Generative Adversarial Networks (GAN): Applications in Cryptocurrency Price Forecasting

Chris Gough

December 2021

Full Project Proposal

1 Executive Summary

There has been a long history of research into the field of time series forecasting with a specific focus on the field of financial time series. Modelling in finance has proven to be extremely difficult because there are often underlying correlations and statistical properties that are hard to identify. A large body of work has focused on statistical methods of time series forecasting while a newer area has been the development of neural network models such as Long Short Term Memory (LSTM) models and more recently Generative Adversarial Networks (GANs). GANs gave been found to convincingly outperform LSTM on financial time series but has not yet been tested on cryptocurrencies. We believe this research is necessary given the meteoric rise of cryptocurrencies and the increasing attractiveness of it as an investment option.

2 Research Question

The research question proposed is as follows: Can the novel GAN framweork proposed by Zhang et al. (2019) be successfully applied to a new field of foreasting, namely cryptocurrency returns? If so, how does it compare to other established statistical and deep learning methods in the field. Does it outperform these methods, is this statistically significant and by how much does it outperform? We believe that it can be successfully applied to the dataset and will layout a framework to prove this throughout this paper.

3 Survey of Background Literature

3.1 Overview

Although cryptocurrencies have been around for a number of years now they are still severely misunderstood. , and in some cases feared, by the general public. This is largely due to the fact that they don't follow any of the classical theories of finance. Many researchers have attempted to apply forecasting methods to them with varying degrees of success. This project aims to follow the method outlined by (Zhang et al., 2019) and apply it for the first time to a cryptocurrency price forecasting problem.

3.2 Background

In the field of finance and investments, one of the most important ongoing areas of research is that of the prediction of future asset prices. This research is being performed not only in academia but also within the large banks and trading firms, and is hotly contested because anyone who can find an accurate method of forecasting asset returns will have an advantage over those who don't and could trade profitably on this information. Predicting future asset returns however is a notoriously difficult task because of the complex interactions and correlations between assets, as well as seasonal effects and non-linearity in the data. While financial asset returns may seem extremely complex and random at times, there are several statistically significant properties that are common across multiple markets, time frames and asset classes.(Eckerli and Osterrieder, 2021) These are known as stylized empirical facts and it was shown in a study by Cont (2001) who found that there is a set of these characteristics or stylised facts that can consistently be found and thus can be used to replicate the statistical properties of these assets, namely: absence of auto-correlations, conditional heavy tails, gain/loss asymmetry and volatility clustering to name a few. A 2018 paper by Zhang et al. conducted upon 8 cryptocurrencies accounting for 70% of the market displayed clear evidence that these stylised facts also hold true for the for cryptocurrencies as an asset class.

The prediction of future stock prices and market movements has generally been framed as a time series forecasting problem. In classical time series forecasting, the majority of models are based on classic statistical methods and the most well known and common of this group is the Auto Regressive Integrate Moving Average (ARIMA) which has been shown to perform well on data that is stationary and linear. However it performs poorly when the data is non linear, in this case a hybrid approach combining ARIMA and Support Vector Machines (SVM) were shown to provide promising improvements in forecasting accuracy. (Pai and Lin, 2005) More recently statistical models have begun to be outperformed by neural network approaches, Borovykh et al. (2018) proposed a using a Convolutional Neural Network (CNN) to forecast the performance of the S&P500 and found that a CNN was quicker to train, easy to implement and outperformed linear models. A study by Siami-Namini and Namin in 2018, compared the performance of an LSTM model in comparison to ARIMA on financial time series forecasting found that LSTM outperformed ARIMA, with an average error reduction of roughly 84-87%.

A newer approach to time series forecasting using GANs has begun to provide promising results in this field. GANs are a type of generative machine learning model and was first proposed by Goodfellow et al. (2014) who introduced a simple and effective framework, which was initially applied to the problem of image generation but has since been expanded into a wide array of various other tasks. The application of GANs to the field of finance and financial time series forecasting is still a relatively new research prospect and it is thought that "GANs may provide good results, since they can generate data by sampling only from real data, often with no additional assumptions or inputs". (Eckerli and Osterrieder, 2021) It is also thought that this ability of GANs to avoid the need to use assumptions protects it from human biases that may affect the modelling process. (Eckerli and Osterrieder, 2021) The GAN framework that was originally proposed by Goodfellow et al. (2014) employs an adversarial process in order to estimate generative models by training two models simultaneously. The first is a generative model known as the Generator to match the distribution of a dataset and generate new data that matches the sample statistics of this dataset, while the second model is a classifier called the Discriminator which is tasked with "estimating the probability that a sample came from the training data" and was not a fake sample made by the Generator. (Eckerli and Osterrieder, 2021) In order to train the Generator, the process is to "maximise the probability of its output being misclassified by the Discriminator" (Eckerli and Osterrieder, 2021) Both the generator and discriminators have their own loss functions, that penalise them for poor performance at generation/discrimination and these are updated alternately as each network is trained, weights for the networks are then updated by backpropagation from the loss function backwards throughout the network.(Eckerli and Osterrieder, 2021)

Although GANs are achieving very promising results, there also significant challenges that they face in their application. They can be very hard to train Eckerli and Osterrieder (2021) because they are trying to optimise two networks simultaneously and thus they may not reach convergence. Eckerli and Osterrieder (2021) explained this problem as follows: "if the Generator gets good too fast, it may fool the Discriminator and stop getting meaningful feedback, which in turn will make the Generator train on bad feedback , leading to a collapse in output quality". Mode collapse is another issue with GANs where due to a deficiency in training, the Generator gets stuck in a local minimum, producing less varied samples and and thus the Discriminator will learn to quickly discriminate the Generators fakes. Eckerli and Osterrieder (2021). GANs also suffer from the vanishing gradient problem where the discriminator becomes very accurate and does not provide enough feedback for the Generator to learn and the gradients being passed backwards during backpropagation are too small to update the weights of the initial layers in the Generator and training will slow down and eventually end. Eckerli and Osterrieder (2021) Research from Wang et al. (2020)

has shown that there are two main solutions to the issues discussed above, using different architectures and using different loss functions. In order to improve the training issues the network should be made deeper and the batch size increased, while the vanishing gradient problem can be solved by changing the loss function. (Wang et al., 2020) These solutions will need to be tested and applied on a cases by case basis as some models will be affected slightly differently by these issues due to different architectures. An optimizer is used to update the model as result of the loss functions output and help find the combinations of model parameters that minimize the loss function. (Eckerli and Osterrieder, 2021) The majority of neural network models today use a optimiser that is based on gradient descent, where the idea is to take repeated steps in the opposite direction of the loss functions gradient in order to find the parameters that minimise the loss function as quickly as possible. The learning rate during this process is a crucial hyperparameter because a learning rate that is too small will take to long to converge, while a learning rate that is too high will take steps that are too big and may cause the algorithm to begin to diverge. While there are many optimisation algorithms available, in the GAN frameworks and literature the most popular optimisation algorithm currently being used is that of adaptive moment estimation or ADAM as it is more commonly known. (Eckerli and Osterrieder, 2021) ADAM is used to iteratively update the network weights by storing by "both the exponentially decaying average of past squared gradients and exponentially decaying average of past gradients". (Eckerli and Osterrieder, 2021)

In the field of financial time series forecasting there are a few specifically developed GAN architectures that have shown promising results. In 2017, Zhou et al. (2018) developed a framework they called GAN-FD which was developed to forecast high frequency stock market data one period ahead. The GAN-FD architecture made use of an LSTM model as the the Generator and a CNN as the discriminator for adversarial training, and they found that their proposed framework "could effectively improve stock price prediction accuracy and reduce forecast error." Zhou et al. (2018) Zhang et al. (2019) were the first to propose a novel method of implementing a GAN for financial forecasting by using a multi linear perceptron (MLP) as the discriminator and a LSTM as the generator. In their model, the generator mines the distribution of stock prices to generate data from a similar distribution and then the discriminator has the task of identifying which data comes from the real data distribution and which has been generated Zhang et al. (2019). They found that their GAN model was able to achieve competitive results when compared to other models in forecasting the daily closing prices of the S&P500 and other individual stocks over a wide range of trading days Zhang et al. (2019). More recently in 2021 two new GAN architectures have been proposed for financial time-series forecasting, namely MTSGAN and ST-GAN. Multi Time Series GAN or MTSGAN was proposed by Wu et al. (2021) in order to provide accurate future forecasts for financial time series by capturing the complex interaction between multiple time series and the temporal dependcies within each of these time series. The MTSGAN architecture consists of an interaction matrix generator, a prediction generator and a time series discriminator, they also used a graph convolutional network (GCN) in order to extract interactional dependencies and long short-term memory (LSTM) networks to extract the temporal dependencies. Wu et al. (2021) They found that when comparing the performance of MTSGAN as a predictor on various datasets to other benchmarks that the MTSGAN model consistently outperforms other current state of the art methods when applied to multiple related time series forecasting problems. Wu et al. (2021) Muthukumar and Zhong (2021) proposed a novel framework call ST-GAN or Stochastic-GAN which they used to analyze both financial news texts as well as financial numeric data in order to predict stock price trends and direction movements. Their implementation was different to that of those before because of the use of a Naive Bayes Classifier to perform sentiment analysis on the financial news text data and then feed this into the GAN as additional input. (Muthukumar and Zhong, 2021) They found that their model performed significantly better then the current models and research on deep neural network when applied to stock price forecasting (Muthukumar and Zhong, 2021)

3.3 Relevance/Impact

The relevance of this research is clear because it will allow a better understanding of the behaviour of cryptocurrency price movements intraday and whether a GAN can be used as a better predictor of these movements then other traditional forecasting methods. It is important to apply this new methodology to cryptocurrencies and contribute to the body of knowledge for cryptocurrencies. The potential impact of a successful forecasting method may also have clear financial implications for those looking to create a trading strategy based on the GAN framework. Further while the GAN framework proposed by the original authors Zhang et al. provided promising forecast results it is unclear how extensible this framework would be to other areas of forecasting, which this paper aims to prove by reproducing their framework and attempting to produce similiar results on a completely different dataset. This research will thus also contribute to proving how robust the original framework is for producing accurate time series forecasts.

4 Proposed Methodology

The methodology proposed for this research project will be follow that of the novel methodology proposed by Zhang et al. (2019) in their paper. The GAN architecture that they proposed implemented a LSTM model as the discriminator and a MLP as the discriminator, which was then applied to to 7 financial factors over 20 years in order to predict a future closing price one day ahead. (Zhang et al., 2019). Where this paper will differ is that we will be applying this methodology to forecasting the returns of cryptocurrencies. This is a field of research which has not previously seen the implementation of this specific forecasting method. We will also attempt to further develop additional factors to use as inputs to the model which may help improve performance. One potential issue that may be faced is that because cryptocurrencies are a relatively new asset class, there will be much fewer years of data with which to train our model and also that there may be a large degree of inter-asset correlation present that will need to be test and accounted for with statistical methods.

The methodology proposed by Zhang et al. (2019) supposed an input of $X = \{x_1, ..., x_t\}$ where the number of days of data is denoted as t and thus each x_k in X is a vector with length the same as the number of factors n as follows in equation 1.

$$[X_{k,i}]_{i=1}^{n} = [X_{k,1}, ..., X_{k,n}] \tag{1}$$

The generator proposed by Zhang et al. (2019) is shown in figure 1 below and is defined by equations 2 and 3 below, the inputs X are passed to the LSTM layer and the output h_t is then passed on to a fully connected layer which computes x_{t+1}, the one day ahead forecast. A Leaky Rectified Linear Unit (RELU) activation function is used and is denoted by δ, a dropout layer is also implemented for regularisation and W_h and b_h denote the weights and biases for the fully connected layer. (Zhang et al., 2019)

$$h_t = g(X) \tag{2}$$

$$G(X) = \hat{x}_{t+1} = \delta(W_h^t h_t + b_h) \tag{3}$$

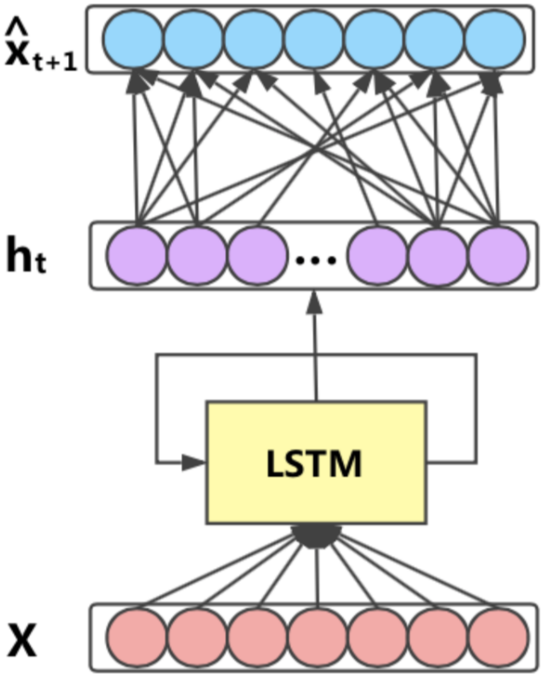

Figure 1: The generator model as proposed by Zhang et al. (2019).

The discriminator proposed by Zhang et al. (2019) is shown in figure 2 and has the purpose of of "constituting a differentiable function D to classify the input data". It will output 0 when the input data is fake and 1 when the input data is real. (Zhang et al., 2019) They chose to implement a MLP with 3 three hidden layers consisting of 72, 100 and 10 neurons respectively, with a Leaky Relu activation function for the hidden layers and a Sigmoid function for the output layer, as well as cross entropy loss used as the loss function for optimisation. (Zhang et al., 2019) The output of the discriminator as defined by Zhang et al. (2019) is defined by equations 4 and 5 below:

$$D(X_{fake}) = \sigma(d(X_{fake})) \tag{4}$$

$$D(X_{real}) = \sigma(d(X_{real})) \tag{5}$$

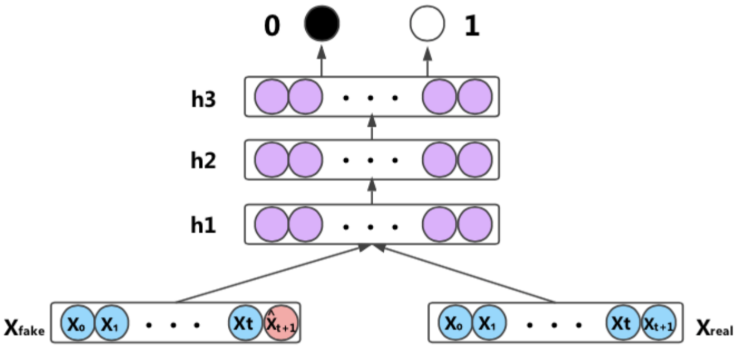
0
1
h3
h2
h1
X_{fake}
X_0 X_1 . . . Xt $\hat{X}_{t+1}$
X_0 X_1 . . . Xt X_{t+1}
X_{real}

Figure 2: The discriminator model as proposed by Zhang et al. (2019).

The GAN architecture as proposed by Zhang et al. (2019) is shown in figure 3 below and is composed of both the Generator and Discriminator models defined above. The value function $V(G, D)$ that needs to be optimized as per Zhang et al. (2019) is defined by equation 6. The generator loss and discriminator loss are defined to optimise the value function and are shown in equations 7, 8 and 9 with the generator loss being a combination of the generator MSE and the generator loss of a classical GAN with the hyperparameters λ_1 and λ_2 being set manually. (Zhang et al., 2019)

$$\underset{G}{min}\,\underset{D}{min}\,V(G, D) = E[logD(X_{real}] + E[log(1 - D(X_{fake})] \tag{6}$$

$$D_{loss} = \frac{1}{m}\sum_{m}^{i=1} logD(X^i_{real}) - \frac{1}{m}\sum_{m}^{i=1} log(1 - D(X^i_{fake}) \tag{7}$$

$$g_{MSE} = \frac{1}{m}\sum_{m}^{i=1} (\hat{x}^i_{t+1} - x^i_{t+1})^2 \tag{8}$$

$$g_{loss} = \frac{1}{m}\sum_{m}^{i=1} log(1 - D(X^i_{fake})) \tag{9}$$

$$G_{loss} = \lambda_1 g_{MSE} + \lambda_2 g_{loss} \tag{10}$$

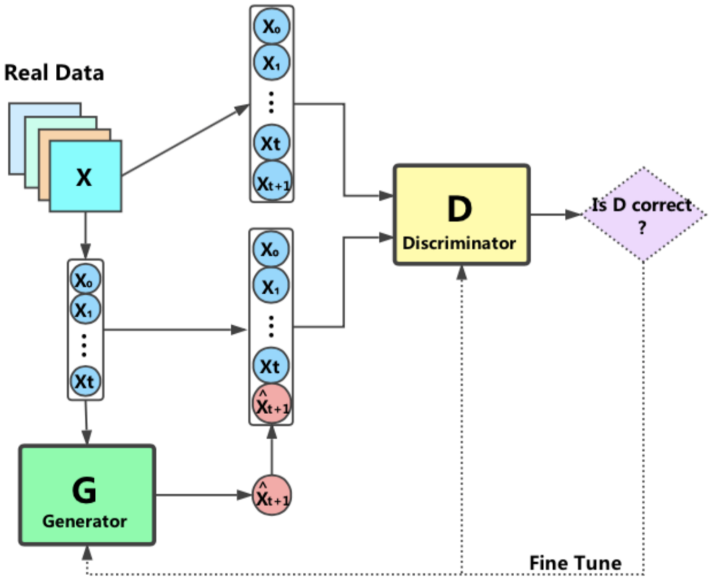

Figure 3: The Gan archictecture as proposed by Zhang et al. (2019).

We will then apply this GAN model architecture as defined by Zhang et al. (2019) to our cryptocurrency daily price price data set. This data will include a few of the largest cryptocurrencies available today by market cap (Bitcoin, Ethereum, Bitcash to name a few), we will also exclude any cryptocurrencies that do not have more then 2 years of data in order to ensure that there are enough data points to accurately train the GAN. The data will also need to be normalised as per Zhang et al. (2019). The data will then be split into a training and test set with roughly 90% of the data in the training set and the remaining 10% in the test set as per the framework defined by Zhang et al. (2019).

The code will be implemented in Google Colab which gives access to an environment similar to a Jupyter Notebook but with access to either a GPU or TPU run-time which will significantly speed up training times for the GAN, and reduces the need to have an access to an expensive GPU. In order to evaluate the accuracy of our models forecasts on the training set we will follow the framework set out by Zhang et al. (2019) and apply the following metrics: Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE) and Average Return (AR). Zhang et al. (2019) computed the mean performance of their model across the 5 datasets they used and this study will also make us of this method, as per the original model framework we will also compare the forecasting performance of the GAN model to that achieved by neural networks, LSTM and ANN, as well as Support Vector Regression, another well known statistical modelling method for forecasting future prices of financial assets.

5 Research Plan

The research plan and the significant milestones for this project are as set out out in the figure 4 below. I will briefly outline the tasks I have identified and give an estimation of the time I believe each task will take.

1. Research Proposal - 10 weeks
2. Literature Review - 16 weeks
3. Planning Methods and Writing Methods Chapter - 10 weeks
4. Data Collection - 2 weeks
5. Exploratory Data Analysis - 2 weeks
6. Modelling and Coding - 2 weeks
7. Visual Results - 1 week
8. Write Results Chapter - 2 weeks
9. Write Discussion Chapter - 2 weeks
10. Write Introduction and Conclusion - 1 week
11. Exam - 1 week
12. Major Editing - 2 weeks
13. Final Proof Reading - 1 week
14. Submit - 1 week

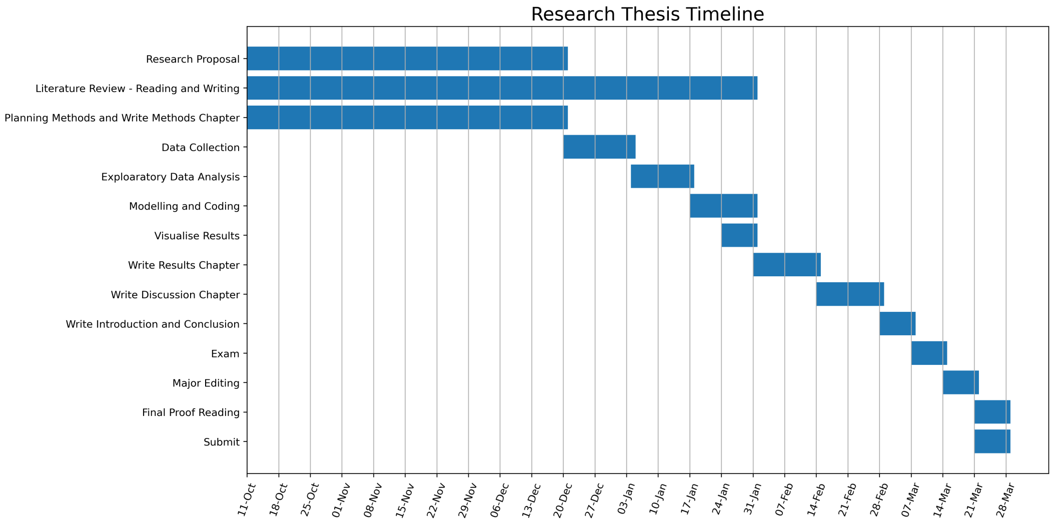

Figure 4: Gantt chart of research project timeline.

6 Ethical Considerations

To the best of my knowledge and after completing the required ethics questionnaire, there are no ethical concerns or considerations in relation to this project. No personal data or specific persons are involved in this study and no survey research is to be undertaken. All data to be used to achieve the outcomes of this project are freely available on the internet and are generated by global cryptocurrency markets.

7 Resources

There will be no financial resources needed to complete this project as all financial datasets that are needed to complete this project are freely available online and all software used for modelling and analysis will be open source. Python is the preferred language for completion of this research project and Google Colaboratory will be used in order to gain access to a GPU runtime that will assist with speeding up the training the deep neural networks that make up the GAN architecture. The cryptocurrency price datasets are freely available at the following url: `www.cryptodatadownload.com`

Word Count - 3063 words

References

Borovykh, A., Bohte, S. and Oosterlee, C. W. (2018), 'Conditional time series forecasting with convolutional neural networks'.

Cont, R. (2001), 'Empirical properties of asset returns: stylized facts and statistical issues', *Quantitative Finance* **1**, 223 – 236.

Eckerli, F. and Osterrieder, J. (2021), 'Generative adversarial networks in finance: an overview'.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y. (2014), 'Generative adversarial networks'.

Muthukumar, P. and Zhong, J. (2021), 'A stochastic time series model for predicting financial trends using nlp'.

Pai, P.-F. and Lin, C.-S. (2005), 'A hybrid arima and support vector machines model in stock price forecasting', *Omega* **33**(6), 497–505.
URL: *https://www.sciencedirect.com/science/article/pii/S0305048304001082*

Siami-Namini, S. and Namin, A. S. (2018), 'Forecasting economics and financial time series: ARIMA vs. LSTM', *CoRR* **abs/1803.06386**.
URL: *http://arxiv.org/abs/1803.06386*

Wang, Z., She, Q. and Ward, T. E. (2020), 'Generative adversarial networks in computer vision: A survey and taxonomy'.

Wu, W., Huang, F., Kao, Y., Chen, Z. and Wu, Q. (2021), 'Prediction method of multiple related time series based on generative adversarial networks', *Information* **12**, 55.

Zhang, K., Zhong, G., Dong, J., Wang, S. and Wang, Y. (2019), 'Stock market prediction based on generative adversarial network', *Procedia Computer Science* **147**, 400–406. 2018 International Conference on Identification, Information and Knowledge in the Internet of Things.
URL: *https://www.sciencedirect.com/science/article/pii/S1877050919302789*

Zhang, W., Wang, P., Li, X. and Shen, D. (2018), 'Some stylized facts of the cryptocurrency market', *Applied Economics* **50**(55), 5950–5965.
URL: *https://doi.org/10.1080/00036846.2018.1488076*

Zhou, X., Pan, Z., Hu, G., Tang, S. and Zhao, C. (2018), 'Stock market prediction on high-frequency data using generative adversarial nets', *Mathematical Problems in Engineering* **2018**, 1–11.