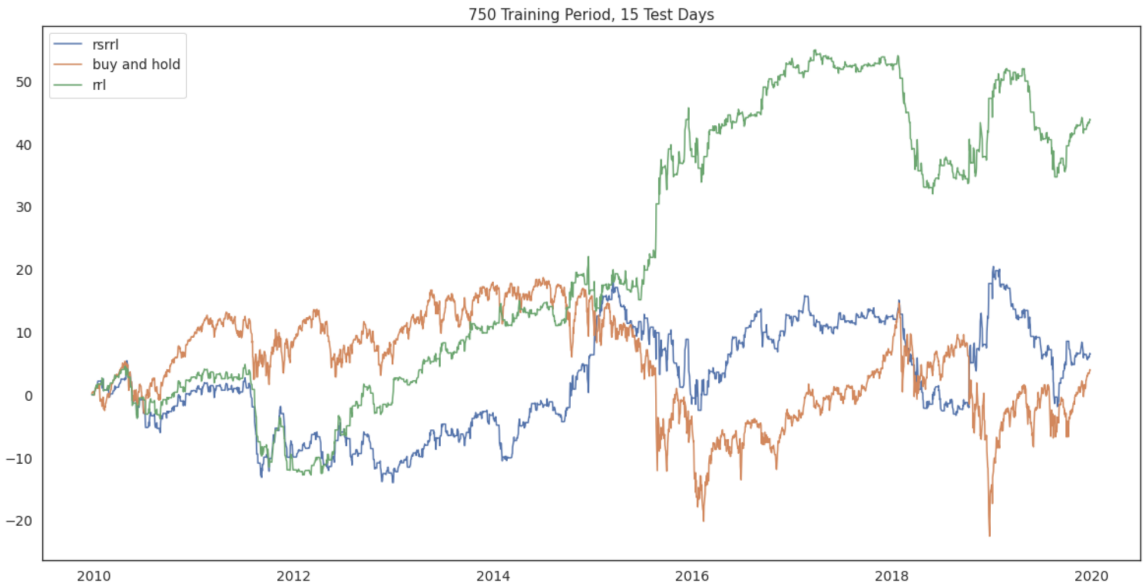


Figure: Rolling Window Exercise

Experiment 4

- For our fourth model we tried to determine the performance of our model during market crashes
- We performed a rolling window exercise to see how our model performed in the case of financial crisis. We segmented the data into rolling windows of 200 days with subsequent 15 test days.
- We observed how our returns curves changed during covid period along with the regimes
- Based on observations, the model showed positive results during the crisis periods

Performance during 2008 crisis

- The graph for 2008 crisis showed an upward trend when markets fell

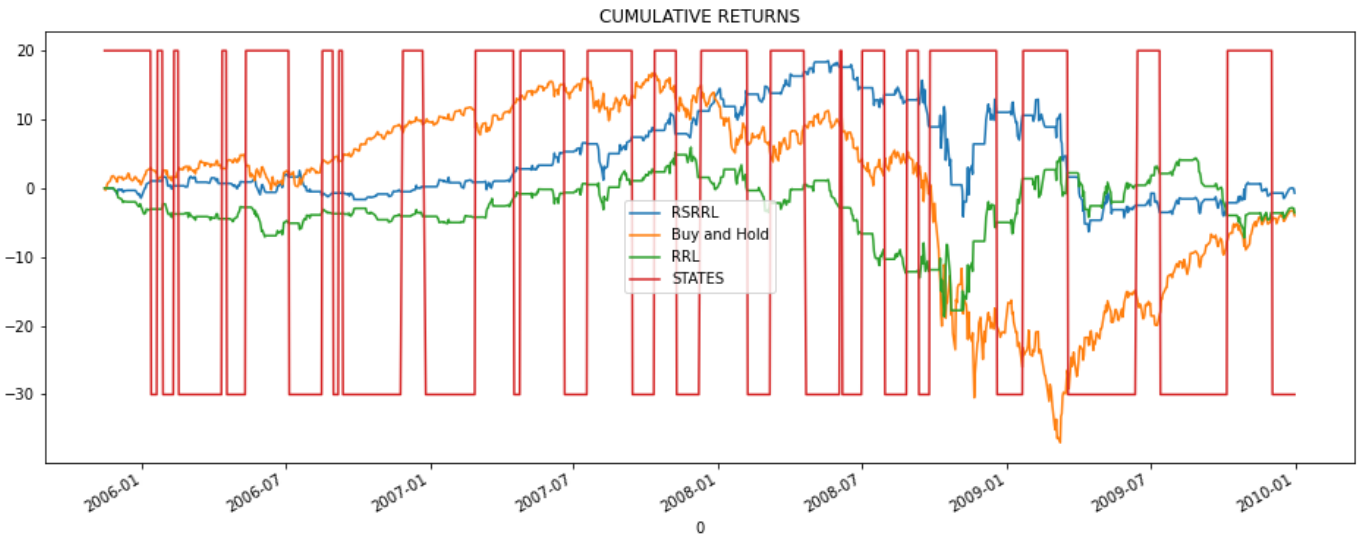


Figure: Cumulative Returns for SPY ETF

Performance during 2008 crisis

- The graph for 2008 crisis showed an upward trend when markets fell

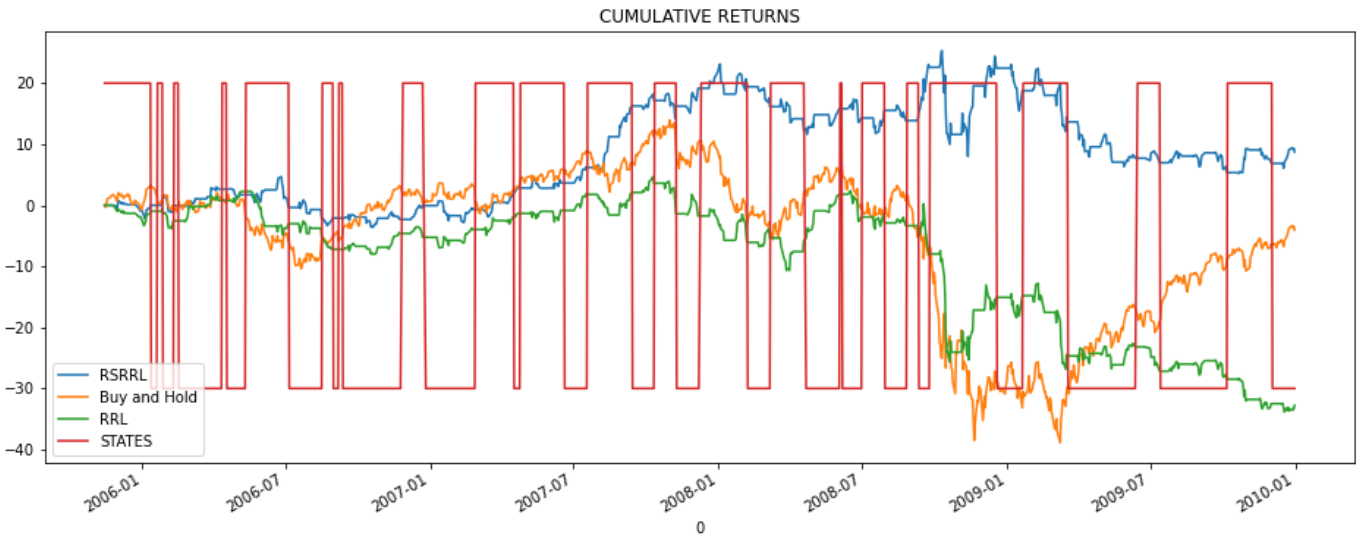


Figure: Cumulative Returns for QQQ ETF

Performance during covid period

- Similar to 2008 crisis the graph for ETFs showed a positive trend when markets fell

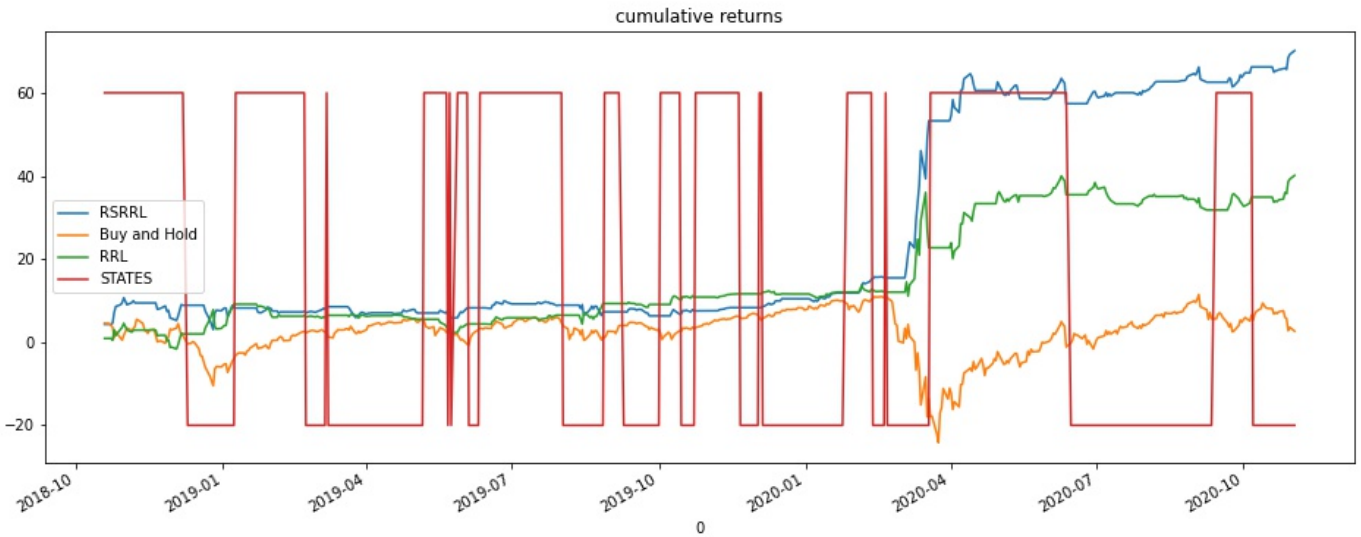


Figure: Cumulative Returns for SPY ETF

Performance during covid period

- Similar to 2008 crisis the graph for ETFs showed a positive trend when markets fell

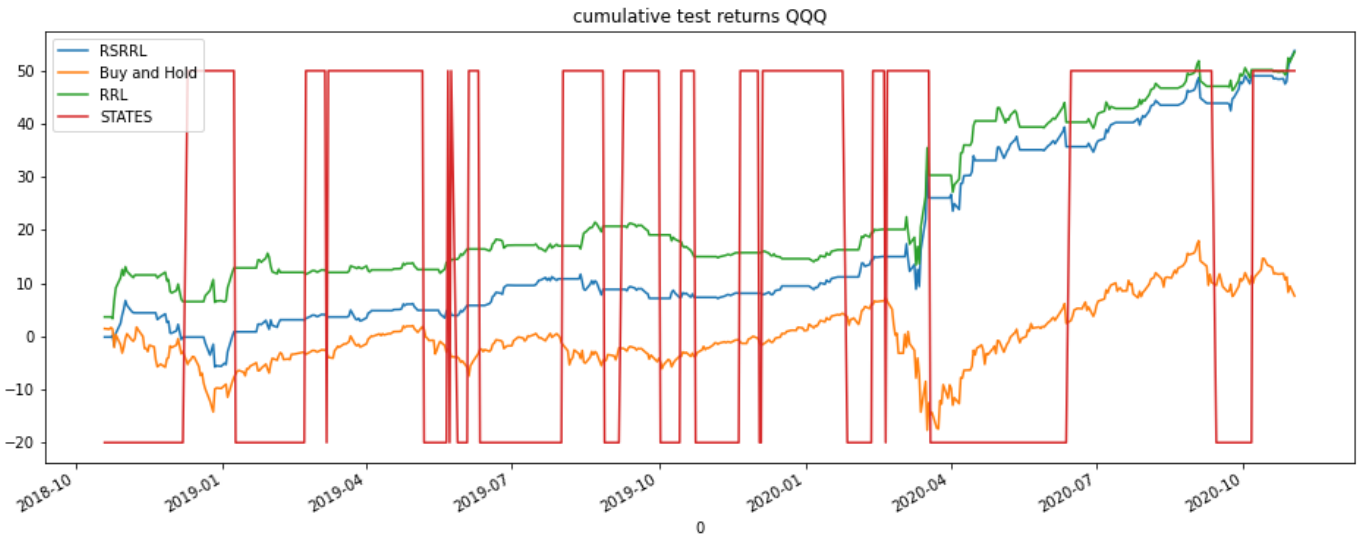


Figure: Cumulative Returns for QQQ ETF

Performance during covid period

- Our model switched states quickly when crisis period started
- This helped us to change the position accordingly as the model trades on the downward regimes

Dietmar Maringer and Tikesh Ramtohul. Regime-switching recurrent reinforcement learning for investment decision making. *Computational Management Science*, 9(1):89–107, 2012.

John Moody and Matthew Saffell. Reinforcement Learning for Trading Systems and Portfolios. In *Kdd*, 1998.

John Moody and Matthew Saffell. Reinforcement learning for trading. *Advances in Neural Information Processing Systems*, (1998):917–923, 1999.

Jin Zhang. Automating Transition Functions : A Way To Improve Trading Profits. *IFIP International Conference on Artificial Intelligence Applications and Innovations*, pages 39–49, 2014.