Automated Trading of Cryptocurrencies with Time-Series Forecasting and Reinforcement Learning

Jeffrey Gradner W01312787

March 2021

1 Supervisory Committee

Chair: Dr. Kyle Feuz	kylefeuz@weber.edu
Member: Dr. Arpit Christi	arpitchristi@weber.edu
Member: Dr. Abdulmalek Al-Gahmi	aalgahmi@weber.edu
Member: Dr. Robert Ball	robertball@weber.edu

2 Thesis Question

What effect does incorporating time series forcasting and technical indicators in a reinforcement learning agent's state space have on cummulative return? I hypothesize that allowing an agent to observe forecasted prices and one or more technical indicators will result in an increase in cummulative reward.

2.1 Purpose

In recent years, the world has seen cryptocurrencies become a phenomenon that intrigues retail and institutional investors alike. With a growing global market cap of $\$2.27T$, an ever-growing distrust in traditional centralized stock markets, and the support of private stakeholders like Tesla, Harvard University, and Yale University, cryptocurrencies appear to be here for the long haul. The ability to identify and capture cryptocurrency price trends can lead to significant returns and is the motivation for this research into how using reinforcement learning with time series forecasting can effect a trading agent's cumulative return.

3 Approach

The proposed novel approach will use a modular regression model first proposed by Taylor and Letham (2017) to generate predictions that will be used in the state space of a Deep Deterministic Policy Gradient (DDPG) agent [1]. The agent will take actions to either buy, sell, or hold, some k amount of a coin given the observable state space. I'll compare cumulative return given an agent who can observe forecasted prices to an agent who does not. In addition, I'll explore how the cumulative return is effected using different combinations of techinncal indicators, with and without the forecasted data. Also, I'll compare results of a simple state space, where for a given coin, the agent can observe the current price, the quantity owned, and the available balance.

3.1 DDPG

DDPG is an off-policy actor-critic network designed for environments with continuous action spaces. The deep Q-learning algorithm has been implemented by OpenAI for both PyTorch and TensorFlow. The state space will build off the state space used by Yang et al. (2020) who used an ensemble of deep reinforcement methods to trade stocks[2]. The observable space will comprise of:

- The available balance .
- The current value.
- Coins owned.
- The following timesteps forecasted prediction.
- Moving Average Convergence Divergence (MACD).
- Relative Strength Index (RSI).
- Commodity Channel Index (CCI).
- Average Directional Index (ADX).

3.2 Data

I'll collect historical data from the Binance.US API and CoinMarketCap, and use Binance.US's API to simulate trading with Paper Money. Then, explore several seasonalities, 15 minutes, hourly, and daily.

3.2.1 Coins

The agents will trade nine coins with the following symbols, BTC, ADA, LINK, EOS, ETH, LTC, XLM, TRX, and XRP.

3.2.2 Seasonality

I'll obtain price data where the time between each data point is 15 minutes, hourly, and daily. The agents will make observations and take actions at each time step according to the interval between data points in each dataset. Given the nine coins listed above I'll have daily price data starting October 2, 2017 - February 27, 2021. The 15 minute, and hourly data will start January 31, 2020 and end Arpil 30, 2021.

3.3 Approach Summary

- Obtain and preprocess historical price data for various cryptocurrencies.
- Tune and train Fbprophet prediction models given multiple seasonalities (e.g., 15 minutes, hourly, and daily).
- Tune, train, and evaluate the proposed DDPG agent with the regression model's predictions in the state space.
- Use the same configuration for the first DDPG agent but exclude the regression predictions and compare performance.
- Compare performance of the agents when actions are taken in relation to the different seasonalities.
- Simulate trading through a Paper Trading API for cryptocurrencies.
- Summarize all results.

4 Research / Related Works

I've read several papers that cover a variety of different approaches to applying machine learning to trading either stocks or cryptocurrencies. Xiong et al. (2018) used a Deep Deterministic Policy Gradient agent to train and trade on data spanning nine years[3]. Their agent produced a 97% return on a starting invesment balance of $\$10,000$ over a testing period of 33 months and an annualized return of 25.87%. The authors used three elements in their state space, the price per stock, number of stocks held, and remaining available balance. They did not consider cryptocurrencies which like stocks can be treated as a time series but are fundamentally different than stocks in how they're governed, their volatility, and the available times to buy and sell.

Another paper writen by Yang et al. (2020), used an ensemble of deep reinforcement learning agents to automate stock trading[2]. In that paper the authors' agent trained on data starting in 2009 and started trading from January 2016 until May 2020. Their ensemble approach was meant to be an improvement on the paper from [3], but performed worse in terms of the Sharpe Ratio, cummulative return and annual return. The state space used in this paper is what the state space I'm proposing adds to and again they consider stocks not currencies. The three agents these authors used in their ensemble are the following from the top-down: Proximal Policy Optimization (PPO), Advantage Actor Critic (A2C), and Deep Deterministic Policy Gradient (DDPG). Interestingly the PPO alone outperformed the ensemble in cummulative return by 13% and by 2% in annual return.

Indulkar (2021) used a long short-term memory (LSTM) network, a varient of the reccurent neural network (RNN), to plot forecasted prices of five different cryptocurrencies[4]. The author also used Fbprophet separately to plot forecasted prices for the same five cryptocurrencies. There was no simulation of trading and the models used were mostly bare-bones, meaning they're out-of-the-box implementations and minimal tuning was performed to increase performance. The approach I'm proposing will use a deep reinforcement learning agent rather than a deep learning agent and the modular regression model provided from Fbprophet will be used in decision making by the RL agent. I also plan to simulate actual trading as the agent takes actions.

5 Criteria

The success of this research will depend on how well the agent performs in terms of cummulative return given various starting balances and when considering different seasonalities. I'll compare the results from my proposed method that integrates Fbprophet with DDPG to an agent without the Fbprophet model's forecast in the state space. Results will be compared for when Fbprophet is used with and without technical indicators, and when Fbprophet is and isn't used with a simple state space that excludes the technical indicators. In summary, there will be four different configurations that will be compared (note: simple state space refers to just three elements, current price, quantity owned, available balance):

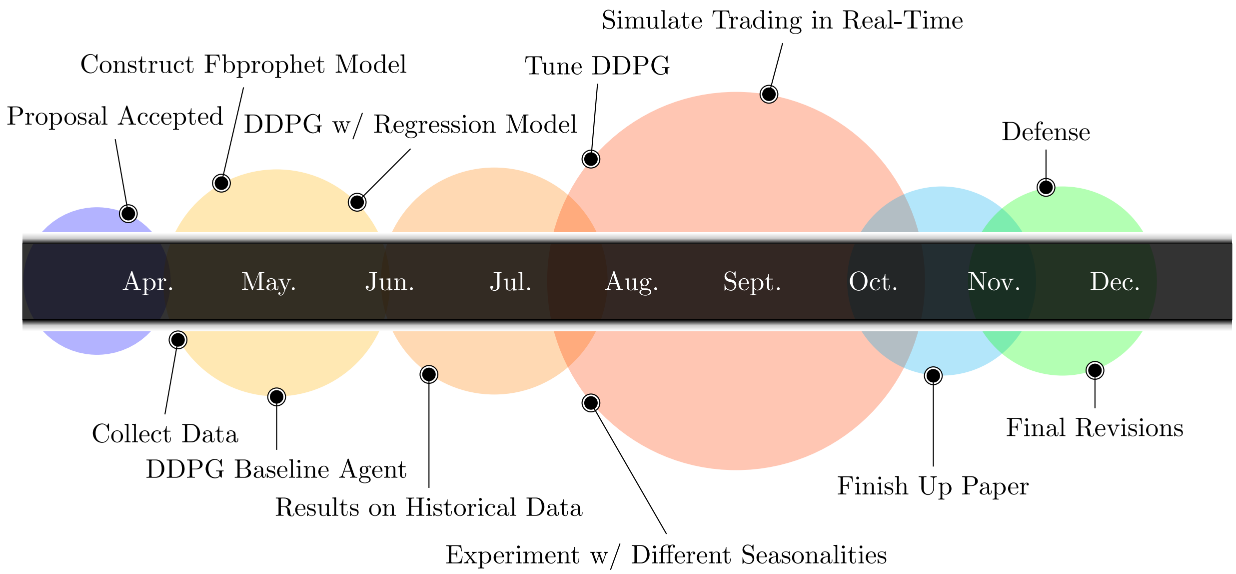
- DDPG with Fbprophet using the simple state space
- DDPG without Fbpophet using the simple state space
- DDPG with Fbprophet using the extended state space (technical indicators)
- DDPG without Fbprophet using the extended state space (techincal indicators)

The 4 configurations will be done for each of the different datasets, 15 minutes, hourly, and daily.

5.1 Measureable Outcomes

- A new approach proposed for autonomous cryptocurrency trading.
- Implementation of the proposed approach with Python, TensorFlow, and Fbprophet.
- Instantiation and use of a DDPG agent that does not use an Fbprophet model's forecasts for comparison.
- Use and tuning of a Fbprophet regression model that considers different seasonalities.
- Compare and report results of the techniques above when trained and tested on historical data.
- Simulate trading within a real-time market using a Paper Trading API (duration TBD).
- Compare results when the models focus on different seasonalities.
- Report seasonal profitability (i.e., hourly return, daily return, and weekly return).
- Report robustness by measuring the percentage of actions that result in an increase of the overall portfolio balances.

6 Tentative Timeline



7 Committee Signatures

Chair ________________________________ Date ____________
Dr. Kyle Feuz

Member ________________________________ Date ____________
Dr. Arpit Christi

Member ________________________________ Date ____________
Dr. Abdulmalek Al-Gahmi

Member ________________________________ Date ____________
Dr. Robert Ball

References

[1] S. J. Taylor, B. Letham, Forecasting at scale, The American Statistician 72 (1) (2018) 37–45.

[2] H. Yang, X.-Y. Liu, S. Zhong, A. Walid, Deep reinforcement learning for automated stock trading: An ensemble strategy, Available at SSRN (2020).

[3] Z. Xiong, X.-Y. Liu, S. Zhong, H. Yang, A. Walid, Practical deep reinforcement learning approach for stock trading, arXiv preprint arXiv:1811.07522 (2018).

[4] Y. Indulkar, Time series analysis of cryptocurrencies using deep learning & fbprophet, in: 2021 International Conference on Emerging Smart Computing and Informatics (ESCI), IEEE, 2021, pp. 306–311.